Parametrization for Surfaces with Arbitrary Topologies

A thesis presented

by

Xianfeng Gu

to

Division of Engineering and Applied Sciences

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of

Computer Science

Harvard University
Cambridge, Massachusetts

December 2002

Parametrization for Surfaces with Arbitrary Topologies

Abstract

Surface parametrization is a fundamental problem in computer graphics. It is essential for operations such as texture mapping, texture synthesis, interactive 3D painting, remeshing, multi-resolution analysis and mesh compression. Conformal parameterization, which preserves angles, has many nice properties such as having no local distortion on textures, and being independent of triangulation or resolution. Existing conformal parameterization methods partition a mesh into several charts, each of which is then parametrized and packed to an atlas. These methods suffer from limitations such as difficulty in segmenting the mesh and artifacts caused by discontinuities between charts.

This work presents a method that was developed with collaboration with Professor Shing-Tung Yau to compute global conformal parameterizations for triangulated surfaces with arbitrary topologies. Our method is boundary free, hence eliminating the need to chartify the mesh. We compute the natural conformal structure of the surface, which is determined solely by its geometry. The parameterization is stable in the sense that if the geometries are similar, then the parameterizations on the canonical domain are also close. The parameterization is conformal everywhere except on $2g-2$ number of points, where g is the number of genus. We prove the gradients of local conformal maps form a

linear space and develop practical algorithms to compute its basis. By linearly combining the bases we can construct any global conformal parametrization of the surface. The algorithms only involve solving linear systems and are easy to implement.

In this work, we also propose to remesh an arbitrary surface onto a completely regular structure, we call a *geometry image*. This part of the thesis was completed under the supervision of Professor Steven Gortler. A geometry image captures geometry as a simple 2D array of quantized points. Surface signals like normals and colors are stored in similar 2D arrays using the same implicit surface parameterization-texture coordinates are absent. To create a geometry image, we cut an arbitrary mesh along a network of edge paths, and parametrize the resulting single chart onto a square. Geometry images unify the geometry and image format and make it possible to apply many techniques in the image processing field to geometry directly.

Contents

1 Introduction **6**
- 1.1 Problem Statement . . . 6
  - 1.1.1 Geometry Images . . . 6
  - 1.1.2 Global Conformal Parameterization . . . 8
- 1.2 Contributions . . . 9
- 1.3 Overview . . . 10
- 1.4 Motivating Applications . . . 15
  - 1.4.1 Efficient Hardware rendering . . . 16
  - 1.4.2 Texture Mapping . . . 16
  - 1.4.3 Remeshing . . . 17
  - 1.4.4 Compression . . . 17
  - 1.4.5 Visualization . . . 17
  - 1.4.6 Shape Analysis . . . 17
- 1.5 Overview of Thesis . . . 18

2 Previous Work **18**
- 2.1 Parameterization . . . 18
- 2.2 Remeshing . . . 20
- 2.3 Global Conformal Parameterization for Genus Zero Surfaces . . . 21
- 2.4 Global Conformal Parameterization for Surfaces with Arbitrary Topologies . . . 22
- 2.5 Computational Topology . . . 23
- 2.6 Conformal Structure . . . 24
- 2.7 Spectral Compression . . . 24

3 Geometry Images **24**
- 3.1 Creation of Geometry Images . . . 25
  - 3.1.1 Parameterization . . . 26
  - 3.1.2 Cutting . . . 28
  - 3.1.3 Topological sideband . . . 32
- 3.2 Applications . . . 33
- 3.3 Results . . . 35
- 3.4 Summary and Future Work . . . 36

4 Discrete Riemann Geometry **46**
- 4.1 Surface Triangulations . . . 46
- 4.2 Simplicial Homology . . . 49
- 4.3 Simplicial Cohomology . . . 50
- 4.4 Interaction Between Homology and Cohomology . . . 51
- 4.5 Conformal Structure for Meshes . . . 52
- 4.6 Harmonic Maps on Smooth Riemann Surfaces . . . 53
- 4.7 Discrete Harmonic Energy . . . 54
- 4.8 Discrete Laplacian Operator . . . 55
- 4.9 Discrete Harmonic 1-Forms . . . 57

4.10 Discrete Hodge Star Operator 60
4.11 Discrete Laplacian Spectrum 63
4.12 Discrete Holomorphic Differentials 64
4.13 Surface Complex Structure 64

5 Conformal Mapping for Genus Zero Surfaces 66
5.1 Constrained Variational Problem 67
5.2 Steepest Descendent Algorithm 68
5.3 Results 73

6 Computing Conformal Structures for Non-zero Genus Meshes 79
6.1 Overview 79
6.2 Computing Homology 82
6.3 Computing Cohomology 86
6.4 Diffusion 90
6.5 Compute Holomorphic 1-Forms 92
6.6 Results 94

7 Global Conformal Parameterization 99
7.1 Branch Points 99
7.2 Modular Structure of Each Handle 101
7.3 Handle Separation 102
7.4 Algorithm for Parameterization 103
7.5 Results 106

8 Performance Analysis and Applications 110
8.1 Performance Analysis 110
8.1.1 Independence 110
8.1.2 Extruding Region 110
8.1.3 Triangulation 111
8.2 Potential Applications 114
8.2.1 Constructing Geometry Images 114
8.2.2 Texture Mapping and Synthesis 114
8.2.3 Surface Classification 116
8.2.4 Laplacian Spectrum Representation 118

9 Summary and Future Research 123
9.1 Homology Independent Global Conformal Parameterization . . . 123
9.2 Multiresolution Parameterization 124
9.3 Generalization to Other Surface Representations 124
9.4 Embedding in Hyperbolic Domain 125

List of Figures

1 Processing the torus mesh 12
2 Process for computing global conformal parameterization 16
3 Columns (a–d) show iterations of the cut improvement algorithm. Upper images show the mesh M with the current cut ρ (blue except red where occluded). Bottom images show the Floater parameterization (over circle) of the corresponding M', together with the shortest path to an extremal point,which will be added to ρ. Column (e) shows the final cut ρ and the geometric-stretch parameterization (over square). 30
4 Rate distortion for geometric reconstruction from compressed geometry images of the bunny (at 257×257 and 513×513 resolutions, and using a Floater-parameterization), compared to [23]. 35
5 Example artifacts in the Buddha geometry image: aliasing (jaggedness) near sharp features, and regions of high anisotropy. 37
6 Mip-mapping a geometry image. As in all examples, the boundary parameterization is constructed for a 65×65 domain grid. . . 38
7 (a–c) Surfaces reconstructed from a 257×257 geometry image under increasing levels of wavelet compression. (d) Reconstructed from a 257×257 Floater-parametrized geometry image. All models are flat-shaded. 39
8 Creation, compression, and rendering of a geometry image. Images b∗ and f∗ (not shown) are compressed using an image wavelet-coder. Geometry image is 12-bit $[x, y, z]$ visualized as $[r, g, b]$. Normal-map image is 8-bit $[n_x, n_y, n_z]$ visualized as $[r, g, b]$. . . . 41
9 Dragon mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping. 42
10 Buddha mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping. 43
11 Three holes torus mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping. . . . 44
12 Gargoyle mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping. 45
13 David mesh with 20000 faces 46
14 Gargoyle model spherical conformal mapping & barycentric mapping 70
15 Bunny model spherical barycentric morphing 71
16 Spherical conformal map of the bunny mesh. The stretching factors of the bunny ears are much bigger than those of the other parts. 72
17 Conformal texture mapping of the brain mesh. Comparison of (a) and (c) shows that the conformal parameterization is independent of resoultion and triangulation. (e) and (f) show the mapping is angle preserving. 76

18 Brain meshes conformal spherical maps. The shapes of the main geometric features are preserved. 77
19 Conformal mapping of the same geometry with different resultions and triangulations. It demonstrates that the mapping is independent of resolution and triangulation. 78
20 Homology basis curves of a genus g surface 83
21 Index of a common point on two loops b_0 and b_1 85
22 Computing holomogy bases of a torus 86
23 Wedge on mesh and splitted along the colored edges 87
24 Floater embedding vs. Tuette Embedding 88
25 Computing dual cohomology 95
26 Homology bases and holomorphic differential bases 96
27 Holomorphic 1-form and stretching level sets of stretch factors . 97
28 Holomorphic 1-forms from different meshes 98
29 Branch points (degree 2) 100
30 Modular structures of the global conformal parameterization for each handle 104
31 For the genus 3 mesh, there are 4 branch points, and 3 handles are mapped to 3 overlapping patches 105
32 Moduler structure for each handle domain 106
33 Separation of handles. The separator is mapped to both patch domains. 107
34 Examples of global conformal parametrizations 108
35 Global conformal parameterization of David mesh 109
36 Boundary independent conformal mapping 112
37 Convergence analysis: The extruding parts converge slowly. The negative string constants cause divergence. 113
38 Surface texture mapping 115
39 Surface classification by period matrices 117
40 Spectrum independent of resolution 119
41 Surface compression 120
42 Surface reconstruction from Laplacian Spectrums 122

Acknowledgments

First, I would like to thank my advisor Steven J. Gortler for his guidance on research, and his help at various stages of my graduate study. He led me to the field of mesh parameterization and has supervised me in the geometry image part of my thesis work.

I would like to thank my coadvisor Shing-Tung Yau for leading me into the world of pure mathematics. His deep insights and global view of geometry inspired me to break through my research in the graphics field. The global parameterization part of the thesis was conducted under his supervision.

I would also like to thank Hugues Hoppe for teaching me the sophisticated techniques in mesh processing. His collaboration in the work of geometry images is highly appreciated.

I would like to thank July Dorsey, Mich Millan, John Snyder for their great help on my research.

Last but not least, I would like to thank my wife, Yan Li, without her support this thesis would not have been possible.

1 Introduction

1.1 Problem Statement

This work aims to solve the problem of parameterizing surfaces with arbitrary topologies and represent the surfaces using the parameterization. There are two closely related topics in this work, geometry images and global conformal parameterizations. Geometry image modifies the topology of the surface, and resamples the surface by regular pattern using special parameterizations. Global conformal parameterization conformally maps the surface to a canonical domain, while preserving the conformality everywhere.

The research of geometry images was done under the supervision of Professor Steven Gortler. The work for global conformal parameterization was supervised by Professor Shing-Tung Yau.

1.1.1 Geometry Images

Surface geometry is often modeled with irregular triangle meshes. The process of *remeshing* refers to approximating such geometry using a mesh with (semi)-regular connectivity. Resampling geometry onto a regular structure offers a number of benefits. Compression is improved since the connectivity of the samples is implicit. Moreover, remeshing can reduce the non-uniformity of the geometric samples in the tangential surface direction, thus reducing overall entropy. The regularity of sample neighborhoods helps in applying signal processing operations and in creating hierarchical representations of multi-resolution viewing and editing.

However, current techniques for remeshing arbitrary surfaces create only semi-regular meshes. The original mesh is typically decomposed into a set of disk-like charts, onto which the geometry is parametrized and sampled. Although the sampling on each chart follows regular subdivision, the chart do-

mains form an irregular network over the surface. This irregular domain network complicates processing, particularly for operations that require accessing data across neighboring charts.

The major problems need to be solved here are:

1. How to use a single chart to parameterize the entire mesh and remesh it using a completely regular pattern, and represent the geometry in an image format.

2. How to conformally parameterize the entire mesh, and preserve the conformality even across the cut boundary.

Representing surfaces as geometry images presents challenges:

- A cut must be found that opens the mesh into a topological disk, and that also permits a good parameterization of the surface within this disk. We describe an effective, automatic method for cutting arbitrary 2-manifold meshes (possibly with boundaries).

- The image boundary must be parametrized such that the reconstructed surface matches exactly along the cut, to avoid cracks. Traditional texture mapping is more forgiving in this respect, in that color discontinuities at boundaries are less noticeable.

- The parameterization must evenly distribute image samples over the surface, since undersampling would lead to geometric blurring. We do not make a technical contribution in this area, but simply apply the geometric-stretch parameterization of [36, 35].

- Straightforward lossy compression of the geometry image may introduce tears along the surface cut. We allow fusing of the cut by encoding the cut topology as a small data sideband.

1.1.2 Global Conformal Parameterization

Surface parameterization is a fundamental problem in computer graphics. It is essential for many geometric operations, such as texture mapping, remeshing, compression, visualization and shape analysis. A parameterization defines a mapping between regions on the 2D plane and the surface embedded in the 3D space, and enables these operations to be performed as easily as if the surface is flat.

Ideally, the mapping between the triangulated surface and the planar triangulation should be isometric, preserving both angles and distances. Unfortunately, with the exception of developable surfaces, general surfaces are not isometric to the plane. Distortion has to be introduced during the parameterization. One choice is the conformal parameterization, which preserves angles and scales distance isotropicly at local regions. Conformal parameterization has several advantages: It is intrinsic, dependent only on geometry and independent of triangulation and resolution; it scales the geometry locally, so there is no local distortion on textures; it is numerically stable. Therefore, conformal parameterization is desirable and suitable for practical purposes.

Existing algorithms for conformal parameterization can only handle topological disks. The conformality can not be preserved along boundaries. Sophisticated schemes have been designed to segment the mesh and to freely move the boundary. So far, no previous work has achieved global conformality.

For global conformal parameterization, the major difficulties are:

- The parameterization should be global without any seams and singularities. But there are topological obstacles, for all vector fields have singularities on general surfaces.

- Every local region affects the entire surface. It is impossible to find a local conformal parameterization for a patch then extend the parameterization

conformally to cover the whole mesh. So the solution is not local and the parameterization should be solved globally.

- There are infinitely many global conformal parameterizations. In order to find all of them, we have to discover the group structure and find the generators of this group.

- In practice, meshes are used to approximate smooth surfaces. This approximation may introduce inaccuracy and instability for computations. It is difficult to verify if the mesh has a good quality and improve the mesh quality for this purpose.

- The problem itself is a nonlinear problem, and it is difficult to convert it to a pure linear problem. How to handle the non-linearity is a challenging problem.

- Meshes in real application have millions of vertices, faces. Efficiency is critical for processing these data sets.

1.2 Contributions

Under the supervision of Professor Steven Gortler and with collaboration with Hugues Hoppe, I develop a method to parameterize meshes with arbitrary topology to a single chart, and remesh it using regular sampling pattern, represent geometry as an image. This geometry representation may unify the processing methods for both geometry and image. By representing geometry as texture, the architecture of rendering hardware may be simplified and rendering efficiency will be improved.

Under the supervision of and with collaboration with Professor Shing-Tung Yau, I develop discrete global conformal theory completely in this thesis. We show that each smooth surface admits a triangulation with all acute angles and

also demonstrate that each cohomology class has a unique discrete harmonic one-form. Then we define wedge product, and discrete Hodge star operator. We also define the discrete holomorphic differentials, and prove their linear space is $2g$ dimensional. We introduce a systematic method to compute simplicial cohomology, harmonic 1-forms and holomorphic differentials. The methods introduced here are very general and can be applied to other problems in processing geometry. For example, the global conformal structure can be used to classify surfaces, find isometries etc. We introduce a practical algorithm to compute global conformal parameterization without any boundary. This avoids the complicated boundary handlings in previous approaches.

1.3 Overview

Overview for geometry images. Given a mesh M, we unwrap the geometry into a single disk. We cut open the mesh in an appropriate way, so that it can be unwrapped onto a simply connected domain on the plane. The topological complexity of the mesh is transformed into the way that the boundary vertices coincide. For the geometry image applications, the cutting method should be appropriate for both topologically operation and good quality parameterization. We locate the special dense regions and allow the cuts to reach these special regions. Then the mesh is parameterized onto this planar domain, and remeshed by using regular 2D array sample pattern. The parameterization minimizes the ununiformity of the sampling. This representation may unify geometry and image processing methods.

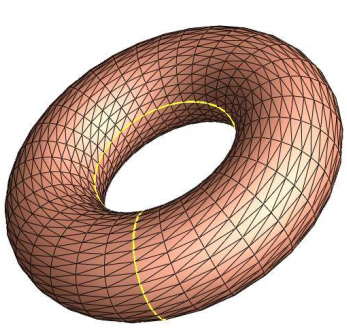
Overview for global conformal parametrization. In order to compute global conformal parameterizations, a special intrinsic natural structure, the *conformal structure* is introduced. Conformally equivalent surfaces share the same conformal structures, which are determined by geometry only, and the

mapping from the metric to the conformal invariants is continuous. So it is feasible to compute conformal structures numerically. This work introduces a systematic way to compute conformal structures and conformally map surfaces to other domains.

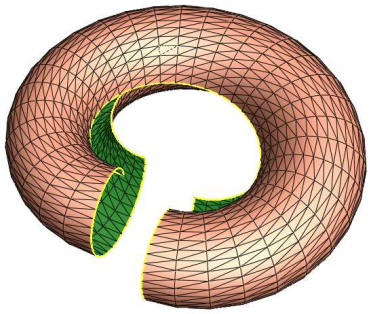
Any genus zero face can be conformally mapped to a sphere. The sufficient and necessary condition for the map to be conformal is that the map is harmonic. Harmonic mappings are the smoothest mapping between surfaces. It can be represented as an energy formula, which measures the length of gradients, and harmonic maps reach the critical points of this energy. In order to compute a conformal map between a genus zero surface to a sphere, one can simply optimize the harmonic energy defined for the mesh. Because the range is a sphere, the gradient field is constrained on the tangent spaces of the sphere during the optimization.

All conformal maps between two genus zero surfaces form a 6 dimensional group, the so called *Mobius Group*, which can be constructed mathematically. The solutions are not unique. Extra constraints are added to ensure the uniqueness of the solution.

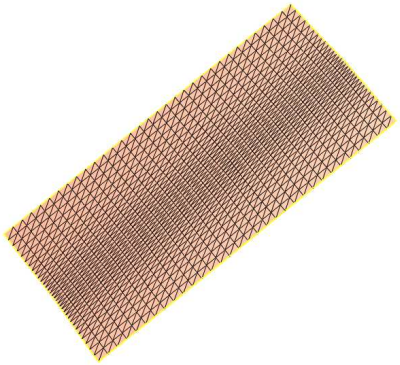
For non-zero genus surfaces, the conformal structure is much more complicated. Instead of examining conformal maps themselves, we first examine their gradient fields, which are the tangential vector fields. These vector fields form a linear space, the dimension of which is two times the number of the genus of the surface. We locate each handle of the surface, and open that handle to a square. The gradient fields of this map are diffused to be harmonic and orthogonalized in each tangent space. The obtained vector fields are the bases of that linear space. By integrating the linear combinations of these vector fields, one can compute the global conformal parameterization with arbitrary topologies. This parameterization has no charts, no boundaries, and is globally conformal everywhere. We use the torus case as an example to explain each step.



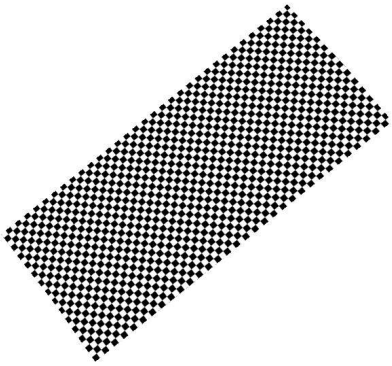
(a) Homology bases



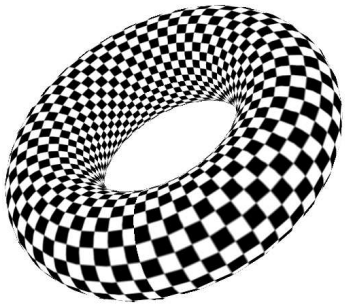
(b) Open handle



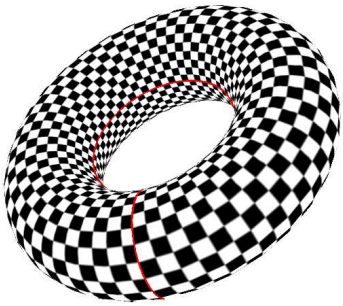
(c) Map to the plane



(d) Texture on the plane



(e) Texture mapping



(f) Conformality across cut boundaries

Figure 1: Processing the torus mesh

Example. This process is illustrated in figure 1. Suppose M is a mesh of genus one. The face set of M is $\{\sigma_0, \sigma_1, \ldots, \sigma_n\}$, the edge set is $\{e_0, e_1, \ldots, e_m\}$, and the vertex set is $\{v_0, v_1, \ldots, v_k\}$. We can define *patch space*, *curve space* and *vertex space* as linear spaces respectivly as follows:

$$C_2M = \{\sum_{i=1}^{n} c_i\sigma_i | c_i \in \mathbb{Z}\}, C_1M = \{\sum_{i=1}^{m} c_ie_i | c_i \in \mathbb{Z}\}, C_0M = \{\sum_{i=1}^{k} c_iv_i | c_i \in \mathbb{Z}\}.$$

They are representations of patches, curves, and points on M. Then we can define *boundary operator* $\partial_2 : C_2K \to C_1K$ intuitively as

$$\partial_2\sigma = [v_0, v_1] + [v_1, v_2] + [v_2, v_0], \sigma = [v_0, v_1, v_2].$$

Similarly for $\partial_1 : C_1K \to C_0K$,

$$\partial_1 e = v_1 - v_0, e = [v_0, v_1].$$

Boundary operators are linear on the patch space, curve space and vertex space. The null space of ∂_1 is the set of all closed curves. The image space of ∂_2 is the set of all the boundaries of patches on M. It is obvious that all the boundaries are closed. The quotient space

$$H_1(M) = \frac{ker\partial_1}{img\partial_2}$$

is the set of special curves. All the boundary operators are represented as linear matrices, so it is easy to find the bases of $H_1(M)$ by linear algebra techniques. Figure 1 (a) shows two such cycles on the mesh. We denote them as e_0 and e_1.

Then we cut the torus open along e_0, e_1 and map the patch to a unit square $[0, 1] \times [0, 1]$, boundary to boundary, corner to corner. This step is shown in figure 1 (b) and (c). Suppose the map is $\mathbf{f} = (f_x, f_y)$, we get the gradient field of f_x and f_y, denoted as ω_0, ω_1. They satisfy

$$\int_{e_i} \omega_j = \delta_i^j, i, j = 0, 1. \tag{1}$$

where δ_i^j is the Kroneck symbol. We represent ω_i as a function defined on edges, such that

$$\omega_i([v_0, v_1]) = f_i(v_1) - f_i(v_0).$$

Then ω_0, ω_1 is a basis of the so called *cohomology* of M, denoted as $H^1(M)$.

Next step, we want to diffuse ω_0, ω_1 to be harmonic, and preserve the above relation 1. We can add another gradient field δF_0 to ω_0, such that $\omega_0 + \delta F_0$ is harmonic. F_0 is defined on vertices of M, then

$$(\omega_0 + \delta F_0)([v_0, v_1]) = \omega_0([v_0, v_1]) + F_0(v_1) - F_0(v_0)$$

By adjusting $F(v_i)$, we can minimize the harmonic energy defined as

$$\sum_{e_i \in M} k_{e_i} ||(\omega_0 + \delta F)(e_i)||^2.$$

Then we assign $\omega_i \leftarrow \omega_i + \delta F_i$. Next step, we want to locally rotate ω_i by a right angle about the normal at each point on M. This can be done by solving a special linear system. We denote the rotated result as ${}^*\omega_i$, then

$$\omega_i + \sqrt{-1}\,{}^*\omega_i, i = 0, 1$$

are the bases of all gradient fields of conformal mappings.

In order to visualize this complex vector field, we use the texture mapping technique. We integrate this complex gradient field over mesh M, and get a planar region as shown in figure 1(c). Then we put a regular checker board pattern over this domain in (d), map the texture back to M, as shown in (e). It is clear that all the right angles for each square are preserved, each of them in the checker board is mapped to a square. The planar region might not be aligned with the checker board image, the texture pattern along the left boundary may be different from that along the right boundary. So the texture is discontinuous across the cut boundaries on the mesh. But the conformality is kept across the

cut boundaries, the stretching factor and the direction of iso-u,iso-v lines are consistent, which is shown in (f). □

The closed curve space is described as *homology group*, the gradient fields ω_i's are represented as *cohomology*. Those gradient fields with the minimum energy are called *harmonic 1-forms*. Local rotation is described as *Hodge star* operator. The gradient fields of conformal maps are called *holomorphic 1-forms*.

In this work, we rigorously define the harmonic 1-forms, holomorphic 1-forms for simplicial complexes and prove their existence, uniqueness, and give the dimensions of their linear spaces. This global discrete conformal theory is completely established in this work.

Using piecewise polygonal models to approximate smooth surfaces, especially for the purpose of discrete harmonic analysis, special triangulations are preferred. In this work, we prove that for any smooth surface, there exists a triangulation with all acute angles. This triangulation is important for computing discrete conformal structures.

The entire computation process can be represented as the diagram in figure 2. In order to locate handles, the homology group is computed explicitly. Then we compute the dual basis of cohomology group. We diffuse the cohomology basis to be harmonic, and locally rotate them, pair them to a holomorphic one-form basis. By linearly combining the bases, we can obtain any holomorphic one-form of conformal maps of the surface. Then intgeration of a holomorphic one-from on a fundamental domain produces a global conformal parametrization.

1.4 Motivating Applications

Parameterization and geometry images have many important applications in computer graphics. The following are some examples of them.

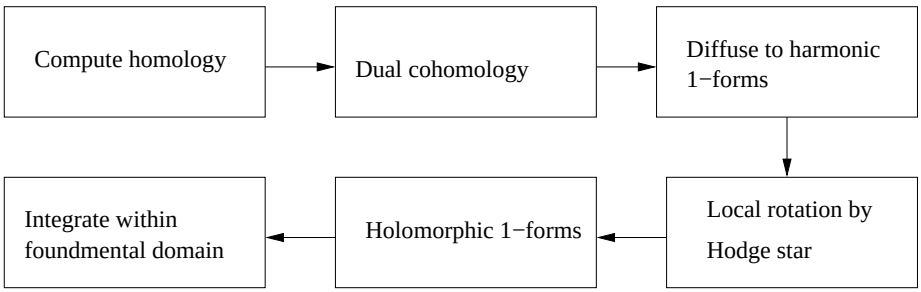


Figure 2: Process for computing global conformal parameterization

1.4.1 Efficient Hardware rendering

Using geometry images to represent meshes, the connectivity is implicit, and the texture coordinates are implicit. Therefore, in the rendering hardware, connectivity buffer and texture coordinates buffer are unnecessary. Because the connectivity pattern is completely regular, no complicated vertex caching scheme is needed. This will save the bandwidth between graphics rendering hardware and the main memory. These factors will improve the efficiency for hardware rendering.

1.4.2 Texture Mapping

A $3D$ paint system makes it possible to enhance the visual appearance of a $3D$ model by interactively adding details to it, such as colors, normals, reflections etc. If the discretization of the surface is fine enough, it is possible to directly paint its vertices. However, in most cases, the desired precision for the colors is finer than the geometric details of the model. Assuming that the surface to be painted is provided with a parameterization, it is convenient to use texture mapping to store colors in the parameter space, and virtually glue the texture image to the model. Texture mapping is at the heart of modern computer graphics rendering techniques.

1.4.3 Remeshing

Geometric models are always represented as irregular meshes. By mapping them to the plane, one can resample the geometric surface description using different planar pattern for different purposes. In order to improve the efficiency for hardware rendering, one can use regular planar sampling pattern; in order to do finite element computations, equilateral triangulations are preferred; in order to filter the geometry, the geometry can be treated as signals over planar domain through the parameterization.

1.4.4 Compression

Geometric models can be represented as functions on a parameter domain, then by using wavelet tools, they can be compressed. By using parameterization, they can be remeshed as subdivision surfaces, and using wavelets defined on these subdivision domain, they can be compressed.

1.4.5 Visualization

By selecting a special parameterization, *conformal parameterization*, a surface can be mapped to the planar domain conformally. Conformal mapping preserves angles, and is described as *similarities in the small*. Locally, shapes are preserved and distances and areas are only changed by a scaling factor. With complicated geometry flattened out, it can be better analyzed by a human viewer.

1.4.6 Shape Analysis

Geometric surfaces can be classified by conformal transformation group. This group is coarser than the isometric group and refiner than topological equivalence and suitable for many applications. Furthermore, by intrinsicly parametrizing surfaces to a canonical domain, one can easily compare them.

1.5 Overview of Thesis

This section introduces the motivation and the central problems solved in this work. Next section will review previous work in parameterization and conformal structure. In section 3, the algorithm for creating geometry images will be explained in details. Section 4 focuses on the theoretic background for global conformal parameterization. Discrete harmonic analysis concepts will be introduced and the theorems of holomorphic differentials, Laplacian spectrum will be proven rigorously. Section 5 explains the method to construct a conformal mapping from a genus one surface to a sphere. Section 6 goes through the algorithms to compute the bases of holomorphic differentials. Section 7 applies the result to parametrize non-zero genus surfaces. Section 8 discusses the advantages and disadvantages of these methods. Future research directions are elaborated in the last section.

The research on geometry images is supervised by Professor Steven Gortler. The research on global conformal parameterization and discrete Riemann geometry is completed under Professor Shing-Tung Yau's supervision.

2 Previous Work

This section will review previous works related to parameterization and computing conformal structures in the fields of computer graphics and mathematics.

2.1 Parameterization

The theory on graph embedding has been studied by Tuette [44], where barycentric maps are introduced. Given a topological disk, the boundary is mapped to a convex polygon on the plane. Tutte embedding maps each vertex to the barycentric center of its neighbors. The solution exists and the bijectivity of this parameterization is guaranteed.

Floater [13] uses a specific weight to improve the quality of the mapping in terms of area deformation and conformality. His method is shape preserving in the sense that for a planar convex mesh the parameterization is affine invariant.

Maillot et al. [30] introduce a deformation energy to measure the distortion introduced by the mapping. Basically they measure the edge length deviations and the area differences. For isometry, the deformation energy is zero. The energy is optimized using the conjugate gradient method. The weighting between edge springs and area-preservation terms must be adjusted to produce an embedding.

Levy and Mallet [27] define a metric as a combination of orthogonality and isoparametric terms. The whole optimization is non-linear. In order to solve it, they iteratively fix one texture component and solve for the other using linear optimization. The solution converges and is unique. The continuity is preserved across the boundary.

Sheffer et al. [40] introduce a angle based flattening method to flaten a mesh to a planar plane so that it minimizes the relative distortion of the planar angles with respect to their counterparts in the three-dimensional space.

Eck et al. [9] introduce the discrete harmonic map. The harmonic energy is represented as a string energy, which measures the norm of gradient of the parameterization. Sometimes, the string constants are negative and the mapping has flipped faces.

Levy et al.[28] compute quasi-conformal parameterizations of topological disks by approximating the Cauchy-Rieman equation using the least square method. They design the segmentation in such a way that the chart boundaries are along geometric features. They show rigorously that the quasi-conformal parameterization exists uniquely, is invariant by similarity, independent of resolution and preserves orientations.

Mathieu et al. [1, 4] compute the discrete Dirichlet energy on triangulations

to achieve discrete conformal parameterization, and apply this parameterization for interactive geometry remeshing. They also prove that this method is equivalent to the one based on approximating the Cauchy-Riemann equation.

Sander et al. [36] develop a texture-stretch metric to minimize texture stretch and texture deviation. Furthermore, Sander et al. [35] design the signal-stretch parameterization metric to measure the signal error. This metric is defined as a tensor on the surface. In order to minimize this nonlinear metric, they use a coarse-to-fine hierarchical solver, followed by fine-to-coarse propagation of the integrated metric tensor.

All of the above methods are in the same framework. A special energy is defined to reflect the deviation of the parameterization from isometry. Then algorithms are used to minimize the energy.

