Constrained conformal/quasi-conformal map and its applications

Zhang, Minqi

2014

Zhang, M. (2014). Constrained conformal/quasi-conformal map and its applications.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/61764>

<https://doi.org/10.32657/10356/61764>

NANYANG TECHNOLOGICAL UNIVERSITY

Constrained Conformal/Quasi-Conformal Map and Its Applications

A thesis submitted
for a degree of Doctor of Philosophy
by

Minqi Zhang

School of Computer Engineering
Nanyang Technological University

Jan 22, 2014

Abstract

With the rapid development of 3D-capture and modeling technologies, a vast number of 3D surface models have been created. More analysis needs to be done on these models, especially for the data from some subjects such as medical imaging. One of the key tools for analyzing these 3D surface models is conformal map. Discrete conformal map is an important parametrization for computer graphics. The surface parametrization refers to the process of mapping the surface onto canonical domains such as 2D plane. However, area or angle distortion would be inevitably introduced in the course of this procedure. The discrete conformal map provides a special parametrization which can minimize angle distortion. From a practical point of view, the conformal map offers a powerful tool to handle a number of geometric problems in various fields.

In this thesis, we achieve various breakthroughs for conformal map in both theorem and applications. In the theorem, we propose a new way to compute the extremal quasi-conformal map; and in practice, we greatly increase the flexibility and the scope of the conformal map by applying the conformal map to a wide range of applications such as medical images, networking routing, the Escher-Droste effect, and so on.

Computing the extremal quasi-conformal map on general surface: The conformal map has desirable properties, but does not always exist; When the conformal map does not exist, the Teichmüller mapping (T-map) is a special extremal quasi-conformal map, which is a perfect substitution for the conformal map. For the first time, we present a simple yet effective technique to compute extremal quasi-conformal maps on general surfaces. Our method is linear, local, easy to implement and has a very natural parallel structure. Furthermore it requires neither the numerical solver nor the global coordinate system.

Surface registration for brainstem by Ricci flow: The morphometry of the brainstem surface is of utmost importance for the disease Adolescent Idiopathic Scoliosis (AIS). We propose an effective method to accurately compute the registration between brainstem surfaces with consistent landmark features by using the conformal map and extremal quasi-conformal map. We also show the efficacy of the proposed registration algorithm through the experiments on 15 AIS models and 15 normal models.

Surface registration for vestibular system by holomorphic 1-form: Another important system related to the AIS is the vestibular system (VS). The non-trivial topology of the VS poses great technical challenges for geometric analysis. We present an effective and practical solution to register vestibular systems by using the holomorphic 1-form and extremal quasi-conformal map techniques. Our method is robust, automatic and efficient.

Greedy routing through distributed parametrization: The greedy routing problem is an important topic for wireless sensor networks. We propose a solution by applying a parametrization similar to the conformal map to build the virtual coordinate system first. The new parametrization, which is modified from the holomorphic 1-form, can embed the network domain to a canonical domain, which allows greedy routing to have guaranteed delivery.

Generalized Escher-Droste effect: The Escher-Droste effect is a special type of recursive picture, which depicts fascinating visual effects. We generalize the classical Escher-Droste effect, such that the self-invariant mappings of the images include general conformal mappings. We regard the planar conformal mappings as the basic image editing tools. The algorithm is simple, fast, flexible, and is capable of generating the Escher-Droste effects for a broad category of images.

Our contribution here is two-fold: on one hand, we make great headway in applying the conformal map as a basic tool, especially in the situation where the conformal map does not exist and the setting that requires delicate change for the traditional conformal algorithm. On the other hand, we introduce the conformal map to the other realm, which also opens a new gate for that realm and brings unexpected progress.

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete my thesis.

I am deeply indebted to my helpful supervisor Prof. He Ying, whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis. It is extremely lucky to have a supervisor who gives me so many helps and instructions on both research and life. His supervision and support are always like the beacon in my research life.

I gratefully acknowledge the help of Luo Jun, who gives me a lot of helpful suggestions and instructions in my study.

I also wish to thank all my friends of the Geometry Modeling Group at gameLab that supported me a lot by providing valuable feedback in many fruitful discussions, in particular Ms. Chen Shuangmin, Dr. Chen Xiaoming, Ms. Dao Thi Phuong Quynh, Mr. Han Shuchu, Mr. Lai Chi-Fu William, Dr. Lin Juncong, Mr. Wang Dayong, Ms. Sun Qian, Dr. Xia Jiazhi, Ms. Yan Xiaoqi, Dr. Xin Shiqing, Mr Li Feng, Dr Zhang Long, Mr Ying Xiang and Mr. Wang Xiaoning.

Especially, I would like to give my special thanks to my family whose enduring love enabled me to complete this thesis.

Publications

Published:

Minqi Zhang, Fang Li, Ying He, Lin Shi, Defeng Wang, Lok Ming Lui. Registration of Brainstem Surfaces in Adolescent Idiopathic Scoliosis using Discrete Ricci Flow, *Proceedings of 15th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI '12)*, LNCS 7511, pp. 146-154, 2012.

Junhui Hou, Lap-Pui Chau, Ying He, **Minqi Zhang**, Nadia Magnenat-Thalmann. Rate-distortion model based bit allocation for 3-D facial compression using geometry video, *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, Vol. 23, No. 9, pp. 1537-1541, 2013.

Qian Sun, Long Zhang, **Minqi Zhang**, Xiang Ying, Shi-Qing Xin, Jiazhi Xia and Ying He. Texture brush: an interactive surface texturing interface, *Proceedings of ACM Symposium on Interactive 3D Graphics and Games (I3D '13)*, pp. 153-160, 2013.

Minqi Zhang, Ying He. A Simple and Local Method For Computing Quasi-conformal On 3D Surfaces, *Computer-Aided Design (CAD)*, Vol 46, No. 1, pp. 192-199, 2014 (**Best Paper Award in SIAM Conference on Geometric and Physical Modeling 2013**)

Minqi Zhang, Ying He, Feng Li, Juncong Lin, Xianfeng David Gu, Jun Luo. GRIP: Greedy Routing through dIstributed Parametrization for Guaranteed Delievery in 2D/3D Sensor Networks. *Wireless Networks*, *Accepted with minor revision*.

Minqi Zhang, Fang Li, Xingce Wang, Shiqing Xin and Ying He. Automatic Registration of Vestibular Systems with Exact Landmark Correspondence. *Geometric Modeling and Processing (GMP) 2014*. *Accepted, to be appeared*.

Ready to submit:

Wei Zeng, **Minqi Zhang**, Qian Sun, Y. He, X.Gu, General Escher Droste Effect.

Contents

Abstract	i
Acknowledgments	iii
Publications	iv
List of Figures	x
List of Tables	xviii
1 Introduction	1
1.1 Conformal Map and Its Challenge	1
1.1.1 Overview	1
1.1.2 Applications	4
1.1.3 Challenges	7
1.2 Contributions	9
1.3 Thesis Organization	12
2 Related work	13
2.1 Overview of Conformal Map History	13
2.2 Related Work for Conformal Map	16
2.2.1 Topological Disk	16
2.2.2 Genus Zero Model	18
2.2.3 High Genus Model	20
2.3 Related Work for Quasi-Conformal Map	22
3 Mathematical Background	25
3.1 Conformal and Quasi-Conformal Map	25
3.1.1 Conformal Map	25

3.1.2	Quasi-Conformal Map	30
3.2	Tools for Computing Conformal Map	32
3.3	From Continuous to Discrete	37
3.3.1	Basic Structure and Discrete Conformal Map	37
3.3.2	Discrete Holomorphic 1-Form and Ricci Flow	39
4	Computing the Extremal Quasi-Conformal Map on General Surface	47
4.1	Introduction	47
4.2	Quasi-Conformal Map and Beltrami Coefficient	49
4.3	Computing the Teichmüller Map on Simply/Multiply Connected Domains	50
4.4	Computing Quasi-Conformal Map on 3D Surfaces	52
4.4.1	Preprocessing	52
4.4.2	Iteratively Improving the Map Quality	56
4.5	Experimental Results	58
4.6	Comparisons & Discussions	63
5	Registration of Brainstem Surfaces in Adolescent Idiopathic Scoliosis using Discrete Ricci Flow	65
5.1	Introduction	65
5.1.1	Related Work on Surface Registration	67
5.2	Algorithm	68
5.3	Overview	68
5.3.1	Landmark Extraction	69
5.3.2	Registration	71
5.3.3	Shape variation detection	74
5.4	Experimental Results	76
6	Automatic Registration of Vestibular Systems with Exact Landmark Correspondence	79
6.1	Motivation	79
6.2	Algorithm	81
6.2.1	Landmark extraction	81

6.2.2	Registration	84
6.3	Results & Discussions	90
7	GRIP: Greedy Routing through dIstributed Parametrization for Guaran- teed Delivery in 2D/3D Sensor Networks	93
7.1	Introduction	93
7.1.1	Overview of the Problem	93
7.1.2	Our Approach and Contributions	94
7.2	Related Work on Greedy Routing	96
7.3	Overview of GRIP	97
7.3.1	Basic Principles	97
7.4	Distributed Parametrization in 2D/3D Wireless Sensor Networks	100
7.4.1	Mathematical Tools	101
7.4.2	Distributed Parametrization	110
7.5	Tunable Greedy Routing	111
7.5.1	Guaranteed Delivery	112
7.5.2	Tuning Greedy Objective	114
7.6	Discussions	116
7.7	Simulations and Experiments	117
7.7.1	Parametrization Efficiency	117
7.7.2	Routing Performance	118
8	Generalized Escher-Droste Effect	123
8.1	Introduction	123
8.2	Related Work on Escher-Droste Effect	126
8.3	Traditional Escher-Droste Effect	127
8.3.1	Mathematical Structure for Escher-Droste Effect	127
8.3.2	Conventional Escher-Droste Algorithm	129
8.4	Generalized Escher-Droste Algorithm	131
8.4.1	Droste Effect	131
8.4.2	Escher Effect	134
8.5	3D Droste Effect	134
8.6	Experimental Results	136

9	Conclusion and Future Work	141
9.1	Conclusion	141
9.2	Future Work	143
9.2.1	Conformal Map on Broken and Noisy Mesh	144
9.2.2	Distortion Control	145
9.2.3	More Applications for Medical Images	146
	References	148

List of Figures

1.1	Riemann map theorem: any surface with a topological disk shape can be conformally mapped to the unit disk, this figure is courtesy from Gu et. al. [35].	2
1.2	Riemann uniformization theorem: all manifold surfaces in real life can be conformally mapped to one of three canonical shapes:the unit sphere (genus-0, left), the Euclidean plane (genus-1, middle), or the hyperbolic space (high genus (>1), right), this figure is courtesy from Gu et. al. [35].	3
1.3	Texture map by using conformal map, this figure is courtesy from Gu et. al. [35].	5
1.4	Face matching by conformal map. ϕ_1, ϕ_2 are conformal maps, $\tilde{f}$ is the 2D surface matching, and f is the result matching, this figure is courtesy from Gu et. al. [35].	6
1.5	Left: knot structure, middle: spline, right: spline surface. By using the conformal map, we can map the knot structure from the 2D plane to the surface, which can help to construct the spline curve, this figure is courtesy from Gu et. al. [35].	7
1.6	The conformal map and its application to the cortical surface. We can analyze the information on the unit sphere, which will be much easier, this figure is courtesy from Gu et. al. [35].	8
2.1	Traditional method like ABF will cut the model to the topological disk first then map it to the canonical domain, this figure is courtesy from Floater et. al. [28].	17
2.2	David's head is mapped to the sphere domain, this figure is courtesy from Gu et. al. [32]	19

2.3	Compute the conformal map for high genus model by holomorphic 1-form, X. Gu et al., this figure is courtesy from Gu et. al. [32].	20
2.4	Compute the conformal map for high genus model by Ricci flow, this figure is courtesy from Gu et. al. [32].	21
2.5	Extremal quasi-conformal map on the planar domain, this figure is courtesy from Weber et. al.[109].	23
3.1	Manifold definition, the transition map $\phi_{\alpha\beta}$ is defined on the intersection of two charts U_α and U_β	27
3.2	The middle is a conformal map, which maps a face to the disk. We can see that the conformal map keeps the details well, unlike the quasi-conformal map in the right, this figure is courtesy from Zeng et. al. [123].	29
3.3	Beltrami coefficient and Beltrami differential.	30
3.4	The surfaces are represented by triangular mesh.	37
3.5	The scalar harmonic field on the bunny, which can induce the harmonic 1-form on the edge.	40
3.6	Left: The harmonic 1-form ω , middle: $*\omega$ without smoothing; right: $*\omega$ after smooth.	43
3.7	The corner angle in one triangle.	43
3.8	The circle packing on the mesh is that each vertex equips a circle c with radius r	44
3.9	Left: Andreev-Thurston circle packing, where each angle between the circles is acute. Right: distance circle packing.	45
4.1	The Teichmüller map from the genus-8 4-Kid model to a polycube domain (see the insect). The circle texture map illustrates the quality of the computed T-map. As shown in the histogram, the magnitude of the Beltrami coefficient $ \mu $ ranges from 0 to 0.8. More than 75% of them are less than 0.2.	49

4.2	Algorithmic pipeline. Our algorithm takes two triangular meshes S_1 and S_2 of the same topology, and an initial map $f_0 : S_1 \rightarrow S_2$ as input. Our algorithm does not require a bijective f_0 . For the two-hole torus and the polycube, we simply project S_1 onto S_2 to get the initial map f_0 . Our algorithm iteratively improves the map quality by diffusing the map. Unlike the existing techniques, our algorithm requires neither the global parameterization nor global numerical solver. Thus, it is proceeded in a completely local manner, and has the advantage of scalability.	53
4.3	Exponential map defines a geodesic polar coordinate system on curved surface.	55
4.4	Computing the local map on the coordinate chart.	56
4.5	The label in one triangle around the vertex $\mathbf{v}$	58
4.6	Our Beltrami coefficient diffusion algorithm converges quickly. The horizontal axis shows the iteration number and the vertical axis shows the maximal change of the vertex position between two consecutive iterations.	60
4.7	T-map from the Head model to the unit square.	61
4.8	T-map from the Bunny to a pyramid shape. We project the Bunny to the pyramid to initialize the map f_0 . Clearly, the initial map is not bijective, due to large overlapped/self-intersect regions. Our method iteratively improves the map quality by diffusing it through the entire mesh. The quality of the body and the head are acceptable after 15 iterations and 100 iterations. Due to the significant geometric difference between the ear and the domain, it does take many iterations to get a bijectively map for the ear.	62
4.9	T-map from the genus-1 Kitten to a torus. These two shapes are not conformally equivalent. The coefficient k ranges from 0.2 to 0.5.	63
4.10	T-map of high genus models. The circle texture is shown on the source surface S_1 and the insects show the target surface S_2 and the histogram of the Beltrami coefficient $ \mu $	64

5.1	Landmark extraction. (A) and (B) show the computed ridge and valley lines respectively. The two strongest ridge and valley lines are chosen as landmarks, as shown in (C). (D) shows the four consistent features extracted on brainstems of a normal and an AIS subject.	68
5.2	The cut surface can be conformally embedded into the universal covering as shown in (A). (B) shows the fundamental domain (one period) of the universal covering. The parameterization naturally induces a segmentation of the brainstem surfaces into six patches as shown in (C). In (D), the checkerboard texture on the parameter domain is mapped to the brainstem surface, which map shows the parameterization is indeed conformal.	71
5.3	(A) Registration between two normal brainstems. (B) Registration between the brainstems of a normal and AIS subject. Landmark features are matched exactly. (C) shows the average shape of 15 AIS brainstems using our registration. The sharp features are well-preserved after averaging, meaning that our registration matches salient features well. (D) shows the average shape using the curvature-based spherical demons registration without landmarks [117]. Note that the sharp features are smoothed out.	72
5.4	Computing the map $\phi_i : F_1^i \rightarrow F_2^i$	75
5.5	Up: Grid texture is drawn on the control brainstem. It is mapped to three different brainstems of the AIS subjects using our proposed registration algorithm. The grid pattern illustrates that our registration result is indeed a homeomorphism. Down: Detection of shape variations. . . .	76
6.1	Computing the geodesic homology basis on the 3-torus model.	81
6.2	The vestibular system is a genus-3 structure (colored in brown) in the inner ear. Its homology basis $\{a_i, b_i\}_{i=1}^3$ of the genus-3 vestibular system contains 6 loops. Among them, the three geodesic <i>tunnel</i> loops a_i are highly consistent regardless of the AIS/normal subjects. The locations of the geodesic <i>handle</i> loops b_i , however, may vary significantly. . . .	82

6.3	Computing the stable geodesic loop c_{ij} across two handles γ_i and γ_j . From left to right: the initial curve $b_i l_{ij} b_j l_{ij}^{-1}$ is continuously deformed into a stable geodesic loop c_{ij} . Note that the final geodesic loop is homologous to the initial curve, since the curve shortening method [112] keeps the curve's topology during the deformation.	83
6.4	Thanks to the intrinsic and geometry-aware nature of the geodesic loops, the extracted landmarks are stable and highly consistent among the AIS and normal subjects.	84
6.5	The geodesic landmarks induce a segmentation of the vestibular system. g : genus, b : number of boundaries.	85
6.6	Registration. (a) The commutative diagram for the map between two topological annuli. (b) The commutative diagram for the map between two topological disks. The red dots are the images of the feature points. To avoid label congestion, only two feature points and their images are labeled, the others can be implied according to the order.	87
6.7	We compute the average shape of 13 normal subjects as the control model, and then measure the difference between the AIS subjects and the control model in terms of various geometric features, such as Gaussian curvature K , mean curvature H , area and conformal representation (λ, H) . The cut-off value for a 95% confidence interval is $t = 2.13$. Among them, the mean curvature H with $t = 2.29$ is an effective measure for distinguishing the normal and AIS subjects.	89
6.8	We normalize the mean curvature to the unit interval and then visualize the difference between the test model and the control model.	89
6.9	Visualizing the registration using checker board texture mapping. With the aid of texture mapping, one can clearly see the exact correspondence of the geodesic landmark features.	91
7.1	A network domain with non-convex holes (left) may lead to routing failure. GRIP first performs distributed parametrization to map the domain to a virtual domain (right), then it applies a greedy routing with tunable objective to achieve load-balancing routing.	98

7.2	Computing harmonic function.	103
7.3	The integral curves are perpendicular to the iso-value curves.	104
7.4	Propagative embedding from a 2-connected triangle mesh into an annulus.	106
7.5	The cut views of the harmonic function on the 3D network domain with a void (a) – (c). Integral curves in a network domain (d) and in the embedding domain, i.e., a solid ball with a spherical hole in the center (e).	108
7.6	Distributed parametrization of a 2D WSN. Each iteration of the algorithm consists of three steps, since the given 2D WSN contains three holes. Initially, all holes are filled virtually. In each step, we open one hole to get a 2-connected domain, and then apply Algorithm 3 to parameterize it to an annulus. The original network domain, the intermediate results, and the final parametrization after two iterations are shown from left to right.	109
7.7	Parametrization of a 3D WSN, where the central void is mapped to a sphere and the others are mapped to the convex polyhedron.	112
7.8	Greedy forward decision (7.3) always succeeds and is loop-free on a convex boundary.	114
7.9	Changes in network topology due to node joining (left) and leaving (right) only have local influences. The dashed circle represents the transmission range of a node.	116
7.10	CPU time comparison between GRIP's parametrization and Ricci flow. $l_1 = l_4 = l_7 = 2$, $l_2 = l_5 = l_8 = 3$, and $l_3 = l_6 = l_9 = 4$, where l_i is the number of holes in the i -th 2D WSN.	118
7.11	Comparing three greedy routing schemes in 2D WSNs.	119
7.12	Comparing different greedy routing schemes. While the pure greedy scheme (gray curve) sticks to the inner boundary, the two tunable schemes (purple and black curves) obviously tend to follow the isosurfaces, and the online tunable scheme (black curve) appears to achieve the most balanced performance.	120