Our method is quite unique, because

1. It handles meshes with arbitrary topologies. All of the above methods deal with genus zero meshes only.

2. Our solution is boundary free. The conformality is preserved across the boundary.

3. We find all the solutions to minimize the objective functional, instead of just finding one solution.

4. Instead of finding the minimum directly, we compute the structure of the solution space.

2.2 Remeshing

There exist several schemes for semi-regular remeshing of arbitrary surfaces.

Eck et al. [8] achieve remeshing by cutting a mesh into multiple charts using a Voronoi-like decomposition. Each chart is parametrized using a harmonic

map, sampled using a regular triangular subdivision pattern, and compressed using a triangular wavelet construction [29].

Khodakovsky et al. [23] use the MAPS scheme [26] to partition the mesh into charts and create the chart parameterizations. They obtain impressive compression results using zero-tree coding of local-frame wavelet coefficients.

Lee et al. [25] create a multi-chart domain using mesh simplification. They define a subdivision surface over this domain and fit it to the original surface. The fit residual is expressed as a semi-regular scalar displacement map over the smooth subdivision surface.

Guskov el al. [19] use a MAPS-like approach to create multiple charts. These charts are recursively subdivided, and newly introduced vertices are expressed using displacements from the previous mesh, mostly as scalar displacements.

In our setting of geometry images, previous semi-regular remeshing approaches can be viewed as representing a surface as a collection of abutting geometry images. The crux of our contribution is to represent the entire surface as a single geometry image, by cutting the surface and sampling it using a completely regular quad grid. We optimize the creation of the cut to allow for a good parameterization.

2.3 Global Conformal Parameterization for Genus Zero Surfaces

Haker et al. [20] introduce a method to compute global conformal mappings from a genus zero surface to a sphere. First a set of conformal coordinates for the surface is chosen, and the sphere is mapped to the complex plane using the stereo-graphic projection. Then the map is defined from the complex plane to the complex plane. Because this map is a conformal map from $\mathbb{C}$ to $\mathbb{C}$, it is in the Mobius group, and has the format of a linear rational. Then the Laplacian operator can be formulated explicitly. The differential equation is solved by the

finite element method.

In this work, a new approach is introduced, which is based on minimizing the harmonic energy. While Haker's method is linear, ours is nonlinear. Haker's method suffers from the following limitations: The stereo-graphic projection is nonliner, using piecewise linear mapping to approximate it will introduce inaccuracies. For the regions near the north pole, this problem is very severe and can cause instability for the computation. For example, when we tested it using the bunny model, Haker's method did not converge. We do all the optimization in the tangent spaces of the sphere, there is no stereo-graphic projection. Hence our method is more stable.

Circle packing [42] has been studied intensively. A circle packing is a configuration of circles with a specified pattern of tangencies. Maps between circle packings which preserve tangency and orientation act in many ways as discrete analogues of analytic functions. Moreover, classical analytic functions and more general classical conformal objects can be approximated using circle packings. But in general surfaces in $\mathbb{R}^3$, the circle packing method only considers the connectivity without geometry, so it is not suitable for our parameterization purpose.

2.4 Global Conformal Parameterization for Surfaces with Arbitrary Topologies

In my paper with Professor Yau [16], a systematic method to compute global conformal structures of closed surfaces is introduced. We approximate the De Rham cohomology by simplicial cohomology and represent the Laplace-Beltrami operator, the Hodge star operator by linear systems. A basis of holomorphic one-forms is constructed explicitly. We then obtain a period matrix by integrating holomorphic differentials along a homology basis. We also study the global conformal mappings between genus zero surfaces and spheres, and between gen-

eral surfaces and planes.

In another paper with Professor Yau [17], we introduce a method to compute global conformal parameterizations for surfaces with arbitrary topologies without partitioning surfaces to topological disks. The method can be applied to surfaces with or without boundaries. Conformality of the parameterization is preserved everywhere except for very few points depending on the genus number of the surface, and there is no boundary of discontinuity. We analyze the structure of the solution space and find all possible solutions instead of finding just one solution.

2.5 Computational Topology

The process of slicing a mesh open of arbitrary topology has been studied in the computational literature.

Lazarus et al. [45, 24] introduce an incremental method to compute a canonical polygonal schema of an oriented mesh. The method is based on Morse theory. The mesh is constructed by adding faces one by one along the current boundary. If the topological type of the boundary is changed, the merging or splitting vertices is recorded. The "gradient" curves connecting these critical vertices are on the boundary of the schema.

Dey et al. [6, 5] compute the polygonal schema and use it to verfy if two closed curves are homotopic equivalent on the surface.

Optimally cutting a mesh of arbitrary genus into a disk is studied in [10]. Erickson and Han-Pelled prove that it is NP-hard to minimize either the total number of cut edges or their total length when cutting a set of edges on a polyhedral manifold surface to obtain a single topological disk.

2.6 Conformal Structure

Conformal structure has been studied in the computational Riemann Geometry literature. In mathematics, Riemann surfaces, algebraic curves and Jacobian varieties are objects represented in different categories. Some research has been done in order to convert one to another and study conformal invariants. [38, 33] Sepala et al. introduce a method to compute period matrices on the Riemann surfaces represented as algebraic curves.

These methods deal with the surfaces defined implicitly. In our application, we handle surfaces represented as meshes, so their methods are not suitable for our purpose.

2.7 Spectral Compression

Fourier analysis has been studied extensively in the signal processing fields. Frequency spectrum can be used to compress signals. The Laplacian eigenfunctions form a basis of the function space defined on a surface, so each function can be decomposed as a linear combination of these eigenfunctions. The spectrum can be applied for compression purposes. In [22], the spectral methods are applied to 3D mesh data to obtain compact representations. This is achieved by projecting the mesh geometry onto an orthonormal basis derived from the mesh topology. The mesh is partitioned into a number of balanced submeshes with minimal interactions, and each is compressed independently. The eigenfunctions are derived from the topological Laplacian operator.

3 Geometry Images

This section describes a novel geometry representation method, *Geometry Images*. The basic idea is to topologically transform a surface to a genus zero disk, and embed the disk in a canonical planar domain, resample the geometry us-

ing regular pattern. This representation unifies geometry and image, the image processing techniques can be applied to geometry directly. Especially, this representation simplifies the architecture of graphics hardware, and the improves the efficiency of rendering process.

3.1 Creation of Geometry Images

From a 2-manifold triangle mesh M, we create a geometry image consisting of an $n \times n$ array of $[x, y, z]$ data values. If we plan to render using normal mapping, we also create another 2D array of normal values $[n_x, n_y, n_z]$. (See Figure 8.)

Our approach is to cut the mesh M to form a new mesh M' that has the topology of a disk (Figure 3). The cut ρ is specified as a set of edges in M. To create M', we split each non-boundary edge in ρ into two boundary edges to form the *opened cut* ρ'. This directed loop of edges ρ' is the boundary of M'. We say that two edges in ρ' are *mates* if they result from the splitting of an edge in ρ.

A vertex v with valence k in ρ is replicated as k vertices in ρ'. Vertices in ρ that have valence $k \neq 2$ in the cut are called *cut-nodes*. (We still refer to these as cut-nodes when replicated in ρ'.) A *cut-path* is the set of boundary edges and vertices between two ordered cut-nodes in the loop ρ'. Each cut-path has a mate defined by the mates of its edges (unless its edges were boundary edges in ρ).

Let D be the domain unit square for the geometry image. The parameterization ϕ is a piecewise linear map from the unit square D to M', defined by associating domain coordinates (s, t) with each mesh vertex in M'. The domain D has a rectilinear $n \times n$ grid, where *grid points* have coordinates $(i/(n-1), j/(n-1))$ with $i, j = 0..n-1$. We evaluate ϕ at the grid points to sample the mesh geometry, as well as any other surface attributes (e.g. color, skinning weights, radiance

transfer coefficients).

The geometry image samples are used to reconstruct an approximation of M. In this work, we use linear basis functions (triangles) to define the reconstruction interpolant for geometry. Our goal is to find a good cut ρ and parameterization ϕ, such that this reconstruction is a good approximation of M for moderate sampling rates.

Approach overview Our strategy for finding a good cut ρ and parameterization ϕ is as follows. We first find a topologically sufficient cut, and create an initial parameterization using this cut. We use information from the parameterization to improve the cut, and reparameterize based on the new cut. This process of cutting and reparametrizing is iterated until the parameterization no longer improves. To aid in the exposition, we first describe how a parameterization is found given any cut ρ (Section 3.1.1). We then describe how the space of cuts is explored (Section 3.1.2).

3.1.1 Parameterization

For now, assume that we are given a cut ρ. To create a parameterization, we first fix a mapping between the opened cut ρ' and the boundary of the unit square D. Next, we solve for a map of M' onto D that is consistent with these boundary conditions. We now describe these two steps in more detail.

Boundary parameterization In order to avoid cracks in the reconstructed geometry, it is necessary that each cut-node in ρ' be exactly sampled in the remesh. This implies that we must map cut-nodes to grid points on the boundary of D. (Other vertices in ρ' are not constrained to lie on grid points.) In addition, cut-path mates must be sampled at identical surface points to avoid cracks, which requires that cut-path mates be allocated the same length on the boundary of D. To accomplish this, we allocate for each cut-path an amount of

the boundary proportional to its length in ρ'. This allocation is then rounded to an integer multiple of $1/(n-1)$. If due to rounding we have over- or under-allocated the boundary, we redistribute the residual to the various cut-paths in units of $1/(n-1)$, making sure to treat cut-path mates identically. Note that an $n \times n$ geometry image can represent a surface with genus at most n.

To avoid degeneracies, we must enforce two more constraints. First, no triangle in M' can have all its three vertices mapped to one of the four sides of the square, for it would become parametrically degenerate. If such a triangle arises, we split the triangle by introducing new vertices at the midpoints of its non-boundary edge(s), and split neighboring triangles so as to avoid T-junctions.

Second, as we lay out ρ' along the boundary of D we must break any edge that spans one of the four corners of D. Otherwise a single boundary edge in M' would map to an "L" shape in D. The edge is broken by introducing a vertex at the domain corner, thus splitting its adjacent triangle into two. To enforce topological consistency across the cut, the same procedure is applied to its mate edge.

Finally, we find that placing a valence-1 cut-node at a corner of D results in poor geometric behavior, so if this occurs we rotate the boundary parameterization.

Interior parameterization Having fixed the boundary of the parameterization, we now solve for its interior. When creating a parameterization, there are numerous metrics that can be used to measure its quality, e.g. [8, 12, 20, 26, 34]. For our application, an ideal metric would be some measure of surface accuracy after sampling and reconstruction. As shown by the analysis in [35], the L^2 geometric-stretch metric introduced in [36] is in fact an approximation of this ideal measure.

Geometric stretch measures the amount of spacing that occurs on the surface

when the parameter domain is uniformly sampled. Thus, minimizing geometric stretch tends to uniformly distribute samples on the surface. In [35], the stretch metric is shown to be related to signal-approximation error (SAE) — the difference between a signal defined on the surface and its reconstruction from a discrete grid sampling. Specifically, the stretch metric corresponds to the first-order Taylor expansion of SAE under the assumption of locally constant reconstruction. In our context, the signal is the geometry itself, and therefore geometric stretch can be seen as a predictor of geometric reconstruction error. In Section 3.3, we show the advantage of using a geometric-stretch parameterization over the Floater "shape-preserving" parameterization.

We compute a geometric-stretch parameterization using the hierarchical optimization algorithm described in [35]. First, the interior of M' is simplified to form a progressive mesh representation [21]. The few interior vertices in the resulting base mesh are optimized within D by brute-force. Then, we apply vertex splits from the progressive mesh to successively refine the mesh. For each inserted vertex, we optimize the parameterization of its neighborhood to minimize stretch using a local, non-linear optimization algorithm.

3.1.2 Cutting

We now describe how we automatically find a good cut ρ for M. Starting with a surface of arbitrary genus, we first find an initial cut that opens M into a disk. Given the resulting topological disk, we use a novel algorithm to augment the cut in order to improve the subsequent parameterization and reconstruction quality.

Initial cut It is well known that any closed surface can be opened into a topological disk (called a polygonal schema) by cutting along an appropriate set of edges [31]. Such a cut was used in [11] as part of a geometric modeling

system for creating smooth surfaces. Piponi and Borshukov [34] describe an interactive system allowing a user to manually cut a genus-zero manifold into a single chart using a tree of edge cuts.

The computational complexity of optimally cutting a mesh of arbitrary genus into a disk is studied in [10]. Algorithms for finding special kinds of cuts (those that form reduced and canonically reduced polygonal schemata) are described in [6, 24, 45].

Our algorithm, which is most similar to that of [6], works as follows. If the mesh has boundaries, let B be the set of original boundary edges. This set remains frozen throughout the algorithm, and is always a subset of the final cut ρ. After removing a single seed triangle from the mesh, we apply two phases.

In the first phase we repeatedly identify an edge $e \notin B$ adjacent to exactly one triangle, and remove both the edge and the triangle. Note that the two remaining edges of the triangle are left in the simplicial complex, even if they are dangling. In order to obtain a result of "minimal radius", we order triangle removals according to their geodesic distance from the seed triangle. When this first phase terminates, we have removed a topological disk that includes all of the faces of the mesh. Thus, the remaining vertices (which is in fact all of them), and the remaining edges must form a topological cut ρ of M. At this point, ρ consists of a set of connected loops along with some unnecessary trees of edges (and is similar to the construction of [43]).

In a second phase, we repeatedly identify a vertex adjacent to exactly one edge (i.e. a dangling edge), and remove both the vertex and the edge. This second phase terminates when all the edge trees have been trimmed away, leaving just the connected loops. Since the resulting cut ρ may be serrated (it is not made up of shortest paths), we straighten each cut-path in ρ by computing a constrained shortest path that connects its two adjacent cut-nodes and stays within a neighborhood of the original cut-path.

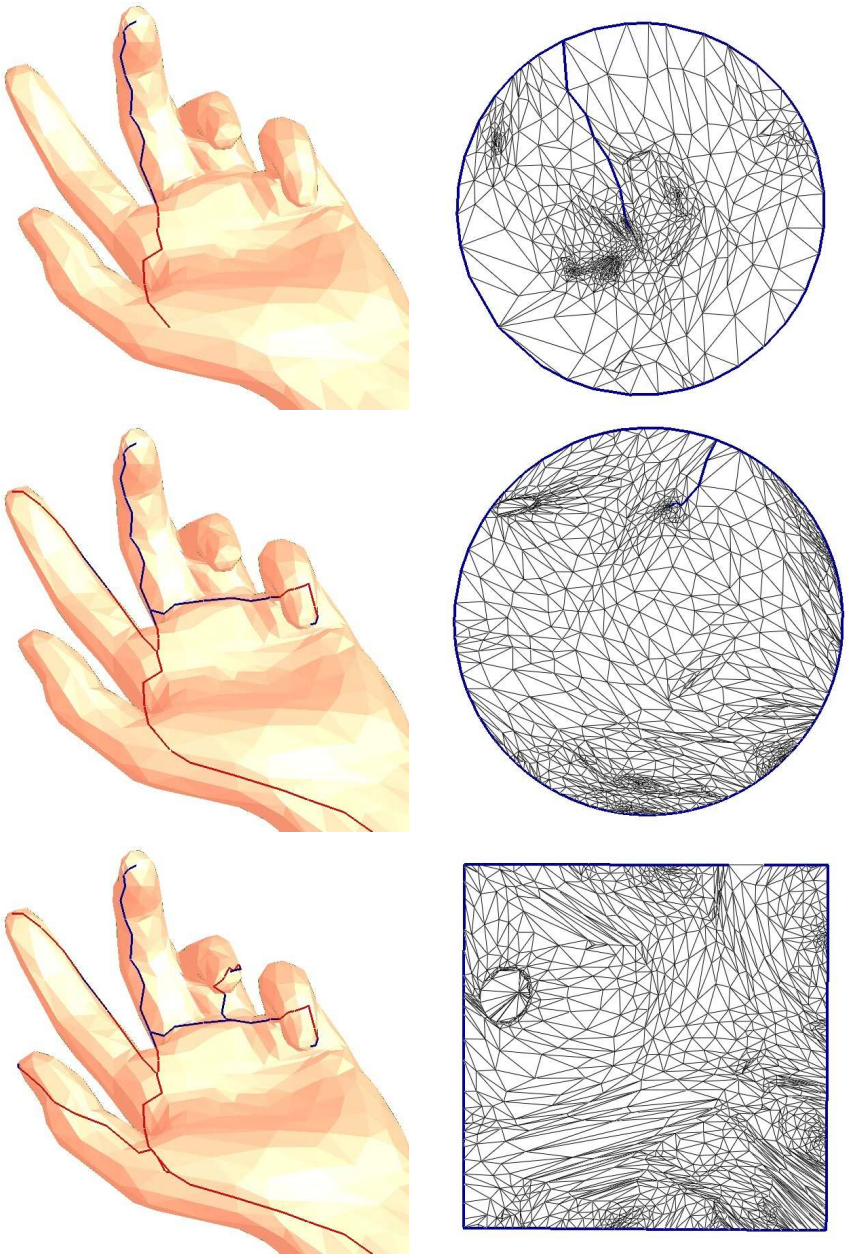


Figure 3: Columns (a–d) show iterations of the cut improvement algorithm. Upper images show the mesh M with the current cut ρ (blue except red where occluded). Bottom images show the Floater parameterization (over circle) of the corresponding M', together with the shortest path to an extremal point,which will be added to ρ. Column (e) shows the final cut ρ and the geometric-stretch parameterization (over square).

For the case of a closed mesh of genus 0, the resulting ρ will consist of a single vertex, since it has no loops. Because our parameterization requires that we map ρ' onto a square, we add back to ρ two adjacent mesh edges.

Iterated cut augmentation Through experiment, we have found that to obtain efficient geometry images, it is important for ρ to pass through the various "extrema" of M. For example, in the hand model a good cut should pass through its five fingers (see Figure 3). Therefore our goal is to find these extrema and augment the cut so that it passes through them. A similar subproblem is investigated by Sheffer [39], who classifies extrema as vertices with high (discrete) curvature. Unfortunately, this type of local method will not be able to find protrusions with widely distributed curvature.

Our approach to finding extrema is to search for mesh regions that behave poorly (have large geometric stretch) under a parameterization using the current cut. Specifically, we map the vertices of ρ' to the unit circle C, spaced according to their edge lengths over the surface. (We use the unit circle at this point instead of the unit square in order to avoid boundary constraints.) The cut mesh M' is parametrized into the interior of C using the shape-preserving parameterization of Floater [12]. Given the resulting map we identify the triangle with maximum geometric stretch, and pick one of its vertices as an extremal vertex.

The intuition for this method is that any protrusion of the mesh experiences high geometric stretch under a Floater parameterization. For instance, it can be shown that when parametrizing a tube closed at its top and open at its base, a triangle at a height h from the base has geometric stretch exponential in h, reaching a maximum at the tube apex. It is important to use the Floater parameterization for protrusion detection, since the geometric-stretch parameterization would evenly distribute stretch, thus hiding the extrema.

Having identified an extremal point, we find the shortest path from it to the current boundary of M' (measuring distance on the mesh), and add this path to ρ. This maintains the invariant that ρ is a valid cut of M.

We repeatedly apply this augmentation process, as shown in Figure 3. To determine when to stop, we run our geometric-stretch parameterization algorithm (Section 3.1.1) after each cut, and stop if the geometric stretch increases.

As a further improvement in the case of genus-zero meshes, when we find the *first* extremal point, we discard the original cut, which was based on an arbitrary random seed point, and replace the cut with a pair of adjacent edges at this extremum.

3.1.3 Topological sideband

A geometry image is a parametric sampling of the topological disk M'. Its reconstruction looks like M because its boundary vertices coincide geometrically. For some applications though, it is important to be able to "fuse" the boundary of D so that it has the original topology of ρ. This fusing could be achieved by searching for geometric correspondences on the image boundary, but this process might be error-prone, particularly if the geometry image undergoes lossy compression.

Since the necessary topological cut information is extremely compact, we record it into a sideband signal as follows. We associate a pair of labels e.g. $\{a, \overline{a}\}$ to each cut-path and its mate. We then store the string of labels corresponding to the sequence of cut-paths on the boundary of M', e.g. $ab\overline{a}\overline{b}c\overline{c}$. From this string, we can recover the topology of the cut, i.e. the valence k of each cut-node in ρ and the ordering of the cut-nodes along ρ'. We also store for each cut-path a its discretized length on the boundary of domain D, and we store the starting boundary location of the first cut-path. From this topological and parametric information, we can later establish the correspondence of all boundary grid

vertices.

The size of this sideband information is $O(q \log n)$ bits, where q is the number of cut-paths and n is the sampling rate over D. For our models, q ranges between 3 and 10, and the sideband is approximately 12 bytes long.

3.2 Applications

Rendering To render geometry images on current hardware, we span each 2×2 quad of grid points using two triangles, by splitting along the shorter of the two diagonals.

Level-of-detail rendering is implemented by mip-mapping the geometry image, as shown in Figure 6. In order to avoid cracks at multiple levels of details, we use geometry images of size $(2^j + 1) \times (2^j + 1)$, and minify using simple sub-sampling. Also, the boundary mapping ϕ of Section 3.1.1 is constructed to place cut-nodes to grid-points of the lowest intended resolution (65×65 for all of our examples). Unlike [34], our boundary samples coincide exactly across the cut so we need no special boundary treatment, even for mip-mapping.

For hardware that implements normal mapping, we also create a normal map using the exact same parameterization ϕ. Usually, we sample the normals into an image of higher resolution than the geometry since the normal-map signal tends to be more detailed. During rendering, the normal-map signal is rasterized over the triangles by hardware texture-mapping, using bilinear reconstruction of each quad in the normal map. (Texture coordinates at the vertices are assigned the range $[(0.5)/n', \ldots, (n'–0.5)/n']$ where n' is the texture resolution, for correspondence with the texture samples.)

Because geometry images have the same regular structure as texture images, one can envision hardware that would use bilinear (or even bicubic) basis functions to reconstruct the geometry. Moreover, the rendering process should be inherently simpler than with traditional texture mapping. The attribute

samples can be accessed in *scan order* rather than backward-mapped through random-access texture coordinates. Also, the attribute samples have a *regular correspondence* with the geometry samples, and therefore do not require general tri-linear interpolation lookup.

Both view-frustum and backface culling could be implemented in a unified setting by constructing hierarchies on the geometry image and the normal image respectively.

Compression and Decompression For compression we use the image-compression coder provided by Davis [3]. For decompression, we decode the wavelet coefficients to recreate an $n \times n$ grid of $[x, y, z]$ values. Our wavelet decoder produces floating-point coordinate values as output. Quantizing these values to 12-bit integers provides sufficient resolution for our models.

Since this wavelet coding is lossy, cut-path mates may be reconstructed differently, leading to cracks in the mesh (see Figure 8d). To address this problem, we also record and losslessly compress the topological sideband (Section 3.1.3). During decompression, we use this topological information to geometrically fuse the cut. We first determine the equivalence classes of boundary grid points. Most boundary grid point are paired up with a single other grid point, while grid points that sample a cut-node are grouped with $k-1$ other grid points, where k is the valence of the cut-node in ρ. We average together the $[x, y, z]$ values of equivalent grid points, and replace their data with this common average. We record the vector displacement added due to this averaging for later error diffusion.

This simple averaging scheme gives rise to a continuous surface, but can lead to unsightly steps in the reconstructed geometry near the cut. In order to smooth these steps, we apply a simple error diffusion technique, spreading the displacements towards the center of the square. The result of this fusing process

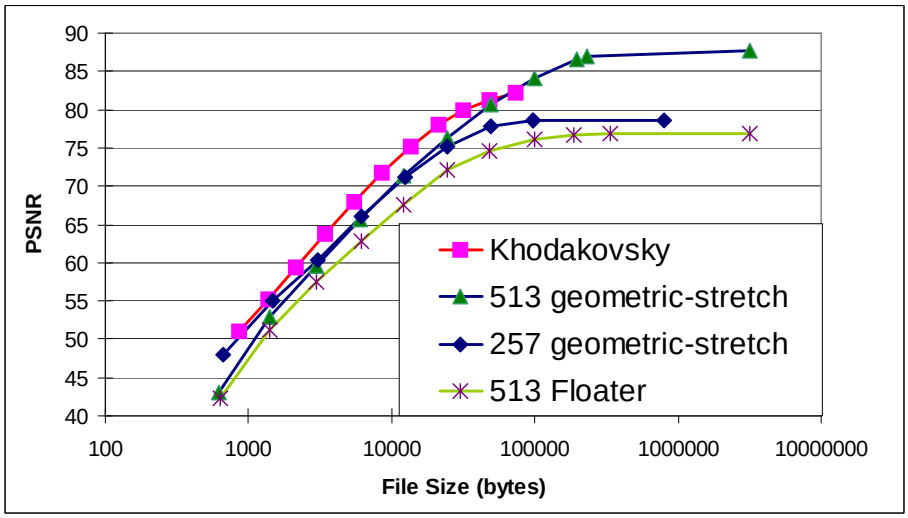


Figure 4: Rate distortion for geometric reconstruction from compressed geometry images of the bunny (at 257×257 and 513×513 resolutions, and using a Floater-parameterization), compared to [23].

is shown in Figure 8e.

3.3 Results

We have run our system on a number of high-resolution models, with and without boundaries. Uncompressed examples are shown in Figure **??**. These required about an hour to convert offline. The conversion bottleneck is the sequence of parameterizations in the iterated cut augmentation process. Currently, we set the geometry image resolution n manually (most often $n = 257$), but this parameter could be set automatically to achieve a desired accuracy.

Geometry images tend to be relatively smooth, and therefore provide opportunity for compression. Even simple image compressors will define basis functions that span the whole surface, and therefore allow high compression ratios. Figure 4 shows rate-distortion curves when using the image wavelet-coder of [3]. These curves measure the reconstruction accuracy for various compression rates applied to the geometry image. Error is measured as Peak Signal to

Noise Ratio PSNR $= 20 \log_{10}(\text{peak}/d)$, where peak is the bounding box diagonal and d is the symmetric rms Hausdorff error (geometric distance) between the original mesh and the reconstructed geometry. The blue curves show results for wavelet-compressed geometry image created using a geometric-stretch parameterization and two different sampling rates. The green curve corresponds to a geometry image formed using the same cut, but with a Floater parameterization, and is noticeably less efficient. For comparison, the red curve is the result of the compression scheme described in [23], which is more efficient by about 3dB. Reconstructions from compressed geometry images are shown in Figure 7.

3.4 Summary and Future Work

We have introduced geometry images, a completely regular representation for approximating the geometry of an irregular mesh. Geometry images can be easily rendered and compressed using current hardware and software. Due to their simplicity, we envision that geometry images may inspire new hardware rendering approaches.

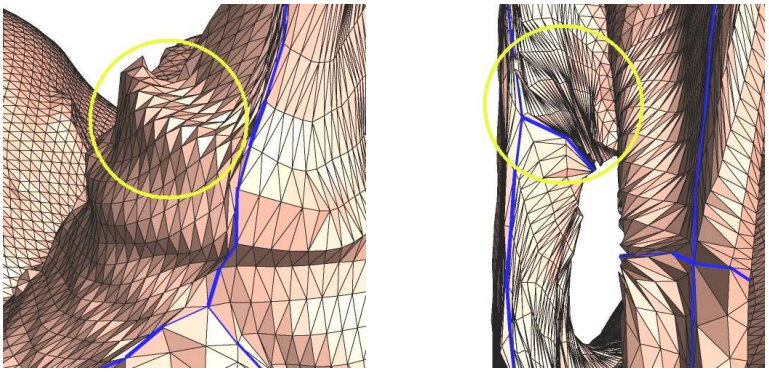
We have found that we can create efficient geometry images on a wide variety of models. However, models of high genus can be problematic. Such models may require long cuts to open up all the topological handles. In that case, much of the surface lies near the cut boundary, making it difficult to create a parameterization without significant geometric stretch and poor resampling.

Figure 5 shows examples of trouble areas in the remeshing of the Buddha model. Our genus-6 Buddha model was obtained by filtering out tiny topological handles from a genus-104 scanned model [48]; working directly on the genus-104 surface would have been impossible.

In general, remeshing techniques can have difficulty capturing sharp surface

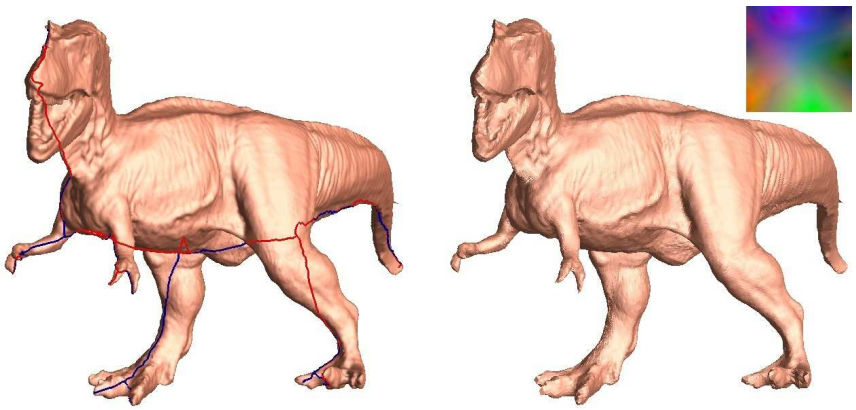
features accurately at low sampling rates. In semi-regular remeshing, one technique to improve accuracy is to make the chart boundaries correspond with the most significant features, so that the subdivided domain edges follow these features [26]. Another technique is feature-sensitive remeshing [46], which warps the parameterization as a post-process to align the remesh edges with the sharp surface features. When creating our geometry images, adding a pass of feature-sensitive remeshing could improve reconstruction results for meshes with sharp geometry.

Since we used off-the-shelf compression code, we did not explore the extra savings that could be obtained using local-frame detail representation [23]. Adding this to our system may improve compression efficiencies.



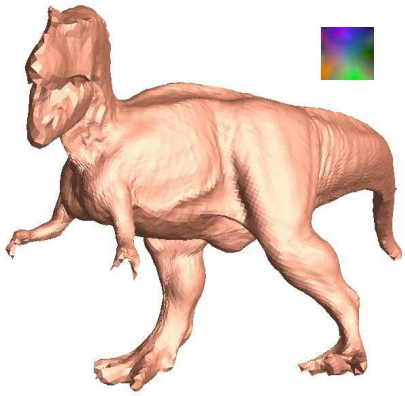
(a) 49 KB (b) 12 KB

Figure 5: Example artifacts in the Buddha geometry image: aliasing (jaggedness) near sharp features, and regions of high anisotropy.

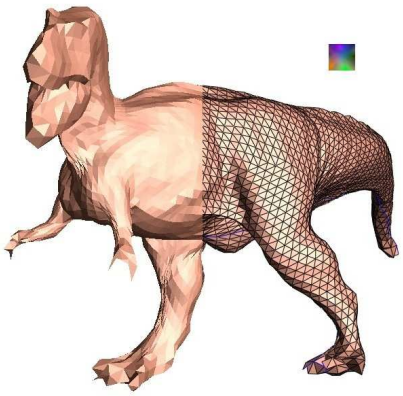


Original mesh (342K faces)

Geometry image (257×257)

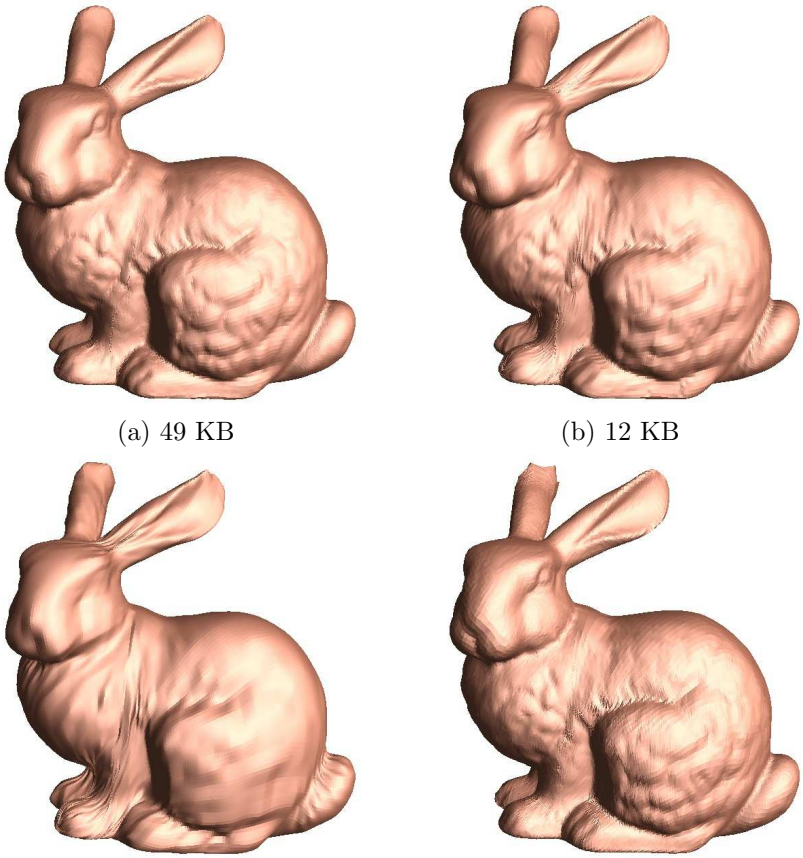


Mip-mapped (129×129)



Mip-mapped (65×65)

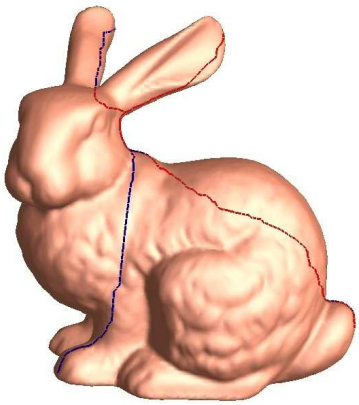
Figure 6: Mip-mapping a geometry image. As in all examples, the boundary parameterization is constructed for a 65×65 domain grid.



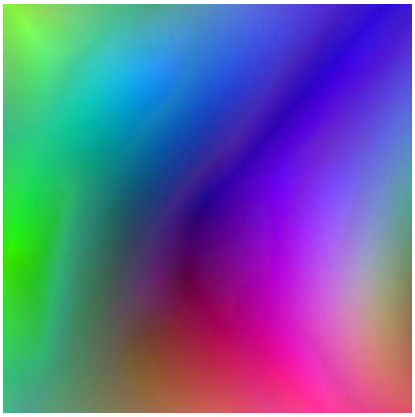
(a) 49 KB (b) 12 KB

(c) 3 KB (d) 49 KB

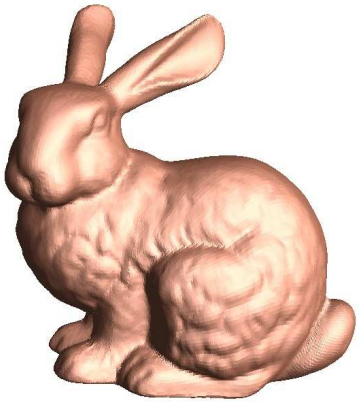
Figure 7: (a–c) Surfaces reconstructed from a 257×257 geometry image under increasing levels of wavelet compression. (d) Reconstructed from a 257×257 Floater-parametrized geometry image. All models are flat-shaded.



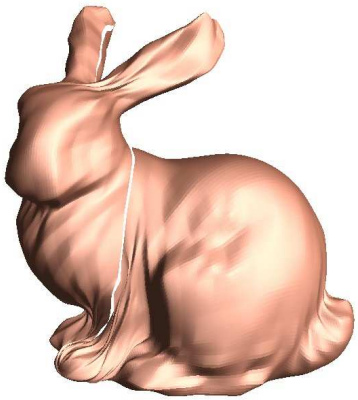
(a) Original mesh with cut
70K faces; genus 0



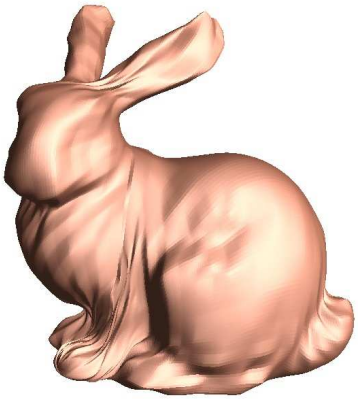
(b) Geometry image 257×257
(b∗) Compr. to 1.5KB (not shown)



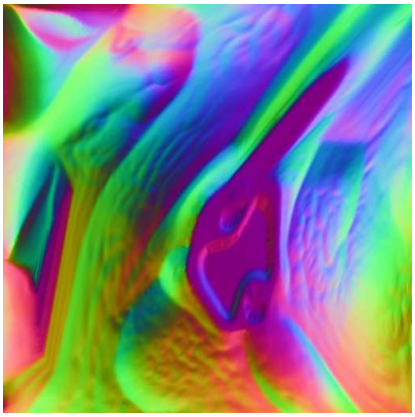
(c) Geometry reconstructed
entirely from b



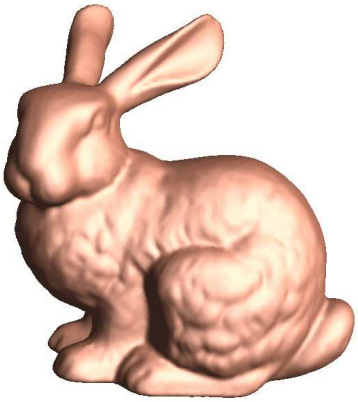
(d) Geometry reconstructed
entirely from b∗



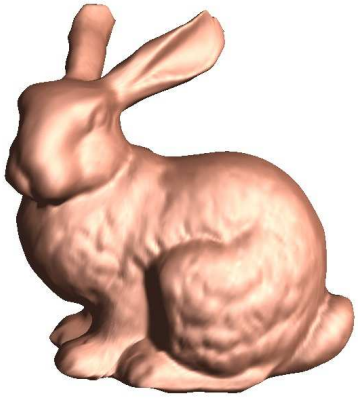
(e) Geometry of d topology-fused
(f∗) Compr. to 24KB (not shown)



(f) Normal-map image 512×512
normal-mapped using f



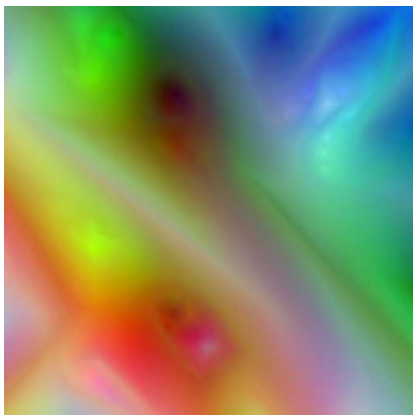
(g) Geometry of c



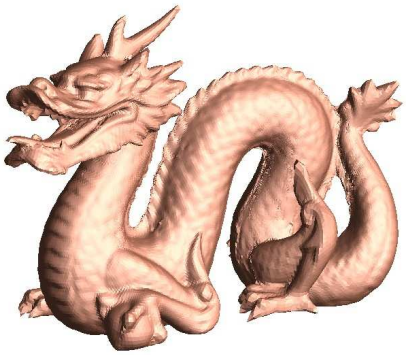
(h) Geometry of e
normal-mapped using f∗

Figure 8: Creation, compression, and rendering of a geometry image. Images b∗ and f∗ (not shown) are compressed using an image wavelet-coder. Geometry image is 12-bit $[x, y, z]$ visualized as $[r, g, b]$. Normal-map image is 8-bit $[n_x, n_y, n_z]$ visualized as $[r, g, b]$.

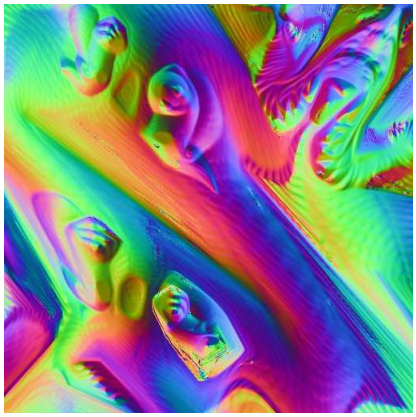
(a) Original (500K faces; genus 1)



(b) Geometry Image [257x257]



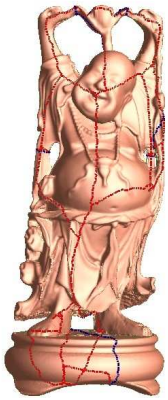
(c) Reconstruction (PSNR=66.8)



(d) Normal map (512x512)

(e) Normal-mapped reconstruction (b+c)

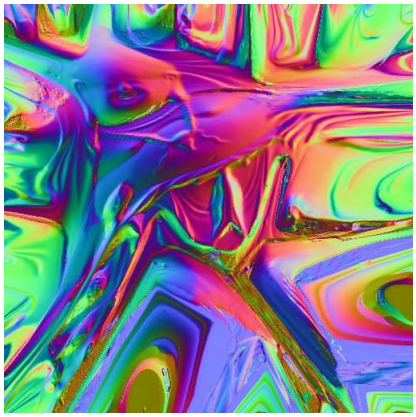
Figure 9: Dragon mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping.



(a) Original (500K faces; genus 6)

(b) Geometry Image (257x257)

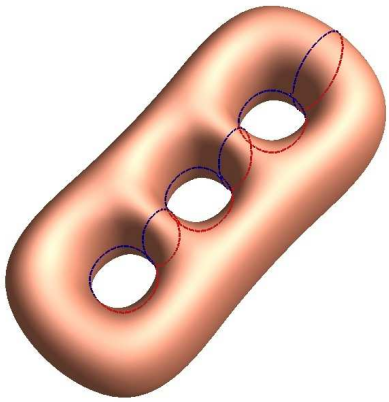
(c) Reconstruction (PSNR=64.9)



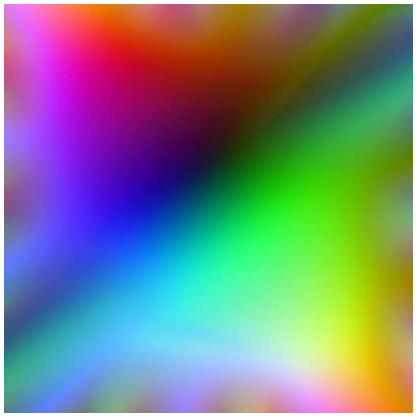
(d) Normal map (512x512)

(e) Normal-mapped reconstruction (b+c)

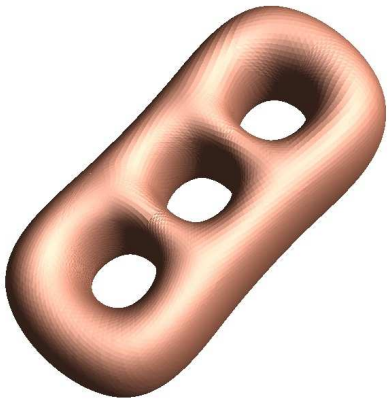
Figure 10: Buddha mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping.



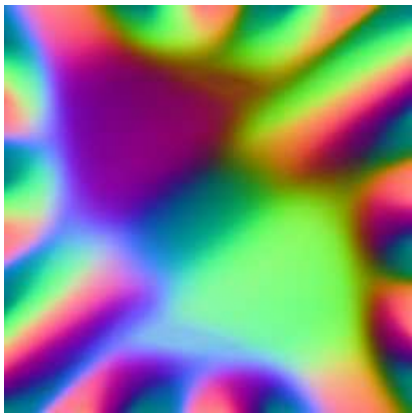
(a) Original (47K faces; genus 3)



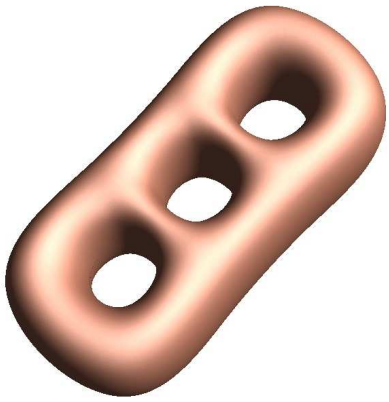
(b) Geometry Image (129x129)



(c) Reconstruction (PSNR=75.2)



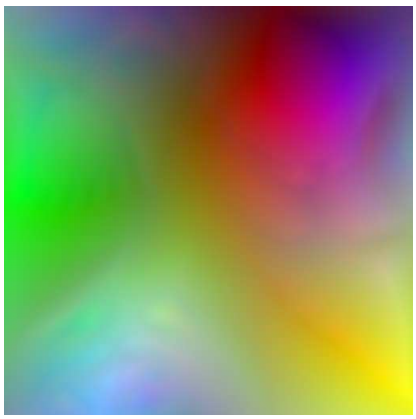
(d) Normal map (256x256)



(e) Normal-mapped reconstruction (b+c)

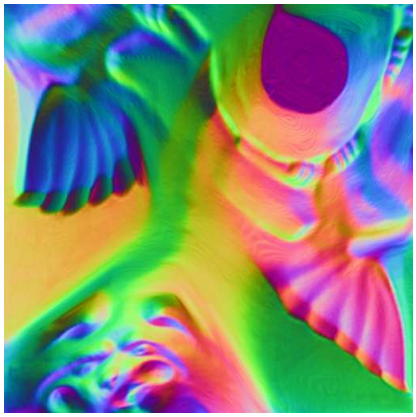
Figure 11: Three holes torus mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping.

(a) Original (480K faces; genus 0)



(b) Geometry Image (257x257)

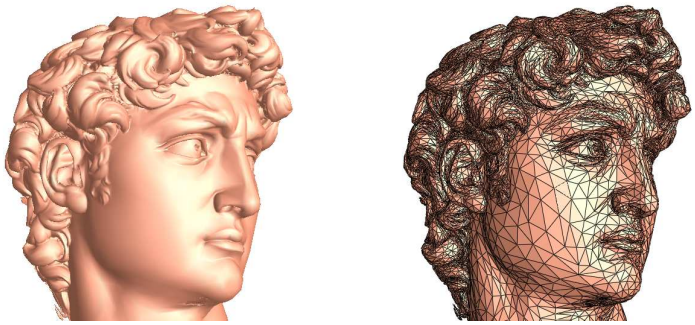
(c) Reconstruction (PSNR=78.2)



(d) Normal map (512x512)

(e) Normal-mapped reconstruction (b+c)

Figure 12: Gargoyle mesh: original meshes with cut, geometry images and their reconstructions, and use of normal-mapping.



(a) Surface (b) Triangulation

Figure 13: David mesh with 20000 faces

4 Discrete Riemann Geometry

Riemann geometry for smooth surfaces has be well established, though there are still many unsolved problems. This section will transfer the thereoms for smooth surfaces to discrete meshes. The mathematical tools are mainly combinatorics and linear algebra. We approximate a smooth surface by a sequence of meshes, and if we show all the results hold for the limit process, then the proofs described in this section can also be used to prove the thereoms in the smooth case.

A sharp difference between the smooth case and the discrete case is embbededness. A smooth surface homeomorphic to a disk can be conformally embedded in a convex planar region. But this is not true in the discrete case. If all the string constants are positive, then the discrete conformal embedding exists. For the genus nonzero case, if all the string constants are positive, then whether the local conformal embedding exists except for $2g-2$ singularities is unknown. I conjecture that such local embedding exists as long as all the string constants are positive.

4.1 Surface Triangulations

Triangulation is the division of a surface or plane polygon into a set of triangles, usually with the restriction that each triangle side is entirely shared by two adjacent triangles. It has been proven that every surface has a triangulation, but it might require an infinite number of triangles and the proof is difficult [14]. A surface with a finite number of triangles in its triangulation is *compact*.

In order to approximate smooth surfaces by triangular meshes, special triangulation is preferred. We want to prove that each smooth surface admits a triangulation with all acute angles. The result is intuitive, but the proof is nontrivial.

Theorem 4.1 *Suppose S is a closed surface, then S admits a triangulation where all angles are acute.*

Proof. A smooth genus zero surface M can be conformally mapped to a sphere. Suppose the metric on the sphere is (E, F, G), then there is a continuous function ρ from S^2 to $\mathbb{R}$, such that the metric on M is $(\rho E, \rho F, \rho G)$. Furthermore, ρ is positive everywhere on the sphere. Because S^2 is compact and ρ is uniformly continuous, ρ reaches its minimum on S^2. Suppose l is the lower bound of ρ. Given any ϵ, we can find δ, such that if $p_1, p_2 \in S^2$, $||p_1 - p_2|| < \delta$ then $|\rho(p_1) - \rho(p_2)| < \epsilon$. We choose ϵ to be far less than l. We can triangulate S^2 by subdividing an icosahedron, and map the edges to geodesics of the sphere. Because this triangulation is symmetric, all angles of this triangulation are either of 60 or of 72 degrees on the sphere.

Suppose the triangulation of S^2 is refine enough, such that the diameter of each curved triangle is less than δ. Then we map this triangulation back to S. Because the map is conformal, all the angles are of either 60 or 72 degrees on S. Also, within each spherical triangle on S^2, the scaling factor ρ is almost a constant. The change of aspect ratio is between $1 - \frac{2\epsilon}{l-\epsilon}$ and $1 + \frac{2\epsilon}{l-\epsilon}$. If $\frac{\epsilon}{l}$ is small enough, the above range is close to 1. Because the triangulation on sphere

is almost equilateral, the triangulation on M is near equilateral.

Suppose M is a topological torus, then M can be conformally mapped to a parallelogram on $\mathbb{R}^2$. We can easily triangulate the parallelogram with all acute angles, and map back the triangulation to M.

For surfaces with higher genus and boundaries, the proof can be conducted in a similar way. First we subdivide the surface to several patches, each of which is homomeomorphic to a torus with several disks removed(a pant). Each pant can be conformally mapped to an annulus, and the conformality is preserved across the boundaries. We use equilateral triangles to telleslate the whole plane, and triangulate the annulus, then map the triangulation back to surface M. $\square$

Any smooth surface can be approximated by a simplicial complex with arbitrary accuracy in the sense of Haudorff metric. In order to perform harmonic analysis on the polygonal model, the angles are more important, and the $Hausdorff$ approximation is not good enough. We require to use triangulations with all acute angles. In later sections, this will be explained in more details.

A mesh is a piecewise linear triangular polyhedron, and can be used to approximate a smooth surface. Arbitrary precision can be achieved by subdividing a mesh.

Definition 4.2 *Let K be a simplicial complex whose topological realization $M = |K|$ is homeomorphic to a compact 2-dimensional manifold. We call such a complex a simplicial surface. Suppose that in addition there is a piecewise linear embedding*

$$F : |K| \to R^3, \tag{2}$$

we call the pair (K, F) a triangular mesh.

Figure 4.1(b) shows a triangular mesh, and the surface it approximates (a). The surface is Michelangelo's David.

4.2 Simplicial Homology

All closed curves on a surface form a linear space. Different curves can be merged together and deformed among each other. Homology is the mathematical tool to represent the curve space and study the equivalence among curves. The merging of curves is expressed algebraicly, the deforming from one to another is represented as the quotient space. Homology is deeply related to the topology of the surface. In our setting, surfaces are represented as triangular meshes. All curves are represented as linear combinations of edges. For the purpose of locating each handle on a mesh, we apply simplicial homology tools.

A *chain complex*

$$C = \{C_q, \partial_q\} \tag{3}$$

is a collection of (additive) Abelian groups C_q one for each integer q, and of homoemorphism $\partial_q : C_q \to C_{q-1}$ such that $\partial_q \cdot \partial_{q-1} = 0$ for each q. Elements of C_q are called $q-chains$ of C, and ∂_q is called the *boundary operator*.

The kernel of ∂_q is denoted by $Z_q(C)$, and its element is called a $q-cycle$ of C. The image of ∂_{q+1} is denoted by $B_q(C)$, and its element is called a $q-boundary$ of C. The relation $\partial_q \cdot \partial_{q+1} = 0$ implies $B_q(C) \subset Z_q(C)$. The quotient group $Z_q(C)/B_q(C)$ is denoted by $H_q(C)$, the *qth homology group* of C. Elements of $H_q(C)$ are called $q-dimensional\ homology\ classes$ of C. Two cycles representing the same homology class are said to be *homologous*. The direct sum $\bigoplus_q H_q(C)$ is denoted by $H_*(C)$ and is called the *homology group* of C.

Suppose K is a simplicial complex, the oriented faces, edge, and vertices are simplices. Let C_qK denote the Abelian group generated by the oriented q-simplexes of K. Each element of C_qK is written uniquely as a finite sum

$$\sum_i g_i\sigma_i,\ g_i \in \mathbb{Z}.$$

The boundary operator is a homeomorphis

$$\partial_q : C_q K \to C_{q-1} K.$$

Intuitively,

$$\partial_2[v_0, v_1, v_2] = [v_0, v_1] + [v_1, v_2] + [v_2, v_0], \partial_1[v_0, v_1] = v_1 - v_0, \partial_0 v_0 = 0.$$

Then the incidence number $[\sigma_i^q : \sigma_j^{q-1}] \in \mathbb{Z}$ is defined by

$$\partial_q(\sigma) i^q) = \sum_j [\sigma_i^q : \sigma_j^{q-1}] \sigma_j^{q-1}.$$

Then $\partial_q \dot{\partial}_{q-1} = 0$, the q-th homology group is

$$H_q K = \frac{ker \partial_q}{img \partial_{q+1}}.$$

H_1 is the set of curves that are closed but not boundaries of any surface patches on M.

4.3 Simplicial Cohomology

Cohomology is the linear functional space of homology. Intuitively, it is the gradient of functions on M and can be represented as tangential vector fields. Cohomology has similiar group structures as homology, but it has wedge products, so it has a ring structure.

A *cochain complex*

$$C = \{C^q, \delta^q\} \tag{4}$$

is a collection of Abelian groups C^q, one for each integer q, and of homomorphisms

$$\delta^q : C^q \to C^{q+1} \tag{5}$$

such that $\delta^{q+1} \circ \delta q = 0$. Elements of C^q are called $q - cochain$s, and δ^q is called the *coboundary operator*.

Let $Hom(A, B)$ denote the group of homomorphism from an Abelian group A to an Abelian group B. Given a chain complex C and an Abelian group G, a cochain complex $C^* = Hom(C, G)$ is defined by

$$C^q = Hom(C_q, G) \tag{6}$$

$$(\delta^q u)(c) = u(\partial_{q+1} c)(u \in C^q, c \in C_{q+1}) \tag{7}$$

.

Suppose M is a simplicial complex. Let Δ_p be simplexes, define linear functional $\Delta^q \in Hom(C_p, R)$, such that

$$\Delta^q(\Delta_p) = \delta_{pq}. \tag{8}$$

Then $\{\sum f_q \Delta^q, f_q \in R\}$ are cochains. The coboundary operator δ is

$$\delta\Delta^q = \sum [\Delta_q : \Delta_p]\Delta^p. \tag{9}$$

The cohomology group is

$$H^q(K) = \frac{ker\delta^q}{img\delta^{q-1}}. \tag{10}$$

4.4 Interaction Between Homology and Cohomology

We can integrate a cocyle on a cycle, the result only depends on the homology class of the cycle and the cohomology type of the cocyle.

Theorem 4.3 *Suppose $r, r' \in Z_1(K)$, $\omega, \omega' \in Z^1(K)$, r and r' are in the same homology class, ω and ω' are in the same cohomology class, then*

$$\int_r \omega = \int_{r'} \omega'.$$

Proof. There exists $\sigma \in C_2 K$, such that $r - r' = \partial\sigma$. Also there exists $\tau \in C^2(K)$ such that $\omega - \omega' = \delta\tau$,

$$\int_r \omega - \int_{r'} \omega = \int_{\partial\sigma} \omega = \int_\sigma \delta\omega = 0.$$

On the other hand,

$$\int_{r'} \omega - \int_{r'} \omega' = \int_{r'} \delta\tau = \int_{\partial r'} \tau = 0.$$

So equation 4.3 holds. □

The integration between cycle and cocyles is only determined by their homology class and cohomology class.

4.5 Conformal Structure for Meshes

Every simplicial surface has a natural underlying complex analytic structure. This is proven in [7].

The embedding F endows M with a *conformal* structure. This means that there is a natural notion of complex analytic function on $|K|$, therefore, many of the techniques in complex analysis can be used to study $|K|$.

A conformal structure is given by the following data: a collection of open sets $U_j \subset M$ covering M, together with a collection of homeomorphisms

$$\Phi_i : U_i \to C \tag{11}$$

onto open sets in the complex plane, called a *complex analytic atlas*. The individual maps Φ_i's are called *charts* and they are required to satisfy the compatibility condition that

$$\Phi_j \circ \Phi_i^{-1} : \Phi_i(U_i \cap U_j) \to \Phi_j(U_i \cap U_j) \tag{12}$$

are all holomorphic (complex analytic).

The conformal structure on $|K|$ can be defined as below. For each vertex $v \in K$, let $U_v \subset |K|$ denote the interior of the simplicial neighborhood of v, and

let $\phi_v : U_v \to C$ be the chart constructed as follows:

If v is an interior vertex of valence n, write U_v as the union

$$U_v = \bigcup_{k=1}^{n} T_k, \tag{13}$$

where $T_k = |\{v, v_k, v_{k+1}\}| - |\{v_k, v_{k+1}\}|$, and $v_{n+1} = v_1$.

Next let θ_k be the angle of T_k at the vertex v, and set

$$\Theta_1 = 0, \Theta_k = \sum_{j=1}^{k-1} \theta_j, \Theta = \sum_{j=1}^{n} \theta_j. \tag{14}$$

Let $f_k : T_k \to C$ be the isometry defined by the equations

$$f_k(v) = 0, f_k(v_k) = ||v_k - v||, f_k(v_{k+1}) = ||v_{k+1} - v||e^{i\theta_k}, \tag{15}$$

where $||\cdot||$ is the Euclidean norm in R^3. Finally let $\psi : C \to C$ be the map $\psi(z) = z^a$, where $a = 2\pi/\Theta$, v to $0 \in C$. Then the chart ϕ_v is defined as follows:

$$\phi_v(p) = e^{i2\pi/\Theta_k}\psi \circ f_k(p) \ \forall p \in T_k \tag{16}$$

Then the maps $\phi_v : U_v \to C$ form a complex analytic atlas.

4.6 Harmonic Maps on Smooth Riemann Surfaces

Harmonic maps have been intensively studied in mathematics. The following concepts are defined in [37] and [7].

Let M be a smooth compact surface with boundary ∂M, and let g be a Riemannian metric on M. Suppose $f : M \to R$ is a smooth function. The *harmonic energy* of f is the quantity

$$E_g[f] = \frac{1}{2}\int_M |df|_g^2 dA_g, \tag{17}$$

where $|\cdot|_g$ is the norm with respect to g and dA_g is the area element on M induced by g. Let $g = g_{ij}dx^i dx^j$ where $x^i, i = 1, 2$ are local coordinates on M. Then the integrand has the local coordinate expression

$$|df|^2 dA = \sum_{1 \le i,j \le 2} \frac{\partial f}{\partial x^i}\frac{\partial f}{\partial x^j}\sqrt{det(g)}dx^1 \wedge dx^2, \tag{18}$$

where g^{ij} denotes the inverse of the matrix g_{ij}, and $det(g)$ denotes the determinant of g_{ij}.

Consider the functional

$$E : H \to R \tag{19}$$

on the space of smooth functions on M with fixed boundary value $f_\partial : \partial M \to R$, where f_∂ is a smooth function. The critical points for E are called *harmonic functions*. There is a unique harmonic map f assuming the prescribed boundary values f_∂.

The harmonic energy depends only on the conformal class of g. That is if $\tilde{g} = e^\lambda g$, for $\lambda : M \to R$ smooth, then

$$E_g[f] = E_{\tilde{g}}[f]. \tag{20}$$

Also, Let $\phi : (M, g) \to (N, h)$ be a biholomorphism between two compact Riemann surfaces, and let $f : N \to R$ be a smooth function, then

$$E_h[f] = E_g[f \circ \phi]. \tag{21}$$

4.7 Discrete Harmonic Energy

Discrete harmonic energy is defined in [7]. Consider a mesh $F : |K| \to R^3$, the underlying space $M = |K|$ has two structures: a piecewise linear structure given by the simplicial complex K, and a conformal structure (M, g) induced by the embedding F. The functional E_g depends on the conformal structure, but it can be computed in terms of the PL structure.

Let $T_\alpha \subset |K|, \alpha = 1, \ldots, n$ be the collection triangular faces of $|K|$. Notice that the union of the interiors of the faces is an open subset of $|K|$ of full measure. It follows that

$$E_g[f] = \frac{1}{2}\int_M |df|_g^2 dA_g = \frac{1}{2}\sum_\alpha \int_{T_\alpha} |df|_g^2 dA_g, \tag{22}$$

where $f : M \to R$ is a piecewise smooth function. Then the following formulae hold:

$$\int_{T_\alpha} |df|_g^2 dA_g = \frac{1}{2} \sum_{i,j} a^\alpha_{v_i,v_j}) |f(v_i) - f(v_j)|^2 \tag{23}$$

$$a^\alpha_{v_1,v_2} = \frac{1}{2} \frac{(v_1 - v_3) \cdot (v_2 - v_3)}{|(v_1 - v_3) \times (v_2 - v_3)|} \tag{24}$$

$$a^\alpha_{v_2,v_3} = \frac{1}{2} \frac{(v_2 - v_1) \cdot (v_3 - v_1)}{|(v_2 - v_1) \times (v_3 - v_1)|} \tag{25}$$

$$a^\alpha_{v_3,v_1} = \frac{1}{2} \frac{(v_3 - v_2) \cdot (v_1 - v_2)}{|(v_3 - v_2) \times (v_1 - v_2)|} \tag{26}$$

.

The final formula for $E[f]$, in the case where f is piecewise linear function on a complex K, involves certain *spring constants*. For $e = \{u, v\}$, an internal edge of K, let

$$k_e = a^\alpha_{u,v} + a^\beta_{u,v}, \tag{27}$$

where T_α and T_β are the faces of K adjacent to e. If e is a boundary edge let

$$k_{u,v} = a^\alpha_{u,v}. \tag{28}$$

Let $F : k \to R^3$ be a mesh and let (M, g) denote the conformal structure on $M = |K|$ induced by F. If $f : K \to R$ is a piecewise linear function then

$$E_g[f] = \frac{1}{2} \sum_{u,v} k_{u,v} |f(u) - f(v)|^2. \tag{29}$$

$E_g[f]$ may be thought of as a "spring energy", where each edge $\{u, v\}$ of K is viewed as a spring with spring constant $k_{u,v}$.

4.8 Discrete Laplacian Operator

Discrete Laplacian operator is defines in [7]. The harmonic energy defines a quadratic form

$$Q : C^0 K \times C^0 K \to R \tag{30}$$

given by the equation

$$Q(f,g) = \frac{1}{2}\sum_{\{u,v\}} k_{u,v}(f(u) - f(v))(g(u) - g(v)). \tag{31}$$

Q is the positive definite inner product on C^0K.

Harmonic function f is the unique function in $C^{PL}(K)$ which minimizes the quantity $Q(f,f)$. For all $h \in C^0K$, the derivative

$$\frac{d}{dt}E_g[f + th] = 2Q(f,h) + 2tQ(h,h) \tag{32}$$

vanishes at $t = 0$.