7.13	Comparing three greedy routing schemes in 3D WSNs.	121
7.14	Illustration of the load balancing effect in two WSNs. Cut views are applied to reveal the loads inside the 3D WSN.	121
8.1	Generalized Escher-Droste effect. (a) <i>The Persistence of Memory</i> , Salvador Dali, 1931. (b) Droste effect on the clock. (c) Close-up view of the clock. (d) Another Droste effect.	124
8.2	An image exhibiting the Droste effect depicts a smaller version of the image within itself in a recursive manner. (a) Dutch chocolate box in 1904. (b) A computer-generated Droste image. (Courtesy of webdesignerdepot.com) (c) Escher's <i>Print Gallery</i> . (d) The revealed structure shows that the visitor is both the guy looking at the picture and the guy in the picture.	125
8.3	Escher-Droste effect of an office photo.	127
8.4	Escher effects with different transformation matrices.	131
8.5	Pipeline for computing discrete conformal map by holomorphic 1-form.	131
8.6	Given the input image (<i>Sting Caused by the Flight of a Bee</i> , Salvador Dali, 1944), the user specifies the region of interest by sketching two closed curves (in yellow). The inner curve is the mouth of the left tiger. Then, our program automatically generates a conforming constrained Delaunay triangulation.	132
8.7	Conformal maps. The first row shows the triangulation of the region of interest in the original image, and the second row shows the conformal parameterization results.	133
8.8	Droste effect for region of interest with irregular boundaries. (a) The Droste effect. (b) The triangulation of the repeated region in the tiger mouth. (c) The close-up view of the tiger mouth.	133
8.9	Möbius transformation is able to map the fixed point from the origin (see (a)) to arbitrary location (see (b)).	135
8.10	Escher effects with (row 2) and without (row 1) Möbius transformation.	135

8.11	The algorithmic pipeline for 3D Escher-Droste effect. (a) Enclose the input mesh M into a layered space of concentric spheres. (b) Map the layered space to a cuboid volume by using spherical coordinates. (c) Rotate the cuboid volume around the θ axis. (d) The rotated cuboid volume can be duplicated and then glued along the ϕ -axis, which forms a cuboid of infinite length. For better visualization, only three copies are rendered in this figure. (e) By evaluating the surface along the infinite cuboid and converting the spherical coordinates back to the Euclidean coordinates, we can get the 3D Escher-Droste effect.	137
8.12	3D Droste effect.	138
8.13	Escher-Droste effect of <i>Nine Goldfish</i> , where γ_1 is the boundary of a stone. The Droste effect is shown in (c) and various Escher effects are shown in (e)-(h).	139
8.14	Escher effect of <i>King Tutankhamun's Golden Mask</i> , where the inner curve γ_1 is the boundary of lips. Note that the conventional Escher-Droste algorithm does not work for this case, since the region bounded by the inner curve γ_1 is degenerate.	139
8.15	Given Vincent van Gogh's self-portrait, we mark his ear as the region of interest (see (a)). The Droste effect is shown in (b). And various Escher effects are shown in (c)-(f).	139
8.16	Escher-Droste effect of <i>The Scream</i> , Edvard Munch, 1893. The irregular hole is cut along the boundary of open mouth.	140
9.1	The face captured by 3D camera will have noise and holes.	144
9.2	Picture morphing uses conformal map, although conformal map can keep local feature isometry, the area distortion still causes serious visible defects.	145
9.3	The Willis's circle (courtesy of Wikipedia) and its MRI model.	146

List of Tables

4.1	Mesh complexity and performance. The preprocessing and diffusion times are measured on a GPU with 512 cores and a CPU with 4 cores respectively.	62
4.2	Comparison to other quasi-conformal methods.	64
7.1	Simulated WSNs	120

Chapter 1

Introduction

1.1 Conformal Map and Its Challenge

1.1.1 Overview

Given a surface, one fundamental structure on the surface is the topological structure, which gives the concept for the continuous property and genus of surface. Another structure more rigid than the topological structure is the conformal structure, which allows us to measure angles between two curves intersecting at the same point. The most powerful structure is the metric structure, which carries more detail and enables us to measure the area, angle, length of the surface, and so on. Hence, intuitively speaking, the conformal structure is the structure between the topological structure and the metric structure. The conformal geometry is the geometry which studies the conformal structure.

One of the important topics for conformal geometry is the conformal map, which is the map between two surfaces preserving angles. From the perspective of conformal geometry, a conformal map studies the map keeping the conformal structure invariant, similar to the homomorphism, which keeps the topological structure invariant, and the isomorphism which preserves the metric.

The study of the conformal map starts from the complex analysis, where the conformal map is regarded as the holomorphic function $\mathbb{C} \rightarrow \mathbb{C}$. So from this point of view, a conformal map can be the generalization of the holomorphic function on the general manifold.

Figure 1.1: Riemann map theorem: any surface with a topological disk shape can be conformally mapped to the unit disk, this figure is courtesy from Gu et. al. [35].

There are many important theorems for the conformal map, one of which is the Riemann map theorem, which claims that any simply connected surface that is topologically equivalent to the disk can have the conformal map to the unit disk (see Figure 1.1). Another important theorem for conformal map is the Riemann uniformization theorem, which states that all 2-D manifold surfaces can be conformally mapped to one of the three canonical shapes according to their genus: the unit sphere, the Euclidean plane, and the hyperbolic plane (see Figure 1.2). All these canonical shapes have the constant Gaussian curvature: +1, 0, or -1. This theorem is quite amazing as it means that all surfaces can have a canonical Riemannian metric, which has the same conformal structure with the original metric but can be embedded into the canonical space.

Due to the fast development of computer technology in recent decades, the continuous conformal map theorem was introduced to computer graphics. Since the computer can

Figure 1.2: Riemann uniformization theorem: all manifold surfaces in real life can be conformally mapped to one of three canonical shapes: the unit sphere (genus-0, left), the Euclidean plane (genus-1, middle), or the hyperbolic space (high genus (>1), right), this figure is courtesy from Gu et. al. [35].

only handle the discrete model, the graphics community has built the discrete conformal map theorem, which is analogous to its continuous counterpart. In computer graphics, surface parametrization means mapping the surface to another domain, which has simple structures such as 2D planes, spheres, and so on; Most of operations on the original surface can then be finished on the target domain, which will become much easier given its simple structure. However, most of the surfaces cannot be isometrically mapped to the target domain; there will be some area or angle distortion unless the surface is developable. The discrete conformal map provides a special parametrization which can minimize angle distortion. There will be no angle distortion for the conformal map when the surface is smooth enough, although it is impossible for computer graphics.

Here are the factors that have made discrete conformal parametrization become so popular for the computer graphics community:

- (i) *Angle preserving*: Angle preserving is the key property for conformal map. The angle preserving property is quite desirable for many applications like the texture mapping in computer graphics.
- (ii) *Flexible*: As the Riemann uniformization theorem has stated: No matter how complicated the surface is, it can be conformally mapped to one of three canonical shapes: spherical, Euclidean or hyperbolic.
- (iii) *Rigid*: According to the Riemann theorem, the conformal map between a topological disk and the unit disk can be uniquely determined by fixing 3 points (up to the Mobius transformation), which means that the conformal map can be controlled by finite parameters
- (iv) *Intrinsic*: The conformal map is determined by its metric rather than by the embedding space. For example, the human body having different limb motions can have the same conformal map result, and the result is also independent of the mesh resolution.
- (v) *Solid and extensible*: The study on continuous conformal maps in differential geometry is quite mature and the discrete conformal map has solid mathematical background. For instance, it can be proved that we can find the conformal map between all the genus-0 closed surfaces. In addition, for the situation where conformal map does not exist, there is still a well-defined extremal quasi-conformal map, which has the clear mathematical description, to replace.

1.1.2 Applications

After the discrete conformal map was first introduced, this technique has demonstrated its great power and changed a broad range of geometric problems in different areas

like geometric modeling, computer graphics, computer vision, visualization, medical imaging, scientific computing, etc.

Computer Graphics

Figure 1.3: Texture map by using conformal map, this figure is courtesy from Gu et. al. [35].

Conformal map has numerous applications in computer graphics, including mesh re-pairing, texture map and synthesis, surface re-meshing, mesh matching, mesh-spline conversion, geometric morphing, efficient rendering, animation, and many more.

As an example, conformal map provides a special parametrization without the angle distortion. This property is quite favored for applications such as texture mapping, since the most important property for texture mapping is to transfer the details of the texture on the original image to the target surface (see Figure 1.3).

Computer Vision

In computer vision, the conformal map has been applied to surface matching, shape comparison, shape classification, geometric analysis, and tracking.

For instance, surface matching is a fundamental task for computer vision. By using the conformal map, we can map both the original 3D surface to the 2D plane and then use the 2D surface matching technique to match two surfaces on the parametrization domain. The final surface matching can be obtained from the compound map of conformal parametrization and 2D surface matching, (see Figure 1.4).

Figure 1.4: Face matching by conformal map. ϕ_1, ϕ_2 are conformal maps, $\tilde{f}$ is the 2D surface matching, and f is the result matching, this figure is courtesy from Gu et. al. [35].

Geometry Processing

Traditional 3D models in computer graphics are usually represented as triangular mesh; however, in some other fields like geometric modeling, the surfaces are usually represented as piecewise polynomials or rational polynomials with higher order continuity called splines. The conformal geometric method can be a useful tool to convert the piecewise linear meshes to the Beizer surfaces (see Figure 1.5).

Medical Image

The modern technology in medical imaging is developing rapidly and a large number of 3D models for medical imaging data have been produced. The conformal map can be a powerful tool to analyze these data; for example, the traditional way to visualize the information on a cortical surface is quite difficult due to its complex surface. However, after we bring the conformal map to the analysis and map the surface to the sphere, the visualization procedure will be quite convenient, as illustrated in Figure 1.6.

Figure 1.5: Left: knot structure, middle: spline, right: spline surface. By using the conformal map, we can map the knot structure from the 2D plane to the surface, which can help to construct the spline curve, this figure is courtesy from Gu et. al. [35].

1.1.3 Challenges

The conformal map has solid theoretical foundation and powerful effect on a broad range of applications. Nevertheless, there remain a lot of challenges for this topic, both in theorem and practice.

In theorem, as we have mentioned as its advantage, the conformal map is rigid and can be determined by a few parameters. However, this advantage could also be its drawback for some other applications. The rigid property could also mean that we only have limited constraints, e.g., when we try to build the conformal map between two high genus models, it does not always exist. Hence, the most important challenge for the conformal map theorem is how to compute a map as close as possible to the conformal map when the traditional conformal map does not exist.

Figure 1.6: The conformal map and its application to the cortical surface. We can analyze the information on the unit sphere, which will be much easier, this figure is courtesy from Gu et. al. [35].

In practice, there are two main problems which limit the application of conformal map. Firstly, the procedure to apply conformal map to an application is not so straightforward. We would encounter a lot of complicated situations in real-life applications. For example, for the situation like sensor networks, the bottleneck is that no single sensor can afford the huge computation because every sensor has limited computational capacity. Furthermore, in applications like surface registration on medical images, we cannot apply the conformal map directly to the surface as we need to match the landmark, otherwise the registration will be meaningless. Because if the landmark matching can not be guaranteed, the registration will map the surface to the totally different area, which makes the geometry difference meaningless. We can not get any valuable information from the geometry difference between one man's face and another man's backside of head. In this situation, we may need to develop a divide-and-conquer method to use the conformal map. Secondly, applying the conformal map to the other application is a multi-disciplines topic, especially for using the conformal map to problems like network routing and medical image registration. It always relates to more than two subjects, and the one who needs this powerful tool may not be aware of the existence of the tool when each researcher focus on his/her own domain.

1.2 Contributions

In this thesis, we achieve a breakthrough for conformal map in both theorem and applications. In the theorem, we propose a new way to compute the extremal quasi-conformal map; and in practice, we greatly increase the flexibility and the scope of the conformal map by applying it to a wide range of applications such as medical images, networking routing, Escher-Droste effect, and so on.

Computing the extremal quasi-conformal map on general surfaces: The conformal map has desirable properties but does not always exist. When the conformal map does not exist, the Teichmüller map (T-map) will be a perfect substitution for it. Teichmüller mapping (T-map) is an extremal quasi-conformal map associated with a holomorphic function as the Beltrami coefficient. The T-map between two Riemann surfaces of the same topology always exists and is unique. The existing techniques to compute T-map require a global coordinate system, thus, they are limited to models with simple topological types, such as genus-0 or 1 surfaces, for which one can obtain the global coordinates by using the global parametrization. To our best knowledge, for the first time, we present a simple yet effective technique to compute extremal quasi-conformal map on surfaces of non-trivial topology. Our method extends the linear Beltrami solver from the complex plane to the manifold setting: it computes the local T-map on the local coordinate chart and then diffuses the map through the entire mesh. Our method is linear, local, easy to implement and has a very natural parallel structure. Furthermore it requires neither numerical solver nor the global coordinate system.

Surface registration for brainstem by Ricci flow: Adolescent Idiopathic Scoliosis (AIS) characterized by the 3D spine deformity affects about 4% schoolchildren worldwide. Several studies have demonstrated the malfunctioning of postural balance, proprioception, and equilibrium control in patients with AIS. Since these functions are closely

related to structures in and around the brainstem, the morphometry of the brainstem surface is of utmost importance. We propose an effective method to accurately compute the registration between brainstem surfaces. Four consistent features, which describe the global geometry of the brainstem, are extracted to guide the surface registration. Using the discrete Ricci flow method, brainstem surfaces are parameterized conformally onto the quadrilaterally-faced hexahedron; The final brainstem registration can be obtained by matching each of their faces through the extremal quasi-conformal map. Our registration algorithm can guarantee the exact landmark correspondence between brainstem surfaces. With the obtained registration, a shape energy can be defined to measure the local shape difference between different brainstem surfaces. We have tested our algorithms on 30 real brainstem surfaces extracted from MRIs of 15 normal subjects and 15 AIS patients. Experimental results show the efficacy of the proposed algorithm to register brainstem surfaces, which matches landmark features consistently. The computed registration can be used for the morphometry of brainstems.

Surface registration for vestibular system by holomorphic 1-form: Another important system related to the AIS is the vestibular system (VS). The morphometry of the VS is to study the geometric differences between the normal and AIS groups, which plays a critical role for analyzing the AIS disease. However, the non-trivial topology of the VS poses great technical challenges to the geometric analysis. We presents an effective and practical solution to register the vestibular systems. We first extract six geodesic landmarks for the VS, which are stable, intrinsic, and insensitive to the VS's resolution and tessellation. Moreover, they are highly consistent regardless of the AIS and normal groups. The detected geodesic landmarks partition the VS into three patches, a topological annulus and two topological disks. For each pair of patches of the AIS subject and the control, we compute a bijective map using the holomorphic 1-form and extremal quasi-conformal techniques. With a carefully designed boundary condition, the three

individual maps can be glued in a seamless manner, so that the resulting registration is a homeomorphism with exact landmark matching. Our method is robust, automatic and efficient, which takes only a few seconds on a PC.

Greedy routing through distributed parametrization: The greedy routing problem is an important topic for the wireless sensor network. One of the challenge for using conformal map in sensor network routing is that each sensor have limited computational capacity. We propose a solution by applying a parametrization similar with the conformal map to build the virtual coordinate system first. The new parametrization, which is modified from the holomorphic 1-form, can embed the network domain to the canonical domain, which allows greedy routing to have guaranteed delivery. Furthermore, the algorithm can be easily generalized to the 3D case, which can be very helpful in solving the routing problem in 3D wireless networks.

Generalized Escher-Droste effect: The Escher-Droste effect is a special type of recursive picture, which delivers fascinating visual effects. The limitation of the self-invariant mappings confines the scope of the application of the Escher-Droste effect. We generalize the classical Escher-Droste effect by introducing the holomorphic 1-form to deform the image first. This greatly increases the flexibility and the scope of the method. We use the planar conformal mappings as image editing tools, which can locally preserve shapes, and globally deform the whole image. The pipeline we propose is simple, fast and flexible, and is also capable of generating the Escher-Droste effect for large category of images. Also, inspired by the spirit, which regards the Droste effect and a conformal map from original image to the basic element of tiling pattern, we further extend Escher-Droste effect to the 3D space and develop a simple algorithm to create interesting visual effects from general 3D models.

Our contribution here is two-fold: on one hand, we have made great headway in applying the conformal map as a basic tool, especially for situations, where the conformal map

does not exist, and settings, which require delicate change for the traditional conformal algorithm. On the other hand, we introduce the conformal map to the other realm, which also opens a new gate for that field and brings unexpected progress. For example, people can enjoy results, which have never been seen before, like the 3D Escher-Droste effect.

1.3 Thesis Organization

In the following report, we first review the related work on conformal parametrization for computer graphics in Chapter 2, then we introduce the mathematical background for conformal map in Chapter 3. In Chapter 4 we will discuss our work to compute extremal quasi-conformal map on general surface. Chapter 5 presents the work by using ricci flow and extremal quasi-conformal map to build the surface registration on brainstem for the study of AIS, in addition in Chapter 6 we use the holomorphic 1-form and extremal quasi-conformal map to build the registration of the vestibular system. In Chapter 7, we propose a solution for 2D/3D sensor network greedy routing by applying a parametrization similar to the conformal map to build the virtual coordinate system first. In Chapter 8, we use the conformal map as a basic image editing tool to generalize the traditional Escher-Droste effect. Lastly, we conclude our work and provide some future directions for conformal maps in Chapter 9.

Chapter 2

Related work

This chapter surveys some related work regarding the conformal map and its extension quasi-conformal theorem. We only focus on the work concerning the conformal map theorem itself here. Other work related to the application of the conformal map, for example, regarding the conformal map as the surface registration tool, will be mentioned in the related chapter.

2.1 Overview of Conformal Map History

The conformal map is a special topic in parametrization. The main task in parametrization is to map a surface to a simple canonical domain. For general parametrization, the map will inevitably introduce distortion, either area distortion or angle distortion. The conformal map minimizes the angle distortion. We will mainly review the work on the conformal map here. For other work, such as distance preserving and area preserving, we refer the reader to [28, 101], which are good surveys of the previous work on general parametrization.

The conformal map firstly appeared in the complex analysis [1], related to the holomorphic function. Following work by several talented mathematicians, such as Rieman, the

conformal map theorem on the complex plane was generalized to the general manifold and became an important part of the Riemannian surface theorem [25].

The study of discrete conformal map started in [19] and [20]. In the 1940s and 50s, Duffin used the simple difference equation, which is analogous to Laplace's equation and Cauchy-Riemann equations, to develop a discrete version of the harmonic and analytic function. Also, Tutte [107] developed a theory on graph embedding in 1963, where he introduced the barycentric map, which inspired much research work on parametrization and conformal map later.

Before 2002, some works such as [84, 76, 22, 61, 26, 3] started to study the conformal map for computer graphics. At this early stage, most work focused on maps for models with simple topology, such as topological disk or sphere. In addition, most work followed a similar pattern to compute the map, firstly they defined the energy function to describe the distortion, then they used the variational method to minimize the energy. Some other researchers, such as Haker et al. [37] solved the map by using the discrete Laplace-Beltrami operator, which is the linear system in their solution.

In 2002, Gu et al.[33] proposed a method to compute the conformal structure on the high genus surface. This work was completely different to the previous work and was regarded as a milestone in conformal map history. For the first time, a practical algorithm to compute a conformal map for a high genus model was proposed, and this method had a strong theoretical guarantee.

After 2002, conformal parametrization became a very hot topic and witnessed its prosperous times. Many viable results have been achieved based on the traditional way of computing the conformal map by minimizing specific energy. Representative work includes the least square conformal map [62], the ABF [120], and its more efficient version ABF++ [100] and LinABF [90]. Meanwhile, the discrete conformal map also

introduced many new topics from its continuous counterpart, such as the holomorphic 1-form [33], Ricci flow [75], circle packing [118], and so on. Many creative ways for computing a conformal map were proposed.

Basically all these methods can be classified by their output: the vector coordinate, the derivative of the map and the metric. Most methods include the early stage work, which tries to output the vector valued position for each vertex, which is exactly the map itself. And some methods output the differential form, which is the derivative of the map. The final map can be obtained by integrating the differential 1-form. In addition, for some work like Ricci flow, the circle pattern [51], the discrete conformal equivalence in [104] or the traditional ABF [120], we simply compute the metric for the target domain (angle or edge length), then recover the map by embedding the surface into a canonical space.

After the rapid development in the last decade, the traditional discrete conformal map theorem owns a wealth of useful results. In contrast, study on the quasi-conformal, which is the extension of the conformal map, has been lacking and there are few results. For many situations in real life, the conformal map does not exist, which means that most of existing algorithms for computing the conformal maps are useless.

After 2010, some work [109, 70] started to focus on research into the extremal quasi-conformal map and the Teichmüller map. Intuitively speaking, the extremal quasi-conformal map is a map, which is as close as possible to the conformal map if a conformal map does not exist. One of important reasons why quasi-conformal map is so popular is that the quasi-conformal always exists for the most constraint situations while the conformal map can not accept the strong constraints more than three points. As far as we know, our work is the pioneer in computing extremal quasi-conformal map, since for the first time it has proposed a practical algorithm for the computing of a quasi-conformal map on a high genus model.

In the next section we will introduce the detail work for conformal map by its target domain, namely, the topological disk, genus zero surface and high genus surface.

2.2 Related Work for Conformal Map

2.2.1 Topological Disk

Most work in the early stage of the conformal map was on the topological disk. When the model is not a topological disk, we apply an independent cutting stage to cut the model into a topological disk, although it may result in the discontinuity of the cutting boundary. A lot of work uses the heuristic approach to approximate the angle preserving property of the conformal map.

In regard to parametrization of the topological disk, the early work actually did not focus on the conformal map, but it gave the inspiration to many subsequent works on the conformal map. Several early works on the parametrization [26, 27, 22] used Tutte's method in [107]. The basic idea of their method is that firstly we map the boundary to the target domain, then build the interior boundary using the linear system. Since solving the linear system is efficient and easy to implement, these methods are quite popular. However, these methods also suffer many problems. For example, they require the boundary to be convex to obtain a bijective map and the map quality will be quite poor when the target boundary is far from the original one. Later, some works [57, 55, 126] improved the result by considering the boundary as a part of the solution.

One important work following this line is the LSCM, which was proposed by Levy et al. [62] in 2002. Their method used an approximation of the Riemann-Cauchy equation to compute the result; and they proved that the result map exists uniquely. Another similar work is that of Desbrun et al. [73]. They computed the map from the discrete Dirichlet

energy and used the conformal parametrization as a tool for interactive geometry. Ray et al.[89] proposed the HLSCM, which improved the LSCM with its hierarchical solver. The MIPS method [47, 41] is another work that optimizes the nonlinear function, which measures the mesh conformality to compute the map.

Liu et al. [68] proposed another way to compute the map by minimizing the distortion energy. The advantage of their method is that they could control different types of distortion in an unified framework; and they proved their method achieved output, which was the same as LSCM when using as-similar-as-possible energy. Their method also involved solving a sparse linear system and the key for their algorithm is to represent the distortion by the transformation matrix for each triangle.

Figure 2.1: Traditional method like ABF will cut the model to the topological disk first then map it to the canonical domain, this figure is courtesy from Floater et. al. [28].

ABF [99] is another famous method for computing the angle preserving for a topological disk. Unlike previous work which solved the map based on the vertex coordinates directly, ABF used the angle values to represent the final map. They specify the planar constraints on each vertex; and the sum of the angles in each triangle should be π , which makes the energy highly non-linear. After we get the angle values, we can embed the result into the plane to get the final result.

The original ABF method has many weaknesses: the solution is slow and it is unstable when we try to converge the angle value to the coordinates for large meshes. Sheffer

et al. [100] proposed the ABF ++ to address these two problems. After using the hierarchical solution, the ABF ++ speeded up ABF significantly, and they also proposed a stable algorithm to convert the angle to the planar coordinate. Zayer et al. [120] proposed a way to guarantee the global bijectivity for the ABF [90] and they proposed a linear version of the ABF in 2007.

There are also some other works, which try to compute the angle-preservation problem via the circle-packing pattern. Bowers and Stephenson [11] used the inversive distance circle packing to flatten the topological disk mesh. And later, Hurdal et al. [42] and Bowers and Hurdal [10] computed angle-preserving parametrization using a variation of the circle-packing formulation described by Collins and Stephenson [14]. Kharevych et al. [52] use a circle pattern approach whereby each circle corresponds to a mesh face.

2.2.2 Genus Zero Model

Compared with other genus models, one advantage of computing the conformal map on the genus zero model is that the harmonic map is exactly a conformal map [97] on the genus zero model. So computing the conformal map on genus zero model is easier than in the other case. Another advantage of the spherical model is that it has a natural seamless continuous map for the genus-0 models.

Since many models used in computer graphics are genus-0 models, such as the bunny, brain surface, and so on, the conformal map between the genus-0 models has received much attention. Many papers [31, 37, 94, 85, 92] have been published on this subject. A rigorous theory for the map on the genus-0 model was also built [32, 31].

One common approach to developing a spherical map is to use the barycentric iteration, which is similar to applying Tutte's [107] on the planar domain. Many works

Figure 2.2: David's head is mapped to the sphere domain, this figure is courtesy from Gu et. al. [32]

[2, 32, 33, 54] have used the Gauss-Seidel iterations to obtain the result. In [31], Gotsman et al. showed a way to generalize the barycentric coordinates to the sphere with all the advantages. The foundation of their generalization was the results from the spectral graph theory developed by Colin de Verdiere [17] and Lovasz and Schrijver [69]'s extension.

Haker et al. [37] used a method which mapped the planar parametrization first, then use the stereographic projection to obtain the result. Their method suffered from the instability limitation. The stereo-graphic projection is non-linear, so it will cause some accuracy problems if we use the piecewise linear map to approximate the projection, the problem will be quite severe in the region near the north pole.

Later, Gu et al. [32] proposed a way to compute the global conformal map on genus-0 surfaces by a nonlinear optimization method. Their method does not need the stereographic projection, which makes it more accurate and stable. The key to their optimization is the gradient descent and they compute the gradient on the sphere's tangent spaces.

Recently, R. Lai et al. [56] proposed a fast and robust algorithm to compute the conformal map for genus-0 surface. Their method significantly improves the efficiency of the previous method [32] and removes the undesired folding with the model having sharp features.

Birkholz [7] applied the hierarchical approach to the embedding of spherical angles. The hierarchical method makes the convergence more stable. However, their method can only get an approximated solution.

The ABF method [98] is also extended to spherical embedding, which is formulated by spherical angles. Later, Zayer et al. [119] proposed a spherical parametrization method, which can have user-prescribed poles.

2.2.3 High Genus Model

Figure 2.3: Compute the conformal map for high genus model by holomorphic 1-form, X. Gu et al., this figure is courtesy from Gu et. al. [32].

The first work on the conformal map for the high genus model is [33]. In their paper, they used the holomorphic 1-form to solve the conformal map for the high genus model. Intuitively speaking, the holomorphic 1-form can be regarded as the derivative of the conformal map. They computed the homology bases on the triangular mesh first, which is the simplicial complex structures. After that, they built the cohomology bases and

applied a diffusion procedure to make these become harmonic 1-forms. Then they could get the bases for the holomorphic differentials. The conformal parametrization on the mesh can be found by the integral of the holomorphic 1-form. They generalized their work to the arbitrary model with holes at [34].

And later, W. Zeng et al. [125] use the holomorphic 1-form to implement the generalized Koebe's method, which map the multiply connected domains to canonical domain. The canonical domain is the disk with round hole inside. Suppose the multiply connected domain have n holes, they iteratively closed the $n - 1$ holes and map the hole left to the annulus by holomorphic 1-form.

Figure 2.4: Compute the conformal map for high genus model by Ricci flow, this figure is courtesy from Gu et. al. [32].

Another important tool for computing the conformal map on the arbitrary genus model is the Ricci flow. According to the universal covering space theorem, all the genus one models can be conformally mapped to the Euclidean plane, while all the other high genus models can be conformally mapped to the hyperbolic plane. All the above canonical space admit the constant curvature, which makes the Ricci flow quite powerful for such situations.

The intrinsic curvature flow was first studied in differential geometry, and was transplanted to computer graphics in the last decades. The Ricci flow appeared in R. Hamilton's work [38] for the first time, which is defined on the general Riemannian manifolds. Later he proposed the surface Ricci flow in [39]. Perelman used the Ricci flow as an important tool for the proof of Poincaré's conjecture [83], which brought the Ricci flow much attention. The first work on the discrete surface Ricci flow is Chow and Luo's [13]. While Jin et al. [75] for the first time brought the Ricci flow to the computer graphics community.

Circle packing is closely related to the Ricci flow and has played an important role in the development of the discrete Ricci flow. Thurston introduced the circle packing method [105] in 1976. In 1985 Thurston [106] proposed his conjecture that the circle packing can be used to approximate the Riemann map, which was proved by Rodin and Sullivan [91] in 1987. Collins and Stephenson [14] implemented the algorithm for this idea in 2003.

Guo [36] proved that Euclidean and hyperbolic circle packing on the closed triangular mesh with the prescribed inversive distance had the local rigidity. Bowers and Hurdal [10] proposed the circle pattern in 2003 and Bobenko et al. [8] proved that it is a minimizer of some convex energy. Kharevych et al. [52] proposed an efficient circle pattern algorithm for the computer graphics community. Luo [72] studied the discrete version of Yamabe flow for surfaces, and Springborn et al. [104] used a similar formulation to compute the conformal equivalent metrics with prescribed curvature.

2.3 Related Work for Quasi-Conformal Map

This part reviews the related work on quasi-conformal maps. Conformal map has the desirable property but also have strong restrictions, which make it unsuitable for some

Figure 2.5: Extremal quasi-conformal map on the planar domain, this figure is courtesy from Weber et. al.[109].

real life applications. So the study of quasi-conformal map is an important topic as the extension of the conformal map. Compared to the large body of literatures in conformal maps, there are only a few work in quasi-conformal maps.

Daripa [15] proposed an algorithm to solve the Beltrami equation in unit disk, their algorithms are based on a representation of the solution in terms of double integral, and they provided the numerical method for their algorithm.

Li et al. [65] computed a diffeomorphism between two surfaces of the same topology by computing a harmonic map using the uniformization metric, which globally minimizes the stretching energy. The resulting map converges to the extremal quasi-conformal map if the surface is of genus-1.

Zeng et al. [123] computed the quasi-conformal maps by solving the Beltrami equation on simply and multiply connected domains. Their strategy contains three steps, first they compute the conformal map from the original surface to the canonical domain, then they deform the metric on the canonical domain by the Beltrami coefficient, and finally, they compute the new conformal map to the canonical domain under the new metric, which is the final quasi-conformal map.

Later, Zeng et al. [122] proposed the discrete Beltrami flow to compute quasi-conformal map of non-trivial topology. Their method defines a vector field by an intrinsic PDE in the local conformal coordinate. Then the vector field is used to adjust the map according to the perturbation of the intrinsic Beltrami differential.

Weber et al. [109] computed the Teichmüller map for genus-0 models with boundaries. With a harmonic map as input, their method deforms the harmonic map to a T-map by minimizing the discrete least square Beltrami energy with user-provided boundary condition. Although the energy is nonlinear, they showed that their algorithm converges reliably for a broad range of simply and multiply connected domain shapes and boundary data.

Recently, Lui et al. [70] proposed a linear method to compute Teichmüller map on simply or multiply connected domains. Their algorithm is based on the iterative scheme, which is called quasi-conformal iteration. They iteratively compute the Beltrami coefficient and recover the map from the Beltrami coefficient. The limitation of their method is that they solve the map by using the global coordinates and need global parametrization to map the whole surface to the planar domain first.

Chapter 3

Mathematical Background

In this chapter, we first give a detailed introduction about conformal map and review its mathematical background, then we give a brief introduction to two basic tools for computing the conformal map, the holomorphic 1-form and Ricci flow, which will be used directly in the application part. We follow Gu et al. [35]’s symbols and notations for the most of definitions.

3.1 Conformal and Quasi-Conformal Map

3.1.1 Conformal Map

The conformal map appeared in the complex analysis theorem for the first time, which is associated with a holomorphic function. Suppose we have a function $f : \mathbb{C} \rightarrow \mathbb{C}, (x, y) \rightarrow (u, v)$ and $z = x + iy$, we use the following notation:

$$\begin{aligned} dz &= dx + idy, \\ d\bar{z} &= dx - idy, \\ \frac{\partial}{\partial z} &= \frac{1}{2} \left(\frac{\partial}{\partial x} - i \frac{\partial}{\partial y} \right), \\ \frac{\partial}{\partial \bar{z}} &= \frac{1}{2} \left(\frac{\partial}{\partial x} + i \frac{\partial}{\partial y} \right). \end{aligned}$$

A function f is called **holomorphic**, if it satisfies the following Cauchy-Riemann equations:

$$\begin{cases} \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \\ \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \end{cases}$$

If $f = u + iv$ is holomorphic, then $\frac{\partial f}{\partial \bar{z}} = 0$.

If a holomorphic function $f : D \rightarrow \mathbb{C}$ is bijective and f^{-1} is holomorphic, then f is biholomorphic, or conformal map.

The modern theorem of conformal map extends the theorem from complex plane to the general 2-D manifold. The **manifold** is the topological space with a special property, where each point can be covered by an open set, which is homeomorphic to the Euclidean space.

Suppose we have a topological space M , which is covered by the open sets $\{U_\alpha\}$; when the topological space M is manifold, for each U_α , The homeomorphism function ϕ_α is defined on the open set and maps the open sets to the Euclidean space $\mathbb{R}^n$. The open sets with the map function (U_α, ϕ_α) is called to be the **coordinate chart** of M , and all coordinate charts form the atlas of M . The **transition map** is defined on the two overlap coordinate charts $U_\alpha \cap U_\beta \neq \emptyset$:

$$\phi_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1} : \phi_\alpha(U_\alpha \cap U_\beta) \rightarrow \phi_\beta(U_\alpha \cap U_\beta) .$$

If the transition map $\phi_{\alpha\beta}$ is holomorphic, then the manifold is known as **Riemann surface**. The Riemann surface means that the surface can have the measurement to measure the angle on it, since the transition map is conformal the from one complex plane to another.

The atlas on Riemann surface is called **conformal atlas**. For two conformal atlas $\mathcal{U}_1, \mathcal{U}_2$, which covers the same topological surface S , we say they are **compatible** when their

Figure 3.1: Manifold definition, the transition map $\phi_{\alpha\beta}$ is defined on the intersection of two charts U_α and U_β .

union $\mathfrak{U}_1 \cup \mathfrak{U}_2$ is still a conformal atlas. It is not hard to prove that the compatible relation is the equivalent relation.

The conformal structure can be defined on the conformal atlas:

Definition 1 (Conformal Structure). Suppose we have a topological surface S , and $\mathfrak{S}$ is the set of all the conformal atlases on S , $\sim$ represents the compatible relation. Then $\mathfrak{S}$ can be classified by $\sim$. The **conformal structure** is the equivalence class in $\mathfrak{S} / \sim$.

Conformal structure means that we define a measurement to measure the angle on the surface. Suppose we have two arbitrary curves, which intersect at a point $\mathbf{p}$, the curves and their tangent at point $\mathbf{p}$ can be mapped to the planar domain, where we can calculate the angle. The map between the atlas is consistent (holomorphic), hence the angle can be uniquely determined. And each Riemann surface is associated with a conformal structure.

Definition 2 (Conformal Map). Given two Riemann surfaces M and $\tilde{M}$, suppose we have the map $f : M \rightarrow \tilde{M}$. For each $p \in M$, $\tilde{p} = f(p) \in \tilde{M}$, and any local charts (U, ϕ) covers p , $(\tilde{U}, \tilde{\phi})$ covers $\tilde{p}$, we have the coordinate function $z = \phi(p), \tilde{z} = \tilde{\phi}(\tilde{p})$, if $\tilde{z} = \tilde{\phi} \circ f \circ \phi^{-1}(z)$ is holomorphic, then we call the map f **conformal map**.

The metric is another important structure on the manifold. The **Riemannian Metric** on a smooth manifold M is the tensor $g_p : T_p M \times T_p M \rightarrow \mathbb{R}, \forall p \in M$, such that

- (i) $g_p(a_1 X_1 + a_2 X_2, b_1 Y_1 + b_2 Y_2) = \sum_{i,j=1}^2 a_i b_j g_p(X_i, Y_j)$,
- (ii) $g_p(X, Y) = g_p(Y, X)$,
- (iii) g_p is non-degenerate,
- (iv) $\forall p \in M$, there exists a local coordinates $\{x^i\}$ that $g_{ij} = g_p(\frac{\partial}{\partial x_i}, \frac{\partial}{\partial x_j})$ are C^∞ .

When the surface M has a Riemannian metric $\mathbf{g}$, the Riemannian metric is required to be compatible with the conformal structure, namely on each local chart of $\{U_\alpha, z_\alpha\}$

$$\mathbf{g} = e^{2\lambda(z_\alpha)} d\mathbf{z}_\alpha d\bar{\mathbf{z}}_\alpha .$$

Obviously, the local parameters are isothermal coordinates.

Suppose $f : (M, g) \rightarrow (N, h)$ is a smooth map between two Riemannian manifolds, $\forall \mathbf{p} \in M, f_* : T_{\mathbf{p}} M \rightarrow T_{f(\mathbf{p})} N$ is the tangent map. The pull back metric $f^* h$ induced by the map f is given by

$$f^* h(X_1, X_2) := h(f_* X_1, f_* X_2), \forall X_1, X_2 \in T_{\mathbf{p}} M .$$

When $f : (S_1, g_1) \rightarrow (S_2, g_2)$ is a conformal map, then we have $f_* g_1 = e^u g_2$. This metric equation provides two important facts, angle preserving and circle preserving.

Firstly, we can derive the most important properties for conformal map from this equation, which is *angle preserving*.

$$\begin{aligned}
 \cos \theta &= \frac{g(X_1, X_2)}{\sqrt{g(X_1, X_1)} \sqrt{g(X_2, X_2)}} \\
 &= \frac{e^{2u} g(X_1, X_2)}{\sqrt{e^{2u} g(X_1, X_1)} \sqrt{e^{2u} g(X_2, X_2)}} \\
 &= \frac{f^* h(X_1, X_2)}{\sqrt{f^* h(X_1, X_1)} \sqrt{f^* h(X_2, X_2)}}.
 \end{aligned}$$

Figure 3.2: The middle is a conformal map, which maps a face to the disk. We can see that the conformal map keeps the details well, unlike the quasi-conformal map in the right, this figure is courtesy from Zeng et. al. [123].

Angle preserving is the most important property of conformal map. This is the key reason why conformal map is widely used by the graphics community in many applications such as texture map (see Fig 3.2).

And on the other hand, $f_* g_1 = e^u g_2$, this equation is not related with the tangent vector, which inspires that the conformal map only scale the infinitesimal area around a point. It also means that the map will map the infinitesimal circle around the point p to the infinitesimal circle. This property will be generalized in the next part to the general map, which is called quasi-conformal map.

3.1.2 Quasi-Conformal Map

The quasi-conformal map is the generalization of conformal map. The conformal map will map the infinitesimal circle to the infinitesimal circle, but the quasi-conformal map will map the infinitesimal circle to the ellipse with bounded eccentricity.

Back to the map between two complex plane, suppose the function $f : \mathbb{C} \rightarrow \mathbb{C}, (x, y) \rightarrow (u, v)$. The function f is **quasi-conformal**, if it satisfies the following **Beltrami equation**:

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z},$$

where μ is a complex-valued function called the **Beltrami coefficient**. When $\mu = 0$, $f_{\bar{z}} = 0$ is the Cauchy-Riemann equation, so f is holomorphic and conformal.

Figure 3.3: Beltrami coefficient and Beltrami differential.

The Beltrami coefficient μ controls the conformality distortion. In the infinitesimal area around a point $p \in \mathbb{C}$, $f(z)$ can be approximated as follows by dropping the high order terms of the Taylor expansion:

$$\begin{aligned} f(z) &= f(p) + f_z(p)z + f_{\bar{z}}(p)\bar{z} \\ &= f(p) + f_z(p)(z + \mu(p)\bar{z}). \end{aligned}$$

The stretch map $S(z) \triangleq z + \mu(p)\bar{z}$ causes the conformality distortion, i.e., an infinitesimal circle is mapped to an infinitesimal ellipse. The maximal expansion (resp. shrinkage) has a factor of $|f_z|(1 + |\mu|)$ (resp. $|f_z|(1 - |\mu|)$) with direction $\arg(\mu)/2$ (resp. $\arg(\mu)/2 + \pi/2$). So the dilation of the ellipse K is given by the ratio $K = \frac{1+|\mu|}{1-|\mu|}$. See Figure 3.3(a).

The general Riemann surfaces does not have the unique global coordinate system; to define the quasi-conformal map between two general Riemann surfaces, the Beltrami coefficient is generalized to the Beltrami differential with the form $\mu(z)\frac{d\bar{z}}{dz}$. Given two overlapped charts (U_α, ϕ_α) and (U_β, ϕ_β) on a Riemann surface S , the Beltrami differential satisfies

$$\mu_\alpha(z_\alpha)\frac{d\bar{z}_\alpha}{dz_\alpha} = \mu_\beta(z_\beta)\frac{d\bar{z}_\beta}{dz_\beta},$$

where $\frac{dz_\beta}{dz_\alpha} = \frac{d}{dz_\alpha}\phi_{\alpha\beta}$ and $\phi_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1}$ is the transition function. See Figure 3.3(b).

An orientation preserving diffeomorphism $f : S_1 \rightarrow S_2$ is called quasi-conformal associated with $\mu(z)\frac{d\bar{z}}{dz}$ if for any chart (U_α, ϕ_α) on S_1 and any chart (V_β, ψ_β) on S_2 , the map $f_{\alpha\beta} = \psi_\beta \circ f \circ \phi_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha(z_\alpha)\frac{d\bar{z}_\alpha}{dz_\alpha}$.