The discrete *Laplacian* is the linear operator

$$\Delta : C^0K \to C^0K \tag{33}$$

on the space of PL functions on K, Δ is defined by the formula

$$\Delta f(u) = \sum_{\{u,v\}\in K} k_{u,v}(f(v) - f(u)). \tag{34}$$

Then

$$Q(f,g) = -\frac{1}{2}\sum_{u\in K} f(u)\Delta g(u). \tag{35}$$

A function $f \in C^0K$ is harmonic if and only if

$$\Delta f(v) = 0 \tag{36}$$

for every interior vertex v of K. This is equivalent to the $mean-value$ property that

$$f(v) = \frac{\sum_{j=1} dk_{v,v_j} f(v_j)}{\sum_{j=1} dk_{v,v_j}}. \tag{37}$$

4.9 Discrete Harmonic 1-Forms

The harmonic 1-forms are the gradient fields of harmonic functions. Therefore, harmonic 1-forms should minimize the harmonic energy.

Definition 4.4 *Suppose $\omega \in Z^1K$, if*

$$\Delta\omega|_u = \sum_{[u,v]\in K} \omega([u,v]) = 0, \forall u \in K$$

then ω is harmonic.

Each cohmology class has a unique harmonic form, and all harmonic forms form a linear space, the dimension of this space is two times the genus of K.

Theorem 4.5 *Given $\omega \in H^1(K,\mathbb{R})$, there exists a unique ω', such that ω' is cohomologous to ω and ω' is harmonic.*

Proof. Construct a $\tau \in C^0K$, such that $\omega + \delta\tau$ is harmonic. Then τ satisfies the relation

$$\sum_{[u,v]\in K} k_{u,v}(\omega([u,v]) + \tau(u) - \tau(v)) = 0, \ \forall u \in K.$$

Then we construct a linear system

$$D\tau = b.$$

D is indexed by the vertices in K. Suppose u, v are different vertices in K, u, v are connected, then $D_{u,v} = -k_{u,v}$, otherwise it is zero. $D_{u,u} = -\sum_{u\neq v} D_{u,v}$. $\tau_u = \tau(u)$. $b_u = -\sum_{[u,v]\in K} \omega([u,v])$. The null space of D is

$$H = \{\lambda(1,1,\ldots,1)^T | \lambda \in \mathbb{R}\}.$$

Suppose $D\psi = 0$, then

$$\sum_{[u,v]\in K} k_{u,v}||\psi(u) - \psi(v)||^2 = 0.$$

Then ψ is constant on each vertex. $\psi \in H$. Therefor D maps the normal complement $H^{\perp}$ of H to itself. By definition,

$$H^{\perp} = \sum_{u \in K} \psi(u) = 0.$$

Now we want to verify $b \in H^{\perp}$.

$$\sum_{u \in K} \sum_{[u,v] \in K} \omega([u,v]) = \sum_{[u,v] \in K} (\omega([u,v]) + \omega([v,u])) = 0.$$

So equation 4.9 has a unique solution. Hence for each cohomology class, there exists at least one harmonic form.

Now we prove the uniqueness. Suppose ω and ω' are cohomologous and harmonic. Then $\psi = \omega - \omega'$ is equal to $\delta\tau$,

$$\Delta\psi = \Delta(\omega - \omega') = \Delta\omega - \Delta\omega' = 0.$$

Then

$$\sum_{[u,v] \in K} k_{u,v} ||\tau(u) - \tau(v)||^2 = 0.$$

Hence $\tau(u)$ is a constant for all $u \in K$. $\delta\tau = 0$. So $\omega = \omega'$. $\square$

Theorem 4.6 *A mesh S is of genus g, then the discrete harmonic 1-forms form a real linear vector space, the dimension of which is $2g$.*

Proof. Laplacian operator $\Delta : Z^1K \to \mathbb{R}$ is a linear operator. Its null space is also a linear space. Harmonic 1-forms are in its null space. Suppose S is a genus g surface, we choose a set of homology bases H_1, $\{e_1, e_2, \ldots, e_{2g}\}$, then we construct the dual cohomology bases, $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$, such that

$$\int_{e_i} \omega_j = \delta_i^j,$$

where δ_i^j is the Kroneck symbol. According to theorem 4.5, we can diffuse ω_i to be harmonic. We want to show these harmonic forms are linearly independent.

Assume there are $2g$ coefficients $c_i \in \mathbb{R}$, such that

$$\sum_{i=1}^{2g} c_i \omega_i = 0.$$

Then

$$0 = \int_{e_j} \sum_{i=1}^{2g} c_i \omega_i = \sum_{i=1}^{2g} c_i \int_{e_j} \omega_i = c_j.$$

Therefore $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$ are linearly independent. Now we want to show any harmonic 1-form can be represented as a linear combination of ω_i's. Suppose ω is harmonic, then

$$c_i = \int_{e_i} \omega \tag{38}$$

$$\tau = \omega - \sum_{i=1}^{2g} \omega_i, \tag{39}$$

τ is harmonic. Furthermore,

$$\int_{e_i} \tau = \int_{e_i} \omega - \int_{e_i} \sum_{j=1}^{2g} c_j \omega_j = c_i - \sum_{j=1}^{2g} c_j \delta_j^i = 0.$$

Given any cycle $r \in Z_1 K$, r can be represented as a linear combination of e_i's, suppose r is cohomologous to $\sum_{i=1}^{2g} r_i e_i$, then integration

$$\int_r \tau = \sum_{i=1}^{2g} r_i \int_{e_i} \tau = 0.$$

We want to show that $\tau \equiv 0$. Suppose there exists an edge $[v_0, v_1] \in C_1 K$, such that $\tau([v_0, v_1]) > 0$, then we can expend $[v_0, v_1]$ to a path $\{v_0, v_1, \ldots, v_n\}$, such that $\tau([v_k, v_{k+1}]) > 0, k = 0, \ldots, n-1$. We extend this path as long as possible. Then there must be no self intersection of this path. Otherwise, we can find a cycle $r = \{v_j, v_{j+1}, \ldots, v_k\}$, $1 \leq j < k \leq n v_j = v_k$, then

$$0 = \int_r \tau = \sum i = j^{k-1} \tau([v_i, v_{i+1}] > 0.$$

On the other hand, because the number of vertices of M is finite, the path has to stop at some vertex v_n. By our construction,

$$\tau([v_n, u]) \leq 0, \forall u, [v_n, u] \in K.$$

We know v_{n-1} is adjacent to v_n, $\tau([v_n, v_{n-1} > 0$. Then

$$\Delta\tau|_{v_n} \neq 0.$$

This is contradictory to the fact that τ is harmonic. So τ must always be zero. Hence the harmonic 1-form space is $2g$ dimensional. □

4.10 Discrete Hodge Star Operator

Simplicial cohomology has a wedge operator, which is a linear map

$$\wedge : Z^1K \times Z^1K \to Z^2K.$$

Theorem 4.7 *Suppose $\omega, \tau \in Z^1K$, for each face $[v_0, v_1, v_2]$,*

$$\omega \wedge \tau([v_0, v_1, v_2]) = \frac{1}{6} \begin{vmatrix} \omega([v_0, v_1]) & \omega([v_1, v_2]) & \omega([v_2, v_0]) \\ \tau([v_0, v_1]) & \tau([v_1, v_2]) & \tau([v_2, v_0]) \\ 1 & 1 & 1 \end{vmatrix}$$

.

Proof. Suppose ω and τ are two closed 1-forms. We construct local isometric coordinates of a face $T = [A, B, C]$, $A(0, 0), B(a, 0), C(b, c)$, here $a = ||B - A||$, $b = ||C - A||cosA$, $c = ||C - A||sinA$. ω and τ can be represented as piecewise constant 1-forms with respect to the coordinates

$$\omega = \frac{1}{ac}(c\omega[A, B]dx + (a\omega[A, C] - b\omega[A, B])dy) \quad (40)$$

$$\tau = \frac{1}{ac}(c\tau[A, B]dx + (a\tau[A, C] - b\tau[A, B])dy). \quad (41)$$

By direct wedge product defined for De Rham 1-forms, we get

$$\omega \wedge \tau = \frac{1}{ac}(-\omega[A, B]\tau[C, A] + \omega[C, A]\tau[A, B])dx \wedge dy. \quad (42)$$

Then

$$\int_T \omega \wedge \tau = \frac{1}{2}(-\omega[A, B]\tau[C, A] + \omega[C, A]\tau[A, B]). \quad (43)$$

Because A, B, C are circular symmetric, similarly, we can get

$$\int_T \omega \wedge \tau = \frac{1}{2}(-\omega[B,C]\tau[A,B] + \omega[A,B]\tau[B,C]) \quad (44)$$

$$\int_T \omega \wedge \tau = \frac{1}{2}(-\omega[C,A]\tau[B,C] + \omega[B,C]\tau[C,A]) \quad (45)$$

. Then

$$\int_T \omega \wedge \tau = \frac{1}{6}\begin{vmatrix} \omega[A,B] & \omega[B,C] & \omega[C,A] \\ \tau[A,B] & \tau[B,C] & \tau[C,A] \\ 1 & 1 & 1 \end{vmatrix} \quad (46)$$

□

If we treat each 1-form ω as a tangential vector field of M, then locally in each tangent space, we can rotate the vector by a right angle about the normal at that point, then construct a new tangential vector field ${}^*\omega$. This operator is called *Hodge star* operator. In simplicial cohomology, the cocycles can not be represented as smooth tangential vector fields directly, but we can compute the wedge product of ${}^*\omega$ with any $\tau \in Z^1K$ directly. So we define a linear operator $\wedge^* : Z^1K \times Z^1K \to Z^2K$, and define ${}^*\omega$ as follows:

Definition 4.8 *Given $\omega, \tau \in Z^1K$, the wedge star is a linear map*

$$\wedge^* : Z^1K \times Z^1K \to Z^2K$$

such that

$$\omega \wedge^* \tau([v_0, v_1, v_2]) = \omega M \tau.$$

Here $\omega = (\omega([v_0,v_1]), \omega([v_1,v_2]), \omega([v_2,v_0]))$, $\tau = (\tau([v_0,v_1]), \omega([v_1,v_2]), \omega([v_2,v_0]))^T$, M is

$$\frac{1}{24S}\begin{pmatrix} 2(b^2+c^2) & a^2+b^2-c^2 & a^2+c^2-b^2 \\ b^2+a^2-c^2 & 2(c^2+a^2) & b^2+c^2-a^2 \\ c^2+a^2-b^2 & c^2+b^2-a^2 & 2(a^2+^2) \end{pmatrix}, \quad (47)$$

where $a = ||v_1 - v_0||, b = ||v_2 - v1||, c = ||v_0 - v_2||$, and S is the area of the triangle.

Definition 4.9 *Given $\omega \in Z^1K$, the Hodge star of ω is denoted as ${}^*\omega$, for any $\tau \in \mathbb{Z}^1$*

$$\int_M \tau \wedge ({}^*\omega) = \int_M \tau \wedge^* \omega$$

.

These definitions are derived from De Rham cohomology. Suppose $\tau \in Z^1$, then we build local a coordinates system and represent τ as formula [6] using the formula of Hodge star

$$\begin{cases} {}^*dx &= +dy \\ {}^*dy &= -dx \end{cases}$$

Suppose T is a face, and the lengths of its three edges are $\{a, b, c\}$, then

$$\int_T \omega \wedge {}^*\tau = aMb^t,$$

where $a = (\omega[AB], \omega[CA])$ and $b = (\tau[AB], \tau[CA])$

$$M = \frac{1}{8S} \begin{pmatrix} 2b^2 & b^2 + c^2 - a^2 \\ c^2 + b^2 - a^2 & 2c^2 \end{pmatrix}. \tag{48}$$

Because a, b, c are circular symmetric, by ciculating a, b, c, we get other two equations. Adding them together, we get the definiton 47.

We also would like to show the uniqueness of the Hodge star operator.

Theorem 4.10 *For each $\omega \in Z^1K$, there exists a unique ${}^*\omega \in Z^1K$. Also, the Hodge star operator $* : Z^1K \to Z^1K$ is linear.*

Proof. We choose a set of cholomology bases $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$. In order to obtain ${}^*\omega$ we need to solve the linear system

$$\omega_i \wedge (\ *\omega) = \omega_i \wedge^* \omega,\ i = 1, 2, \ldots, 2g.$$

Because ${}^*\omega$ is in $H^1(K, \mathbb{R})$,

$${}^*\omega = \sum_{i=1}^{2g} c_i \omega_i.$$

Then we can build the linear system

$$\Omega \mathbf{c} = \mathbf{d},$$

where Ω has entries $\int_M \omega_i \wedge \omega_j$, $\mathbf{c} = (c_1, c_2, \ldots, c_{2g})$, and $\mathbf{d}$ has entries $\int_M \omega_i \wedge {}^*\omega$. Because ω_i's are cohomology bases and Ω is non-degenerated, the above equation has a unique solution. Suppose we want to compute ${}^*(a\omega_0 + b\omega_1)$, where $a, b \in \mathbb{R}$, then we can construct the linear system for ω_0, ω_1 individual. $\Omega \mathbf{c_i} = \mathbf{d_i}, i = 0, 1$, then $\Omega(a\mathbf{c_0} + b\mathbf{c_1}) = a\mathbf{d_0} + b\mathbf{d_1}$. This shows

$$ {}^*(a\omega_0 + b\omega_1) = a\,{}^*\omega_0 + b\,{}^*\omega_1.$$

□

4.11 Discrete Laplacian Spectrum

The Laplacian operator is positive semidefinite, all the eigenvalues are real and nonnegative. All the eigenfunctions are orthogonal to each other. Furthermore, they form a basis of all functions on the mesh. This is valuable for potential signal processing and data compression directly on meshes.

Theorem 4.11 *A mesh M is with all sharp angles. Given the discrete Laplacian operator Δ on M, all the eigenvalues of Δ are positive real numbers except for one zero. All the eigenfunctions of Δ are orthogonal and form a complete basis of C^0K.*

Proof. From the definition of Δ, we can construct the matrix representation of it as D.

$$\sum_{u,v \in K} k_{u,v} ||f(u) - f(v)||^2 = \mathbf{f}^T D \mathbf{f}.$$

Then D is symmetric, so all its eigenvalues are real. D is semi-positive, all the eigenvalues are nonnegative. Furthermore, all the eigenvectors of D are orthorgnal and form the bases for the whole space,

$$\Delta \mathbf{f} = \lambda_i \mathbf{f}, \Delta \mathbf{g} = \lambda_j \mathbf{g}$$

implies

$$\sum_{u \in K} f_u g_u = 0.$$

□

4.12 Discrete Holomorphic Differentials

The gradient fields of conformal mappings are represented as holomorphic differentials or holomorphic one-forms, which are harmonic and locally orthogonal with equal lengths. All holomorphic differentials also form a real linear space, the dimension of which is also $2g$, where g is the genus of M.

Definition 4.12 *Suppose ω is in Z^1K and harmonic, then the holomorphic 1-form is*

$$\omega + \sqrt{-1} * \omega$$

The Hodge star operator $*$ is linear so the linear relation of holomorphic forms is determined by its real part. We already know the harmonic 1-forms are $2g$ dimensional, so are the holomorphic 1-forms.

Theorem 4.13 *All holomorphic 1-forms form a real linear space, which is $2g$ dimensional.*

Proof. We choose a set of harmonic 1-form bases $\{\omega_1, \omega_2, \ldots, \omega_{2g}\}$. Then given any holomorphic 1-form $\omega + \sqrt{-1}\,{}^*\omega$, $\omega = \sum_{i=1}^{2g} c_i \omega_i$, then

$$\omega + \sqrt{-1}\,{}^*\omega = \sum_{i=1}^{2g} c_i (\omega_i + \sqrt{-1}\,{}^*\omega).$$

□

4.13 Surface Complex Structure

A 2-dimensional manifold M has a natural complex structure. In our setting where M is a mesh, the complex structure is constructed explicitly in [7].

Any genus zero surface M is conformally equivalent to S^2. $u : M \to S^2$ is conformal if and only if u is harmonic. The conformal auto-morphism group of S^2 is 6 dimensional, which is the Mobius transformation group defined on the complex plane $\mathbb{C}$. If we fix the images of 3 points, then there is a unique conformal map from M to S^2.

For non-zero genus surfaces, we study the structure of its holomorphic differential group. The following form

$$\tau + \sqrt{-1}\omega, \tau, \omega \in \Omega^1(M; TM) \tag{49}$$

is called a holomorphic form if both τ and σ are harmonic and $*\tau = \omega$, here $*$ is the Hodge star operator. Suppose $\{v_1, v_2\}$ are orthonormal bases of a tangent space on M, then

$$\omega(v_1) = *\omega(v_2) \tag{50}$$

The set of holomorphic 1-forms is denoted as $H^{1,0}(M, \mathbb{C})$. Let M be a compact Riemann surface of genus g and $B = \{e_1, e_2, \cdots, e_{2g}\}$ be an arbitrary basis of $H_1(M, Z)$. The intersection matrix C of the above basis has entries

$$c_{ij} = -e_i \cdot e_j \tag{51}$$

where the dot denotes the algebraic number of intersections. A basis $B^* = \{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ of real vector space $H^{1,0}(M, \mathbb{C})$ is the dual of B if

$$Re \int_{e_i} \omega_j = c_{ij}. \tag{52}$$

From Riemann bilinear relations [2] it follows that matrix S with entries

$$Im \int_{e_i} \omega_j = s_{ij} \tag{53}$$

is symmetric and positive definite. The complex structure in $H^{1,0}(M, \mathbb{C})$ is given by a matrix R with respect to the basis B and satisfies $R^2 = -I$. The following relation holds

$$CR = S. \tag{54}$$

After Weyl [47] and Siegel [41], the matrix R is called the period matrix of M with respect to the basis B. Let a be a holomorphic automorphism of M, and let $[a]$ denote the matrix of its action on the homology and cohomology with respect to the above basis, then

$$[a]^{-1}R[a] = R, [a]^T C[a] = C. \tag{55}$$

The pair (R, C) determines the analytic structure of a given Riemann surface in the following sense: Two such pairs, (R_1, C_1) and (R_2, C_2) determine the same structure if and only if there exists an integral matrix N whose determinant is ± 1 such that

$$N^{-1}R_1N = R_2, N^t C_1 N = C_2. \tag{56}$$

If the bases B_1 and B_2 are canonical ones, then both C_1 and C_2 are identities, and N is an integral symplectic matrix.

5 Conformal Mapping for Genus Zero Surfaces

Given two genus zero meshes M_1, M_2, there are many conformal mappings between them. The algorithm for computing conformal mapping is based on the fact that harmonic maps are conformal for genus zero surfaces. All conformal mappings between M_1, M_2 form a group, which is the so-called Mobius group. Our method is as follows: First find a homeomorphism $\mathbf{h}$ between M_1 and M_2, then diffuse $\mathbf{h}$ so that $\mathbf{h}$ minimizes the harmonic energy. In order to ensure the convergence of the algorithm, special constraints are added so that the solution is unique.

5.1 Constrained Variational Problem

Suppose M_1 and M_2 are genus zero meshes, $\mathbf{h} : M_1 \rightarrow M_2$ is a degree one mapping. We would like to minimize the harmonic energy $E(\mathbf{h})$,

$$E(\mathbf{h}) = \sum_{[u,v]\in K_1} k_{u,v}||\mathbf{h}(u) - \mathbf{h}(v)||^2, \mathbf{h} = (h_0, h_1, h_2). \tag{57}$$

The Laplacian for $\mathbf{h}$ is simply

$$\Delta^{PL}\mathbf{h} = (\Delta^{PL}h_0, \Delta^{PL}h_1, \Delta^{PL}h_2). \tag{58}$$

Then if $\mathbf{h}$ is harmonic, the tangential component of $\Delta^{PL}\mathbf{h}$ is zero. Define the projection operator

$$P_{\mathbf{v}} = I - \frac{\mathbf{v}\otimes\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}, \mathbf{v} \in R^3, \tag{59}$$

where $\otimes$ is tensor product and I is an identity matrix. Then $\mathbf{h}$ is harmonic if and only if

$$P_{\mathbf{n}\circ\mathbf{h}}\Delta^{PL}\mathbf{h} = 0, \tag{60}$$

where $\mathbf{n}$ is the normal on M_2.

In order to ensure that the process converges to a unique solution, we have to add extra constraints. We force the center of mass of the surface to be at its origin, that is,

$$\int_{M_2} \mathbf{h} d\sigma_{M_1} = \mathbf{0} \tag{61}$$

where $d\sigma_{M_1}$ is the area element on M_1. This constraint will guarantee the solution is unique up to a rotation. Then we can construct the partial differential equation

$$\frac{\partial\mathbf{h}}{\partial t} + P_{\mathbf{n}\circ\mathbf{h}}\Delta^{PL}\mathbf{h} = 0 \tag{62}$$

with constraints 61. The steady state solution of $\mathbf{h}$ is the conformal mapping from M_1 to M_2. Equation 62 can be solved by iterative methods.

5.2 Steepest Descendent Algorithm

In our implementation, we fix M_2 as S^2. In order to compute the initial homeomorphism from M_1 to S^2, we first compute the spherical baricentric embedding, which minimizes the barycentric string energy. The barycentric energy is defined as in eqn. 29, here we let

$$k_{u,v} \equiv 1. \tag{63}$$

The corresponding Laplacian is defined as eqn. 34 with constant unit $k_{u,v}$. Then the following algorithm computes the spherical barycentric embedding,

Input mesh M, step length δt, threshold ϵ.

Output sphereial barycentric mapping $\mathbf{h}$.

1. Compute the Gauss map $\mathbf{n}$ from M to S^2, $\mathbf{h} \leftarrow \mathbf{n}$.
2. Compute the barycentric energy $E(\mathbf{h})$, if $\delta E < \epsilon$ then return $\mathbf{h}$.
3. Compute the tangential Laplacian of $\mathbf{h}$, $\delta h \leftarrow P_{\mathbf{n}\circ\mathbf{h}}\Delta^{PL}\mathbf{h}$
4. Update $\mathbf{h}$ by $\mathbf{h} \leftarrow \mathbf{h} - \delta t \times \delta\mathbf{h}$.
5. Repeat 2 through 4.

Algorithm 1. Spherical barycentric embedding

In practice, the barycentric embedding converges faster than the spherical harmonic embedding, and there are no extra constraints. Hence we use it as the initial embedding to compute spherical conformal mappings. The spherical conformal embedding algorithm is more complicated. In each iteration an extra normalization step is inserted so that the mass center of the surface stays in the

origin during the entire process.

Input Mesh M, step length δt, threshold ϵ.

Output Spherical harmonic map $\mathbf{h}$.

1. Compute the spherical barycentric map $\mathbf{b}$ from M to S^2, $\mathbf{h} \leftarrow b$.
2. Compute the harmonic energy $E(\mathbf{h})$, if $\delta E < \epsilon$ then return $\mathbf{h}$.
3. Compute the tangential Laplacian of $\mathbf{h}$, $\delta\mathbf{h} \leftarrow P_{\mathbf{n}\circ\mathbf{h}}\Delta^{PL}\mathbf{h}$.
4. Update $\mathbf{h}$ by $\mathbf{h} \leftarrow \mathbf{h} - \delta t \times \delta\mathbf{h}$.
5. Compute a Mobius transformation $\mathbf{m}$, such that $\mathbf{m} \circ \mathbf{h}$ satisfies the center of mass constraint equation 61.
6. Repeat 2 through 5.

Algorithm 2. Spherical Conformal Embedding

In step 5 above, the Mobius transformation on S^2 is in the form $\phi^{-1} \circ f \circ \phi$, where ϕ is the stereo-graphic projection from S^2 to the complex plane,

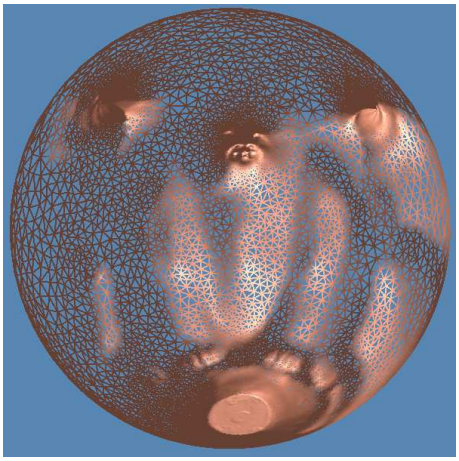
$$\phi(x_0, x_1, x_2) = (\frac{x_0}{1+x_2}, \frac{x_1}{1+x_2}), (x_0, x_1, x_2) \in R^3. \tag{64}$$

f is a Mobius transformation on $\mathbb{C}$,

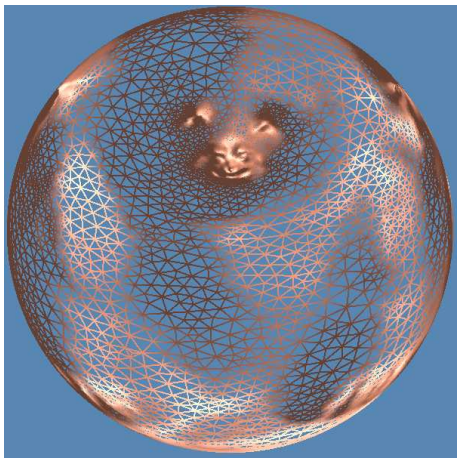
$$f(z) = \frac{az+b}{cz+d}, a, b, c, d \in \mathbb{C}, ad - bc \neq 0. \tag{65}$$

In practice, it is computationally expensive to normalize $\mathbf{h}$ by a Mobius transformation, we simply shift the center of mass of $\mathbf{h}(M_1)$ to the origin and normalize $\mathbf{h}(v), v \in K$ to the unit vector.

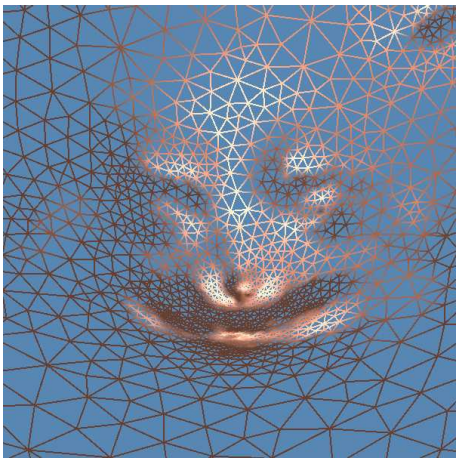
(a) Gargoyle mesh (20000 faces)



(b) Conformal spherical mapping

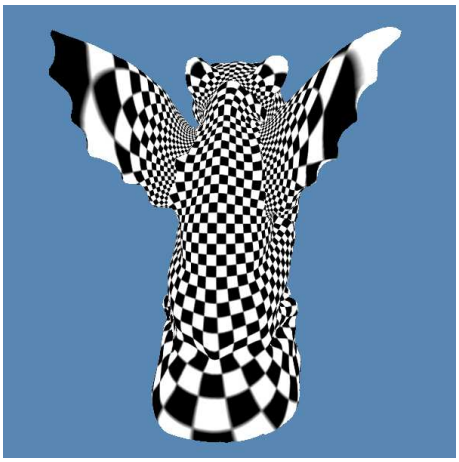


(c) Barycentric mapping



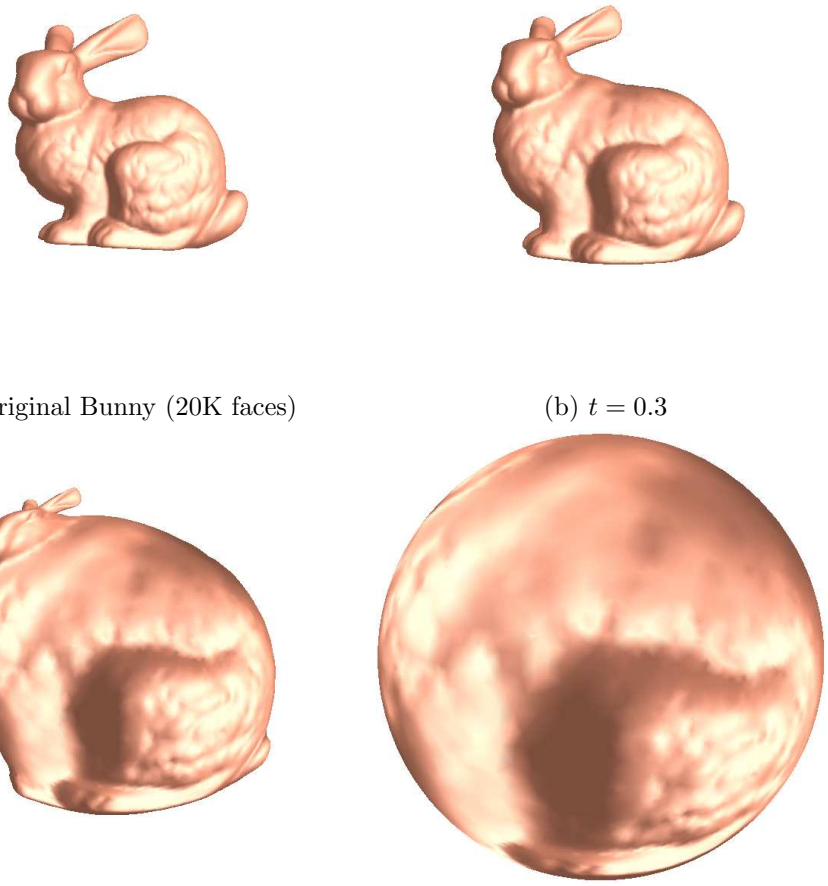
(d) Zoom into the face region of (b)

(e) Conformal texture mapping (front view)



(f) Conformal texture mapping (back view)

Figure 14: Gargoyle model spherical conformal mapping & barycentric mapping



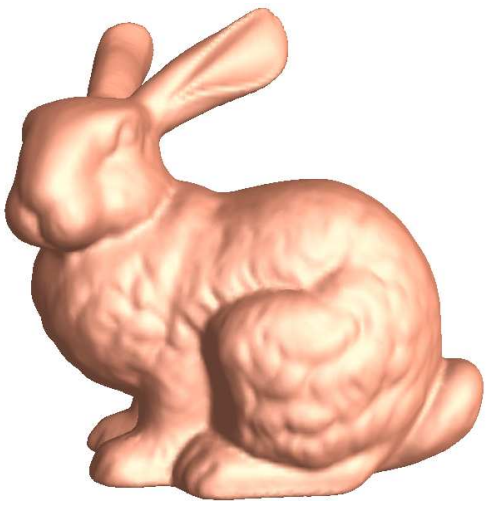
(a) Original Bunny (20K faces)

(b) $t = 0.3$

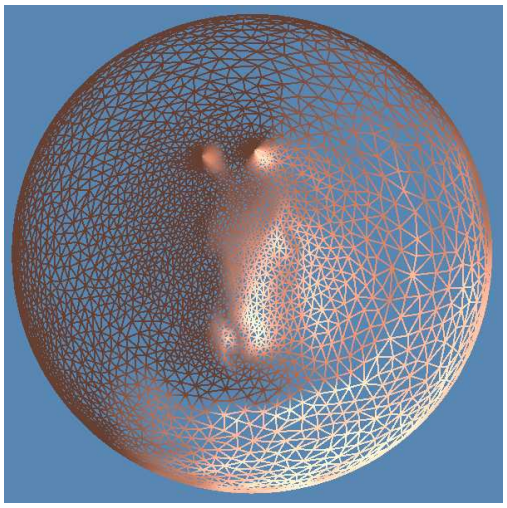
(c) $t = 0.8$

(d) $t = 1.0$ onto sphere

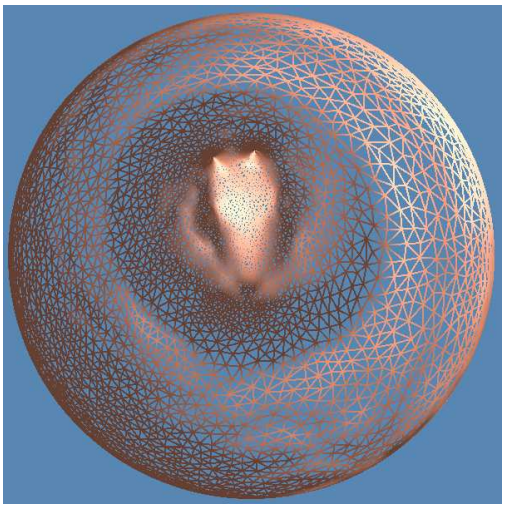
Figure 15: Bunny model spherical barycentric morphing



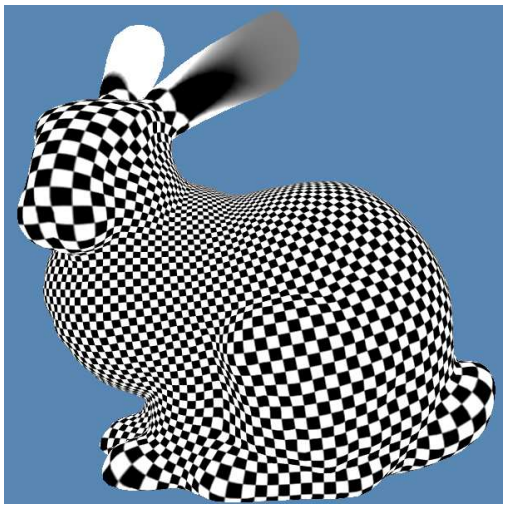
(a) Original Bunny model



(b) Conformal map to sphere



(c) Mobius transformation of (b)



(d) Conformal texture mapping

Figure 16: Spherical conformal map of the bunny mesh. The stretching factors of the bunny ears are much bigger than those of the other parts.