Extremal quasi-conformal maps minimizes the maximal conformality, which is highly desired by many real-world applications when the conformal map does not exist. A Teichmüller map (T-map) is an extremal quasi-conformal map associated with a holomorphic function. The T-map between two Riemann surfaces S_1 and S_2 is defined as follows:

Definition 3 Teichmüller map: Let $f : S_1 \rightarrow S_2$ be a quasi-conformal map. f is said to be a Teichmüller map (T-map) associated to the integrable holomorphic function $\phi : S_1 \rightarrow \mathbb{C}$ if its associated Beltrami coefficient is of the form $\mu(f) = k\frac{\bar{\phi}}{|\phi|}$ for some

constant $k < 1$ and the holomorphic function $\varphi \neq 0$ with $\|\varphi\|_1 = \int_{S_1} |\varphi| < \infty$.

Extremal quasi-conformal map always exists but may not be unique. A T-map associated with a holomorphic function is the unique extremal map in its homotopic class.

3.2 Tools for Computing Conformal Map

In this section we will introduce two different tools for computing the conformal map, these tools will be important for computing the conformal map in some applications in the following chapter.

Holomorphic 1-form

Given a point $\mathbf{p}$ on the Riemann surface M , we have the tangent space $T_p M$, it is 2-dimensional vector space for the 2-D manifold. and $T_p^* M$ is the dual space defined on $T_p M$, which is also referred as cotangent space for $T_p M$. If $\omega \in T_p^* M$. then $\omega(v), v \in T_p M$ is the linear function defined on $T_p M$. We call ω **differential 1-form** at $\mathbf{p}$.

Just like the vector field on the surface, there are differential 1-form field for the surface. And combined with the coordinate map, which maps the surface to the complex plane, the differential 1-form will have the local representation:

$$\omega(x, y) = \omega_x(x, y)dx + \omega_y(x, y)dy,$$

where dx, dy are the dual vector in the cotangent space to $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}$, such that

$$\begin{cases} dx(\frac{\partial}{\partial x}), &= 1, & dx(\frac{\partial}{\partial y}) &= 0 \\ dy(\frac{\partial}{\partial x}), &= 0, & dy(\frac{\partial}{\partial y}) &= 1. \end{cases}$$

For the point $\mathbf{p}$ on the manifold M , the **tensor** $\mathcal{T}$ is a multi-linear map defined on the tangent space $T_p M$ and its dual cotangent space $T_p^* M$.

$$\mathcal{T}_p : T_p M \times T_p M \cdots \times T_p M \times T_p^* M \times \cdots \times T_p^* M \rightarrow R.$$

It is called tensor of type (m, n) , when it have m $T_p M$ and n $T_p^* M$ in the parameter. The tensor is a generalization of the vector and dual vector. It is not hard to find that the dual vector is actually the $(1, 0)$ tensor and the vector is the $(0, 1)$ tensor, the differential n -form is the $(n, 0)$ tensor. A Riemannian metric on the manifold M is an inner product on $T_p M$, which can be also regarded as differential 2-form.

The **tensor product** is an operator to combine the (k, l) tensor and (k', l') tensor to be a $(k + k', l + l')$ tensor:

$$T \otimes T'(v_1, \dots, v_{k+k'}, \omega_1, \dots, \omega_{l+l'}) = T(v_1, \dots, v_k, \omega_1, \dots, \omega_l) T'(v_{k+1}, \dots, v_{k+k'}, \omega_{l+1}, \dots, \omega_{l+l'}).$$

One of the important operations related to the tensor product is the wedge product, which is used to combine two low dimension differential form to a high dimension differential form.

Given the m_1 -form ω_1 and an m_2 -form ω_2 , the **wedge product** $\wedge$ is an operator to produce a $m_1 + m_2$ form which can be represented by:

$$(\omega_1 \wedge \omega_2)(\xi_1, \dots, \xi_{m_1+m_2}) = \sum_{\sigma \in S_{m_1+m_2}} \frac{(-1)^\sigma}{m_1! m_2!} \omega_1(\xi_{\sigma(1)}, \dots, \xi_{\sigma(m_1)}) \cdot \omega_2(\xi_{\sigma(m_1+1)}, \dots, \xi_{\sigma(m_1+m_2)}).$$

The σ means the parity of permutation.

The computation of conformal map will need the differential 1-form and 2-form, it is not hard to see that

$$\begin{aligned} dx \wedge dx &= dy \wedge dy = 0, \\ dx \wedge dy &= -dy \wedge dx, \\ dx \wedge dy(\xi_1, \xi_2) &= \frac{1}{2} [dx(\xi_1)dy(\xi_2) - dx(\xi_2)dy(\xi_1)]. \end{aligned}$$

Suppose $\omega_1 = adx + bdy$, $\omega_2 = cdx + ddy$. The wedge product will be

$$\begin{aligned}\omega_1 \wedge \omega_2 &= (a dx + b dy) \wedge (c dx + d dy), \\ &= ac dx \wedge dx + ad dx \wedge dy + bc dy \wedge dx + bd dy \wedge dx, \\ &= (ad - bc)dx \wedge dy.\end{aligned}$$

Given an n-dimension manifold, if the manifold is embedded in $\mathbb{R}^{n+1}$, then it has an induced Euclidean metric.

Suppose an n-dimension manifold M is equipped with a Riemannian metric. There exists the oriented orthonormal basis of vector fields locally, such that

$$\left\{ \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \cdots \frac{\partial}{\partial x_n} \right\}$$

form an oriented orthonormal basis. Let

$$\{dx_1, dx_2, \dots, dx_n\}$$

be the dual 1-form basis.

Suppose there is a scalar function $f : M \rightarrow \mathbb{R}$ and differential form $\omega = \omega_I dx_I$. The **exterior derivative** can be defined as:

$$\begin{aligned}df &= \sum_i \frac{\partial f}{\partial x_i} dx_i, \\ d\omega &= d(\omega_I dx_I) = d\omega_I \wedge dx_I.\end{aligned}$$

Let the space of the differential k-form to be $\Omega^k(M)$, the **Hodge star operator** $*$ is the linear map from $\Omega^k(M) \rightarrow \Omega^{n-k}(M)$, defined as

$$*(dx_1 \wedge dx_2 \wedge \cdots \wedge dx_k) = dx_{k+1} \wedge dx_{k+2} \wedge \cdots \wedge dx_n.$$

Suppose $\sigma = (i_1, i_2, \dots, i_n)$ is the new permutation of $(1, 2, \dots, n)$. the Hodge star will be

$$*(dx_{i_1} \wedge dx_{i_2} \wedge \dots \wedge dx_{i_k}) = (-1)^\sigma dx_{i_{k+1}} \wedge dx_{i_{k+2}} \wedge \dots \wedge dx_{i_n}.$$

The ***codifferential operator*** is another linear map:

$$\delta : \Omega^k(M) \rightarrow \Omega^{k-1}(M),$$

which can be represented by the exterior derivative and Hodge star:

$$\delta = (-1)^{k+1+k(n-k)} * d *.$$

Based on the codifferential operator and the exterior derivative, we can define the ***Laplace operator*** $\Delta : \Omega^k(M) \rightarrow \Omega^k(M)$:

$$\Delta = \delta d + d \delta.$$

And the ***harmonic form*** is the differential k-form, which satisfied that

$$\Delta \omega = 0.$$

The ω is known as ***closed*** if $d\omega = 0$ and ***co-closed*** if $d*\omega = 0$. An n-form ω is an ***exact form*** if $\omega = df$, where f is an (n-1)-form. The necessary and sufficient condition for a differential k-form to be a harmonic form is that the form is both closed and co-closed.

Definition 4 Holomorphic 1-form *Given a Riemann surface S , the holomorphic 1-form is the $\omega + *\omega i$, here the ω is harmonic 1-form and $*$ is the Hodge star.*

Ricci flow

Ricci flow is another important tool for conformal map, it is a geometric flow, which is similar to the heat diffusion. In history, it is the primary tool for Grigori Perelman to

solve the the Poincare conjecture. The discrete Ricci flow for the computation of the conformal map have a lot of advantages, such as robust and fast. In this part we will introduce the basic concept for Ricci flow.

Yamabe Problem:[115]

Given a surface (S, g) whose Gauss curvature is K , how can we change the metric g to $\tilde{g} = e^\mu g$, while $(S, \tilde{g})$ has constant Gauss curvature $\bar{K}$?

This problem is hard to conquer and Yamabe has introduced his famous equation, the Yamabe equation, to solve this problem.

$$e^{2\mu} \bar{K} = K - \Delta_g \mu .$$

When K and $\bar{K}$ is known, we can get μ by solving this question, and then we will obtain the $\tilde{g}$ we want.

However, this equation cannot inspire us any practical solution to solve the Yamabe problem since it is highly non-linear; In 1981, Hamilton [38] proposed his famous Hamilton's Ricci flow, which is a key to approach this equation.

Hamilton's Surface Ricci Flow: Given a closed surface, which is equipped with the Riemannian metric $\mathbf{g}$, the Ricci flow on the surface is defined as

$$\frac{d\mathbf{g}}{dt} = -K\mathbf{g},$$

The Ricci flow will converge to the metric, which have the constant Guassian curvature everywhere, if we keep the total area of the surface.

Given $\mathbf{g}(t) = e^\mu \mathbf{g}(0)$, the Ricci flow will become

$$\frac{d\mu}{dt} = \bar{K} - K.$$

3.3 From Continuous to Discrete

In the previous section, we have introduced the basic mathematical theorem, which we will use in this thesis. All theorems are built under the smooth setting. However, for computer graphics, it is not practical to require our models to be smooth. Hence it is important for us to build a discrete counterpart for conformal map. After several years of development, the discrete conformal map and discrete differential geometry became a very rich topic and can almost be regarded as an independent subject.

3.3.1 Basic Structure and Discrete Conformal Map

Figure 3.4: The surfaces are represented by triangular mesh.

In computer graphics, for the most situations, the surface is represented by the piecewise linear mesh like the triangular mesh, as shown in Figure 3.4. When the number of triangles is limited, there will always be some errors between the discrete mesh and the smooth surface.

The triangular mesh is a set of triangles who are glued together edge-to-edge; we use (V, E, T) to represent the sets of vertices, edges and triangle. Triangular mesh (V, E, T)

will form the CW complex, where the vertices, edges, and faces are regarded as 0-cells, 1-cells, and 2-cells, respectively. And the metric on the triangular mesh is a positive function defined on the edge set E , $l : E \rightarrow \mathbb{R}_{>0}$; and the metric should satisfy the triangular inequalities for all the faces in T . So the whole triangular mesh with the metric can be represented by (T, l) . The embedding of the triangular mesh (T, l) is the function defined on the $\mathcal{E} : V \rightarrow \mathbb{R}^3$ and $|\mathbf{v}_i \mathbf{v}_j| = l_{ij}$.

The **Hausdorff distance** is a good measurement to measure the distance between two surfaces. Given two surfaces M_1, M_2 , the Hausdorff distance $d_H(M_1, M_2)$ will be

$$d_H(M_1, M_2) = \inf\{\varepsilon > 0 \mid M_1 \subset B_\varepsilon(M_2), M_2 \subset B_\varepsilon(M_1)\}.$$

The $B_\varepsilon(M)$ is the point set $\{x \in \mathbb{R}^3 \mid \exists y \in M : d(x, y) < \varepsilon\}$. A fundamental fact for the discrete mesh is that: given any $\varepsilon > 0$, there will be a discrete mesh $d_H(M_\varepsilon, S) < \varepsilon$. This means that if the number of triangle meshes is not limited, we can always find a discrete mesh M , which is as close as we want to the smooth surface S .

The definition for continuous conformal map on Riemannian surface can also be transplanted to the discrete triangular surface. Suppose there is a map $f : S_1 \rightarrow S_2$ between Riemannian manifold $(S_1, g_1), (S_2, g_2)$, the map is conformal if and only if $f_* g_1 = e^u g_2$. For the discrete situation, suppose there are two triangular surfaces $(T_1, l_1), (T_2, l_2)$, which are embedded in $\mathbb{R}^3$. The discrete map g maps the vertex set of T_1 onto the triangular surface T_2 , which assigns the T_1 a new metric $\tilde{l}_1$, the map is **conformal** only when

$$\tilde{l}_{1ij} = e^{\frac{1}{2}(u_i + u_j)} l_{1ij}.$$

The scalar function u is defined on the vertices set, $u : V \rightarrow \mathbb{R}$. This definition introduces an equivalent relation between two triangular meshes which have the same topological connection $(T, l), (T, \tilde{l})$. The equivalent class from this relation can be regarded as discrete conformal class for the triangular mesh T .

Remark: Due to the intrinsic difference from the continuous situation to the discrete, we should also notice that there are some other definitions for the discrete conformal map, which starts from different aspects of the continuous conformal map, like the conformal deformation from the circle packing in discrete Ricci flow. Some computer graphics papers do not even care about the detailed definition of discrete conformal map and just mention the continuous case. It does not have problem when we regard the discrete conformal map on the triangular mesh as an extension of the continuous case, which simply makes the smooth problem available for the discrete setting. We put this definition here for the soundness of thesis.

3.3.2 Discrete Holomorphic 1-Form and Ricci Flow

Discrete holomorphic 1-form

For the discrete setting, 1-form on the surface is the function $\omega_E : E \rightarrow \mathbb{R}$, and 0-form is $\omega_V : V \rightarrow \mathbb{R}$, and similarly the 2-form is defined on the surface as $\omega_T : T \rightarrow \mathbb{R}$. Given a triangle $[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$ and ∂ the operator to obtain the boundary, the exterior operator d will be

$$\begin{aligned} d\omega_V([\mathbf{v}_0, \mathbf{v}_1]) &= \omega_E([\mathbf{v}_0, \mathbf{v}_1]) = \omega_V(\partial[\mathbf{v}_0, \mathbf{v}_1]) = \omega_V(\mathbf{v}_1) - \omega_V(\mathbf{v}_0) \\ d\omega_E([\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]) &= \omega_T([\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]) = \omega_E(\partial[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]) \\ &= \omega_E([\mathbf{v}_0, \mathbf{v}_1]) + \omega_E([\mathbf{v}_1, \mathbf{v}_2]) + \omega_E([\mathbf{v}_2, \mathbf{v}_0]). \end{aligned}$$

The closed 1-form is the 1-form, which has the property $d\omega = 0$. Suppose $[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$ is an arbitrary face. Then

$$d\omega([\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]) = \omega\partial[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2] = \omega[\mathbf{v}_0, \mathbf{v}_1] + \omega[\mathbf{v}_1, \mathbf{v}_2] + \omega[\mathbf{v}_2, \mathbf{v}_0] = 0.$$

And the δ operator will be

$$\begin{aligned}\delta\omega_E([\mathbf{v}]) &= \omega_V([\mathbf{v}]) = \sum_{[\mathbf{v}, \mathbf{w}] \in E} \omega_E([\mathbf{v}, \mathbf{w}]), \\ \delta\omega_T([\mathbf{v}_0, \mathbf{v}_1]) &= \omega_E([\mathbf{v}_0, \mathbf{v}_1]) = \sum_{[\mathbf{v}_0, \mathbf{v}_1, \mathbf{w}] \in T} \omega_T([\mathbf{v}_0, \mathbf{v}_1, \mathbf{w}]).\end{aligned}$$

The co-closed 1-form is the 1-form which satisfied that $\delta\omega = 0$.

After we have the discrete definition for $d\omega$ and $\delta\omega$, we can introduce the discrete harmonic 1-form. Just exactly like its continuous situation. The discrete harmonic 1-form is the function on E_T , which satisfies both $d\omega = 0$ and $\delta\omega = 0$.

Figure 3.5: The scalar harmonic field on the bunny, which can induce the harmonic 1-form on the edge.

From these two equations the harmonic 1-form can be computed in a very easy way. Firstly, we compute the harmonic function defined on each vertex $\mathbf{v}$ by the linear equation system:

$$\sum_{\mathbf{w} \in E} (f(\mathbf{w}) - f(\mathbf{v})) = 0.$$

Then the harmonic 1-form ω can be obtained by $\omega(\mathbf{v}_i \mathbf{v}_j) = f(\mathbf{v}_j) - f(\mathbf{v}_i)$ for each edge $\mathbf{v}_i \mathbf{v}_j$. It could be easy to verify that it satisfies both $d\omega = 0$ and $\delta\omega = 0$.

Suppose our mesh M embedded in $\mathbb{R}^3$, and if given the closed 1-form ω defined on the edges E , then there exists a unique vector $\mathbf{w}$ on the plane spanned by vertices $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$, such that

$$\langle \mathbf{w}, \mathbf{v}_1 - \mathbf{v}_0 \rangle = \omega([\mathbf{v}_0, \mathbf{v}_1]),$$

$$\langle \mathbf{w}, \mathbf{v}_2 - \mathbf{v}_1 \rangle = \omega([\mathbf{v}_1, \mathbf{v}_2]),$$

$$\text{and } \langle \mathbf{w}, \mathbf{v}_0 - \mathbf{v}_2 \rangle = \omega([\mathbf{v}_2, \mathbf{v}_0]).$$

Suppose $\mathbf{n}$ is the normal vector for the plane triangle $[\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2]$, then the vector $\mathbf{w}$ can be explicitly calculated as:

$$\mathbf{w} = -\omega[\mathbf{v}_0, \mathbf{v}_1] \mathbf{v}_2 \times \mathbf{n} - \omega[\mathbf{v}_2, \mathbf{v}_0] \mathbf{v}_1 \times \mathbf{n} - \omega[\mathbf{v}_1, \mathbf{v}_2] \mathbf{v}_0 \times \mathbf{n}.$$

Therefore, any closed 1-form ω corresponds to a vector valued 2-form

$$\mathbf{w} : F \rightarrow \mathbb{R}^3.$$

We should notice that $\mathbf{w}$ must fall into the plane spanned by the triangle $\mathbf{v}_0 \mathbf{v}_1 \mathbf{v}_2$.

Hence, given any two closed 1-form ω_1 , and ω_2 , suppose the area of the face is A , we can define the vector value 2-form on the surface by

$$\omega_1 \wedge \omega_2([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]) = \mathbf{w}_1 \times \mathbf{w}_2 \cdot \mathbf{n} A.$$

Now we can compute the approximation of the Hodge star. The discrete Hodge star can be approximated in numerous ways, here we will introduce one simple and efficient method, which can get the accurate result.

Given 1-form ω , we can get the vector 2-form $\mathbf{w}$ for each face, rotate the $\mathbf{w}$ in the plane spanned by the face by 90 degrees to get $*\mathbf{w}$, and we can obtain the another vector 2-form. But in this case, we cannot convert $*\mathbf{w}$ to a closed 1-form directly, because the consistency relation cannot hold any more.

$$\langle *\mathbf{w}([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]), \mathbf{v}_j - \mathbf{v}_i \rangle \neq \langle *\mathbf{w}([\mathbf{v}_j, \mathbf{v}_i, \mathbf{v}_l]), \mathbf{v}_j - \mathbf{v}_i \rangle.$$

Input: The mesh $M = (V, E, F)$

Output: The holomorphic 1-form $\omega : E \rightarrow \omega + *\omega i$

- 1 Compute the harmonic field for each vertex $f : V \rightarrow R$ by $\sum_{\mathbf{v}\mathbf{w} \in E} (f(\mathbf{w}) - f(\mathbf{v})) = 0$.
- 2 Compute the harmonic 1-form for each edge $\omega(\mathbf{vw}) = f(\mathbf{w}) - f(\mathbf{v})$.
- 3 Compute the vector 2-form by $\omega_1 \wedge \omega_2([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]) = \mathbf{w}_1 \times \mathbf{w}_2 \cdot \mathbf{nA}$.
- 4 Rotate the vector 2-form on each face by 90 degree and take the average to get $*\omega$ for each edge.
- 5 Solve the minimization in (3.1) to obtain new $*\omega$.
- 6 Output $\omega + *\omega i$ for each edge.

Algorithm 1: The algorithm pipeline to compute the holomorphic 1-form.