5.3 Results

A spherical barycentric embedding result is shown in figure 14. A mesh model of gargoyle with 20000 faces is mapped to a sphere using algorithm 1. Figure (14)(b) shows the conformal spherical embedding. (c) shows the barycentric spherical embedding. The normal information is preserved so the shading indicates the correspondence. When we zoom in to examine the neighborhood of a vertex in (d), we can see that the vertex is at the barycentric center of all its neighbors. During the optimization, the head and wing parts converge more slowly than the other regions. Special local optimization is performed for these parts.

Figure 14 (e) and (f) show the conformal texture mapping. First, the gargoyle mesh is mapped to a sphere using conformal mapping (b). The upper and lower hemispheres are projected to the tangent planes through its north pole and south pole individually. The projections are stereo-graphic projections. Then the gargoyle mesh is mapped to 2 planes conformally. The texture coordinates are defined as the plane coordinates. The texture is a regular checker board image. From the snapshot, we can tell that all the right angles of parametric curves are preserved. For the head and wing regions, the texture coordinates are scaled. So the entire texture mapping is a multi-resolution texture-mapping.

Figure 15 demonstrates the spherical barycentric embedding. In order to show the correspondances, a geometric morphing process is illustrated. The morphing is constucted as a linear combination of the original surface and the final one. Parameter t gives the blending factors. From the morphing, we can tell that the ear parts are mapped to relatively small regions on the sphere.

Figure 16 shows the spherical conformal embedding of the bunny mesh. (c) is obtained by spherical Mobius transforming of (b). We can tell that the head part is zoomed in uniformaly. On the conformal texture mapping result (d), the

texture scaling is quite different.

We define the stretching factor function as the scaling factor of the first fundamental forms. Suppose the first fundamental form of the mesh is ds^2, and (u, v) are the conformal coordinates (or texture coordinates in our case), then

$$ds^2 = \lambda(u, v)^2(du^2 + dv^2), \tag{66}$$

where $\lambda(u, v) : M \to \mathbb{R}$ is the stretching factor function.

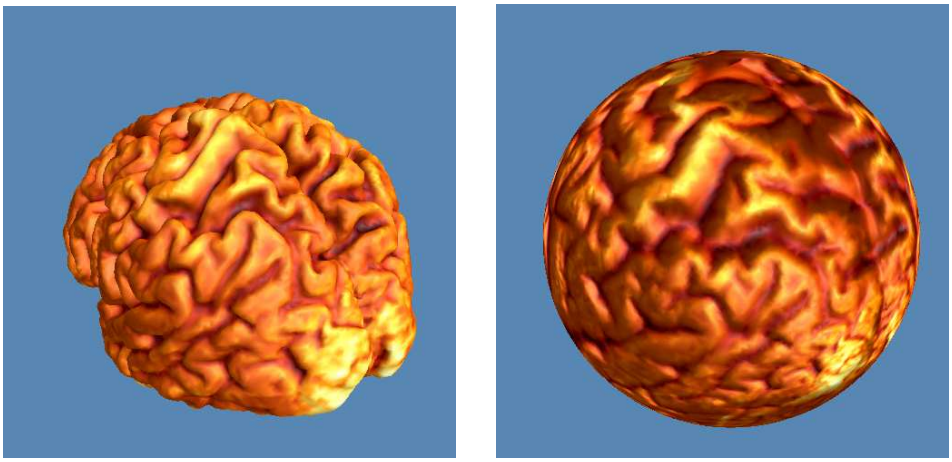
The stretching factor is distributed non-uniformly. On the ears of the bunny, stretching factors change rapidly. This is indicated by the texture on those parts. In general, any extruding parts have greater gradients of stretching factors. In practice, multi-resolution texture mapping should be applied to mitigate this problem.

Figure 18 shows the spherical conformal mapping of a brain mesh and its conformal texture-mapping. (a) and (b) are the same brain scanned twice. There are some minor deviations caused by the noise during data aqquisition and inaccuracy of reconstruction algorithms. (c) and (d) are the spherical conformal mapping results of (a) and (b) respectively. Because a conformal mapping is a local scaling of the first fundamental form, it preserves the shape locally. There is no distortion between the mesh and its image on a sphere. By comparing figures 18 (a) and (c), one can find the correspondence of major geometric features easily. Also, the conformal parameterization is stable, the small deviation on the original mesh does not affect the final parameterization too much.

Figure 17 demonstrates that the conformal parameterization is resolution independent. (a) is the original brain mesh with 50000 faces. (c) is a simplified version of (a), with only 20000 faces. (b) and (d) are the conformal embedding of (a) and (c) respectively. By comparing (b) and (d), we can recognize the correspondences of major features of the brain. Figure 17 (e) and (f) demonstrate that the embedding is conformal. We stereographicly project the sphere to the

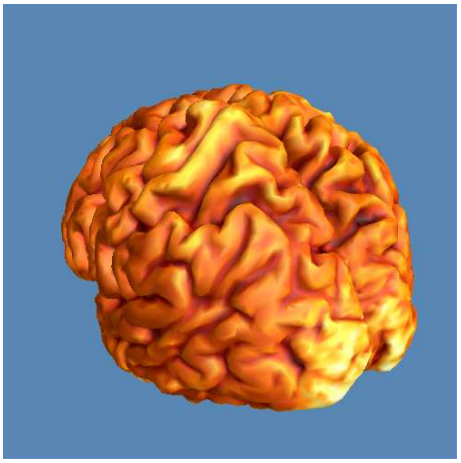
complex plane, use planar coordinates as texture coordinates, and conformally map the plane to the sphere, as shown in (f). Then we assign the texture coordinates of the sphere to the brain mesh, and texture map the brain mesh using the checker board texture. It is clear that all the right angles of each square are preserved on the highly convoluted brain surface. Figure 19 also demonstrates the mapping's independence of resolution and triangulation on a male face.

I implemented the algorithm in Haker's paper [20] on the same bunny mesh model in order to compare the efficiency and convergence of the algorithms, the program ran for hours and did not converge.

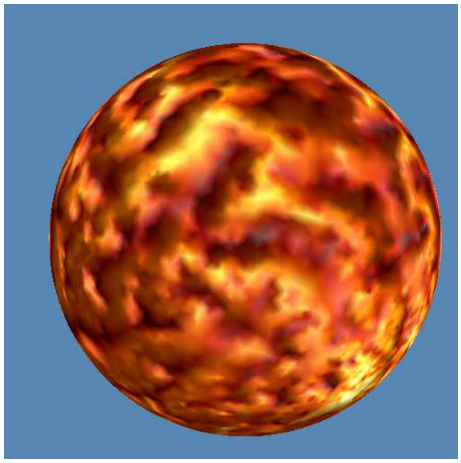


(a) 50K faces mesh

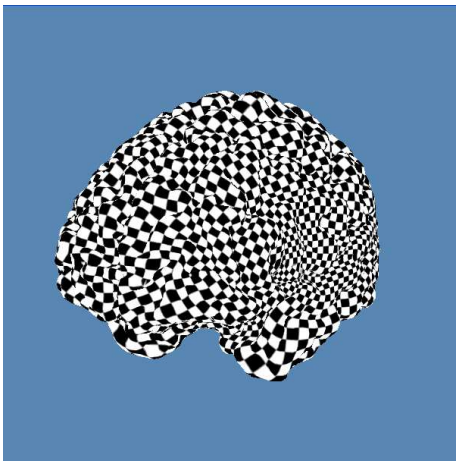
(b) Spherical conformal map of (a)



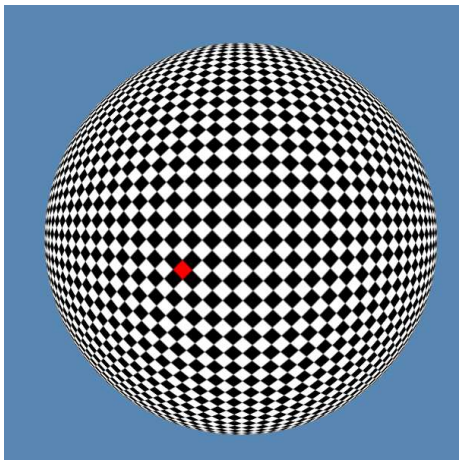
(c) 20K faces mesh



(d) Spherical conformal map of (c)

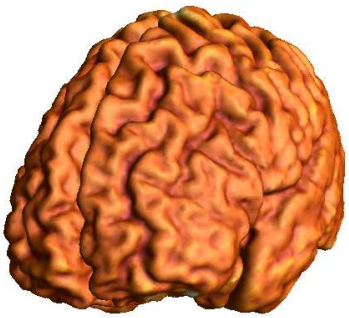


(e) Conformal texture mapping

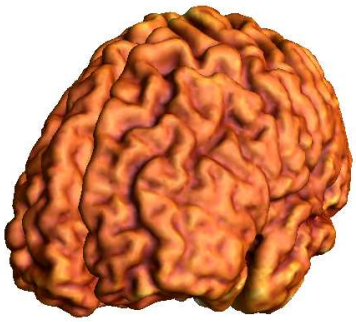


(f)Conformal texture map of the sphere

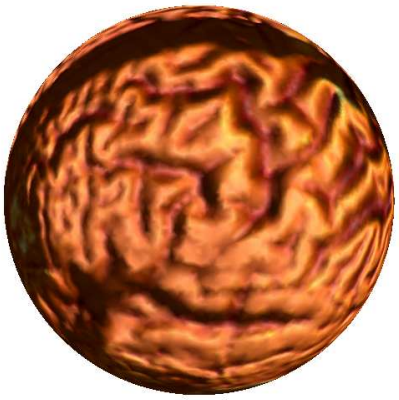
Figure 17: Conformal texture mapping of the brain mesh. Comparison of (a) and (c) shows that the conformal parameterization is independent of resoultion and triangulation. (e) and (f) show the mapping is angle preserving.



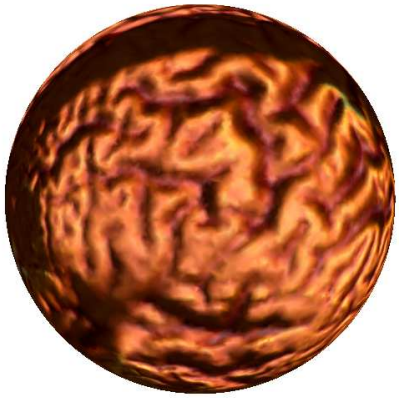
(a) Brain mesh (40K faces)



(b) Brain mesh, scanned different times

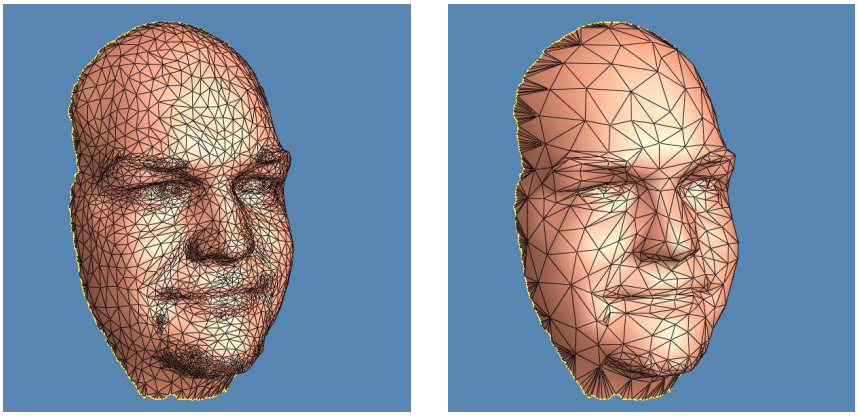


(c) Conformal map of (a) to a sphere



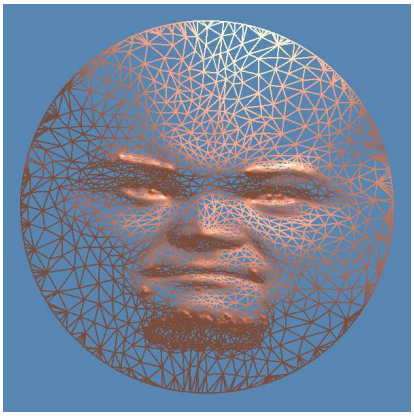
(d) Conformal map of (b) to a sphere

Figure 18: Brain meshes conformal spherical maps. The shapes of the main geometric features are preserved.

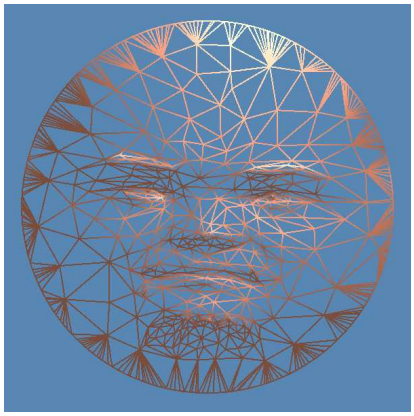


(a) Mesh with 5K faces

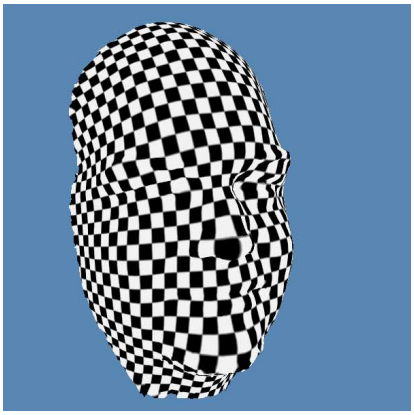
(b) Mesh with 2K faces



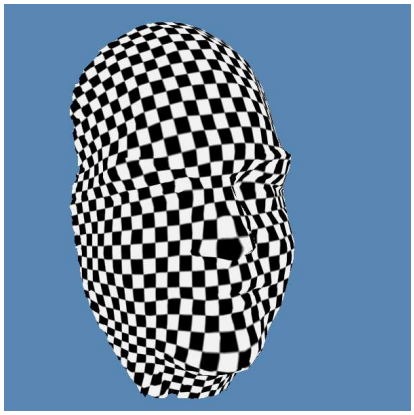
(c) Conformal mapping of (a)



(d) Conformal map of (b)



(e) Texture mapping of (c)



(f) Texture mapping of (d)

Figure 19: Conformal mapping of the same geometry with different resultions and triangulations. It demonstrates that the mapping is independent of resolution and triangulation.

6 Computing Conformal Structures for Non-zero Genus Meshes

6.1 Overview

For the purpose of computing global conformal parameterizations, we need to apply some abstract mathematical concepts. We first explain the main ideas in an intuitive way.

Instead of studying maps directly, we first study their gradient vector fields. Suppose M is a surface, $f : M \to \mathbf{R}$, then its gradient $\bigtriangledown f$ is a tangential vector field on M. f can be recovered from $\bigtriangledown f$ by integration. In [15], it is shown that a special set of curves C can be found, such that their complement M/C is a topological disk. We fix a base point p_0, for any point p, choose an arbitrary path r from p_0 to p, and integrate $\bigtriangledown f$ along r. The following relation holds as long as r has no intersection with C,

$$f(p) - f(p_0) = \int_r \bigtriangledown f dr. \tag{67}$$

The above definition is independent of the choice of r, because the divergence of $\bigtriangledown f$ is zero. For any tangential vector field with zero divergence, this definition is also valid. We call such vector fields *closed 1-forms*.

Suppose there exists a map $\mathbf{f} : M \to \mathbf{R}^2$, if $\mathbf{f} = (f_0, f_1)$ is conformal, then the gradient fields

$$\bigtriangledown \mathbf{f} = (\bigtriangledown f_x, \bigtriangledown f_y) \tag{68}$$

satisfy the following constraints,

1. Both $\bigtriangledown f_x$ and $\bigtriangledown f_y$ are harmonic.

2. At each point p, $\bigtriangledown f_y(p) = \mathbf{n}(p) \times \bigtriangledown f_x(p)$, where $\mathbf{n}$ is the normal on M.

The definition of *harmonic* in constraint (1) is given below. Suppose ω is a

vector field on M, then *harmonic energy* is defined for vector fields as

$$E(\omega) = \int_M ||\omega||^2 d\sigma, \tag{69}$$

where $d\sigma$ is area element on M, $||\cdot||$ is Euclidean norm. If ω minimizes the harmonic energy, then we call it *harmonic*. We can diffuse any closed 1-form ω to be harmonic by adding a gradient field $\bigtriangledown f$ to it, such that $\omega + \bigtriangledown f$ minimizes the harmonic energy. ω and $\omega + \bigtriangledown f$ have the same values for integration along any closed curves on M.

Constraint (2) means that $\bigtriangledown f_y$ can be obtained by rotating $\bigtriangledown f_x$ a right angle about the normal at each point on M. We call a pair of closed 1-forms (ω_0, ω_1) *holomorphic 1-form*, or *holomorphic differential* if they satisfy the above constraints. Our goal is to find all holomorphic 1-forms on M. In Chapter 3, we have proven that all harmonic 1-forms on M form a linear space, the dimension of which is two times of the genus of M. Constraint (2) is also linear. Therefore, there is a linear vector space of all holomorphic forms on M. We introduce a method to find a set of bases of this space, construct holomorphic 1-forms by linearly combining the bases, then build conformal maps by integrating them. Amazingly, for a genus g surface, the dimension of this holomorphic one-form space is $2g$. Furthermore, each base corresponds to one handle respectively.

For the genus one case, we find two oriented loops a, b on M, a, b intersect at only one point p, cut M along them and open M to a topological disk D, the boundary of D is $aba^{-1}b^{-1}$, where a^{-1} is the reverse of a, and the neighborhood of p is split to four corners. Then we map D to a unit square, corner to corner, boundary to boundary. Suppose the map is $\mathbf{f} : D \to \mathbb{R}^2$, $\mathbf{f} = (f_0, f_1)$, then we can diffuse $\bigtriangledown f_0$ and $\bigtriangledown f_1$ to harmonic 1-forms (ω_0, ω_1). $\{\omega_0, \omega_1\}$ are the bases of harmonic 1-forms on torus M. In order to meet constraint (2), we rotate ω_0 at each point $\mathbf{n} \times \omega_0$. The rotation does not change the harmonity, the resulting 1-form is still harmonic. So we can find a linear combination of ω_0 and ω_1,

$p, q \in \mathbb{R}$, such that

$$p\omega_0 + q\omega_1 = \mathbf{n} \times \omega_0. \tag{70}$$

Then $(\omega_0, p\omega_0 + q\omega_1)$ is a holomorphic form.

For the genus $g > 1$ case, we can select one handle at each time, and "collapse" the other handles. We treat the collapsed M as a torus, and compute a holomorphic 1-form on it. The g holomorphic 1-forms corresponding to g handles are the bases of all holomorphic 1-forms on M.

The pipeline of our algorithm is as follows: First we locate the handles, then open each handle, map it to a unit square, then diffuse the gradient vector fields of these maps to harmonic 1-forms. Finally, we rotate the harmonic forms and pair them to holomorphic forms.

The rigorous definition of handles is *homology group*. The linear space structure of tangential vector fields is described as *De Rham cohomology group*. The gradient, divergence operators are the sepcial cases of *exterior differentiation*. The operator to rotate a vector field is *Hodge star operator*.

The above discussion assumes that the surface M is smooth. In practice, surfaces are represented as piecewise linear meshes. We approximate the smooth vector fields by piecewise linear vector fields. It is natural to use simplicial homology and cohomology to represent the algorithms. But the computations are derived from De Rham cohomology.

For non-zero genus meshes, the computation of conformal structure is much more complicated. The goal is to find the complete bases of the holomorphic 1-form group. The algorithm can be summarized in the following steps:

Input A mesh M

Output A set of bases of holomorphic differentials.

1. Compute homology group bases $B = \{e_1, e_2, \cdots, e_g, e_g{+}1 \cdots, e_{2g}\}$.
2. Compute cohomology group bases $\Omega = \{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ which are the dual of B.
3. Compute harmonic 1-forms $\zeta = \{\zeta_1, \cdots, \zeta_{2g}\}$, such that ζ_i is homologous to ω_i.
4. Apply Hodge star on ζ_i, and compute holomorphic 1-forms

$$\zeta_i + \sqrt{-1}({}^*\zeta_i).$$

Algorithm 3. Compute holomorphic differentials

The following subsections explain each step in details.

6.2 Computing Homology

There are many methods for computing homology groups H_*K of a simplical complex K. In our implementation, we use the classic algorithm, which is based on reducing boundary operator matrices ∂_q to their Smith normal forms [32]. In order to avoid the substantial computational cost of the reduction to Simth normal form, the mesh is simplified by using progressive mesh algorithm introduced in [21]. Once the homology bases B are found on the coarser mesh, they are mapped back to the finer mesh through a sequence of vertex splits. At each vertex split step, we check the neighborhood of current split vertex, and preserve the connectness of each homology base cycle in B. Finally, on the finer mesh, we use the Dijkstra algorithm to shorten each base cycle, and perturb them such that they intersect transversely.

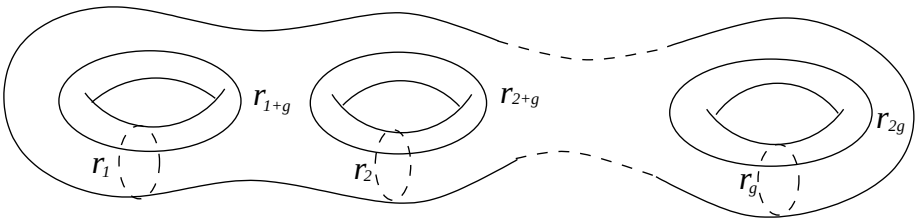


Figure 20: Homology basis curves of a genus g surface

For genus g surface M, there are $2g$ homology classes in H_1. Becuase geometric data are represented as triangular meshes, they have a natural simplicial complex structure. Denote a vertex as σ_i^0, an edge as σ_j^1, a face as σ_k^2

$$\partial_1 \circ \partial_2 = 0. \tag{71}$$

We represent boundary operators by integer matrices,

$$\partial_k = \begin{bmatrix} [\sigma_0^k : \sigma_0^{k-1}] & [\sigma_0^k : \sigma_1^{k-1}] & \cdots & [\sigma_0^k : \sigma_n^{k-1}] \\ [\sigma_1^k : \sigma_0^{k-1}] & [\sigma_1^k : \sigma_1^{k-1}] & \cdots & [\sigma_1^k : \sigma_n^{k-1}] \\ \cdots & \cdots & \cdots & \cdots \\ [\sigma_m^k : \sigma_0^{k-1}] & [\sigma_m^k : \sigma_1^{k-1}] & \cdots & [\sigma_m^k : \sigma_n^{k-1}] \end{bmatrix} \tag{72}$$

where $[\sigma_i^k : \sigma_j^{k-1}]$ is $+1$, -1 or 0.

Then homology group H_1 is defined as

$$H_1(M, Z) = \frac{Ker\partial_1}{Img\partial_2}. \tag{73}$$

In general cases, the canonical base curves are like the ones in figure 20.

By computing the Smith canonical forms of these matrices, we can get the kernal space and image space of the boundary oerpators. Therefore, we can find the bases in the homology group by taking the quotient.

The bases obtained so far are not canonical. The following algorithm will transform $(b_1, \cdots, b_{2g})$ to a canornical one. We represent each chain b_i as a cyclic

list of vertices $\{v_1, v_2, \cdots, v_m\}$. First we compute the intersection number of two chains.

Input Two chains $c_0 = \{v_1, v_2, \cdots, v_m\}$ and $c_1 = \{u_1, u_2, \cdots, u_n\}$.

Output Intersection number of c_0 and c_1.

1. Compute common vertices $c_0 \cap c_1$.
2. For each $v \in c_0 \cap c_1$, compute its index, $ind(v) \in \{\pm 2, \pm 1, 0\}$
3. Compute intersection number $Int(c_0, c_1)$

$$Int(c_0, c_1) = \frac{1}{2} \sum_{v \in c_0 \cap c_1} Ind(v)$$

.

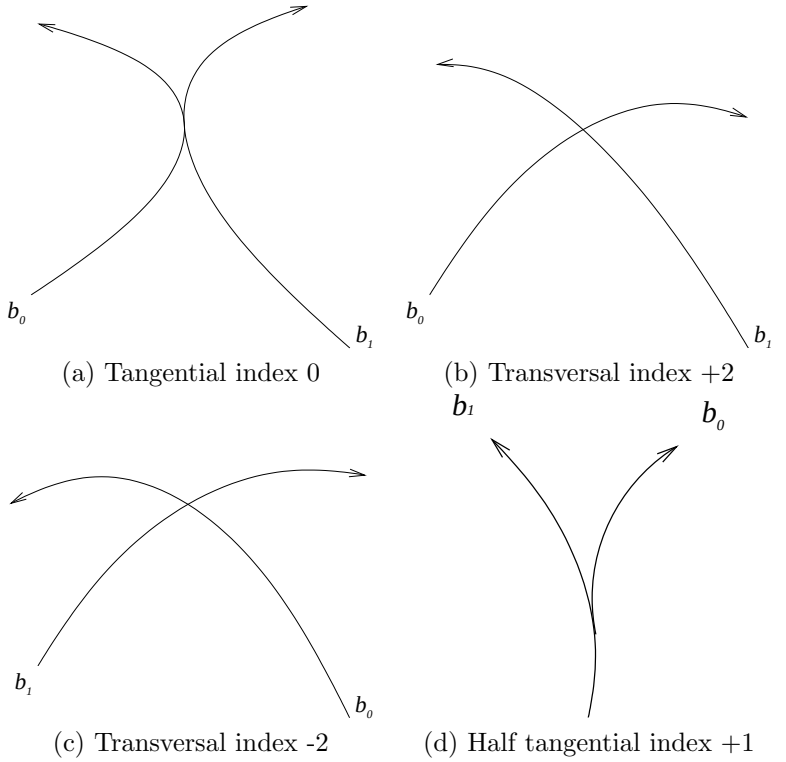
Algorithm 4. Compute intersection number of two cycles.

The index of each common vertex is computed easily. If two curves are tangentially touched, then the index is zero. If c_0 traverses c_1 from its left side to the right side, then the index is $+2$. If c_0 crosses c_1 from the right side to the left side, the index is -2. The tricky case is when c_0 and c_1 merge or split. Suppose c_0 is splitted from c_1 and goes to the right side, the index is $+1$, otherwise the index is -1. If c_0 merges with c_1 from left side, the index is $+1$, otherwise it is -1. Figure 21 depicts these different situations.

Then the intersection matrix can be constructed easily. Suppose the bases computed so far are $\Gamma = \{b_1, b_2, \cdots, b_{2g}\}$, the intersection matrix

$$\Gamma^t \cdot \Gamma = \begin{pmatrix} b_1 \cdot b_1 & \cdots & b_1 \cdot b_{2g} \\ \cdots & \cdots & \cdots \\ b_{2g} \cdot b_1 & \cdots & b_{2g} \cdot b_{2g} \end{pmatrix} = Q,$$

where $b_i \cdot b_j$ is the intersection number of b_i and b_j. Q is skymetric, $Q^t = -Q$,



(a) Tangential index 0 (b) Transversal index +2

(c) Transversal index -2 (d) Half tangential index +1

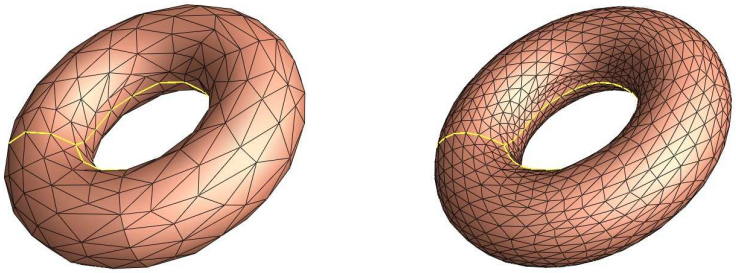
Figure 21: Index of a common point on two loops b_0 and b_1

then there exists an integer matrix A, such that

$$A^t \Gamma^t \cdot \Gamma A = \begin{pmatrix} 0 & I_g \\ -I_g & 0 \end{pmatrix} = \Lambda, A \in Z^{2g \times 2g}, \tag{74}$$

where I_g is a $g \times g$ identity matrix. Q is congruent to the normal form Λ. The congruence matrix A can be computed by using the Gaussian elimination method. Then we transform the current bases Γ to $\Gamma' = \Gamma A$, Γ' is a set of canornical bases of K.

The computation of the Smith normal form could be computationally expensive for large scale meshes. In practice, we simplify the mesh first, and use the highly simplified mesh to compute the homology. Then we map the coarser curves back to the finer mesh. The following meshes are examples of a simplified mesh, and its corresponding refined mesh.



(a) Simplified mesh with 500 faces (b) Original mesh with 4000 faces

Figure 22: Computing holomogy bases of a torus

Figure 22 shows the homology bases of a torus mesh. The original mesh has 4000 faces and it is simplified to 500 faces. The base cycles are computed on the coarse mesh then lifted back to the original mesh. Figure 26 (a) shows the homology bases of a genus 2 mesh. Figure 26 (b) shows the fundamental domain of the 2 hole torus mesh.

6.3 Computing Cohomology

Once we obtain a set of homology bases B, we can compute a set of cohomology bases Ω which is dual to B, such that

$$\int_{e_i} \omega_j = \delta_i^j. \tag{75}$$

We chose a handle and the pair of conjugate homology cycles on it, denoted as $\{e_i, e_{i+g}\}$. Then we split the mesh along these 2 cycles, map the mesh boundary to the boundary of a unit square, embed the interior to the unit square by the Floater embedding algorithm described in [13]. Suppose the embedding is (f_x, f_y) respectively. Then the 1-forms $\{df_x, df_y\}$ are the duals of $\{e_i, e_{i+g}\}$.