This is mainly caused by the discrete approximation. In order to mitigate the problem, we can compute $*\omega[v_i, v_j]$ by taking the average:

$$*\omega([\mathbf{v}_i, \mathbf{v}_j]) = \frac{1}{2}(*\mathbf{w}([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]), \mathbf{v}_j - \mathbf{v}_i) + \langle *\mathbf{w}([\mathbf{v}_j, \mathbf{v}_i, \mathbf{v}_l]), \mathbf{v}_j - \mathbf{v}_i \rangle.$$

The result of the average is better but still inaccurate. We can apply a smooth process to the algorithm, in order to make the vector 2-form more consistent. The hill climbing, which starts from $*\mathbf{w}$ to minimize the following energy, will be efficient for this problem.

$$E = \sum_{[\mathbf{v}_i, \mathbf{v}_j] \in E} (*\mathbf{w}([\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]), \mathbf{v}_j - \mathbf{v}_i) - \langle *\mathbf{w}([\mathbf{v}_j, \mathbf{v}_i, \mathbf{v}_l]), \mathbf{v}_j - \mathbf{v}_i \rangle)^2 \quad (3.1)$$

The result is much better after the smooth process (see Figure 3.6).

The holomorphic 1-form on the discrete surface can be computed by the discrete harmonic 1-form and Hodge star, which is $h : E_T \rightarrow \mathbb{C}$, where the real part is harmonic 1-form while the image part is the Hodge star of its real part.

Figure 3.6: Left: The harmonic 1-form ω , middle: $*\omega$ without smoothing; right: $*\omega$ after smooth.

Ricci flow

This section introduces the representation of the discrete Ricci flow on the triangular mesh. The discrete Ricci flow can be used to design the Riemannian metric for the mesh, which is conformal to the original one and has constant Gaussian curvature.

Figure 3.7: The corner angle in one triangle.

Given a triangular mesh $\{T, l\}$, the corner angle for vertex $\mathbf{v}_i$ on the triangle $[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]$ can be determined like this, see Figure 3.7:

$$\theta_i = \arccos \frac{l_{ij}^2 + l_{ki}^2 - l_{jk}^2}{2l_{ij}l_{ki}}.$$

The discrete Gaussian curvature is defined as angle deficiency; And the definition on the interior vertex is slightly different with the definition on the boundary. Suppose $\mathbf{v}_i \in V$,

with surrounding faces $[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \in T$. We use the θ_i^{jk} to represent the corner angle in $[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]$ at the vertex $\mathbf{v}_i$. Then the discrete Gaussian curvature of $\mathbf{v}_i$ is given by

$$K_i = \begin{cases} 2\pi - \sum_{j,k} \theta_i^{jk}, & v_i \notin \partial M \\ \pi - \sum_{j,k} \theta_i^{jk}, & v_i \in \partial M. \end{cases}$$

where ∂M means the boundary of the mesh.

It is not hard to verify that for the discrete Gaussian curvature, the following equation still hold:

$$\sum_i K_i = 2\pi \xi(M),$$

where $\xi(M) = |V| + |F| - |E|$ is the Euler number of M . That means that the total of discrete Gaussian curvature is determined by the topology of the mesh, which is exactly like the smooth situation.

Figure 3.8: The circle packing on the mesh is that each vertex equips a circle c with radius r .

Before we move to the discrete Ricci flow, we need to introduce the circle packing (see Figure 3.8), which is the key component for the discrete Ricci flow. The *circle packing*

can be regarded as that each vertex $\mathbf{v}_i \in V$ is associated with a circle c_i with radii γ_i . There will appear two different special situations according to the radii of two circles c_i, c_j and the edge length $\mathbf{v}_i \mathbf{v}_j$: if any two circles c_i and c_j on the edge $\mathbf{v}_i \mathbf{v}_j \in E$ will intersect and the angle they form ϕ_{ij} is acute; for this situation, the circle packing is the Andreev-Thurston circle packing. Otherwise if any two circles are disjoint, the circle packing will become the distance circle packing, see Figure 3.9.

Figure 3.9: Left: Andreev-Thurston circle packing, where each angle between the circles is acute. Right: distance circle packing.

For both cases, the *inversive distance* can be defined by

$$I(c_i, c_j) = \frac{l_{ij}^2 - \gamma_i^2 - \gamma_j^2}{2\gamma_i\gamma_j}.$$

It is not difficult to tell that $I(c_i, c_j)$ is the generalization of the $\cos\phi_{ij}$, and $I(c_i, c_j)$ will become exactly $\cos\phi_{ij}$ when the circle packing is Andreev-Thurston circle packing.

When the $I(c_i, c_j)$ is fixed for each edge, the discrete metric l_{ij} will be determined by c_i . Suppose the $\mathbf{v}_i$ have the circle c_i with the radius γ_i . Then the edge length of l_{ij} will be

$$l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2I_{ij}\gamma_i\gamma_j}.$$

So that the mesh (T, l) can also be represented by (S, Γ, I) , where $\Gamma : V \rightarrow \mathbb{R}_{>0}$ and $I : E \rightarrow \mathbb{R}$.

When we keep I invariant and deform Γ , the discrete conformal deformation can be obtained, which means that the new metric will be conformal with the original one. The *discrete Ricci flow* can be represented by the deformation of $\gamma_i = e^{u_i}$, then

$$\frac{du_i}{dt} = \bar{K}_i - K_i,$$

where $\bar{K}_i$ is the target Guassian curvature.

Input: The mesh $M = (V, E, F)$, target curvature $\bar{K}$, threshold ε
Output: The new metric $l : E \rightarrow R_{>0}$ with target curvature.

- 1 Compute the initial circle radius for each vertex $\gamma_i = e^{u_i}$.
- 2 Compute the inversive distance I_{ij} for each edge.
- 3 **while** $\bar{K}_i - K_i > \varepsilon$ **do**
- 4 Deform u_i by $\bar{K}_i - K_i$.
- 5 Compute the new $\gamma_i = e^{u_i}$.
- 6 Compute the new metric for each edge $l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2I_{ij}\gamma_i\gamma_j}$.
- 7 Compute the K_i for the new metric.
- 8 **end**

Algorithm 2: Discrete surface Ricci flow.

Chapter 4

Computing the Extremal Quasi-Conformal Map on General Surface

The conformal map has demonstrated its value in a broad range of applications with the property of minimizing the angle distortion. However, for some real-world applications, the conformal map may not exist, especially when the map needs strong landmark matching or it is between two high genus models. For these kinds of situation, the Teichmüller mapping (T-map) is a special extremal quasi-conformal map, which is a perfect substitution for the conformal map. In this chapter, we present a novel way to compute extremal quasi-conformal maps on general surfaces, which is linear, local, easy to implement and has a very natural parallel structure.

4.1 Introduction

It is well known that all genus-0 surfaces can be conformally mapped to the unit sphere $\mathbb{S}^2$, thus, they are conformally equivalent. But the conformal maps between surfaces of high genus usually do not exist. For example, two tori are conformally equivalent if and only if their ratios of the radius of the tube and the radius of the swept circle

agree. In general, conformal maps are really rare for high genus models in real-world applications. A natural generalization of conformal map is the quasi-conformal map, which has bounded conformality distortion. Intuitively speaking, a conformal map sends an infinitesimal circle to an infinitesimal circle, while a quasi-conformal map sends an infinitesimal circle to an infinitesimal ellipse with bounded eccentricity.

Among all quasi-conformal maps, the extremal quasi-conformal map is of particular interests, since it minimizes the maximal conformal distortion. Extremal quasi-conformal map always exists but may not be unique. Teichmüller map (T-map) is a quasi-conformal map associated with a holomorphic function and it is unique in its homotopic class. Lui et al. [70] formulated the T-map between two simply (or multiply) connected domains as a variational problem, which minimizes the conformality distortion and the harmonic energy of the magnitude of the Beltrami coefficient. They proposed the quasi-conformal iteration algorithm to compute the Beltrami coefficient and reconstruct the T-map by using the linear Beltrami solver. Due to the global coordinate system required in computing the Beltrami coefficient, their method is limited to models with simple topology, such as simply or multiply connected domains, genus-0 and genus-1 models, for which one can construct the global coordinate system by using the global parameterization technique.

In this chapter, we propose a simple yet effective technique to compute the extremal quasi-conformal map on surfaces of *non-trivial topology*. Our method extends the linear Beltrami solver from the complex plane to the manifold setting. For each mesh vertex, we define a local coordinate chart, which is computed efficiently by using exponential map in the preprocessing stage. Rather than the global method [70], which solves the T-map via a global linear system, we iteratively compute the map in the local coordinate chart and then diffuse it through the entire mesh. Our method is linear, local, easy to

implement and has a very natural parallel structure. It is also robust and insensitive to the initial map. For the initial map of poor quality (i.e., non-bijection with large number of overlapped/self-intersected regions), our method can efficiently diffuse it to a high quality quasi-conformal map. We tested our method on both synthetic and real-world models of various topologies. The promising experimental results demonstrate the efficacy of our method. Figure 4.1 shows an example of Teichmüller map of a genus-8 model. To our knowledge, this chapter presents the first implementation of computing the extremal quasi-conformal map on surfaces of arbitrary topology.

Figure 4.1: The Teichmüller map from the genus-8 4-Kid model to a polycube domain (see the inset). The circle texture map illustrates the quality of the computed T-map. As shown in the histogram, the magnitude of the Beltrami coefficient $|\mu|$ ranges from 0 to 0.8. More than 75% of them are less than 0.2.

4.2 Quasi-Conformal Map and Beltrami Coefficient

As we have discussed in the previous mathematical background chapter, for the complex function $f : \mathbb{C} \rightarrow \mathbb{C}$, $f(z) = u(x, y) + \sqrt{-1}v(x, y)$, the Beltrami coefficient is:

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z}.$$

When $\mu = 0$, $f_{\bar{z}} = 0$ is the Cauchy-Riemann equation, so f is holomorphic and conformal. The Beltrami coefficient μ controls the conformality distortion.

Given the Beltrami coefficient μ , the quasi-conformal map f can be determined uniquely. With the Beltrami equation, we can write μ as

$$\mu(f) = \frac{(u_x - v_y) + \sqrt{-1}(v_x + u_y)}{(u_x + v_y) + \sqrt{-1}(v_x - u_y)}.$$

Let $\mu \triangleq \rho + \sqrt{-1}\tau$, then we can write v_x and v_y as linear combination of u_x and u_y

$$\begin{pmatrix} -v_y \\ v_x \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix}, \quad (4.1)$$

where

$$\begin{aligned} a_{11} &= \frac{(\rho - 1)^2 + \tau^2}{1 - \rho^2 - \tau^2}, \\ a_{12} &= a_{21} = \frac{-2\tau}{1 - \rho^2 - \tau^2}, \\ a_{22} &= \frac{(\rho + 1)^2 + \tau^2}{1 - \rho^2 - \tau^2}. \end{aligned}$$

Let us denote $A = (a_{ij})_{1 \leq i, j \leq 2}$. Similarly, u_x and u_y can also be written as linear combination of v_x and v_y , i.e., $(-u_y \ u_x)^T = A(v_x \ v_y)^T$. Since $\nabla \cdot (-v_y \ v_x)^T = 0$, we have

$$\nabla \cdot (A(u_x \ u_y)^T) = 0 \quad (4.2)$$

$$\nabla \cdot (A(v_x \ v_y)^T) = 0, \quad (4.3)$$

The above equations play a critical role in our algorithm, since it allows us to reconstruct the T-map f from the Beltrami coefficient $\mu(f)$.

4.3 Computing the Teichmüller Map on Simply/Multiply Connected Domains

As we have mentioned in the previous chapter, extremal quasi-conformal maps minimize the maximal conformality, which is highly desired by many real-world applications when the conformal map does not exist. A Teichmüller map (T-map) is an extremal

quasi-conformal map associated with a holomorphic function. There are two important work on the topic about computing the Teichmüller map on simply/multiply connected domains.

Let D_1 and D_2 be two domains on the complex plane with the same topology. Let $f : D_1 \rightarrow D_2$ be the Teichmüller map with given boundary condition $g : \partial D_1 \rightarrow D_2$.

To compute the extremal quasi-conformal map, Weber et al. [109] minimized the following least-squares Beltrami(LSB) energy

$$\int_{D_1} |f_{\bar{z}} - k \frac{\bar{\phi}}{|\phi|}| dA,$$

where f, ϕ and k are unknowns and ϕ is holomorphic $\phi_{\bar{z}} = 0$ and the boundary map $f|_{\partial D_1} = g$ agrees with a given function g . As the LSB energy is highly non-linear and non-convex, the standard methods (such as the quasi-Newton method) do not behave well [109]. Weber et al. suggested an alternating-descent algorithm, which alternatively minimizes one variable while fixing the other two. Although this approach works well in practice, there is no guarantee of the convergence to the global minimum.

Lui et al. [70] adopted a different strategy, which formulates a variational problem

$$f := \mathbf{argmin}_{f:D_1 \rightarrow D_2} \{ \|\mu(f)\|_{\infty} + \|\nabla |\mu(f)|\|_2^2 \}$$

subject to $f|_{\partial D_1} = g$ and $\mu(f) = k \frac{\bar{\phi}}{|\phi|}$ for some constant k and holomorphic function $\phi : D_1 \rightarrow \mathbb{C}$. The first term minimizes the maximal conformal distortion and the second term minimizes the harmonic energy of $|\mu(f)|$, since the Beltrami coefficient of the Teichmüller map f has a constant magnitude. However, minimizing the above function with respect to the space of diffeomorphisms between D_1 and D_2 is difficult. Therefore, Lui et al. [70] reformulated the energy functional with respect to the space of all Beltrami coefficients

$$(v, f) := \mathbf{argmin}_{v:D_1 \rightarrow \mathbb{C}} \{ \|v\|_{\infty} + \|\nabla |v|\|_2^2 \},$$

where the Beltrami coefficient $\nu = \mu(f) = k \frac{\bar{\phi}}{\phi}$ satisfies $\|\nu\|_\infty < 1$ and $f|_{\partial D_1} = g$ agrees with a given function g on the boundary. Lui et al. [70] proposed the quasi-conformal (QC) iteration to solve the above variational problem. Given an initial map f_0 satisfying the given boundary condition, the QC iteration first computes its Beltrami coefficient $\nu_0 = \mu(f_0)$ for each vertex. Then it updates μ_1 by averaging the Beltrami coefficients of each vertex's 1-ring neighbors. Next, the new map f_1 can be computed from μ_1 by solving a linear system. It repeats these steps until the difference $\|\nu_{i+1} - \nu_i\|$ is less than the user-specified tolerance ε . The QC iteration method [70] guarantees the convergence to the global minimum.

4.4 Computing Quasi-Conformal Map on 3D Surfaces

Given two triangle meshes S_1 and S_2 of the same topology and an initial map $f_0 : S_1 \rightarrow S_2$, our algorithm iteratively computes the Teichmüller map from f_0 . Note the initial map is not necessarily bijective. Our algorithm contains two phases: in the preprocessing phase, it computes the local coordinate charts for each input mesh; in the iteration phase, we compute the map on the local coordinate chart and then diffuse it through the entire mesh. Figure 4.2 illustrates the iterative diffusion procedure on the two-hole torus model. Algorithm 1 presents the pseudo-code of our T-map diffusion technique.

4.4.1 Preprocessing

Many local parameterization techniques can be used for constructing the local coordinate charts on triangle meshes. In this paper, we adopt the exponential map for the chart construction due to its simplicity. Let $\mathbf{p} \in M$ be an arbitrary point on a smooth surface M . The exponential map $\exp_{\mathbf{p}} : T_{\mathbf{p}} \rightarrow M$ at $\mathbf{p}$ is a map from the tangent plane at $\mathbf{p}$ to $\mathbf{p}$'s local neighborhood. Given a radial line on the tangent plane, which is originated at $\mathbf{p}$

Figure 4.2: Algorithmic pipeline. Our algorithm takes two triangular meshes S_1 and S_2 of the same topology, and an initial map $f_0 : S_1 \rightarrow S_2$ as input. Our algorithm does not require a bijective f_0 . For the two-hole torus and the polycube, we simply project S_1 onto S_2 to get the initial map f_0 . Our algorithm iteratively improves the map quality by diffusing the map. Unlike the existing techniques, our algorithm requires neither the global parameterization nor global numerical solver. Thus, it is proceeded in a completely local manner, and has the advantage of scalability.

and has direction $\mathbf{v} \in T_{\mathbf{p}}$, the exponential map sends $\mathbf{p} + t\mathbf{v}$ to a unique geodesic curve γ originating at $\mathbf{p}$ such that $\gamma'(0) = \mathbf{v}$. Conversely, given an arc-length parameterized geodesic γ originating at $\mathbf{p}$, $\gamma(0) = \mathbf{p}$, there is a unique tangent direction $\gamma'(0)$ on the tangent plane $\exp_{\mathbf{p}}^{-1}(\gamma) = \gamma'(0)$.

Exponential map naturally defines a geodesic polar coordinate system on curved surface. The discrete exponential map can be computed efficiently by modifying the existing discrete geodesic algorithms [45, 114]. These geodesic algorithms partition each mesh edge into a set of intervals, called windows, which encode the shortest distance from the source point. The windows are maintained in a priority queue and propagated across the mesh faces: pops a window from the queue and then computes its children windows which can add, modify, or remove existing windows, and updates the queue accordingly. To compute the exponential map on meshes, we slightly modify the ICH algorithm [114] by adding the orientation into the window data structure. The geodesic algorithm terminates when the wavefront touches the boundary of the coordinate chart. Each vertex in the coordinate chart has both the geodesic distance to the source point and the orientation of the geodesic path, which are used to define the polar coordinate system. The modified algorithm has the same time $O(n^2)$ and space $O(n)$ complexity as the original ICH algorithm, where n is the number of vertices in the local chart. Figure 4.3 shows an example of exponential map on the Bunny model. We implement the ICH discrete geodesic algorithm on GPU such that each GPU thread maintain its own windows, and two GPU threads do not have data conflicts even though the coordinate charts are overlapped. As a result, we can compute the local coordinate chart in a fully parallel manner.

For each vertex $\mathbf{v}$ of the input mesh, we denote its l -ring neighbors by $N_l(\mathbf{v})$, where l is the user-specified parameter controlling the size of the chart. We usually set $l = 4$ for

Figure 4.3: Exponential map defines a geodesic polar coordinate system on curved surface.

a well tessellated mesh and apply the discrete exponential map to $N_l(\mathbf{v})$. Our exponential map program terminates when the user-specified l -ring neighbors have been fully covered by the geodesic wavefront. In general, the vertices in $N_l(\mathbf{v})$ form a topological disk. But for a low-resolution model with very thin handles, $\mathbf{v}$'s local neighborhood may contain part or even the entire handle. To handle such a case, our exponential map program checks the following condition: if the geodesic wavefront does not intersect itself during the propagation, the l -ring vertices of $\mathbf{v}$ do form a topological disk. Otherwise, there is some point q , called a cut point, where the geodesic wavefront intersects itself. Intuitively speaking, there are multiple geodesic paths of equal length from the source to q . All cut points form a 1-dimensional graph, which is called cut graph. Cutting $N_l(\mathbf{v})$ along the cut graph makes it a topological disk.

Remark The exponential map has no guarantee of the bijectivity due to two reasons: first, the globally shortest geodesic path between two points may not be unique, especially when the two points are far from each other. Second, when passing through a saddle vertex (with a total angle more than 2π), the inbound geodesic path may split into two or more outbound paths. In spite of the theoretical defects of the discrete exponential map, we have observed that computing the exponential map is highly efficient

Figure 4.4: Computing the local map on the coordinate chart.

and its induced parametrization has good quality for a small neighborhood (e.g., the 4-ring neighbors as in our implementation). In case the bijectivity does not hold for a coordinate chart, we re-parameterize that chart by using the Ricci flow method [75], which is more computationally expensive than the exponential map but can guarantee that the local parametrization is a diffeomorphism.

4.4.2 Iteratively Improving the Map Quality

With the exponential map for each vertex on the source and target shapes, we can form the coordinate chart. Initially, we set the Beltrami coefficient to zero, then our method iteratively computes the quasi-conformal map on the local coordinate chart and diffuses the map and Beltrami coefficient through the entire mesh. Our diffusion method involves only the local computation, which is completely linear and does not require any numerical solver.

To compute f from the current Beltrami coefficient μ , we extend the linear Beltrami

solver [70] from the complex plane $\mathbb{C}$ to manifold. As shown in Figure 4.4, given a vertex $\mathbf{v}$ on S_1 , the exponential map at $\mathbf{v}$ defines a coordinate chart for $\{U(\mathbf{v}), \phi\}$, where $U(\mathbf{v})$ is the l_1 -ring neighborhood of $\mathbf{v}$ and ϕ sends $U(\mathbf{v})$ to the complex plane $\mathbb{C}$. Let $\phi(\mathbf{v}) = \mathbf{v}' = x + \sqrt{-1}y$.

The image $f(\mathbf{v})$ is a point on the target surface S_2 . Usually, $f(\mathbf{v})$ is not a vertex of S_2 . Let $T \in S_2$ be the triangle containing $f(\mathbf{v})$. We choose one of T 's vertices to form the coordinate chart for $\{V(f(\mathbf{v})), \varphi\}$. Then the composite map $g : \mathbb{C} \rightarrow \mathbb{C}$, $g = \varphi \cdot f \cdot \phi^{-1}$ sends $\mathbf{v}'$ to $g(\mathbf{v}') = u + \sqrt{-1}v$, i.e.,

$$g : (x, y) \rightarrow (u, v) .$$

The discrete partial derivatives u_x, u_y, v_x, v_y and the discrete divergence $\nabla \cdot$ are well defined on triangle meshes. Discretizing Eq. (4.2) and Eq. (4.3) leads to the following equation,

$$\begin{aligned} \Sigma_{\triangle \in N(\mathbf{v})} \Sigma_{k=1}^3 g(\mathbf{v}'_k) [(C_k^\triangle a_{11}^\triangle + D_k^\triangle a_{12}^\triangle) A_k^\triangle + \\ (C_k^\triangle a_{21}^\triangle + D_k^\triangle a_{22}^\triangle) B_k^\triangle] = 0, \end{aligned} \quad (4.4)$$

where $N(\mathbf{v})$ is the 1-ring neighbor of $\mathbf{v}$ and the coefficients $A_k^\triangle$, $B_k^\triangle$, $C_k^\triangle$, and $D_k^\triangle$ are defined for the triangle $\triangle$ which is one of $\mathbf{v}$'s one-ring neighbors.

$$\begin{aligned} A_k^\triangle &= \frac{y_{k+1} - y_{k+2}}{2\text{Area}(\triangle)} \\ B_k^\triangle &= \frac{x_{k+1} - x_{k+2}}{2\text{Area}(\triangle)} \\ C_k^\triangle &= \cot \theta_{k+1}(x_{k+2} - x_k) + \cot \theta_{i+2}(x_{k+1} - x_k) \\ D_k^\triangle &= \cot \theta_{k+1}(y_{k+2} - y_k) + \cot \theta_{i+2}(y_{k+1} - y_k), \end{aligned}$$

Note (x_i, y_i) are the local coordinates of $\mathbf{v}_i$. See the figure 4.5 for the notations used in the above equation.

Figure 4.5: The label in one triangle around the vertex $\mathbf{v}$.

For each vertex $\mathbf{v}_i$ on S_1 , first we use the local chart to get the map g , and updated the g on the local chart by Equ. (4.4); then we transfer the updated $g(\mathbf{v}_i)$ back the surface S_2 by the function $\exp^{-1}(g(\mathbf{v}_i))$; the map $f(\mathbf{v}_i)$ will be updated. These iterations will diffuse the map $f : S_1 \rightarrow S_2$ to the quasi-conformal map with provided Beltrami coefficient.

4.5 Experimental Results

We implemented our Beltrami coefficient diffusion algorithm in C++ and the exponential map algorithm on CUDA 4.2. We run our program on a 64-bit Workstation with an Intel 3.50GHz CPU (4 cores) and 16GB memory. The graphics card is an NVIDIA GTX 580 with 512 cores and 1.5GB memory. We tested it on both synthetic and real-world models of various topological types, where genus ranges from 0 to 8. Table 1 shows the mesh complexity and execution time of our program. For each test model, we map it to a target surface of the same topology. The target surface is usually chosen either as the standard domain, such as unit square, torus, or the polycubes. The target meshes have similar mesh complexity as the source meshes.

Our algorithm iteratively improves the map quality until the change of map is within the user-specified threshold ε . The convergence threshold ε is set to 10^{-5} for all test models

Input: S_1 and S_2 two triangle meshes of the same topology; an initial map $f_0 : S_1 \rightarrow S_2$; l_1 and l_2 the sizes of local coordinate charts on S_1 and S_2 ; a convergence tolerance ε .

Output: The T-map $f : S_1 \rightarrow S_2$

// Preprocessing
 For each vertex $\mathbf{v}$ on S_1 (resp. S_2), compute an exponential map at $\mathbf{v}$ covering $\mathbf{v}$'s l_1 (resp. l_2) ring neighbors.
 // Initialize the Beltrami coefficient
 For each triangle $\triangle \in S_1$, set $\mu(\triangle) = 0$.

$i \leftarrow 0$; // Iteratively compute the map f by diffusion
do
 for each vertex $\mathbf{v}_j \in S_1$
 form the coordinate chart $\{N_{l_1}(\mathbf{v}_j), \exp^{-1}(\mathbf{v}_j)\}$;
 form the coordinate chart $\{N_{l_2}(f(\mathbf{v}_j)), \exp^{-1}(f(\mathbf{v}_j))\}$;
 compute the map $g_i(\mathbf{v}_j)$ in Fig 4.4.
 compute $g_{i+1}(\mathbf{v}_j)$ by Equ. (4.4).
 compute $f_{i+1}(\mathbf{v}_j)$ by $\exp^{-1}(f(\mathbf{v}_j))$ and $g_{i+1}(\mathbf{v}_j)$.
 compute the $\mu(\triangle)$ and diffuse it if necessary.
endfor
 $i \leftarrow i + 1$;
while $\max_j \|f_i(\mathbf{v}_j) - f_{i+1}(\mathbf{v}_j)\| \geq \varepsilon$

Algorithm 3: The Local Method to Compute T-map on 3D Surfaces.

Figure 4.6: Our Beltrami coefficient diffusion algorithm converges quickly. The horizontal axis shows the iteration number and the vertical axis shows the maximal change of the vertex position between two consecutive iterations.

except the Bunny, where $\varepsilon = 10^{-9}$. We notice that the number of iterations closely depends on the initialization. Two types of initial maps are used in our experiments. We adopt polycube maps [111] as the initial map for the Four Kids, Greek and Fertility models. As the input map is of good quality, it takes our algorithm 20 to 40 iterations to get a high quality T-map (see Figure 4.5). For the Head, Bunny, Two-hole Torus, Decocube and Botijo models, we simply project the source surface to the target. As the quality of initial map is pretty poor, more iterations are usually required to obtain a high quality map. Furthermore, the geometry of the source/target shape is another key factor of the total number of iterations. For example, it takes our algorithm 1500 iterations to obtain the Bunny-to-Pyramid T-map (see Figure 4.8). While the map for the body and the head become fairly good within 100 iterations, the ears need many more iterations to converge due to significant difference between a long tube and a small disk. However, as the computational cost in each iteration is very small, the overall performance of our algorithm is still good.

Figure 4.7: T-map from the Head model to the unit square.

Figure 4.7 shows the quasi-conformal map from the genus-0 Head model (with one boundary) to the unit square. We choose four points on the boundary of the source mesh and map them to the four corners of the square. The other boundary vertices are mapped to the boundary of the square by using arc-length parameterization. Then we project the interior vertices of the source mesh onto the unit square to obtain the initial map. Obviously, the initial map is not bijective. Note that two topological disks are conformally equivalent, which is also confirmed by our result. Our Beltrami coefficient diffusion algorithm converges in 30 iterations. As shown in the circle texture map, the circles from the texture domain are mapped to circles. Furthermore, the majority of the $k = |\mu|$ values are 0, which implies the conformal distortion is really small and again confirms the conformality of the map.

Figures 4.1, 4.10 and 4.9 show the T-map of models with genus $g \geq 1$. As each pair of the source and the target are *not* conformally equivalent, the Beltrami coefficients $|\mu|$ of the computed map vary.

Figure 4.8: T-map from the Bunny to a pyramid shape. We project the Bunny to the pyramid to initialize the map f_0 . Clearly, the initial map is not bijective, due to large overlapped/self-intersect regions. Our method iteratively improves the map quality by diffusing it through the entire mesh. The quality of the body and the head are acceptable after 15 iterations and 100 iterations. Due to the significant geometric difference between the ear and the domain, it does take many iterations to get a bijectively map for the ear.

Model	genus	#vertices	Iteration times	Preprocessing time (GPU)	Diffusion time (CPU)
Bunny	0	14K	1500	12.7s	175.8s
Head	0	25K	195	22.9s	15.3s
Kitten	1	10K	212	9.7s	8.4s
2-hole torus	2	25K	287	27.9s	19.3s
Fertility	4	20K	28	18.7s	7.3s
Greek	4	25K	32	26.3s	8.2s
Botijo	5	29K	298	31.4s	21.7s
Decocube	5	20K	177	20.7s	13.7s
4 Kids	8	34K	39	37.8s	9.8s

Table 4.1: Mesh complexity and performance. The preprocessing and diffusion times are measured on a GPU with 512 cores and a CPU with 4 cores respectively.

Figure 4.9: T-map from the genus-1 Kitten to a torus. These two shapes are not conformally equivalent. The coefficient k ranges from 0.2 to 0.5.

4.6 Comparisons & Discussions

Table 4.1 compares our method to the existing quasi-conformal methods. The techniques in [109] [70] require the global coordinate system, thus, they are limited to models with simple topology. Weber et al.'s method [109] is non-linear since it solves a non-linear least square, which couples the Beltrami coefficient $k_{\frac{\bar{\varphi}}{|\varphi|}}$ and the map. Lui et al. [70] proposed the linear Beltrami solver which decouples the Beltrami coefficient and the map such that one can reconstruct the T-map with fixed Beltrami coefficients. They form a sparse linear system to compute the T-map. Our method extends the linear approach [70] to manifold surface of non-trivial topology. We apply the linear Beltrami solver on each local coordinate chart instead of forming a global linear system. The decomposition is very natural from the manifold's perspective and it indeed brings significant benefits, such as the ease of implementation, parallel structure, scalability, free of global coordinate system. These features distinguish our method from the global algorithm [70].

Figure 4.10: T-map of high genus models. The circle texture is shown on the source surface S_1 and the insets show the target surface S_2 and the histogram of the Beltrami coefficient $|\mu|$.

Method	Topology	Coordinate	Linear	Numerical solver	Parallel	Performance
Zeng et al. [123]	Simply/multiply connected domains	Global	No	Linear system	No	Fair
Li et al. [65]	Arbitrary genus	Global	Yes	No	No	Poor
Zeng et al. [122]	Arbitrary genus	Local	No	Linear system	No	Poor
Webber et al. [109]	Simply/multiply connected domains	Global	No	Non-linear optimization	No	Fair
Lui et al. [70]	Genus ≤ 1	Global	Yes	Linear system	No	Good
Our method	Arbitrary genus	Local	Yes	No	Yes	Good

Table 4.2: Comparison to other quasi-conformal methods.

Chapter 5

Registration of Brainstem Surfaces in Adolescent Idiopathic Scoliosis using Discrete Ricci Flow

We present a surface registration method for brainstem by using the Ricci flow and extremal quasi-conformal map in this chapter, which is important for the study of Adolescent idiopathic scoliosis (AIS).

5.1 Introduction

Adolescent idiopathic scoliosis (AIS) is a three-dimensional structural deformity of the spine that occurs during adolescence, and the prevalence is about 4% of the adolescent population worldwide. Due to the lack of generally accepted scientific theory on the etiology of AIS, the treatment and prognosis of AIS are limited [78]. Many experiments and evidences point toward the nervous regulation of the postural balance in the idiopathic scoliosis patients. The brainstem and vestibular system are two key organs in the balance control system. Significant anomalies of the balance function, proprioception and oculomotor reflexes have been reported [93], and morphological difference has been found in the vestibular system [121]. However, there is no reported work on

studying the morphological alterations in brain stems in the AIS subjects in an objective and quantitative way. Motivated by this, we are interested in developing mathematical models to facilitate the morphometry of brainstem surfaces in AIS and normal controls.

In order to perform shape analysis effectively, meaningful one-to-one correspondences between different brainstems must be obtained. Such a process is called *surface registration*. Landmark-matching based registration approaches have been commonly applied, in which landmark features were required to be consistently matched to guide the registration. Landmark-matching registration has shown to be effective in obtaining accurate point-wise correspondences between 3D medical data. On brainstem surfaces, there are no medical features defined by neuroscientists that can be used as a constraint to establish good correspondences. Therefore, in order to register brainstems effectively, meaningful landmark features that describe the global geometry of the brainstem surfaces must firstly be extracted. On brainstem surfaces, four consistent feature curves can be observed, which are basically ridges and valleys with high curvatures. It motivates us to extract out these features to guide the geometric brainstem registrations. With these obtained features, landmark-matching registrations can be computed to obtain an accurate point registration that matches geometry as accurate as possible.

In this section, we introduce an algorithm to automatically register brainstem surfaces based on the discrete Ricci flow method. Four consistent features are firstly extracted to guide the surface registration. These features effectively describe the global geometry of the brainstem surfaces. Using the discrete Ricci Flow method, brainstem surfaces are parameterized conformally onto the quadrilaterally-faced hexahedron, of which the extracted feature landmarks are mapped to the edges. Quadrilaterally-faced hexahedron is chosen as the parameter domain since the geometry of the brainstem surface is similar to a hexahedron. With that, the distortion under the parameterization can be minimized

so that registration can be obtained accurately. Surface registrations between brainstems can then be obtained through the parameterization, which consistently match the feature landmarks. With the obtained registration, a shape energy can be defined to measure the local shape difference between different brainstem surfaces, which is useful to study shape variation between brainstems for the purpose of disease analysis. We tested our algorithm on real brainstem surfaces extracted from MRIs of 15 normal subjects and 15 AIS patients. Experimental results show the efficacy of the proposed algorithms to register and detect the local shape difference between different brainstem surfaces.

Our contributions are two-folded: first, we propose to delineate four salient features on brainstems that describe the global geometry; second, by mapping the four detected landmarks to the four side edges of a hexahedron, we propose an efficient algorithm to conformally parameterize the brainstem surfaces, which naturally induces the registration among brainstem surfaces. Our registration algorithm can guarantee the exact landmark feature correspondence.

5.1.1 Related Work on Surface Registration

Surface registration, which aims to find a meaningful 1-1 correspondence between different surfaces, has been studied extensively by different groups. Conformal surface registration is commonly used [32][43], which gives a parameterization minimizing the angular distortions. An advantage of this approach is that they preserve local geometry very well. Conformal structure can also measure non-isotropic deformation effectively. However, conformal registrations generally cannot map landmark features, such as sulcal landmarks on brain surfaces, consistently. Landmark-based diffeomorphisms are often used to compute, or adjust, cortical surface parameterizations [44, 59]. For example, Glaunes et al. [44] proposed to generate large deformation diffeomorphisms of the

Figure 5.1: Landmark extraction. (A) and (B) show the computed ridge and valley lines respectively. The two strongest ridge and valley lines are chosen as landmarks, as shown in (C). (D) shows the four consistent features extracted on brainstems of a normal and an AIS subject.

sphere onto itself, given the displacements of a finite set of template landmarks. Leow et al. [59] proposed a level-set based approach to match different types of features, including points and 2D or 3D curves represented as implicit functions. These methods provide good registrations when the corresponding landmark points on the surfaces can be labeled in advance. On surfaces without well-defined landmarks, some authors have proposed driving features into correspondence based on shape information. Lyttelton et al. [80] computed surface parameterizations that match surface curvature. Fischl et al. [4] improved the alignment of cortical folding patterns by minimizing the mean squared difference between the average convexity across a set of subjects and that of the individual. Lord et al. [79] matched surfaces by minimizing the deviation from isometry.

5.2 Algorithm

5.3 Overview

Given a brainstem surface M , we first detect four salient geometric features on M . We then apply the discrete Ricci flow method to parameterize the brainstem surfaces onto

a quadrilaterally-faced hexahedron, through which the four feature curves on the brainstem are mapped to the four vertical sides of the hexahedron. As our parameterization method naturally segments the brainstem surfaces into six patches, we can easily register two different brainstems by finding a diffeomorphism between each corresponding patches. In the following, we explain each step in details.

5.3.1 Landmark Extraction

On brainstem surfaces, four consistent features can be extracted, which are essentially curves with high surface curvatures. We extract these features by computing the ridge-valley lines [6], an effective shape descriptor on the surface along which the surface bends sharply. Figure 5.1(A) and (B) show the computed ridge-valley lines on a brainstem surface. After obtaining the ridge-valley lines, we smooth them using a Gaussian filter (with kernel size 2% of the main diagonal of the object's bounding box) to reduce the artifacts caused by the noise and/or poor triangulation. The detected ridges or valleys are merged, if their endpoints are close and their tangent directions are within the user-specified threshold. In our implementation, we set the distance and direction thresholds to 2% of the object's main diagonal and 15 degree respectively. Next, we measure the strength of each feature by computing the length of the detected ridge-valley line. The longer the curve, the stronger the feature. We then find the two strongest ridges and valleys, which are chosen as landmark features. On brainstem surfaces, there are consistent umbilic points and high-curvature points. These points are chosen as the endpoints of the landmark features. Figure 5.1(C) shows the strongest ridge and valley lines extracted.

Remark: Among the four landmarks, we observed the two feature lines in the valley γ_2 and γ_4 are quite robust and consistent, since their end points fall in regions with highly

negative mean curvatures. For the other two ridges γ_1 and γ_3 , their locations are consistent, but their end points may differ among the subjects. To solve this issue, we specify the length of the ridges so that the tracing stops immediately when the landmark exceeds the threshold. We also allow the users to specify the end points manually and do some manual correction when the feature line detected is not correct. Figure 5.1(D) shows the four consistent features extracted on brainstems of a normal and an AIS subject.

Parameterization using discrete Ricci flow. To ease the computation process, we parameterize the brainstem onto a domain in $\mathbb{R}^2$. The ideal parameter domain for such surface is the quadrilaterally-faced hexahedron, of which each landmark is mapped to an edge of the hexahedron. Discrete Ricci flow method is used to obtain the parameterization.

Given a brainstem surface, we first cut it open along the detected feature lines. As a result, the open brainstem surface is of genus 0 with 4 boundaries, denoted by γ_i , $i = 1, \dots, 4$. We then set the target curvatures to be $-\pi/2$ at the end points of the landmarks and zero elsewhere. Using the discrete Ricci flow method [74], we can parameterize the brainstem surfaces onto the universal covering space embedded in $\mathbb{R}^2$.

To visualize the parameterization, we cut the brainstem surface along the geodesics (i.e. line segments) $e_1^1 e_2^1$, $e_2^1 e_3^1$, and $e_3^1 e_4^1$ (see Figure 5.2(C)), under the new metric $\bar{g}$. Intuitively speaking, we cut along the edges of the bottom face to make the hexahedron open. Now the brainstem surface becomes a genus-0 surface with only one boundary.

We compute the layout of the parameterized mesh by isometrically embedding all the triangles with the new metric $\bar{g}$. Figure 5.2 shows the universal covering space of the brainstem surface with the highlighted fundamental domain. One can see that each landmark is mapped to the straight side of the hexahedron. This feature allows us to

Figure 5.2: The cut surface can be conformally embedded into the universal covering as shown in (A). (B) shows the fundamental domain (one period) of the universal covering. The parameterization naturally induces a segmentation of the brainstem surfaces into six patches as shown in (C). In (D), the checkerboard texture on the parameter domain is mapped to the brainstem surface, which map shows the parameterization is indeed conformal.

easily compute the high quality registration with guaranteed exact landmark matching, which will be described in the next section.

5.3.2 Registration

The above discrete Ricci flow method parameterizes each brainstem surface to a hexahedron, such that the four landmarks are mapped to the four vertical edges. The hexahedron naturally induces a segmentation of 6 patches. With these, we can easily register

Figure 5.3: (A) Registration between two normal brainstems. (B) Registration between the brainstems of a normal and AIS subject. Landmark features are matched exactly. (C) shows the average shape of 15 AIS brainstems using our registration. The sharp features are well-preserved after averaging, meaning that our registration matches salient features well. (D) shows the average shape using the curvature-based spherical demons registration without landmarks [117]. Note that the sharp features are smoothed out.

two brainstem surfaces in a piecewise manner.

Let F_1^i (resp. F_2^i), $i = 1, \dots, 6$, denote the six patches of M_1 (resp. M_2) induced by the parameterization. We want to find a diffeomorphism (i.e., a C^∞ -smooth and bijective map) $\phi_i : F_1^i \rightarrow F_2^i$ between each pair of patches F_1^i and F_2^i . Let f_1 (resp. f_2) denote the Ricci flow parameterization that maps F_1^i (resp. F_2^i) to a quadrilateral D_1^i (resp. D_2^i). Let us also denote the corners of the 3D patch and 2D quadrilateral by p^j and q^j , $j = 1, \dots, 4$, respectively. Then we map each quadrilateral Q to a unit disc D by the extremal quasi-conformal map $g : Q \rightarrow D$ with the boundary condition $g(\partial Q) = \partial D =$

$\mathbb{S}^1$. Let $r^j = f(q^j)$, $j = 1, \dots, 4$, denote the images of the corners on the unit circle. Next, we compute another extremal quasi-conformal map $h : D_1^i \rightarrow D_2^i$ between the two unit discs D_1^i and D_2^i . We set the boundary condition $h(\partial D_1^i) = \partial D_2^i$ by enforcing the exact feature correspondence, i.e., the corners r_1^j are mapped to corners r_2^j . Therefore, a map ϕ_i between the patches F_1^i and F_2^i can be obtained by the composite map $\phi_i = f_2^{-1} \circ g_2^{-1} \circ h \circ g_1 \circ f_1$ (See Figure 5.4).