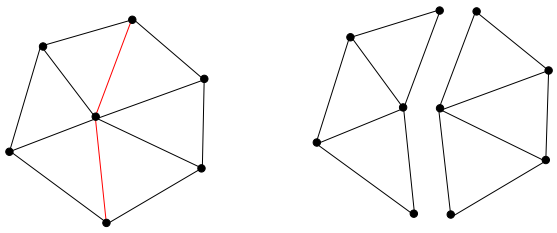


Figure 23: Wedge on mesh and splitted along the colored edges

First, we would like to splice the mesh open. The following is the algorithm for slicing a mesh.

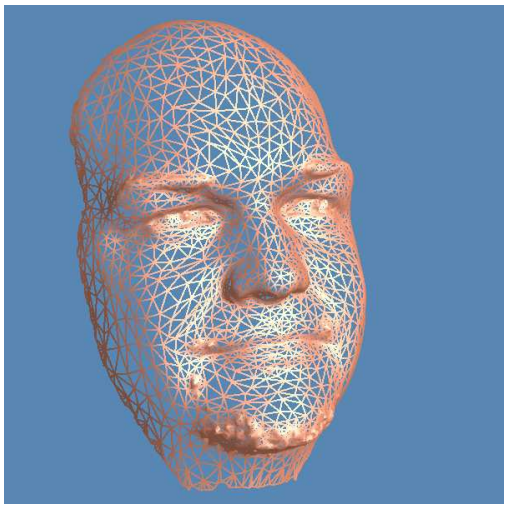
Definition 6.1 *Suppose M is a triangluar mesh, given a face f, the three angles are called corners. All the cornners adjacent to a same vertex v form a wedge w. w is associated with v.*

Input A set of curves C, C is homotopic to 0

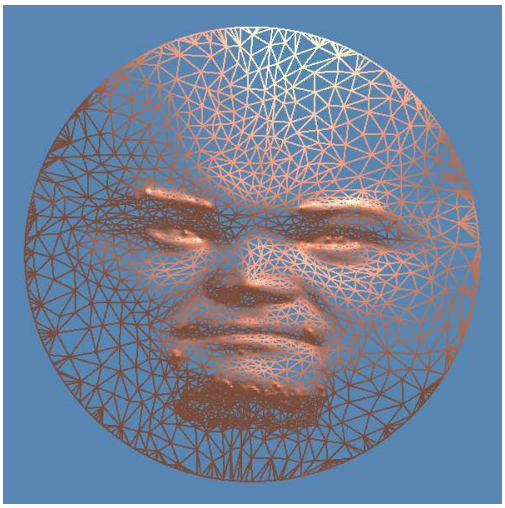
Output Open mesh with boundary along C

1. Label all edges in C.
2. For each vertex v construct a wedge w, all corners adjacent to v are also associated with wedge w.
3. Split w to w_1, w_2 along the labeled edges. Associate all corners in w_1 with w_1, all corners in w_2 with w_2.
4. Construct a new mesh M'. Vertices are all wedges on M. For each face $f \in M$, three corners are $\{c_0, c_1, c_2\}$, the associated wedges are $\{w_0, w_1, w_2\}$. Construct a face $f' \in M'$, using $\{w_0, w_1, w_2\}$ as vertices.

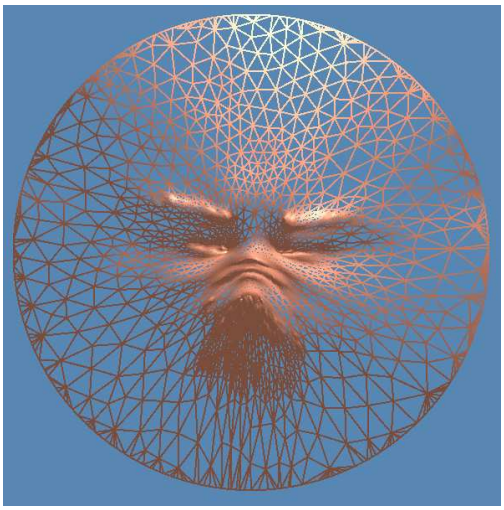
Algorithm 5. Mesh slicing



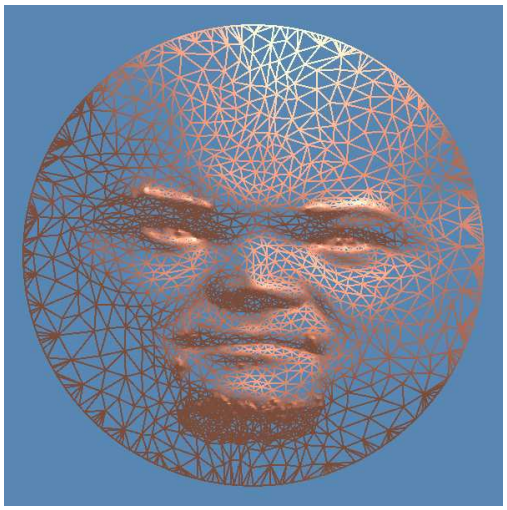
(a) Original triangular Mesh



(b) Conformal embedding



(c) Barycentric embedding



(d) Floater embedding

Figure 24: Floater embedding vs. Tuette Embedding

This algorithm is illustrated in figure 25. A pair of conjugate homology bases is drawn in (a). Then the mesh is cut open along these two cycles, as shown in (b). Suppose mesh M is sliced to M' along curve set C, then the mapping

$$\tau : Edge(M') \to Edge(M)$$

is one to one, τ is defined as follows,

$$\tau([u_0, u_1]) = [v_0, v_1] \; if u_0 \in wedge(v_0), u_1 \in wedge(v_1).$$

The immersion is computed by general Floater or Tuette embeding method. These embeddings are demonstrated in figure 24. The original mesh (a) is mapped to a disk by using Tuette and Floater embeddings with the same boundary condition. From the examples, we can see that the Tuette parameterization tends to shrink exponentially (c) and the Floater parameterization is shape preserving (d), which has rigorous meanings. If the mesh is a convex polygon, the range is also a convex polygon. If the boundary mapping is affine, then the whole mapping is affine.

Given mesh M, with ∂M simple connected, the Floater (Tuette) immsersion $\phi : M \to [0,1] \times [0,1]$ uniquely exists. The entire algorithm for computing comhomology bases is

Input Homology bases $\{r_0, r_1, \cdots, r_{2g}\}$

Output Dual cohomology bases $\{\omega_0, \omega_1, \cdots, \omega_{2g}\}$

1. Select a pair of homology bases $\{r_i, r_{i+g}\}$, slice M along them, get M'.
2. Compute the Floater immersion ϕ.
3. Suppose $\phi = (\phi_x, \phi_y)$, compute 1-forms $\omega_x = d\phi_x$,$\omega_y = d\phi_y$.
4. Pull back ω_x, ω_y from M' to M,

$$\omega_i \leftarrow \omega_x \circ \tau^{-1}, \omega_{i+g} \leftarrow \omega_y \circ \tau^{-1}.$$

5. Repeat until all homology bases are processed.

Algorithm 6. Compute dual cohomology bases

In Figure 25 (c) the entire mesh is mapped to the unit square. The mapping is of degree one but not an immersion. The second handle of the mesh is collapsed to the central region, which is shown in (d) with that region zoomed in.

6.4 Diffusion

Suppose the cohomology bases of mesh M are $\Omega = \{\omega_1, \omega_2, \cdots, \omega_{2g}\}$, we deform them to harmonic forms by adding exact 1-forms δf_i, where $f_i \in C^0K$, such that $\omega_i + df_i$ minimizes the harmonic energy in equation 29. So the Laplacian is zero

$$\Delta(\omega_i + \delta f_i) = 0. \tag{76}$$

Then for each vertex $u \in M$,

$$\sum_{[u,v]\in K} k_{u,v}(\omega_i[u,v] + f_i(u) - f_i(v)) = 0. \tag{77}$$

We construct a linear system to solve f_i. Define matrix D as follows: If $u \neq v$ and $[u, v]$ is an edge of M, then

$$D_{u,v} = -k_{u,v}. \tag{78}$$

If $u = v$, then

$$D_{u,u} = -\sum_{u \neq v} D_{u,v}. \tag{79}$$

It has been proven that the kernel space of D is

$$H = \{\lambda(1, 1, \ldots, 1)^T | \lambda \in \mathbf{R}\}, \tag{80}$$

and D is a real-valued symmetric sparse matrix, D is semi-positive definite. Suppose $Dx = 0$, then $x^T Dx = \sum k_{u,v}||x_u - x_v||^2 = 0$, so $x \in H$. Therefore D maps $H^\perp$, the orthogonal complement of H, bijectively to itself. The linear system of 77 is

$$D\mathbf{f_i} = \Omega, \tag{81}$$

where $\mathbf{f_i}$ is with entries $f_i(u)$, u is a vertex of M. The u-th entry of Ω is $\Omega_u = -\sum_{[u,v] \in K} k_{u,v}\omega[u, v]$. We need to show that Ω is in $H^\perp$, then the solution uniquely exists. $\mathbf{f} \in H^\perp$, if and only if $\sum_u f_u = 0$. Then

$$\sum_u \Omega_u = \sum_u \sum_{[u,v]} k_{u,v}\omega[u, v] = \sum_{[u,v]} k_{u,v}(\omega[u, v] + \omega[v, u]) = 0. \tag{82}$$

Hence $\Omega \in H^{\perp}$. We use the following iterative algorithm to diffuse 1-forms.

Input `1-form` $\omega \in C^1 K$.

Output `Harmonic 1-form` ω.

1 $F \leftarrow 0$.

2 Compute the Laplacian

$$\Delta^{PL} F = \sum_{[u,v] \in K} k_{u,v}(F(u) - F(v) + \omega[u,v]) \tag{83}$$

3 $F \leftarrow F - \Delta F \times \delta t$.

4 Compute the harmonic energy $E(\omega + \delta F)$.

if $\delta E < \epsilon$ then $\omega \leftarrow \omega + \delta F$, return.

5 Repeat 2 through 4.

Algorithm 7. Compute harmonic 1-Forms

This is the most time-consuming step during the whole procedure. In practice, we perform local optimization for the extruding parts on the surface. Because the matrix D is symmetric, positive semidefinite, and sparse, special linear algebra techinques can be applied to improve the efficiency of the algorithm.

6.5 Compute Holomorphic 1-Forms

Given a set of harmonic 1-form basis $\Omega = \{\omega_1, \cdots, \omega_{2g}\}$, we can construct the bases of holomorphic 1-forms directly by pairing ω_i with its conjugate ${}^*\omega_i$. ${}^*\omega_i$ is also harmonic, so it can be represented as a linear combination of ω_j's.

$${}^*\omega = \sum_{i=1}^{2g} \alpha_i \omega_i. \tag{84}$$

We consider the wedge product

$$\int_M \omega_i \wedge {}^*\omega = \sum_{j=1}^{2g} \alpha_i \int_M \omega_i \wedge \omega_j. \tag{85}$$

Then we construct the following linear system

$$W\mathbf{a} = \mathbf{b}, \tag{86}$$

where W has entries

$$w_{ij} = \int_M \omega_i \wedge \omega_j, \tag{87}$$

and vector b has entries

$$b_i = \int_M \omega_i \wedge {}^*\omega. \tag{88}$$

Because $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$ are dual to $\{e_1, e_2, \cdots, e_{2g}\}$, matrix W equals the intersection matrix, i.e. $w_{ij} = e_i \cap e_j$. Therefore W is nondegenerate, linear system 86 has a unique solution. We denote $\zeta_i = \omega_i + \sqrt{-1}\,{}^*\omega_i$ $i = 1, 2, \ldots, g$. Then ζ_i's are a set of basis of the complex linear space of holomorphic forms. The following is the algorithm to compute holomorphic 1-forms.

Input Bases of a harmonic 1-form group $\{\omega_1, \omega_2, \cdots, \omega_{2g}\}$,
a harmonic 1-form ω.

Output Holomorphic 1-form $\omega + \sqrt{-1}\,{}^*\omega$.

1. Compute $\int_M \omega_i \wedge^* \omega$.
2. Compute $\int_M \omega_i \wedge \omega_j$.
3. Solve the linear system ${}^*\omega = \sum_i \alpha_i \omega_i$

$$(\int_M \omega_i \wedge \omega_j)(\alpha_i) = (\int_M \omega_i \wedge^* \omega)$$

4. $\zeta \leftarrow \omega + \sqrt{-1}\,{}^*\omega$, return ζ.

Algorithm 8. Compute holomorphic forms

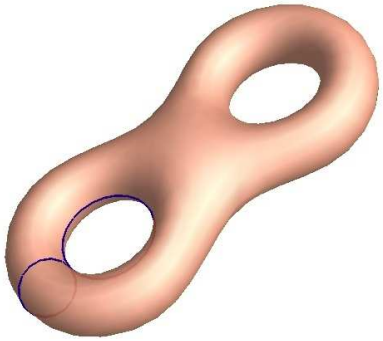
6.6 Results

By applying the above algorithm, we can compute holomorphic differentials on M. Suppose we treat the holomorphic differentials as a complex vector space, we denote a set of bases as $\{\zeta_1, \zeta_2, \cdots, \zeta_g\}$, where g is the genus of M. Figure 26 shows the results of computing holomorphic 1-forms on a genus two mesh. By integrating ζ_1 on a foundamental domain shown in (b), we map the mesh conformally to the complex plane. Then we treat the plane as the texture parameter space, texture map a checker board texture the mesh, and we obtain (c). (d) is constructed in the same way by integrating ζ_2.

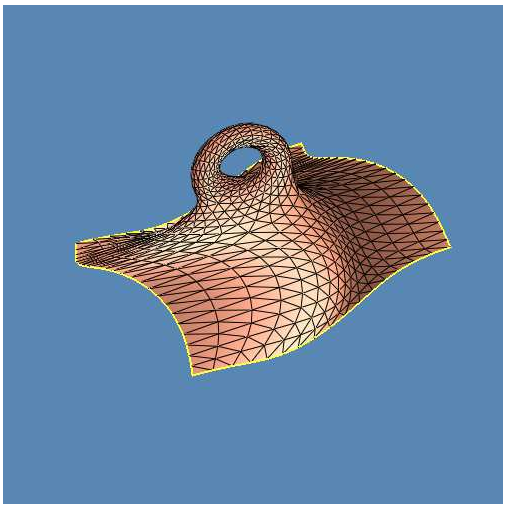
By linearly combining ζ_i's, we can construct all holomorphic 1-forms on M. By integrating holomorphic 1-forms on the fundamental domain, the mesh is globally conformally mapped to the plane with finite singularities. The number of singularities on M is $2g - 2$. Figure 27 (a) shows $\zeta_1 + \zeta_2$, figure 27(b) shows $\zeta_1 - \zeta_2$. The singularities of $\zeta_1 + \zeta_2$ are at the front and back sides of the torus. Those of $\zeta_1 - \zeta_2$ are on the left and right sides.

Figure 27 also shows the level sets of stretching factors as defined in equation 66. (c) and (d) show the stretching factor level sets on the fundamental domains embedded in the complex plane. (e) and (f) show the level sets on the meshes. These level sets have rich geometric information of mesh M.

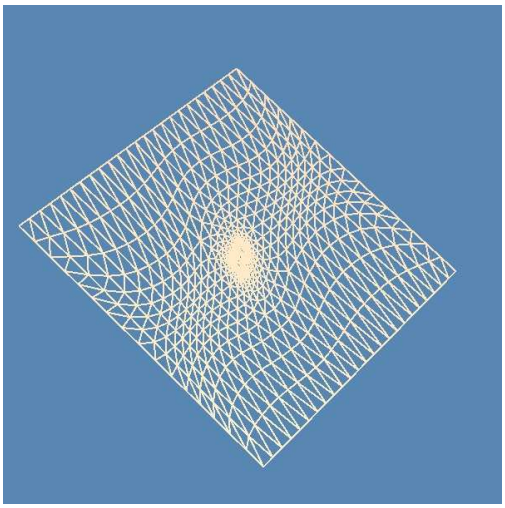
Figure 28 show holomorphic 1-forms on different surfaces. (a) and (b) show the bases of holomorphic 1-forms on a genus 2 vase surface. (c) is a complicated genus 2 mesh with knotted handles. (d) is a holomorphic 1-form visualized by texture mapping. All the parameterizations are globally conformal and boundary free.



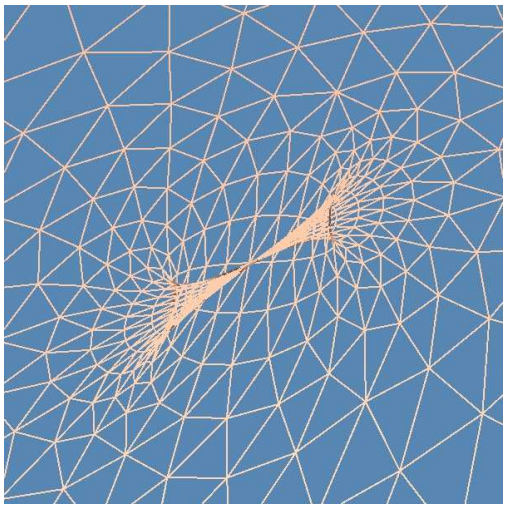
(a) A handle and homology bases on it



(b) Sliced open

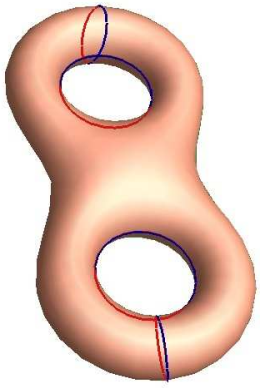


(c) Mapped to the planar domain

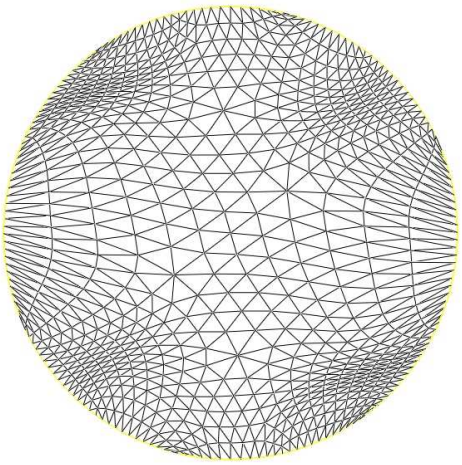


(d) Zoomed in to the other handle

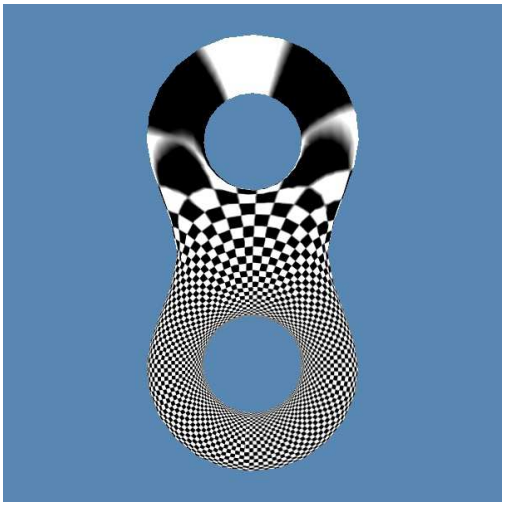
Figure 25: Computing dual cohomology



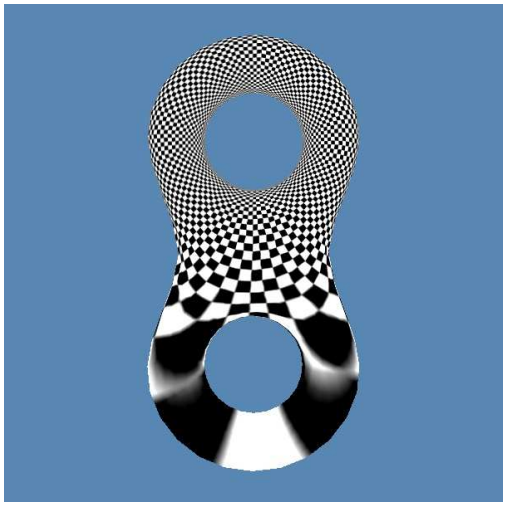
(a) Homology Bases



(b) Foundamental Domain

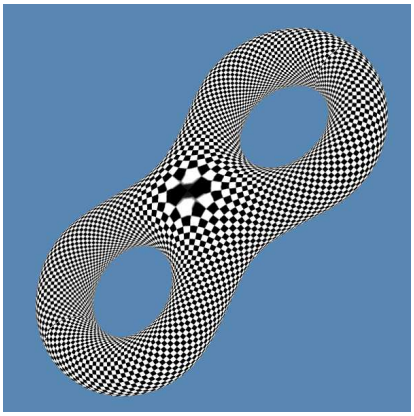


(c) Holomorphic 1-form ζ_0

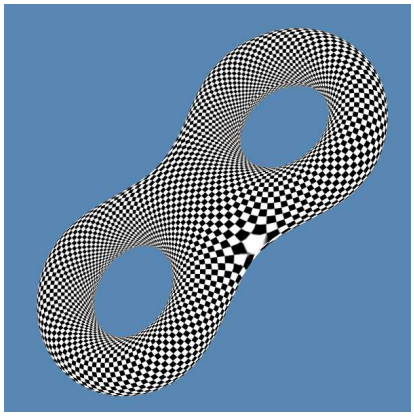


(d) Holomorphic 1-form ζ_1

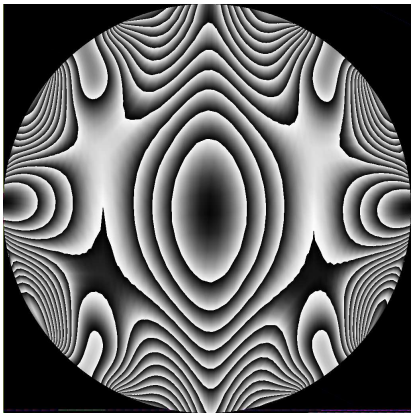
Figure 26: Homology bases and holomorphic differential bases



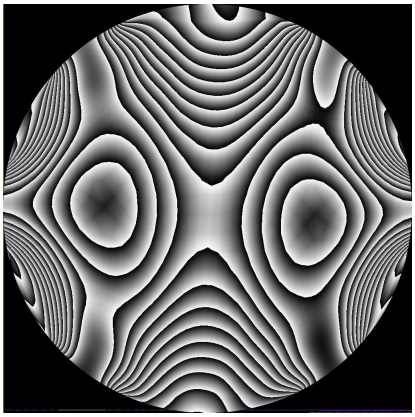
(a) $\zeta_0 + \zeta_1$



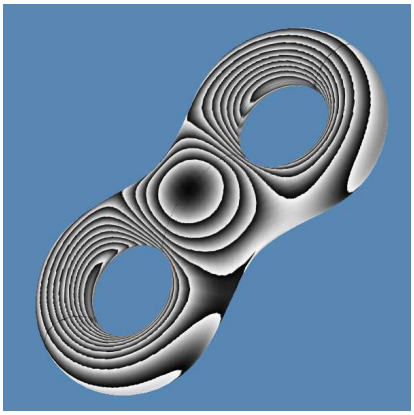
(b) $\zeta_0 - \zeta_1$



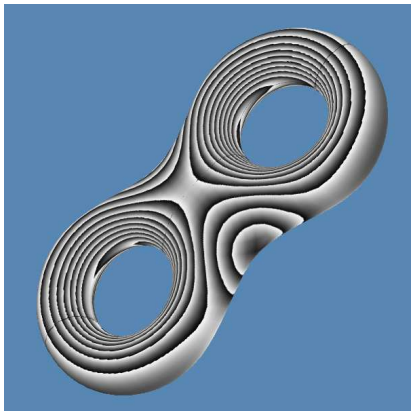
(c) Stetch level set of (a)
on a foundamental domain



(d) Stretch level set of (b)
on a foundamental domain

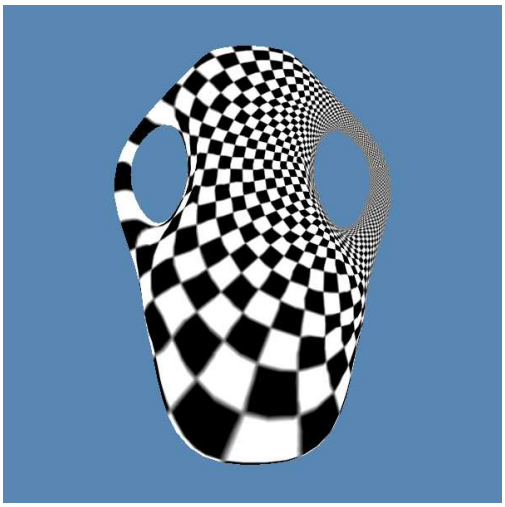


(e) Stretch level set of (a)

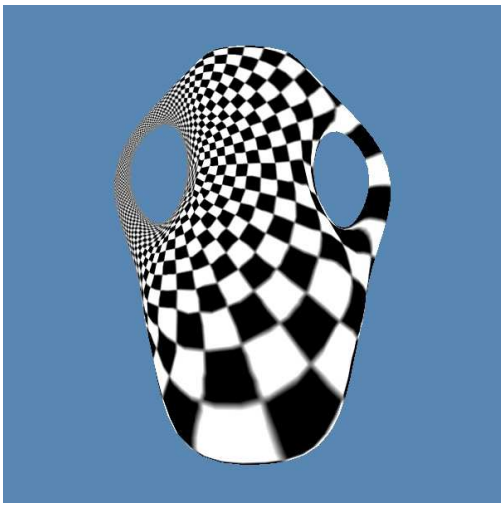


(f) Stretch level set of (b)

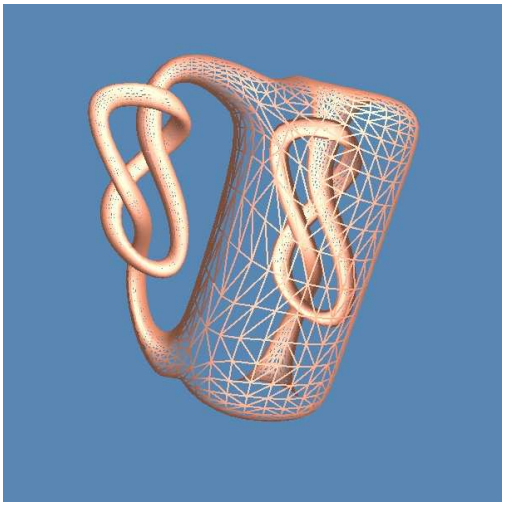
Figure 27: Holomorphic 1-form and stretching level sets of stretch factors



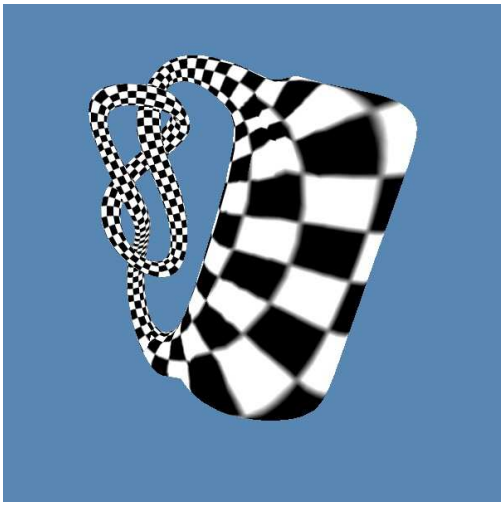
(a) Holomorphic 1-form ζ_0 for the vase mesh



(b) Holomorphic 1-form ζ_1 for the vase mesh



(c) Genus 2 knot mesh



(d) Holomorphic 1-form of knot mesh

Figure 28: Holomorphic 1-forms from different meshes

7 Global Conformal Parameterization

Our goal is to parametrize a mesh with arbitrary topology while preserving the conformality everywhere. In this section this parameterization will be described in details. The common structure of such parameterization will be explained and illustrated. In general, there must be finite number of singularities, which we call *branch points*, and each handle can be parametrized by a modular space, the curves separating the handles are the boundaries on the parameter domain, we call them *handle separators*.

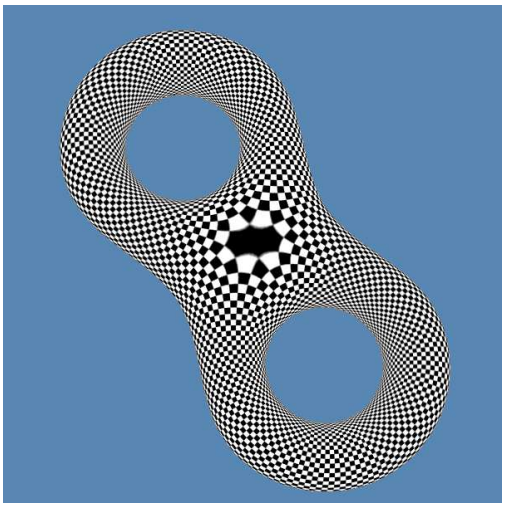
Suppose we have obtained a set of bases of holomorphic 1-forms. We can linearly combine them to get any holomorphic 1-form, and integrate it on the foundamental domain of M. The mapping from M to the complex plane is the desired global conformal parameterization. In this section we will analyze the properties of this kind of parameterization.

7.1 Branch Points

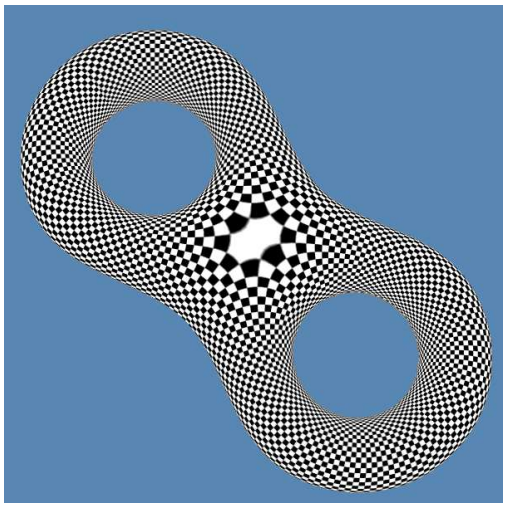
The mapping is conformal everywhere except at finite points. Suppose we choose a holomorphic 1-form $\zeta + \sqrt{-1}\,{}^*\zeta$, where ζ could be treated as a vector field on M. According to the Hope-Poincare theorem [18], ζ must have zero points if M is not homeomorphic to a torus. At the zero points of ζ, the mapping is degenerated. Such points are called *branch points*. For a genus g surface, there are totally $2g-2$ branch points. The map wraps the neigborhood of each branch points twice and double cover the neighborhood of $f(p)$ on the complex plane. In other words, if we draw a closed loop r on M around p, then the image of r $f(r)$ is also a closed loop and around $f(p)$ twice.

In order to find the branch points, we define the following streching factor for each vertex $u \in M$,

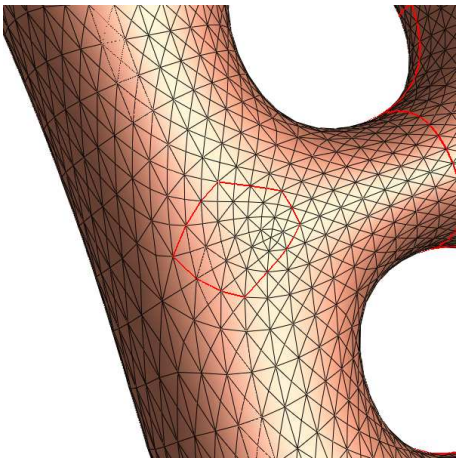
$$stretch(u) = \frac{1}{valence(u)} \sum_{[u,v]\in K} \frac{||\zeta[u,v]||^2}{|||[u,v]|||^2}. \tag{89}$$



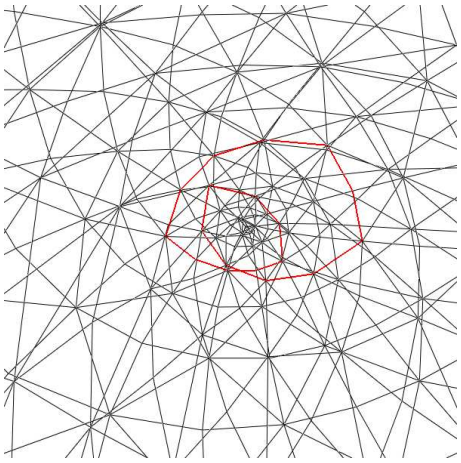
(a) Front view



(b) Back view



(c) On the surface



(d) On the complex plane

Figure 29: Branch points (degree 2)

The branch points are the vertices with local minimum streching factors. Figure 29 illustrates the branch points on a 2 hole torus. (a) and (b) show two singularities at the center of the mesh. (c) and (d) show the winding number is changed to two by the conformal mapping. Figure 31 illustrates the branch points of a genus three mesh and its conformal mapping to the complex plane.

7.2 Modular Structure of Each Handle

Given a torus M, we get the foundamental domain D, and integrate a holomorphic 1-form ζ on D, map D to a region on the complex plane. We can extend this map across the boundary of D. We fix one point $p_0 \in D$ near the boundary, and extend the map on its neighborhood N in the following way: For any point $p \in N$, find a path r from p_0 to p, and r is totally contained in N, then

$$f(p) - f(p_0) = \int_r (\zeta + \sqrt{-1}\,{}^*\zeta. \tag{90}$$

This way we can extend f to cover the whole complex plane. The images of p are a discrete point set, and they satisfy the following relation: Suppose $f^\alpha(p)$ and $f^\beta(p)$ are two images of p, then

$$f^\alpha(p) - f^\beta(p) = ia + jb,\ i, j \in N,\ a, b \in \mathbb{C}, \tag{91}$$

where a, b are two constant complex numbers, which we call the *periods* of M. The periods can be computed explicitly, suppose $\{e_0, e_1\}$ are a set of homology bases, then

$$a = \int_{e_0} \zeta + \sqrt{-1}\,{}^*\zeta, b = \int_{e_1} \zeta + \sqrt{-1}\,{}^*\zeta. \tag{92}$$

In other words, the torus is conformally mapped to a modular space $\mathbb{C}/\Lambda$, where Λ is a lattice defined as

$$\Lambda = \{ia + jb | i, j \in N\}. \tag{93}$$

The mapping is globally conformal, and boundary free.

Suppose the genus of M is greater than one. Then f still maps each handle to a modular space, but each handle has different periods. In our implementation, we can simply treat each modular space as a planar parallelogram, using a, b as its two edges. The next section will explain how to separate each handle, and map each handle to a modular space respectively.

7.3 Handle Separation

Suppose M is of genus g, the homology bases are $\{e_1, e_2, \cdots, e_{2g}\}$. There are g handles $\{h_1, h_2, \cdots, h_g\}$, such that $\{e_i, e_{i+g}\} \subset h_i$. Then f maps h_i to a modular space, represented as a paralleloogram S_i. Our goal is to separate M to handles, such that h_i, h_j are either disjoint, or intersect at a closed curve. More importantly, f is one to one and onto from h_i to S_i.

As shown in figure 30, the three hole torus is separated into three handles in different colors, each handle is conformally mapped to a modular space. The mapping across the boundary is still conformal. Figure 32 shows the modular structure for each domain. The white disks on the meshes are across the boundaries of these domains, and across the corners.

We separate the handles by the following algorithm. First we compute the fundamental domain of M, and integrate the holomorphic 1-forms. f maps D to g overlapping parallelograms, denoted as $S_1, S_2, \cdots, S_g$. If S_i is attached to S_j, there must be two branch points p and q on both of them. Find one planar curve r, from p to q, and $r \in S_i$, $r \in S_j$. Then we map back r from S_i to M get a curve R_i on M, map r back from S_j to M, we get R_j. The closed loop $R_i R_j^{-1}$ is the cycle to separate handles h_i and h_j. Figure 7.1 (a) is a genus 3 mesh. The holomorphic 1-form is visualized by conformal texture mapping. From (c) and (d) we can recognize the 4 branch points. The integration on its foundamental domain gives 3 overlapping planar domains, as shown in (b).

In figure 33, on the mesh, r is the boundary between the red handle and

the green handle. r is mapped to the middle line segment in S_0 and S_1. If we exit from the left side of r on S_0, we enter S_1 to the right side of r. If we exit from the right side of r on S_0, we enter S_1 to the left side of r. The mapping is conformal across R on M.

7.4 Algorithm for Parameterization

We summarize our procedure for conformal parameterization as follows:

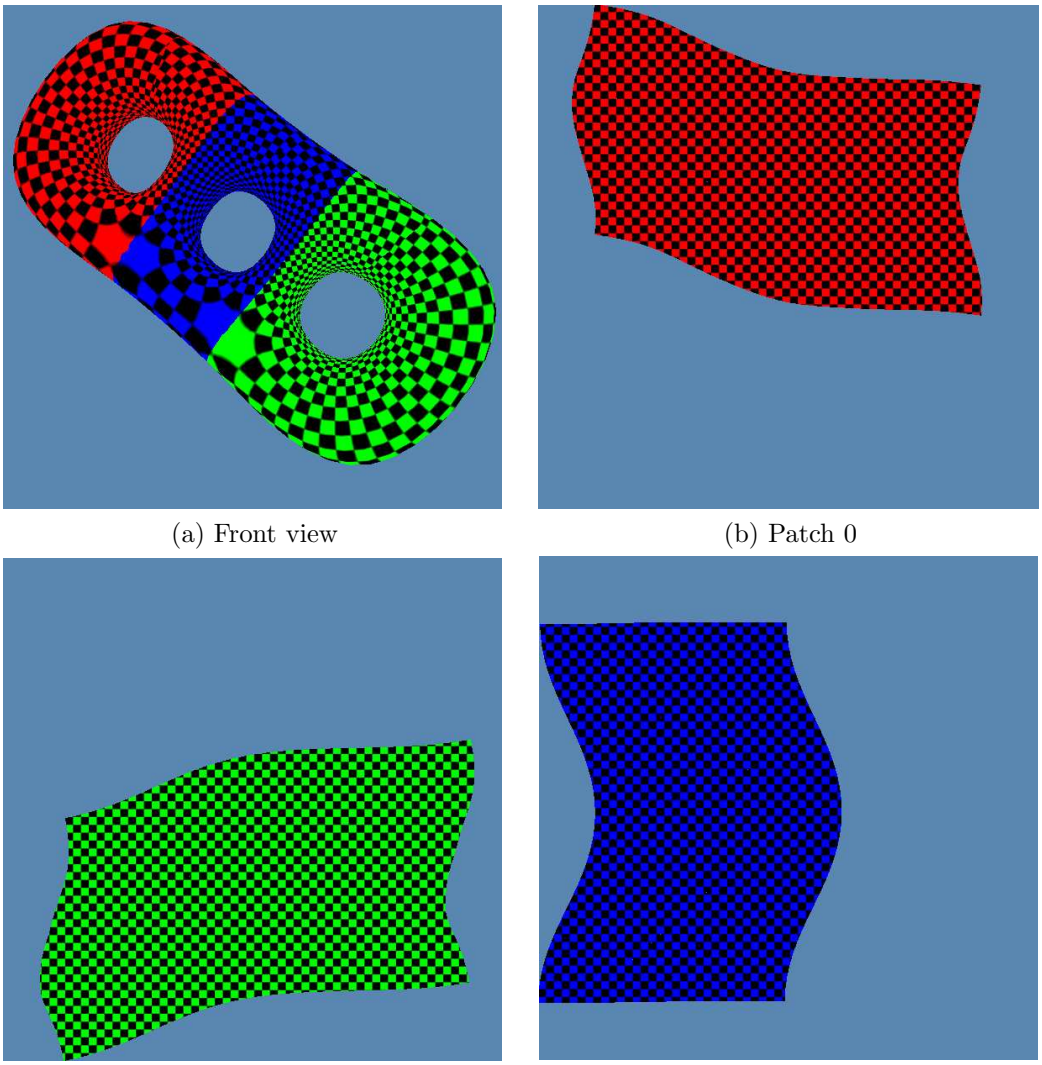
Input Mesh M

Output Conformal parameterization

1. Compute holomorphic 1-form bases.
2. Construct a holomorphic 1-form.
3. Locate the branch points.
4. Separate the handles.
5. Compute Modular space for each handle.

Algorithm 9. Global conformal prametrization

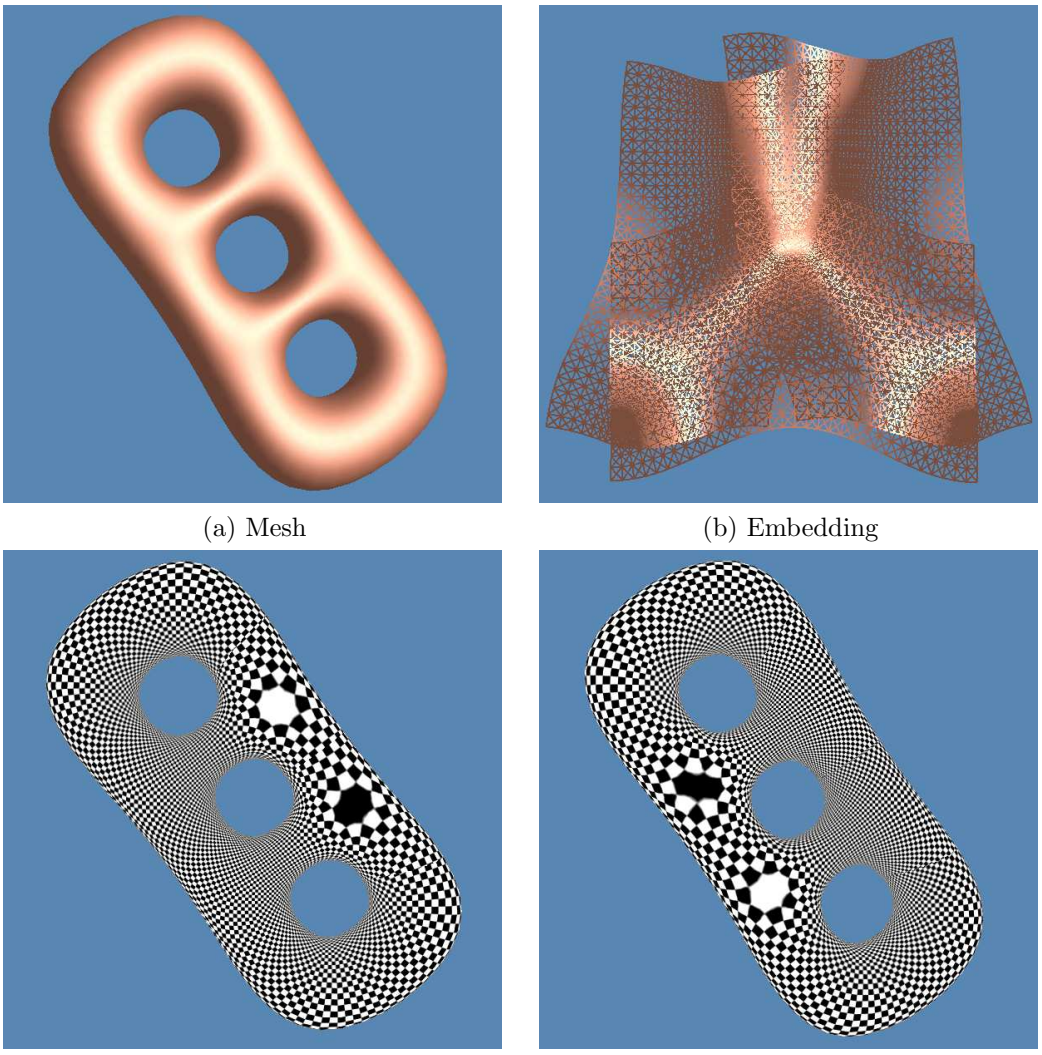
All the computations are based on mesh structures, and it is convenient to use meshes as the base structure. 1-forms are represented as a function defined on oriented edges, so we associate each edge with a real number. We approximate the branch points by vertices. The homology bases are represented as lists of edges. Each face has a unique handle id. Each corner of a face stores the coordinates on the complex plane. The modular space S_i is represented as a piecewise linear polygon on the complex plane. The boundary of S_i is $e_i e_{i+g} e_i^{-1} e_{i+g}^{-1}$. Each half edge on the boundary has a mate on the opposite side. The handle separators are also represented as a list of half edges. Suppose S_i and S_j share two branch points p,q, path $r_i : p \to q$ on S_i is the mate of $r_j : q \to$ on S_j, and vise versa, path r_i^{-1} on S_i is the mate of r_j^{-1}.



(a) Front view (b) Patch 0

(c) Patch 1 (d) Patch 2

Figure 30: Modular structures of the global conformal parameterization for each handle



(a) Mesh (b) Embedding

(c) Front view (d) Back view

Figure 31: For the genus 3 mesh, there are 4 branch points, and 3 handles are mapped to 3 overlapping patches

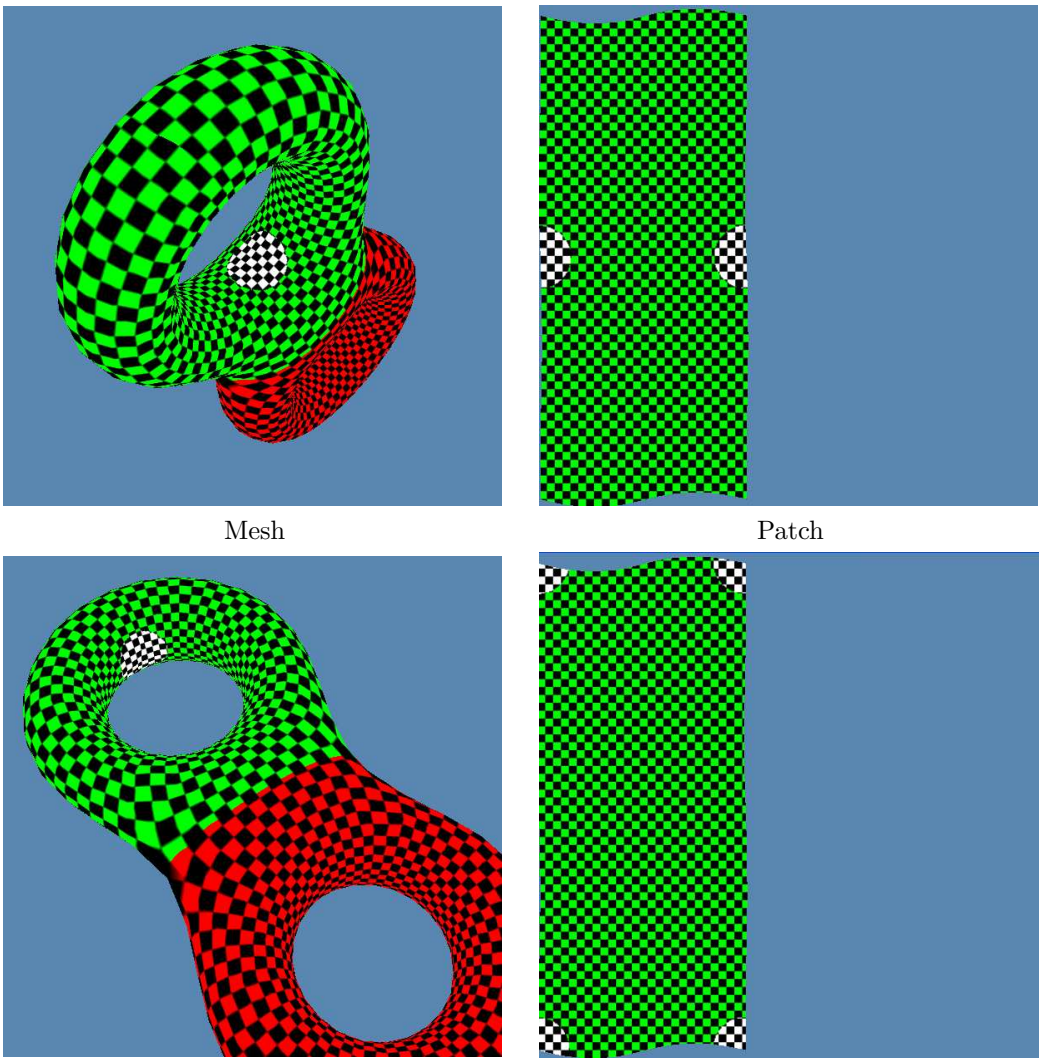
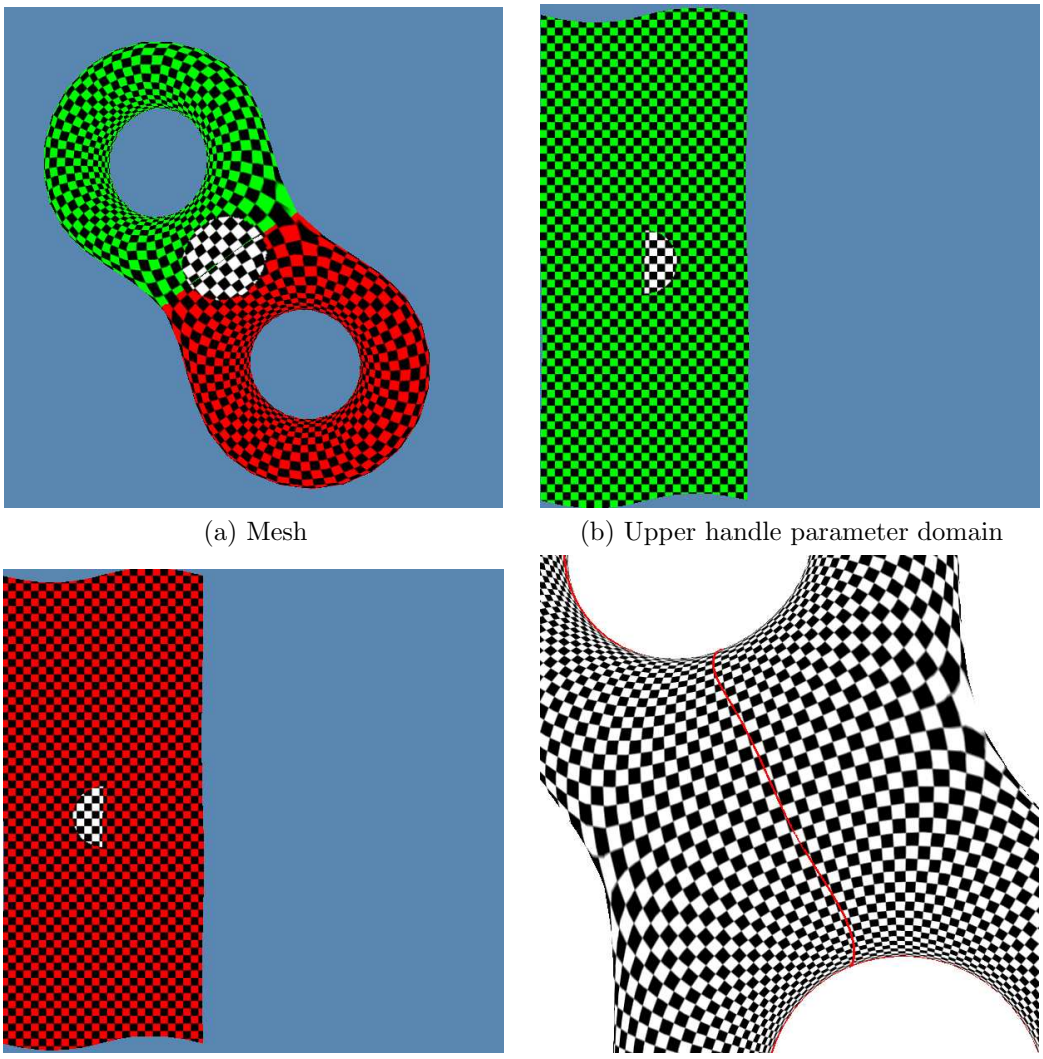


Figure 32: Moduler structure for each handle domain

7.5 Results

Some surfaces with complicated topologies are globally conformally parameterized. The results are shown in figure 34. (a) shows a surface with 3 boundaries.



(a) Mesh

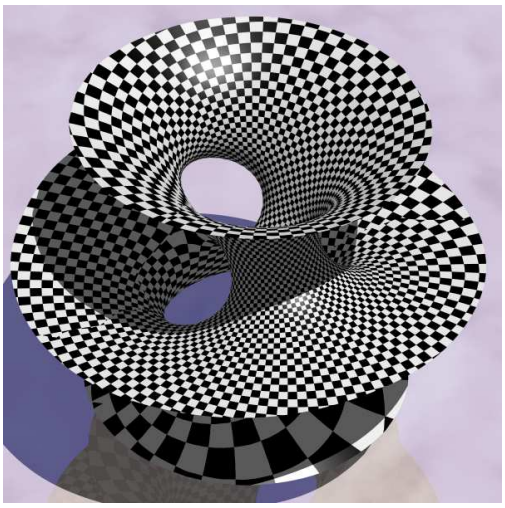
(b) Upper handle parameter domain

(c) Lower handle parameter domain

(d) Conformality preservation across boundaries (red curves)

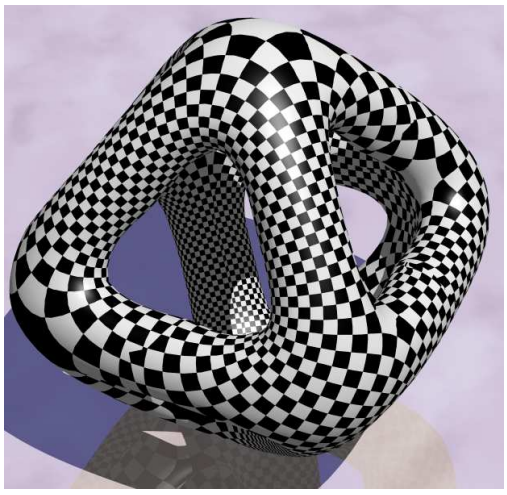
Figure 33: Separation of handles. The separator is mapped to both patch domains.

We first make two copies of it, and glue them together along these three boundaries, the obtained surface is closed and with genus four. We global conformally parameterize the double covering surface, and choose a symmetric holomorphic one-form and visualize it by texture mapping.



(a) Hyperbolic surface

(b) Knot surface

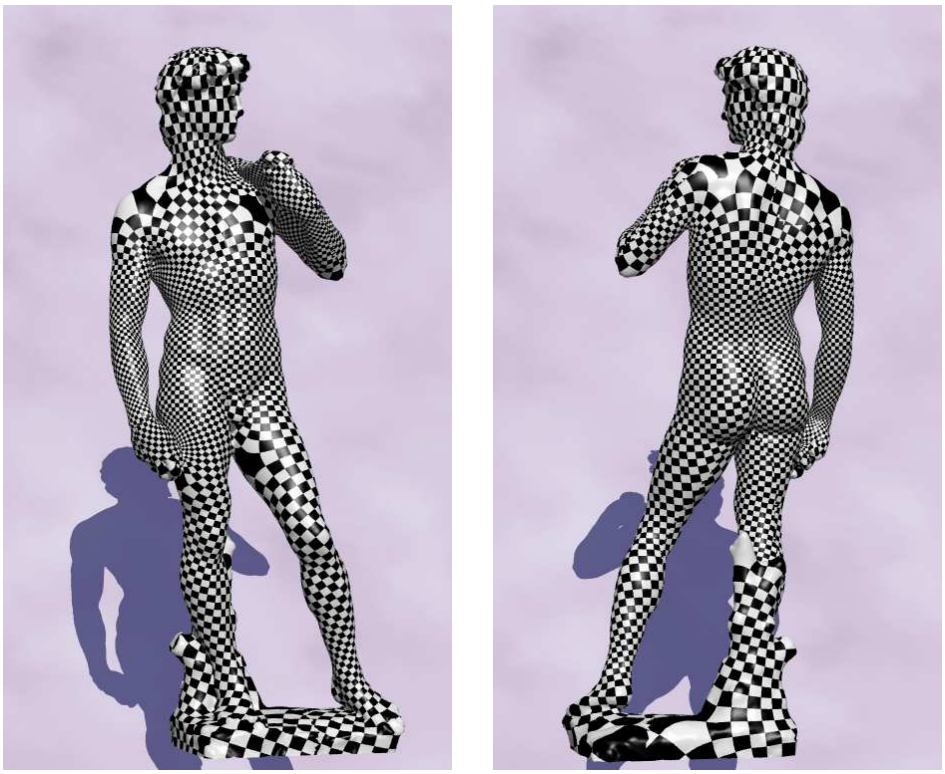


(c) Teri mesh

(d) Genus three scultpure

Figure 34: Examples of global conformal parametrizations

Figure 34 (b) illustrates a genus one surface parameterized by the method described in this work. (c) is a genus seven surface, there are totally 12 singularites, some of them can be recognized easily on the snapshot. (d) is a surface scanned from a real sculpture model, which is of genus three. The streching factors are highly nonuniform.



(a) Front view (b) Back view

Figure 35: Global conformal parameterization of David mesh

The David model is of genus three too. We make 2 holes on the surface in order to improve the uniformity of the parameterization. This only changes the topology, which will lead to big changes of the parameterization without affecting the appearance too much. One is at the bottom, the other is at the top of the head. Then we construct a double covering of the surface, which is of genus seven. From the back view, we can see there are some singularites on the shoulders and the base. The parameterization is quite uniform and conformal.

8 Performance Analysis and Applications

This section will analyze the performance of the algorithms, the major difficulties and ways to tackle them. Some potential applications are also introduced here. Some of the applications are quite novel and demand intensive research in the future.

8.1 Performance Analysis

8.1.1 Independence

The algorithm is independent of the choice of geometric realization of homology cycles, but dependent on their homology classes. In future work, we will give a method to compute a global conformal parameterization which is independent of the choice of the homology classes. Figure 36 shows the result of holomorphic one-forms using different cuts. In figure 36 (a), there are two conjugate homology cycles represented as the colored curves. In (c), one of them is deformed to keep the homology type. (b) and (d) are the resulting holomorphic forms integrated on the fundamental domain where complex coordinates are used as texture coordinates. From these figures, we can see that the texture mapping pattern and the stretching factors are very similar. We can also see that conformal mapping is global, even across the cut boundaries.

8.1.2 Extruding Region

Figure 37 shows the embedding of a cow head to the plane. It is obvious that the extruding parts, like the nose, ears and horns are mapped to relatively small regions. Those planar regions are very dense. During the optimization process, these regions converge more slowly. In general, special local optimization is necessary for these regions.

8.1.3 Triangulation

The energy form $k_{u,v}||f(u) - f(v)||^2$ is determined by $k_{u,v}$. During our experiments, we find that if $k_{u,v}$ are all positive, then the algorithm converges faster. For the harmonic energy minimization, the edge coefficients can be reformulated as

$$k_{u,v} = cot\angle\alpha + cot\angle\beta. \tag{94}$$

Here there are two faces sharing edge $[u, v]$ and α, β are the two angles in these faces opposite to the edge. For the venus model shown in figure 37 (c), the barycentric embedding converges very fast. The harmonic optimization converges much more slowly. In (c), there are too many obtuse angles on the triangle mesh, so many $k_{u,v}$'s are negative. In our implementation, we carry out some preprocessing on the meshes to swap or split edges with negative $k_{u,v}$. This process improves the convergence speed.

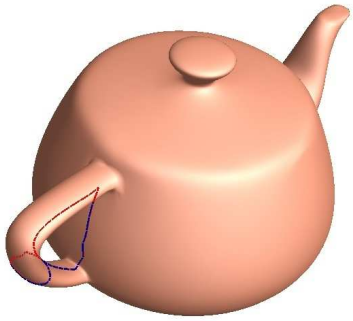
Input A Mesh M.

Output Remesh of M, such that all $k_{u,v}$'s are positive.

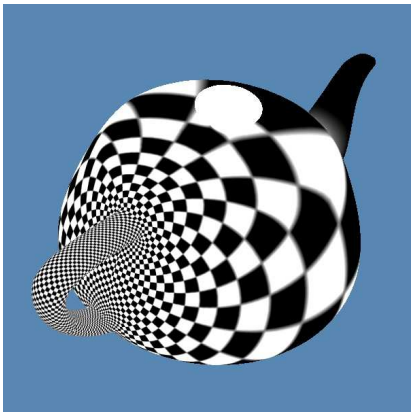
1. For each edge on M, compute $k_{u,v}$.
2. For each edge $[u, v]$ with negative $k_{u,v}$, split $[u, v]$.
3. For the edges with negative $k_{u,v}$, swap the edges.
4. Repeat 2,3 until most $k_{u,v}$'s are positive.

Algorithm 10. Remeshing to improve string constants.

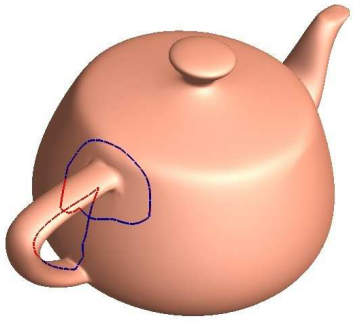
The above algorithm can improve the convergence speed in practice.



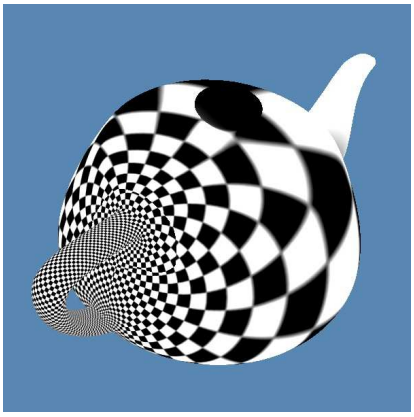
(a) Homology bases



(b) Holomorphic 1-form computed from (a)

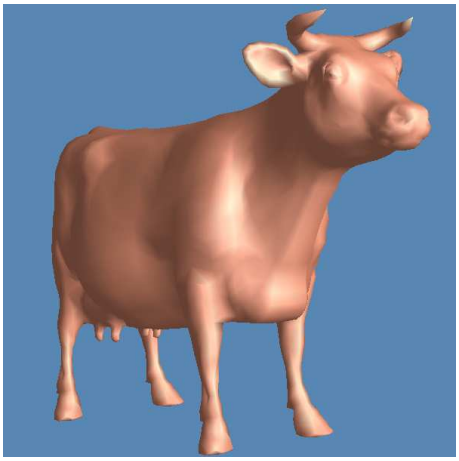


(c) Homology bases

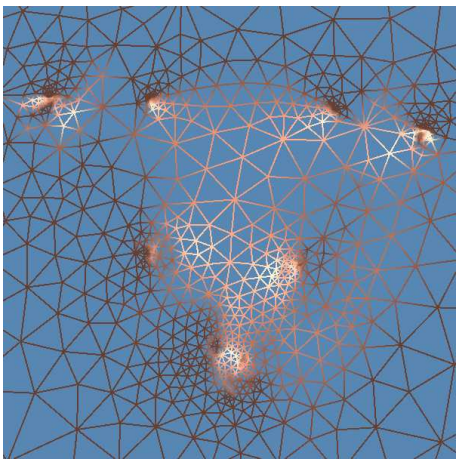


(d) Holomorphic 1-form computed from (c)

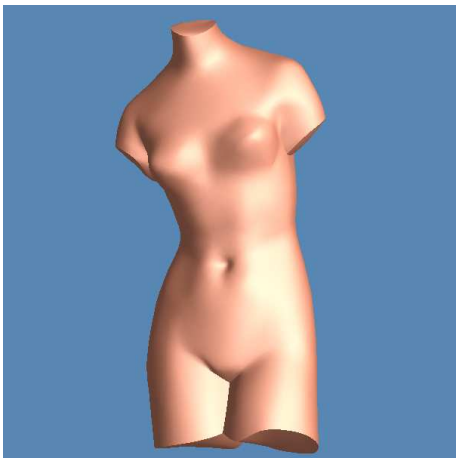
Figure 36: Boundary independent conformal mapping



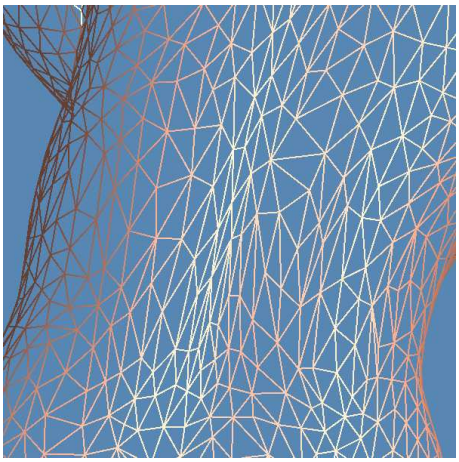
(a) Cow Head



(b) Cow head conformaly mapped to a sphere



(c) Venus mesh



(d) Obtuse angles on the mesh

Figure 37: Convergence analysis: The extruding parts converge slowly. The negative string constants cause divergence.