A homeomorphism $\phi : M_1 \rightarrow M_2$ can then be obtained by $\phi = \bigcup_{i=1}^6 \phi_i$. Note that landmarks are exactly matched for each patches. So, our algorithm guarantees the exact correspondence between the landmark features.

Proposition. The proposed brainstem surface registration algorithm produces a homeomorphism ϕ between M_1 and M_2 , which guarantees the exact correspondence between the landmarks $\phi(\gamma_1^j) = \gamma_2^j$, $j = 1, \dots, 4$.

Proof. We first show that the resulting map ϕ is a homeomorphism. Observe that the discrete Ricci flow induced parameterization f_1 and f_2 are diffeomorphism [13]. And the diffusion process in the extremal quasi-conformal map is quite similar to the heat diffusion for the harmonic function, and according to the classical result [87], the result will be diffeomorphism when the target domain is convex and the boundary map is homemorphism. As Q_j and D_j , $j = 1, 2$, are convex and the boundary conditions in g_j and h are homeomorphic, all the extremal quasi-conformal map g_1 , g_2 and h are diffeomorphic. Therefore, the map $\phi_i = f_2^{-1} \circ g_2^{-1} \circ h \circ g_1 \circ f_1$ is a diffeomorphism (see Figure 5.4).

Although the maps ϕ_i 's are calculated individually, they can be glued seamlessly. We use the arc-length parameterized boundary condition in computing the harmonic map from each patch to the unit disk. Assume two patches F_a and F_b share a boundary γ , and

$p \in \gamma$ is an arbitrary point on γ . When we compute the map g from F_a to the unit disk, we set the boundary condition so that p 's image is given by the arc-length parameterization. Similarly, the map from F_b to the unit disk also sends p to exactly the same location on the circle. With this, our method can guarantee the boundaries of two adjacent patches are mapped consistently and thus all patches are glued seamlessly in a C^0 manner. Thus, the global map, $\phi = \bigcup_i \phi_i$, is in fact a homeomorphism.

Second, we show that the map ϕ preserves the correspondence between the landmarks. By setting the prescribed geodesic curvature on γ_1^j and γ_2^j to zero, $j = 1, \dots, 4$, the metric $g(\infty)$ obtained by discrete Ricci flow can guarantee each landmark is mapped to a *line segment* (see Figure 5.2(b)). Then with the extremal quasi-conformal map, g_1 and g_2 , each line segment is mapped to an arc of the unit circle. Note that the exact feature correspondence are set when computing the harmonic function $h : D_1 \rightarrow D_2$ between two unit disks. Therefore, the composite function ϕ_i maps the boundary ∂F_1 homeomorphically to ∂F_2 and also sends each corner of F_1 to the corresponding corner of F_2 . Thus, the map ϕ_i sends the landmark of F_1 to the corresponding landmark of F_2 . As a result, the map ϕ preserves the exact correspondence between the landmarks $\phi(\gamma_1^j) = \gamma_2^j$, $j = 1, \dots, 4$.

Putting it together, the map $\phi : M_1 \rightarrow M_2$ is a homeomorphism with guaranteed landmark correspondence. $\square$

5.3.3 Shape variation detection

With the obtained registration between brainstems, we can define a shape energy which detect local shape variations. According to Riemannian geometry theories, the local geometry of a Riemann surface can be described by its mean curvature and Gaussian curvature. Let B_1 and B_2 be two brainstem surfaces. Suppose $f : B_1 \rightarrow B_2$ is

Figure 5.4: Computing the map $\phi_i : F_1^i \rightarrow F_2^i$.

the registration between B_1 and B_2 . We can define a shape energy E_{shape} as follows:
 $E_{shape}(f) = \alpha \int_{B_1} |H_1 - H_2(f)| + \beta \int_{B_1} |K_1 - K_2(f)|$, where H_1 and K_1 (resp. H_2 and K_2) are the mean curvature and Gaussian curvature of B_1 (resp. B_2). $E_{shape}(f) = 0$ if and only if B_1 and B_2 are equal up to a rigid motion. Therefore, E_{shape} measures the local shape difference between B_1 and B_2 effectively. This allows us to examine the

Input: Two brainstem surfaces M_1 and M_2

Output: The homeomorphism $\phi : M_1 \rightarrow M_2$ with guaranteed landmark correspondence

- 1 1. Extract the four landmarks from the brainstem surfaces;
- 2 2. Cut the brainstem surfaces open along the detected landmarks;
- 3 3. Set the prescribed Gaussian and geodesic curvatures to each vertex and run Ricci flow;
- 4 4. With the resulting metric, embed the brainstem surfaces into quadrilaterally-faced hexahedral, which naturally induces a segmentation of M_1 and M_2 into six patches.
- 5 5. For each pair of patches $F_1^i \in M_1$ and $F_2^i \in M_2$, find the bijective map $\phi_i : F_1^i \rightarrow F_2^i$ by computing extremal quasi-conformal with exact landmark match at boundary;
- 6 6. Output the map $\phi = \bigcup_{i=1}^6 \phi_i$.

Algorithm 4: Brainstem surfaces registration with guaranteed landmark correspondence.

Figure 5.5: Up: Grid texture is drawn on the control brainstem. It is mapped to three different brainstems of the AIS subjects using our proposed registration algorithm. The grid pattern illustrates that our registration result is indeed a homeomorphism. Down: Detection of shape variations.

region of significant shape difference for the morphometry of the brainstems.

5.4 Experimental Results

Subject and data acquisition: We tested our proposed algorithms on 30 brainstem surfaces, which are extracted from the MRIs. The MRI brain data were acquired from 15 AIS subjects and 15 age-matched normal controls using a 1.5T MRI scanner (Sonata, Siemens, Erlanger, Germany) using a quadrature head-coil. The segmentation of the brain stems was achieved automatically. Two experienced operators verified the seg-

mentation results.

Brainstem registration: Using our proposed algorithms, we compute the registration between 30 brainstems of 15 normal subjects and 15 AIS patients. Experimental results show that our proposed method can compute the registration effectively with exact landmark-matching. Figure 5.3(A) shows the registration results between two normal brainstems. The registration result between the brainstems of a normal and AIS subject is shown in (B). Landmark features are exactly matched. In Figure 5.5(A)-(D), grid texture is drawn on the control brainstem. It is mapped to three different brainstems of the AIS subjects using our proposed registration algorithm. Note that grid patterns have no overlapping, indicating the registration is indeed a homeomorphism. The whole registration procedure can be computed efficiently. The feature landmarks (ridge and valley lines) are computed in real-time for brainstem meshes with 20K vertices. The Ricci flow based conformal parameterization for each brainstem mesh takes about 12 seconds, since the Newton's method with second order convergence rate is adopted in our implementation. The registration between brainstems through the parameterization takes about 5 seconds to compute. The whole process to obtain the landmark-matching brainstem registration takes less than 20 seconds. Hence, our proposed registration algorithm is quite efficient.

Detection of shape variations: With the obtained registration f , the local shape difference between different brainstem surface can be detected from the shape energy $E_{shape}(f)$. Figure 5.5(E)-(H) shows the local shape differences of two brainstem (1 normal and 1 AIS subject) surfaces from the control. The color map is given by the shape energy, which measures their local shape difference. The red (resp. blue) color indicates a higher (resp. lower) degree of shape difference. The local shape difference of the AIS

brainstem from the control tends to be more obvious than that of the normal brainstem. Figure 5.5(H) shows the statistical significance p-map measuring the local shape difference from the control between the normal ($n = 15$) and AIS ($n = 15$) groups, plotted on a control brainstem. The deep red color highlights regions of significant statistical difference. This method can be potentially used to study factors that influence shape changes of brainstems in AIS.

Comparison: We also compare our registration algorithm with the curvature-based spherical demons registration without landmarks [117]. The curvature-based registration without landmarks generally cannot match salient features. Figure 5.3(C) shows the average shape of 15 AIS brainstems using our registration. The sharp features are well-preserved after averaging, meaning that our registration matches salient features well. (D) shows the average shape using the curvature-based registration without landmarks. Note that the sharp features are smoothed out. It indicates the mismatching of salient features.

Chapter 6

Automatic Registration of Vestibular Systems with Exact Landmark Correspondence

Another important system related to the AIS is the vestibular system. The non-trivial topology of the VS poses great technical challenges to the geometric analysis. In this chapter we present an effective and practical solution to register the vestibular systems by using the holomorphic 1-form and extremal quasi-conformal map techniques. Our method is robust, automatic and efficient.

6.1 Motivation

The vestibular system (VS) is the sensory system situated in the inner ear, which contributes to balance and the sense of spatial orientation. The VS is a genus-3 structure with three semicircular canals and the morphometry of VS plays an important role in the analysis of various diseases such as the Adolescent Idiopathic Scoliosis (AIS) disease. One popular hypothesis was suggested to be the structural changes in the VS that induce the disturbed balance perception, and further cause the spinal deformity [12, 103]. Some recent works have revealed the statistical difference in global morphology of the

VS between right-thoracic AIS and normal controls [121, 113]. In order to perform shape analysis effectively, meaningful one-to-one correspondences between different VSs must be obtained. Landmark-matching based registration techniques, in which landmark features were required to be consistently matched to guide the registration, have proven to be effective in obtaining accurate point-wise correspondences between 3D medical data. However, the non-trivial topology of the VS poses great challenge for the landmark based registration.

This chapter presents a novel technique to register the vestibular systems. The contributions are two-folded:

- First, we present a robust algorithm to extract the salient geodesic landmark features from the vestibular systems. Thanks to its intrinsic nature, the geodesic landmarks are totally determined by the metric and insensitive to the VS's resolution and tessellation. Furthermore, they are highly consistent regardless of the AIS and healthy groups.
- Second, we present an algorithm to register the vestibular systems with exact landmark correspondence. We first partition each VS into three patches, a topological annulus and two topological disks, using the extracted landmark features. Then for each pair of patches of the AIS subject and the control, we compute a bijective map using the holomorphic 1-form and extremal quasi-conformal map techniques. With a carefully designed boundary condition, the three individual maps can be glued in a seamless manner. The resulting registration is guaranteed to be a homeomorphism with exact landmark correspondence.

To our knowledge, this is the first work to address the landmark based registration of the vestibular systems. Our method is robust, fully automatic and highly efficient, which takes only a few seconds on a regular PC. We have tested our algorithms on 13 normal

subjects and 15 AIS patients. The promising experimental results demonstrate the efficacy of our method to register the vestibular systems, which matches landmark features consistently.

6.2 Algorithm

Our algorithm contains two stages, the first stage is to detect the landmarks using stable geodesic loops and the second one is to parameterize the VS surface using holomorphic 1-form and extremal quasi-conformal map. Our registration method can guarantee the bijection and the exact correspondence between the landmarks.

Figure 6.1: Computing the geodesic homology basis on the 3-torus model.

6.2.1 Landmark extraction

Let us denote S the boundary surface of the vestibular system, which is a closed oriented 3D surface of genus $g = 3$. S is topologically equivalent to a sphere with three handles, which are horizontal canal, anterior canal, and posterior canal, denoted by γ_1 , γ_2 and γ_3 , respectively. The horizontal canal γ_1 , is roughly orthogonal to the other two regardless of the healthy/AIS groups, see Figure 6.2.

It is well-known that one can cut a genus g surface into a topological disk by $2g$ loops, which form the basis of the homology group. Dey et al. [18] defined two classes of

Figure 6.2: The vestibular system is a genus-3 structure (colored in brown) in the inner ear. Its homology basis $\{a_i, b_i\}_{i=1}^3$ of the genus-3 vestibular system contains 6 loops. Among them, the three geodesic *tunnel* loops a_i are highly consistent regardless of the AIS/normal subjects. The locations of the geodesic *handle* loops b_i , however, may vary significantly.

loops, called tunnel and handle loops in terms of a homology group. A loop is a *tunnel* (resp. *handle*) if it spans a surface, say D , in the *unbounded* (resp. *bounded*) space bordered by S and does not do so in S . Intuitively speaking, if one cuts S along a tunnel loop and fills the boundaries with two copies of D , one eliminates a handle. Similarly, removing a handle loop eliminates a tunnel.

Xin et al. [113] observed that the geodesic tunnels on the VS surface can be used to study the geometric differences between the healthy and AIS groups. In this section, we adopt Xin et al.'s approach [113] to compute the three geodesic tunnels a_1 , a_2 and a_3 (see the red curves in Figure 6.2). As a byproduct, Xin et al.'s algorithm also produces the dual *geodesic* loops b_i , $i = 1, 2, 3$. Together $\{a_i, b_i\}_{i=1}^3$ form the homology basis of the genus-3 VS, such that the intersection numbers of the paths are $a_i \cdot a_j = b_i \cdot b_j = 0$ and $a_i \cdot b_j = \delta_{ij}$.

It is worth noting that the geodesic tunnel loops a_i , $i = 1, 2, 3$, are highly stable, regardless of the AIS and normal groups. However, the locations of the dual loops b_i s may vary significantly, see Figure 6.2. To obtain consistent landmark features, we adopt the

following strategy: for each pair of handles, say γ_i and γ_j , we find a pair of points $p_i \in \gamma_i$ and $p_j \in \gamma_j$. Then we use Dijkstra's algorithm to find a path l_{ij} connecting p_i and p_j . The path $c_{ij} \triangleq b_i l_{ij} b_j l_{ij}^{-1}$ is homologous to $b_i + b_j$. Next, we apply Xin et al.'s curve shortening algorithm [112] to deform c_{ij} into a geodesic loop, see Figure 6.3. As Xin et al.'s algorithm keeps the curve's topology during the length shortening process, c_{ij} is homologous to $b_i + b_j$. Furthermore, as geodesic is intrinsic and geometry aware, the computed loop c_{ij} is insensitive to the initial curve $b_i l_{ij} b_j l_{ij}^{-1}$. We observe the geodesic loops a_i , c_{ij} are highly consistent among AIS/normal subjects, thus, can be used as the landmark features (see Figure 6.4).

Figure 6.3: Computing the stable geodesic loop c_{ij} across two handles γ_i and γ_j . From left to right: the initial curve $b_i l_{ij} b_j l_{ij}^{-1}$ is continuously deformed into a stable geodesic loop c_{ij} . Note that the final geodesic loop is homologous to the initial curve, since the curve shortening method [112] keeps the curve's topology during the deformation.

Input: A vestibular system (VS) surface S

Output: Six geodesic landmark features

- 1 Compute the geodesic homology basis $\{a_i, b_i\}$ for each handle γ_i , $i = 1, 2, 3$.
- 2 For any two handles γ_i and γ_j ,
- 3 Choose two arbitrary points $p_i \in b_i$ and $p_j \in b_j$.
- 4 Find a path l_{ij} connecting b_i and b_j using Dijkstra's shortest path algorithm.
- 5 Deform $b_i l_{ij} b_j l_{ij}^{-1}$ into a geodesic loop c_{ij} using [112].
- 6 End For
- 7 Output $\{a_1, a_2, a_3, c_{12}, c_{23}, c_{31}\}$.

Algorithm 5: Extracting the six geodesic landmarks from a VS surface.

The extracted geodesic landmarks naturally induce a surface segmentation: cutting S along the geodesic loops c_{12} , c_{23} and c_{31} produces a genus-1 surface with three boundaries and a genus-0 surface with three boundaries. Further cutting the two surfaces along the geodesic *tunnel* loops a_1 , a_2 and a_3 , we obtain three patches, i.e., a topological annulus P_1 and two topological disks P_2 and P_3 (see Figure 6.5). The six feature points, $a_1 \cap c_{12}$, $a_2 \cap c_{12}$, $a_2 \cap c_{32}$, ..., defined as the intersection of two landmark curves. Note that each boundary of P_1 , P_2 and P_3 has exactly 6 feature points, which allows us to define a boundary condition in the registration so that the landmark curves can be matched exactly.

Figure 6.4: Thanks to the intrinsic and geometry-aware nature of the geodesic loops, the extracted landmarks are stable and highly consistent among the AIS and normal subjects.

6.2.2 Registration

Given two VS surfaces S^1 and S^2 with the extracted landmarks $\{a_1^i, a_2^i, a_3^i, c_{12}^i, c_{23}^i, c_{31}^i\}$, $i = 1, 2$, our goal is to find a bijective map $\phi : S^1 \rightarrow S^2$ with exact landmark correspon-

dence, i.e., $\phi(a_1^1) = a_1^2$, $\phi(c_{12}^1) = c_{12}^2$, etc. As each VS surface can be segmented into three patches, $S^i = \bigcup_{j=1}^3 P_j^i$, $i = 1, 2$, our idea is to find a bijective map $\phi_i : P_i^1 \rightarrow P_i^2$ between each pair of patches. With a carefully designed boundary condition for each individual map, we can “glue” them together to form the bijective map with exact landmark correspondence.

Figure 6.5: The geodesic landmarks induce a segmentation of the vestibular system. g : genus, b : number of boundaries.

Computing the map between two topological annuli

The patch P_1 containing three semicircular canals is the major component of the VS. Its geometry resembles a long tube and its topology is equivalent to an annulus. We first parameterize the patch P_1 to a canonical annulus (with outer radius equals 1) using the Gu-Yau method [34]. Then we cut the annulus open and map it to a rectangle using the complex logarithm function $z \mapsto \log z$. Let t denote the conformal map from the patch P_1 to the annulus and g denote the conformal map from the annulus to a rectangle. Then the composite map $g \circ t$ maps P_1 to a rectangular domain R .

Now given two patches P_1^1 and P_1^2 , we compute the maps $g_1 \circ t_1 : P_1^1 \rightarrow R_1$ and $g_2 \circ t_2 : P_1^2 \rightarrow R_2$. Then we compute the extremal quasi-conformal map $r : R_1 \rightarrow R_2$ between the two rectangles. Similar to the extremal quasi-conformal map d in the disk map, we specify the exact landmark matching by matching the feature points and parameterize the non-feature points using arc length parameterization. Thus, the annulus map

$\phi_1 : P_1^1 \rightarrow P_1^2$ is defined as $\phi_1 = t_2^{-1} \circ g_2^{-1} \circ r \circ g_1 \circ t_1$, see the commutative diagram in Figure 6.6(b).

Computing the map between two topological disks

Observe that both P_2 and P_3 are topological disks and the boundary of each disk has 6 feature points, which are the intersection points of landmark curves. Here we present the method to compute the map $\phi_2 : P_2^1 \rightarrow P_2^2$. The other map $\phi_3 : P_3^1 \rightarrow P_3^2$ can be constructed in the same way. Let q_j^i , $j = 1, \dots, 6$ denote the six feature points on ∂P_2^i . We map each disk P_2^i to the unit disk $\mathbb{D}$ by computing the harmonic map $f_i : P_2^i \rightarrow \mathbb{D}$. The boundary condition is specified by using the arc length parameterization: Set the corner q_1^i as the reference point with $f_i(q_1^i) = e^{i0}$. Then for any boundary point $q \in \partial D^i$, $f_i(q) = e^{i\theta(q)}$, where $\theta(q) = 2\pi l(q, q_1^i)/L$, and $l(s, t)$ measures the boundary length from s to t , and L is the total length of the boundary ∂D^i . Next, we compute the extremal quasi-conformal map $d : \mathbb{D} \rightarrow \mathbb{D}$, which aligns two unit disks such that the images of the feature points $d(f_1(q_1^2)) = f_2(q_1^2)$ are matched and the non-feature boundary points are determined by the arc-length parameterization. Finally, the map $\phi_2 : P_2^1 \rightarrow P_2^2$ is constructed by the composite map $\phi_2 = f_2^{-1} \circ d \circ f_1$. See Figure 6.6(a) for the commutative diagram.

Constructing the global map

The global map $\phi : S^1 \rightarrow S^2$ is simply constructed by gluing all three individual maps together, i.e., $\phi = \bigcup_{i=1}^3 \phi_i$. Here we show the map ϕ is a homeomorphism with exact landmark correspondence. In the disk maps ϕ_2, ϕ_3 and the annulus map ϕ_1 , the harmonic maps f_1, f_2 and the extremal quasi-conformal map d and r are all diffeomorphic, since the domains and co-domains are all convex and the boundary conditions are homeomorphisms. The holomorphic 1-form induced mappings t_1, t_2 are conformal and diffeomorphic. The complex log functions g_1 and g_2 are also diffeomorphic when restricting

Figure 6.6: Registration. (a) The commutative diagram for the map between two topological annuli. (b) The commutative diagram for the map between two topological disks. The red dots are the images of the feature points. To avoid label congestion, only two feature points and their images are labeled, the others can be implied according to the order.

the argument to $[0, 2\pi]$. Thus, all three maps ϕ_1, ϕ_2 and ϕ_3 are diffeomorphic for all the interior points.