8.2 Potential Applications

There are many applications for global parameterizations and geometry images, because all geometric processing problem are related to parameterizations. Geometry image unifies the geometry and image representation, a lot of image processing techniques can be applied to geometry directly by utilizing this technique. The followings are a few of applications. They will be explored more in the future.

8.2.1 Constructing Geometry Images

Conformal parameterization can be applied to construct geometry images. Compared with other metrics, the conformal parameter is nonuniform, which will cause some inefficiency. But conformal parameterization does not change aspect ratio, and no skewed triangles will be generated. Therefore, the normals can be reconstructed with high quality. In [15], it is difficult to recover normals from geometry images, a special normal map has to be used to metigate this problem. In figure 41, we conformaly map the brain mesh to a sphere using the conformal map, and resample it using regular grids. We apply wavelet compression on the geometry image, and it can be seen that the reconstructed normals have very good quality. For a mesh with general shapes, we can decompose it to different submeshes with different topologies, and construct a geometry image for each submesh.

8.2.2 Texture Mapping and Synthesis

Texture mapping is the most important field in computer graphics. Because the mesh vertices resolution is much lower than the pixel resolution during rendering process, more information can be stored in texture space, and illustrated by texture mapping.

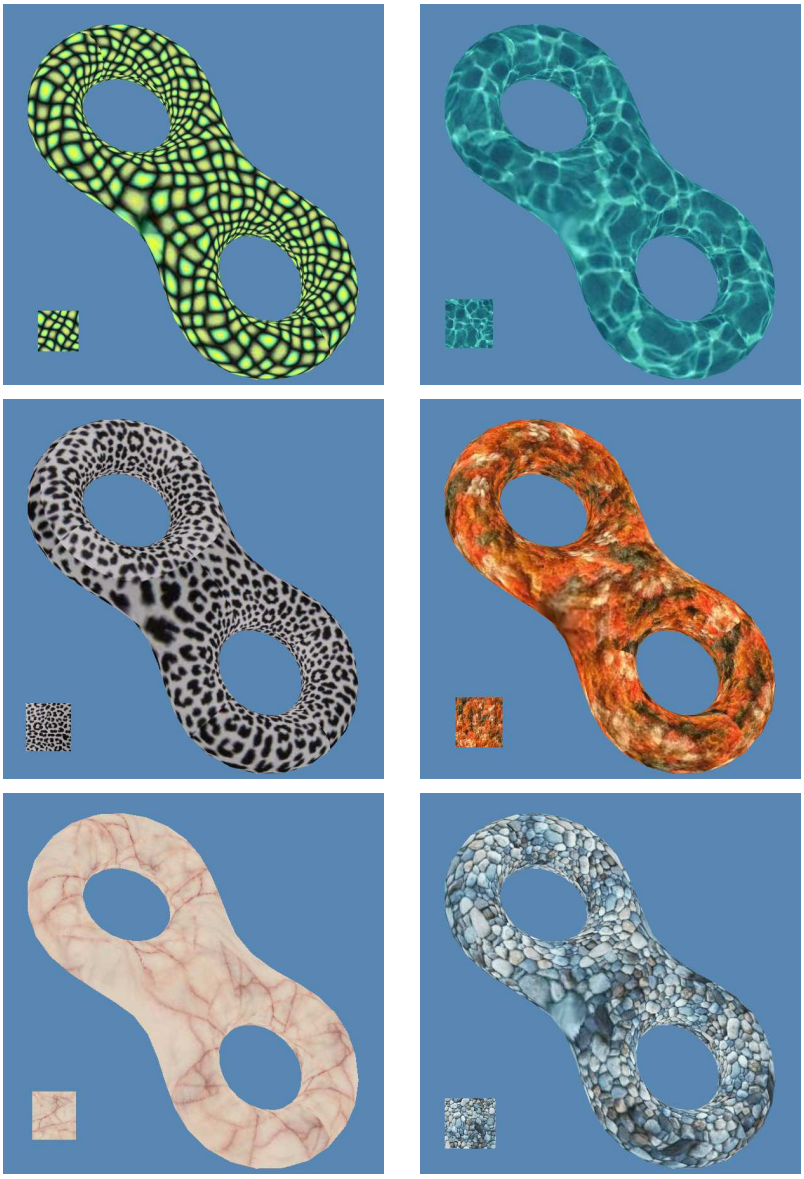


Figure 38: Surface texture mapping

Conformal parameterization is valuable for texture mapping, because it creates no distortion between the texture image and the texture on the surface. It is easy to design the texture without considering compensation for distortion.

For texture synthesis on surfaces, the conformality can simplify the synthesis process a great deal. For the interior part of each modular space, it is as simple as pasting the copies of a local texture. Special treatments are necessary for the regions near the boundaries and handle separators.

Figure 38 shows some texture mappings on the genus 2 torus. Each tile of the texture is shown on the left corner of each snapshot. The texture looks natural without distortion.

8.2.3 Surface Classification

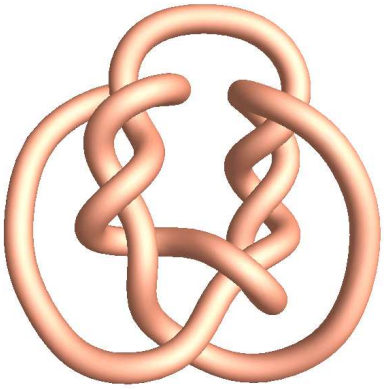
Non-zero genus surfaces can be classified by their conformal structures naturally. After the bases of holomorphic 1-form group are computed, it is straightforward to compute the period matrices. During the construction of homology bases, we can obtain a canonical set of homology bases, that is

$$\begin{cases} r_i \cap r_{g+i} &= +1, i = 1, 2, \cdots, g \\ r_i \cap r_j &= 0, j \neq g + i \end{cases} \tag{95}$$

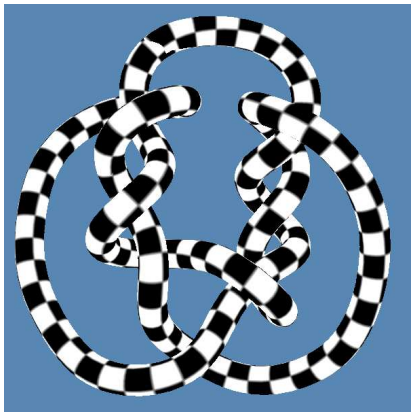
Then the period matrix is

$$\mathbf{P} = \begin{pmatrix} \int_{r_1} \zeta_1 & \int_{r_1} \zeta_2 & \cdots & \int_{r_1} \zeta_{2g-1} & \int_{r_1} \zeta_{2g} \\ \int_{r_2} \zeta_1 & \int_{r_2} \zeta_2 & \cdots & \int_{r_2} \zeta_{2g-1} & \int_{r_2} \zeta_{2g} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \int_{r_{2g}} \zeta_1 & \int_{r_1} \zeta_2 & \cdots & \int_{r_{2g}} \zeta_{2g-1} & \int_{r_{2g}} \zeta_{2g} \end{pmatrix} \tag{96}$$

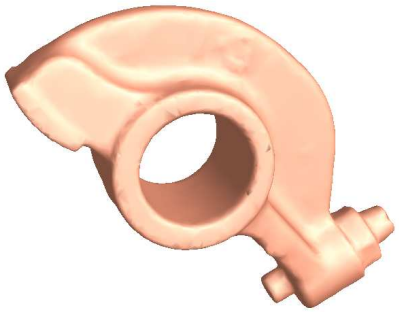
If two surfaces M_1, M_2 are conformally equivalent, then there exists an integral symplectic matrix N, such that $N^{-1}P_1N = P_2$. N is the homology bases transformation matrix.



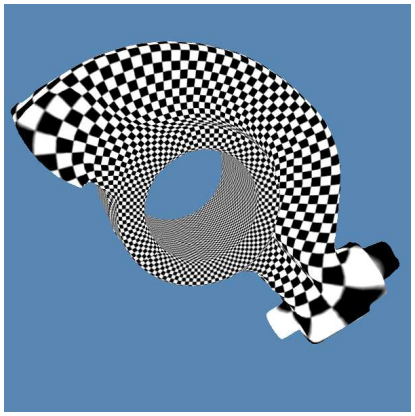
(a) Knotty Surface



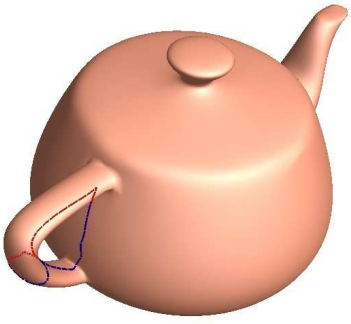
(b) Angle = 85.1, ratio = 31.150



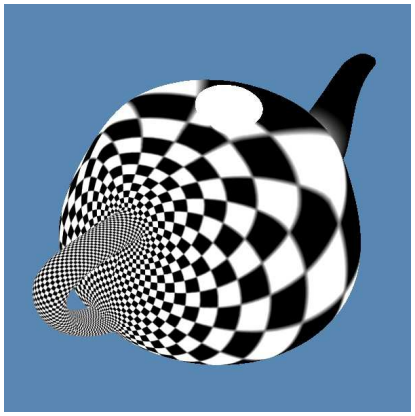
(c) Machine part



(d) Angle = 85.43, ratio = 4.9928



(e) Teapot mesh



(f) Angle = 90, ratio = 2.2916

Figure 39: Surface classification by period matrices

For the genus one case, the period matrix can be equivalently represented as shape factors. Each genus one surface can be conformally mapped to a parallelogram. The shape factors are the edge length ratio and the acute angle that determine the parralogram. The following are the shape factors of our computing results. Figure 39 shows different genus one surfaces, with different shape factors.

Shape Factors of genus one meshes

Mesh	Angle (degree)	Length Ratio	Size
Torus	89.9874	2.2916	1089 vertices, 2048 faces
Teapot	89.95	3.0264	17024 vertices, 34048 faces
Knot	85.1	31.150	5808 vertices, 11616 faces
Machine Part	85.4321	4.9928	3750 vertices, 7500 faces

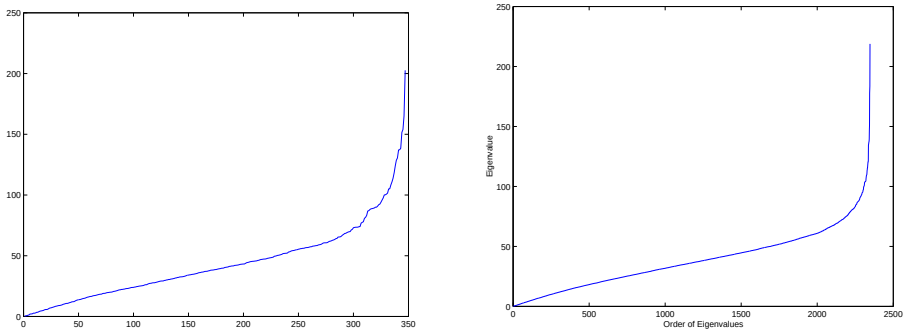
Different genus one meshes can be classified by their shape factors and differentiated without resorting to further geometric features. From the above results, the knot mesh has the greatest length ratio, which is consistent with our expectation. The teapot and the torus are symmetric, so the anglse are right angles.

8.2.4 Laplacian Spectrum Representation

According to the theorem of Laplacian eigenfuctions, all eigenfunctions form a basis of the functional space defined on the surface. Any function can be represented as a linear combination of these eighen functions. If the surface is a torus, then this setting is exactly the 2D Fourier transformation. If the surface is a sphere, then the eigenfunctions are the so called spherical harmonics.

Figure 41 illustrates geometric compression using spherical harmonic functions. The brain mesh is mapped to a sphere conformally and resampled by regular spherical grids. The reconstructed surface is compressed using spherical harmonic functions, which are the eigenfunctions of the Laplace-Beltrami operator on spheres. From the figure we can tell the reconstructed normals are with good qualities.

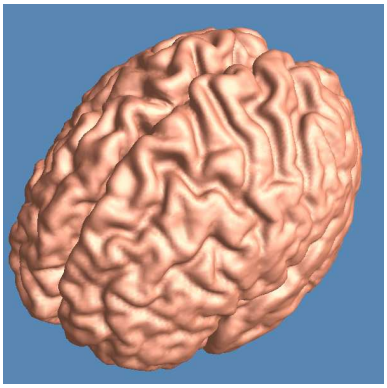
The eigenvalues of a Laplacian operator are determined by geometry, and invariant of the triangulation or resolution. Figure 40 shows two plots of eigenvalue spectrums of two meshes approximating the same geometry. Although the two meshes have quite different resolutions, their spectrum plots look similar. This demonstrates the spectrum is an intrinsic property of the geometry, and invariant of resolutions.



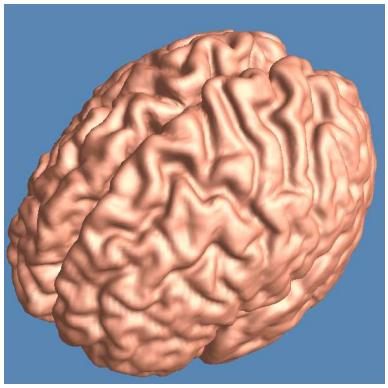
(a) Spectrum for mesh with 1000 faces (b) Spectrum for mesh with 5000 faces

Figure 40: Spectrum independent of resolution

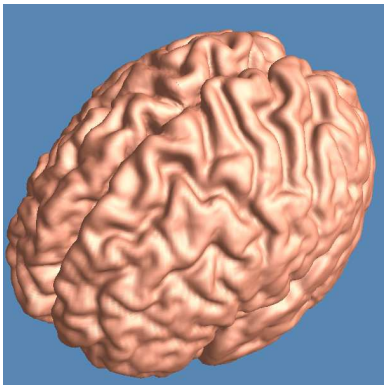
We can map a mesh M to a canonical mesh M_0 conformally, and compute the eigenfunctions on M_0, and treat M as a vector function on M_0. By computing the spectrum and filtering out high frequencies, we can conduct geometric data compression.



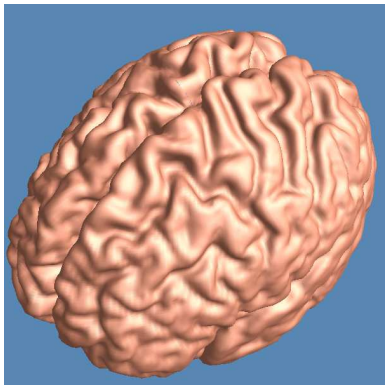
a. Original



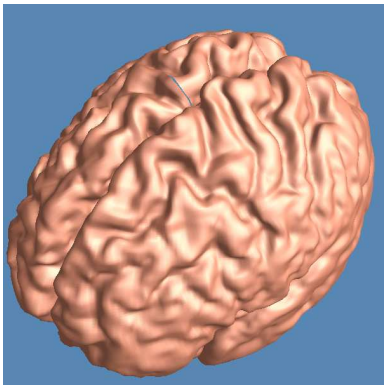
b. Compression ratio 8



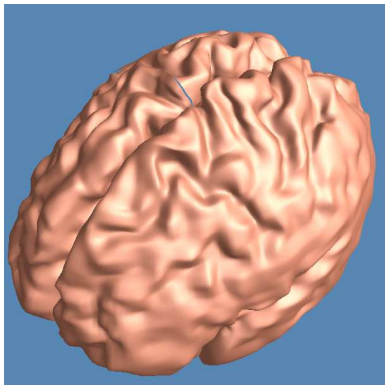
c. Compression ratio 16



d. Compression ratio 32



e. Compression ratio 128



f. Compression ratio 256

Figure 41: Surface compression

The following is an algorithm for data compression using eigenfunctions.

Input Base mesh M_0, a mesh M.

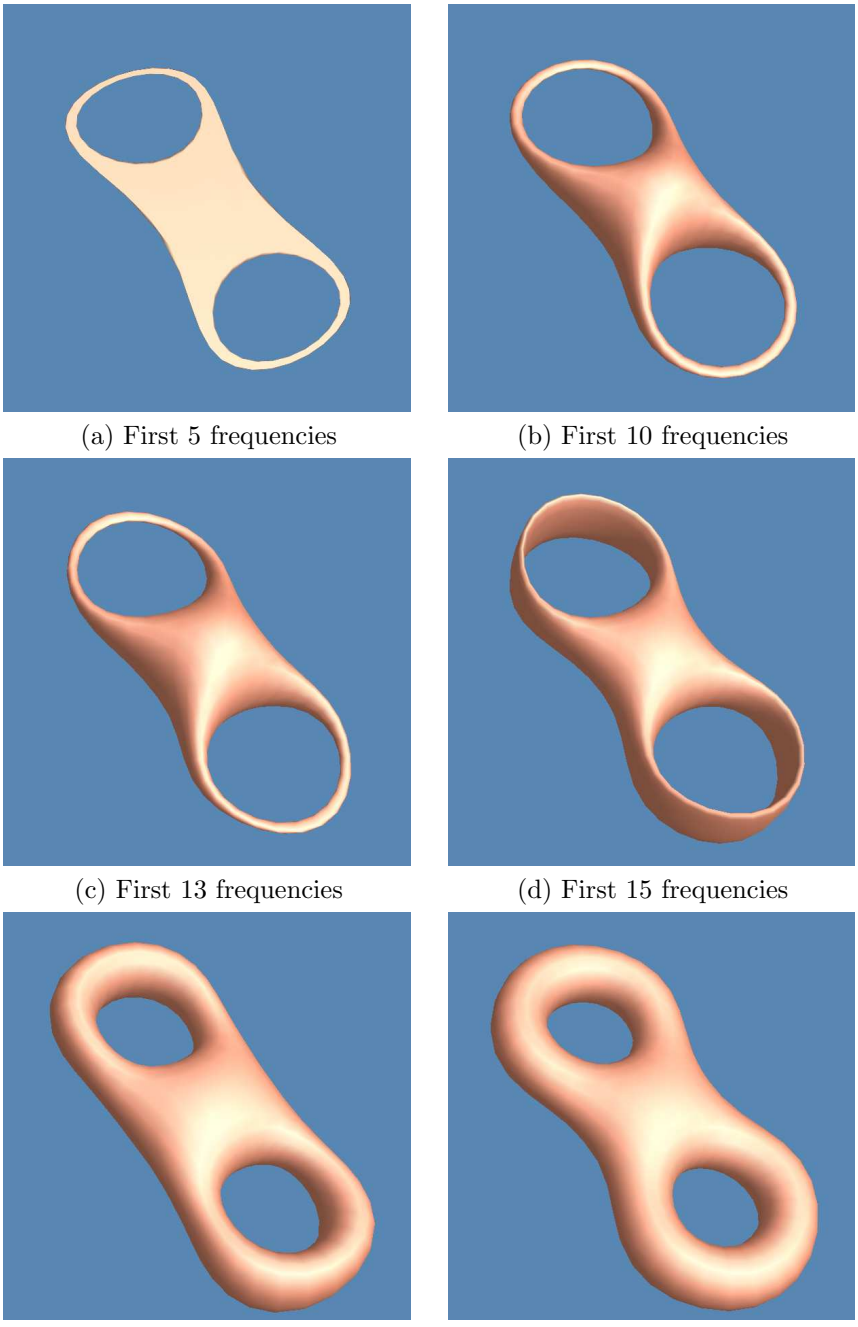
Output Spectrum of M

1. Construct the Laplacian operator matrix of M_0.
2. Compute the eigenvalues and eigefunctions of M_0.
3. Decompose M with resepct to the eigenfunctions on M_0.
4. Return the spectrum.

Algorithm 11. Harmonic anylasis

The harmonic synthesis is similar, using spectrum to linearly combine eigenfunctions. A compression result is shown in Figure 42. Different meshes are reconstructed using different lengths of the spectrums. From the figure, we can tell that only 10% of the spectrums can sufficiently reconstruct the original geometry with high fidelity. Figure 42 shows the reconstruction result from the Laplacian spectrum. The original number of vertices is 766. From the snapshots, we see that the first 30 parameters in the spectrum are good enough to reconstruct the entire mesh with very high accuracy.

This method is different from the topological Laplacian spetrum introduced in [22]. First the eigenfunctions are defined for meshes with arbitrary topologies, not only topological disks. More importantly, the Laplacian operators used in our work are geometric not topologic. If we change the triangulation of the surface, we will get the same eigenfunctions. Therefore, it is intrinsic and connectivity independent. The topological Laplacian operator depends on the triangulation.



(a) First 5 frequencies

(b) First 10 frequencies

(c) First 13 frequencies

(d) First 15 frequencies

(e) First 18 frequencies

(f) First 30 frequencies

Figure 42: Surface reconstruction from Laplacian Spectrums

9 Summary and Future Research

In this work, a set of discrete Riemann geometry concepts is established, and essensial facts are proven. Any surface admits a triangulation with all acute angles. On this kind of mesh, there exists a linear space of all discrete harmonic 1-forms, the dimension of which is two times its genus. The discrete Hodge star operator is defined, and then the real linear space of discrete holomorphic 1-forms is studied. The dimension of this space is also two times its number of genus. Furthermore, the bases are constructed explicitly.

By applying the discrete Riemann theory, a set of pratical algorithms is introduced to find the bases of the discrete holomorphic differentials. Global conformal parameterization is obtained by integrating these holomorphic 1-forms.

A new method for computing conformal map from a genus zero mesh to a sphere is also introduced. Compared to the previous approaches, this method is more stable and flexible for different applications.

There are a lot of directions for future research, the following are the major ones.

9.1 Homology Independent Global Conformal Parameterization

Current parameterization is independent of the geometric realization of homology bases, but it is dependent on the homology class of them. Once the homology bases are replaced by another set of bases, and if the transformation is nontrival, the parameterization will be different.

We can introduce an implicit metric, under which special geodesics can be found. The mesh is sliced open along these geodesics, and each patch can be mapped to anulas conformally. The conformality will be preserved across these boundaries. This global parameterization method is independent of the homology type of the bases.

9.2 Multiresolution Parameterization

From the computing results, it is clear that the streching factors vary too much for a general surface. Using a uniform sampling rate can not satisfy the real purpose. Different resolutions should be introduced for different surface regions. From the bunny model, it is obvious that the ear part has the highest streching factor. Texture with uniform resolution can not cover the ear part sufficiently as it does on the other regions. In practice, we can increase the resolution of the texture for the ear parts only.

Another approach is to find a developable surface as parameter domains, conformally map the surface to these domains, and make the streching factors on these domains as uniform as possible by adjusting these developable surfaces. Then we can unwrap the developable surface to the plane easily. For example, we can conformally map the two ears of the bunny mesh to two cylinders, and unwrap the cylinders to the parameter domain.

9.3 Generalization to Other Surface Representations

Geometric objects can be represented in many formats. Mesh is the dominent one. Sometimes it is convenient to use implicit surfaces or level sets. We would like to generalize our algorithms to these representations. Compared to the mesh representation, level sets are formulated as partial differential equations and easier to deal with mathematically. But level sets and implicit surfaces can not represent a surface immersed in $\mathbb{R}^3$. During the computation, surfaces can intersect itself in the transistion stage. For meshes, the resolution does not change for the whole process, for some surface regions with too sparse vertices, high inaccuracy will be introduced. It will be interesting to explore how to generalize the methods in this work to other geometry representaions.

9.4 Embedding in Hyperbolic Domain

For non-zero genus surfaces, their universal covering space can be embedded in the hyperbolic space. Computing such embedding requires solving a partial differential equation. By curvature flow, a general surface can be converged to its embedding in the hyperbolic space. In the future, this approach will be tested. This method will lead to general conformal maps between surfaces.

Acknowledgement

I gratefully thank Hugues Hoppe for mesh library codes and datasets, Cindy Grimms, Ligang Liu, Stanford University, University of Washington, Georgia Institute of Technology and Cyberware for models.

References

[1] Alliez, P., Meyer, M., and Desbrun, M. Interactive geomety remeshing. In *Proceedings of ACM SIGGRAPH* (2002), Addison Wesley, pp. 347–354.

[2] Arbarello, E., Cornalba, M., Griffiths, P., and Harris, J. *Topics in the Theory of Algebraic Curves*. 1938.

[3] Davis, G. Wavelet image compression construction kit. *http://www.geoffdavis.net/dartmouth/wavelet/wavelet.html*.

[4] Desbrun, M., Meyer, M., and Alliez, P. Intrinsic parametrizations of surface meshes. In *Proceedings of Eurographics* (2002).

[5] Dey, T. K., and Guha, S. Transforming curves on surfaces. *Journal of Computer and System Sciences 58*, 2 (1999), 297–325.

[6] Dey, T. K., and Schipper, H. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete and Computational Geometry 14* (1995), 93–110.

[7] Duchamp, T., Certian, A., Derose, A., and Stuetzle, W. Hierarchical computation of pl harmonic embeddings. *preprint* (July 1997).

[8] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH 95*, pp. 173–182.

[9] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. Multiresolution analysis of arbitrary meshes. In *Proceedings of ACM SIGGRAPH* (Auguest 1995), Addison Wesley.

[10] Erickson, J., and Har-Peled, S. Cutting a surface into a disk. *ACM SoCG 2002*.

[11] Ferguson, H., Rockwood, A., and Cox, J. Topological design of sculptured surfaces. In *SIGGRAPH 92*, pp. 149–156.

[12] Floater, M. Parametrization and smooth approximation of surface triangulations. *CAGD 14*, 3 (1997), 231–250.

[13] Floater, M. S. Parametrization and smooth approximation of surface triangulations. *Comp.Aided Geom. Design* (1997), 231–250.

[14] Francis, G. K., and Weeks, J. R. Conway's zip proof. *Amer.Math.Monthly* (1999), 393–399.

[15] Gu, X., Gortler, S. J., and Hoppe, H. Geometry images. In *Proceedings of ACM SIGGRAPH* (2002), Addison Wesley.

[16] Gu, X., and Yau, S. T. Computing conformal structure of surfaces. In *Communication of Information and Systems Vol 2, No. 2* (2002), IP Press.

[17] Gu, X., and Yau, S. T. Global conformal parametrization. In *Preprint* (2002).

[18] Guillemin, V., and Pollack, A. *Differential Topology*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

[19] Guskov, I., Vidimce, K., Sweldens, W., and Schröder, P. Normal meshes. In *SIGGRAPH 2000*, pp. 95–102.

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

[21] Hoppe, H. Progressive meshes. In *Proceedings of ACM SIGGRAPH* (1996), Addison Wesley, pp. 99–108.

[22] Karni, Z., and Gotsman, C. Spetral compression of mesh geometry. In *Proceedings of ACM SIGGRAPH* (2000), Addison Wesley, pp. 279–286.

[23] Khodakovsky, A., Schröder, P., and Sweldens, W. Progressive geometry compression. In *SIGGRAPH 2000*, pp. 271–278.

[24] Lazarus, F., Pocchiola, M., Vegter, G., and Verroust, A. Computing a canonical polygonal schema of an orientable triangulated surface. In *ACM SoCG 2001*, pp. 80–89.

[25] Lee, A., Moreton, H., and Hoppe, H. Displaced subdivision surfaces. In *SIGGRAPH 2000*, pp. 85–94.

[26] Lee, A., Sweldens, W., Schröder, P., Cowsar, L., and Dobkin, D. MAPS: Multiresolution adaptive parameterization of surfaces. In *SIGGRAPH 98*, pp. 95–104.

[27] Levy, B., and Mallet, J. L. Non-distorted texture mapping for sheared triangulated meshes. In *SIGGRAPH 98* (1998), Addison Wesley.

[28] Levy, B., Petitjean, S., Ray, N., and Maillot, J. Least squares conformal maps for automatic texture atlas generation. In *Proceedings of ACM SIGGRAPH* (2002), Addison Wesley.

[29] Lounsbery, M., DeRose, T., and Warren, J. Multiresolution analysis for surfaces of arbitrary topological type. *ACM TOG 16*, 1 (January 1997), 34–73.

[30] Maillot, J., Yahia, H., and Verroust, A. Interactive texture mapping. In *SIGGRAPH 93*, pp. 27–34. ISBN 0-201-58889-7.

[31] Munkres, J. *Topology.* Prentice Hall, 2000.

[32] Munkres, J. R. *Elements of Algebraic Topology.* Addison-Wesley Co., 1984.

[33] P. M. Gianni, M. Seppala, R. S., and Trager, B. M. Riemann surfaces, plane algebraic curves and their period matrices. vol. 26, pp. 789–803.

[34] Piponi, D., and Borshukov, G. D. Seamless texture mapping of subdivision surfaces by model pelting and texture blending. In *SIGGRAPH 2000*, pp. 471–478.

[35] Sander, P., Gortler, S., Snyder, J., and Hoppe, H. Signal-specialized parametrization. *Proceedings of Eurographics Workshop on Rendering 2002* (2002).

[36] Sander, P., Snyder, J., Gortler, S., and Hoppe, H. Texture mapping progressive meshes. In *SIGGRAPH 2001*, pp. 409–416.

[37] Schoen, R., and Yau, S. T. *Lectures on Harmonic Maps.* International Press, Harvard University, Cambridge MA, 1997.

[38] Seppala, M. Computation of peirod matrices of real algebraic curves. vol. 11, pp. 65–81.

[39] Sheffer, A. Spanning tree seams for reducing parameterization distortion of triangulated surfaces. *Shape Modelling International* (2002).

[40] Sheffer, A., and Sturler, E. Parameterization of faceted surfaces for meshing using angle-based flattening. vol. 17, pp. 326–337.

[41] Siegel, C. L. *Algebras of Riemann Matrices - Tata Institute of Fundamental Research.* Lecture on Mathematics and Physics, Tate Institute Bombay, 1956.

[42] Stephenson, K. Approximation of conformal structures via circle packing. World Scientific, pp. 551–582.

[43] Taubin, G., and Rossignac, J. Geometric compression through topological surgery. *ACM TOG 17*, 2 (1998), 84–115.

[44] Tutte, W. T. How to draw a graph. *Proc. London Math Soc. 13* (1963), 743–768.

[45] Vegter, G., and Yap, C. K. Computational complexity of combinatorial surfaces. In *ACM SoCG 1990*, pp. 102–111.

[46] Vorsatz, J., Rössl, C., Kobbelt, L., and Seidel, H.-P. Feature sensitive remeshing. *Computer Graphics Forum 20*, 3 (2001), 393–401.

[47] Weyl, H. On generalized riemann surfaces. *Ann. of Math. 35* (1934), 714–729.

[48] Wood, Z., Hoppe, H., Desbrun, M., and Schröder, P. Isosurface topology simplification. *Microsoft Research MSR-TR-2002-28* (January 2002).