Although the maps ϕ_i are calculated individually, they can be glued seamlessly, since we use the arc-length parameterized boundary condition in computing the extremal quasi-conformal maps. Note that each boundary has exactly 6 feature points, which are guaranteed to be matched as required in the boundary conditions used in harmonic maps f_1, f_2 and the extremal quasi-conformal maps d and r . For the nonfeature points on the boundary, their images are completely determined by the metric since we parameterize them using the arclength parametrization. Thus, for two adjacent patches sharing a common boundary, both the feature and non-feature points can be matched exactly, which implies that all the boundary maps are homeomorphic. Putting it altogether, the global map, $\phi = \bigcup_i^3 \phi_i$, is in fact a homeomorphism with exact landmark correspondence.

Correctness

Proposition. The proposed VS surface registration algorithm produces a homeomorphism ϕ between S^1 and S^2 , which guarantees the exact correspondence between the six geodesic landmarks $\{a_1^i, a_2^i, a_3^i, c_{12}^i, c_{23}^i, c_{31}^i\}, i = 1, 2$, i.e. $\phi(a_1^1) = a_1^2, \phi(c_{12}^1) = c_{12}^2$, etc.

Proof. The annulus map $\phi_1 = t_2^{-1} \cdot g_2^{-1} \cdot r \cdot g_1 \cdot t_1$ is a diffeomorphism, since $t_i, g_i, i = 1, 2$, are conformal and r is extremal quasi-conformal. The disk map $\phi_2 = f_2^{-1} \cdot d \cdot f_1$ is a diffeomorphism too, since f_1, f_2 are harmonic and d is extremal quasi-conformal map. Similarly, ϕ_3 is also a diffeomorphism. Note that we use the arc-length parameterization based boundary condition in computing these extremal quasi-conformal maps and harmonic maps r, f_1, f_2 , and d . These boundary conditions can guarantee that the boundary of two adjacent patches is mapped consistently and thus all patches are glued seamlessly in a C^0 manner. Thus, the global map, $\phi = \bigcup_i^3 \phi_i$, is in fact a homeomorphism.

Also note that each corner (i.e., the intersection of two landmarks) of the source patch is mapped to the corresponding corner of the target patch. For example, the map ϕ^2 sends the corner $q_1^1 \in P_2^1$ to the corner $q_1^2 \in P_2^2$, and $q_6^1 \in P_2^1$ to $q_6^2 \in P_2^2$. Then, the arc-length parameterization based Dirichlet boundary condition guarantees ϕ_2 sends the boundary curve $q_1^1 q_6^1$ to the boundary curve $q_1^2 q_6^2$. Therefore, the produced registration has guaranteed landmark correspondence. $\square$

Input: Two vestibular system surfaces S_1 and S_2

Output: The homeomorphism $\phi : S_1 \rightarrow S_2$ with guaranteed landmark correspondence

- 1 Compute the six geodesic landmarks for each VS surface using Algorithm 1.
- 2 Cut each VS surface into three genus-0 patches P_1, P_2 , and P_3 , along the extracted landmark features.
- 3 Compute the map $\phi_1 : P_1^1 \rightarrow P_1^2$ between two topological annuli.
- 4 Compute the map $\phi_2 : P_2^1 \rightarrow P_2^2$ between two topological disks.
- 5 Compute the map $\phi_3 : P_3^1 \rightarrow P_3^2$ between two topological disks.
- 6 Output the map $\phi = \bigcup_{i=1}^3 \phi_i$.

Algorithm 6: Registration of vestibular system surfaces registration with guaranteed landmark correspondence.

Figure 6.7: We compute the average shape of 13 normal subjects as the control model, and then measure the difference between the AIS subjects and the control model in terms of various geometric features, such as Gaussian curvature K , mean curvature H , area and conformal representation (λ, H) . The cut-off value for a 95% confidence interval is $t = 2.13$. Among them, the mean curvature H with $t = 2.29$ is an effective measure for distinguishing the normal and AIS subjects.

Figure 6.8: We normalize the mean curvature to the unit interval and then visualize the difference between the test model and the control model.

6.3 Results & Discussions

Experimental results: We implemented our algorithm in C++ on a PC with an Intel Core2 Quad CPU 2.83GHz and 4GB RAM. TAUCS¹ is adopted as the linear system solver for the harmonic map and holomorphic 1-form methods. We tested 28 VS surfaces (15 AIS patients and 13 normal subjects), which were extracted from the MRI. Each VS surface is modeled as a triangle mesh with about 4K vertices. Our method is automatic and efficient: computing the geodesic landmark features takes less than 1 second for each VS surface and registering two VS surfaces takes roughly 6 seconds.

To evaluate the difference of VS between the AIS and normal subjects, we computed the average shape for all 13 normal subjects, which is used as our control model (see Figure 6.7(a)). Let $f : M \rightarrow N$ be the mapping from a VS subject M to the control model N . Then we defined the following functionals to measure the difference between M and N based on various types of geometric properties,

$$\begin{aligned} E_A &= \int_M |d\sigma - d\sigma(f)|^2 \\ E_K &= \int_M |K - K(f)|^2 \\ E_H &= \int_M |H - H(f)|^2 \\ E_C &= \int_M |d\lambda - d\lambda(f)|^2 + s|H - H(f)|^2 d\sigma, \end{aligned}$$

where σ is the area element, K is the Gaussian curvature, H is the mean curvature, λ is the conformal factor and the scalar $s > 0$ balances the terms of conformal factor and mean curvature. The functionals E_A , E_K and E_H measure the distortion of area, Gaussian curvature and mean curvature, respectively. The functional E_C extends E_H by considering the conformal factor, which can measure the stretch distortion of the conformal

¹<http://www.tau.ac.il/~stoledo/taucs/>

representation (λ, H) . Figure 6.7(b)-(e) shows the statistical analysis by using the box plot. We observed that the mean curvature distortion E_H can effectively distinguish the normal and AIS subjects, while the others do not. Figure 6.8 shows the mean curvature difference on test subjects and the control model.

Figure 6.9: Visualizing the registration using checker board texture mapping. With the aid of texture mapping, one can clearly see the exact correspondence of the geodesic landmark features.

Comparison: There is a large body of literatures in landmark-matching registration. Most of them address the registration in the planar domain $\mathbb{R}^2$, and the representative works for 3D shape registration include the conformal mapping [32, 108], the quasi-conformal mapping [123, 71] and the large deformation diffeomorphic metric mapping (LDDMM) [5, 46, 86]. The LDDMM framework places the shapes in a metric space, and provides a diffeomorphic transformation by solving the transport equation of a time dependent vector field. The LDDMM framework is theoretically sound and elegant, which can guarantee a diffeomorphism with exact landmark matching. However, its high computational cost and memory requirement diminishes its application to large surfaces.

Our method is different than the existing works in that our application domain is limited to the genus-3 vestibular system, since both our landmark extraction and registration take advantage of the unique geometric as well as topological features of the VS. Our technique is robust, fully automatic and highly efficient, the resulting registration is

guaranteed to be homeomorphism with exact landmark correspondence. The aforementioned techniques are not able to do that, due to either the high computational cost or the topological challenge. Thus, we consider the proposed method a very *practical* and *effective* solution for registration of the vestibular systems.

Chapter 7

GRIP: Greedy Routing through dIstributed Parametrization for Guaranteed Delivery in 2D/3D Sensor Networks

In this chapter, we modify the way to compute holomorphic 1-form and propose a distributed algorithm to compute a parametrization similar to the conformal map for sensor network routing. The modified algorithm is totally distributed and perfect for the situation like sensor network, where each sensor has limited computational capacity. In addition, the algorithm can be easily generalized to the 3D setting, which distinguishes our method with others and can be used to solve the 3D sensor network routing problem in real life.

7.1 Introduction

7.1.1 Overview of the Problem

Greedy (geographic) routing has long been considered as a scalable data delivery mechanism in large scale multi-hop wireless networks, due to its dependence only on local

information [9, 50]. In a nutshell, knowing the geographic location of the destination, the routing decision selects, among all the one-hop neighbors of a forwarding node, the one that is the closest to the destination. Such a protocol is *stateless* as the routing decision is made upon information local to the forwarding node.

For the 2D setting, although stateless greedy routing could fall into local minimum due to the existence of *communication voids* (or *holes*) in a network [24], combining it with *face* (or *perimeter*) routing (which relies on a planarized connectivity graph [9, 50] or well detected hole boundaries [24]) may still guarantee data delivery. Unfortunately, as *two-dimensional* (2D) structures such as face or perimeter cannot be trivially extended to 3D, routing decisions relying only on local information cannot guarantee delivery for 3D networks in general [21].

Another body of research works looked at whether stateless greedy routing may work under a certain embedding of a connectivity graph [82, 53]. When exists, such an embedding is termed *greedy embedding*. In particular, Kleinberg [53] shows that every finite connected graph has a greedy embedding in the hyperbolic plane. However, embedding the combinatorial structure of a network requires the embedding to be substantially revamped upon any change in the network, making such a scheme impractical [29].

7.1.2 Our Approach and Contributions

Our approach in this chapter extends the idea of mapping a geometric space rather than embedding a combinatorial structure (e.g., [95]). This is motivated by the fact that, in a densely deployed *Wireless Sensor Networks* (WSN), the space covered by the network is much more stable than the connectivity graph of the network. We propose a **fundamentally new geometric map** valid in both 2D and 3D, and we also propose localized algorithms to perform this map in WSNs.

In terms of performing greedy routing in an embedding domain, existing approaches either stick to the distance greedy principle (leading to an unbalanced load distribution [53, 95]) or rely on complicated transformations to produce multiple routing paths [124]. Our approach, in contrast, only relies on a tunable greedy objective to naturally generate routing path diversity. In summary, our *Greedy Routing through dIstributed Parametrization* (GRIP) has two main components:

- (i) A localized algorithm to map a network domain $\mathcal{N} \in \mathbb{R}^d, d = 2, 3$ into a virtual domain $\mathcal{D} \in \mathbb{R}^d$, such that all the boundaries (outer or inner) become circular (2D) or convex (3D).
- (ii) A tunable greedy routing mechanism to guarantee data delivery between any $s - t$ pairs in $\mathcal{N}$, while providing flexibility to fine tune the tradeoff between energy efficiency and load balancing.

In designing these two components of GRIP, we make the following main contributions:

- A fundamentally new parametrization procedure to “regularize” the boundaries (outer or inner) of a multi-connected domain in both 2D and 3D.
- Fully distributed and localized algorithms to perform the geometric parametrization and to obtain virtual coordinates in a WSN.
- A greedy routing objective that combines both the distance and the “position” in the harmonic field resulting from the parametrization, providing tunability towards diversified routing paths.
- A real implementation of GRIP in TOSSIM¹ for feasibility evaluations.

¹<http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM>

7.2 Related Work on Greedy Routing

Our discussions in this section only focus on the approaches of applying embedding (or domain map) to allow stateless greedy routing. The reason is that, the “local tuning” approaches, such as [9, 50], to enhance the stateless greedy routing is proven to be impossible for 3D routing by Durocher *et al.* [21]. What is shown in [21] is that routing decision based only on local information may always lead to infinite loop in 3D Euclidean space, which is essentially the consequence of the increased “freedom” in choosing directions in a high dimensional space. Although randomized algorithms may give a temporary cure to this issue [29], it simply cannot provide guaranteed delivery.

One of the earliest approach of generating virtual coordinates through embedding is proposed in [88]. While the idea there was about location-free routing (rather than getting stateless greedy routing out of local minimums), Rao *et al.* [88] did motivate the idea of using virtual coordinates to enable stateless greedy routing. The question asked by Papadimitriou and Ratajczak [82] is under what condition(s) a connectivity graph admits a *greedy embedding*, i.e., an embedding in a 2D plane under which stateless greedy routing never fails. Their conjecture was that any planar 3-connected graph admits greedy embedding, and the conjecture was later proven by Leighton and Moitra [58].

One of the important results along the line of research is Kleinberg’s statement that every connected finite graph has a greedy embedding in hyperbolic plane [53]. Although this result extends the condition for greedy embedding from 2D to an arbitrarily higher dimension, the idea of embedding network topology graph (a combinatorial structure) has two major drawbacks. Firstly, any change in a network may require a substantial revamp of the embedding; the large overhead makes such a scheme impractical in large scale WSNs [29]: one would be better off using a stateful routing instead. Secondly, as

the embedding is performed on a spanning tree, the path between any $s - t$ pair becomes unique and many $s - t$ pairs may share a substantial amount of links with their routing paths. The consequent unbalanced load again makes this scheme unsuitable for the energy constrained WSNs.

Suppose we can **map the space** covered by a WSN to another one with a geometry that favors the greedy routing, changes in network topology would be limited within their neighborhood; they affect the whole map only if certain communication voids that hamper a greedy routing are created. This is the idea brought forward by Sarkar *et al.* [95], and was later improved with load balancing features by Zeng *et al.* [124]. Unfortunately, these earlier approaches are confined in 2D spaces, even if we consider only 2D WSNs, the heavy computation load imposed by these tools makes themselves less practical. Our approach presented in this paper significantly improves upon these two aspects. Our map works for **both 2D and 3D**. The computations required for sensor nodes are all **arithmetic**; they can be performed in a localized manner and converge quickly. All these features contrast strikingly against the existing approaches that entail computing trigonometric functions and result in a slow convergence rate [95, 124].

7.3 Overview of GRIP

In this section, we first explain the principles that guide the GRIP design. Then we describe the network model and the system construction of GRIP. We also provide some discussions on the services required by GRIP to function properly.

7.3.1 Basic Principles

We use Figure 7.1 to illustrate the principles of GRIP. As shown on the left side, the existence of two lakes in the area covered by a WSN result in two communication voids

(of non-convex shapes),² cause failure of stateless greedy routing. To overcome this

Figure 7.1: A network domain with non-convex holes (left) may lead to routing failure. GRIP first performs distributed parametrization to map the domain to a virtual domain (right), then it applies a greedy routing with tunable objective to achieve load-balancing routing.

difficulty, GRIP first parameterizes the original network domain to a virtual domain, in which all the boundaries are convex. Under the coordinates of this virtual domain, we will show that stateless greedy routing always succeeds. However, the contribution of GRIP goes beyond a simple stateless greedy routing, which can lead to a concentration of routing paths at the internal boundaries [124]. As shown in Figure 7.1 (the right side), GRIP’s tunable greedy routing can get around the holes by having a greedy objective that adapts to the geometry of the network domain.

7.3.1.1 Network Model and System Construction

We assume that WSN nodes are deployed in 2D or 3D spaces and that two nodes can communicate with each other directly iff their distance is below a certain threshold. Without loss of generality, we set this threshold to one. The resulting graph $\mathcal{G}(V, E)$

²Geological objects are only one reason for this; irregular deployment of WSNs may be another.

is termed *unit disk graph* (UDG) for 2D and *unit ball graph* (UBG) in 3D, with V representing the set of nodes and edge (u, v) is in E if u and v can communicate with each other directly. For a 3D WSN, we assume that all its boundaries are genus-0 surfaces. A *stateless greedy routing* relies only on the locations (either exact or virtual) of the source, the destination, as well as every forwarding node u and its one-hop neighbors $N(u)$.

The components of GRIP are as follows:

- Restricted Delaunay Triangulation
- Boundary Detection and Location Service
- Distributed Network Parametrization
- Tunable Greedy Routing

We will briefly introduce the first two components in the following. The remaining components, which are our main contributions in designing GRIP, will be detailed in Sec. 7.4 and 7.5, respectively.

7.3.1.2 Restricted Delaunay Triangulation

In order to parameterize the network domain to a virtual domain with required properties, the topology of the network domain is needed. GRIP gets aware of this information through the WSN nodes. In other words, using the connectivity relations or the location information (if available), GRIP constructs a Delaunay triangulation upon the network domain, which is a discrete representation of the “shape” of the domain.

Straightforward Delaunay triangulation may result in arbitrarily long edges, hence can fail to represent the topology of a domain. For example, triangular faces with very long edges may eventually cover the holes, hence create an illusion of a simply connected domain. Gao *et al.* [30] propose a **restricted Delaunay graph** (RDG): it confines its edge

set $\tilde{E}$ to be a subset of the network edge set E . The outcome is that all the communication voids are represented as non-triangular faces. Although the distributed algorithm proposed in [30] is meant for 2D networks, it can be extended to 3D to construct tetrahedron mesh, again based on the principle of the restricted Delaunay graph.

Remark: The RDG only serves our distributed parametrization; it is by no means used to assist routing.

7.3.1.3 Other Supporting Services

Similar to other routing mechanism relying on virtual coordinates, GRIP needs the support from two services, namely boundary detection and location management. While boundary detection lets a node at a boundary to be aware of its state and the boundary surface it sits on, location management allows any node to query the (virtual) coordinates of other nodes. A location management service is not constrained by the dimensionality, so we can use any existing solutions, such as [64]. An effective 3D boundary detection scheme has recently been proposed [127]. In addition to getting aware of the network boundaries, it also supports distributed coordinations among boundary nodes. For example, a boundary surface can be constructed in a distributed way and certain consensus can be reached among nodes belonging to the same boundary surface.

7.4 Distributed Parametrization in 2D/3D Wireless Sensor Networks

In this section, we focus on the distributed algorithms to realize the map illustrated in Figure 7.1. In Sec. 7.4.1, We first introduce the mathematical representation for individual tools, as well as corresponding distributed algorithms to implement these

tools. Then we summarize the distributed parametrization procedures for 2D and 3D in Sec. 7.4.2.

7.4.1 Mathematical Tools

We have introduced the traditional holomorphic 1-form in previous chapter. The way to compute the holomorphic 1-form is to compute the harmonic field first, which can induce the harmonic 1-form; then we compute the Hodge star of the harmonic 1-form to get the complex 1-form. For the algorithm we proposed in this section, we firstly compute the harmonic field, that is exactly the same with what we do in holomorphic 1-form. However, in the second step we compute the integral curve along the gradient of the harmonic field directly rather than the Hodge star which rotates it to 90 degrees; this does not need the optimization to smooth the vector field and the tracing can be implemented by a distributed and propagated way, which makes our algorithm easily distributed for the sensor network routing.

We hereby explain the individual mathematical tools, namely harmonic function, gradient vector field, integral curve, and annulus embedding. For each tool, we propose a distributed algorithm to perform the computation; these algorithms are executed by individual nodes and they require only the access of local information. To simplify the exposition, we first introduce the concepts and tools for 2D cases in Sec. 7.4.1.1 to 7.4.1.4, and then discuss the generalization to $\mathbb{R}^3$ in Sec. 7.4.1.5.

7.4.1.1 Harmonic Function

Given a network domain $\mathcal{N} \subseteq \mathbb{R}^2$ with holes, we consider it as a multi-connected domain with boundaries $\partial\mathcal{N} = \gamma_0 - \gamma_1 - \cdots - \gamma_l$, where γ_0 is the outer boundary and γ_i ,

$1 \leq i \leq l$, are the hole boundaries. The harmonic function $f : \mathcal{N} \rightarrow \mathbb{R}$ is twice continuously differentiable and satisfies the Laplace's equation

$$\Delta f(p) = 0, \quad \forall p \notin \partial \mathcal{N},$$

with Dirichlet boundary condition $f|_{\gamma_i} = f_i, \quad 0 \leq i \leq l$.

According to Sec. 7.3.1.2, $\mathcal{N}$ is represented by a triangle mesh (i.e., an RDG) $M = (V, \tilde{E}, F)$ where V , $\tilde{E}$ and F are the set of vertices, edges and faces, respectively. As pointed out in [95], the boundaries ∂M of the mesh may well approximate $\partial \mathcal{N}$. Let $\tilde{E}(p)$ be the subset of $\tilde{E}$ incident to p , the discrete Laplace operator of an interior vertex $p \notin \partial M$ is defined as $\Delta f(p) = f(p) - \sum_{(p,q) \in \tilde{E}(p)} \omega_{pq} f(q)$, where ω_{pq} is the weight of directed edge (p, q) . The choice of the weight is usually application dependent. We choose a straightforward weight determined by vertex degree, i.e., $\omega_{pq} = \frac{1}{\deg(p)}$, with $\deg(p)$ being the degree of vertex p .

Observing the above Laplace equation is equivalent to the heat diffusion

$$\frac{\partial f}{\partial t} = -\Delta f, \tag{7.1}$$

we can solve it using the Euler method in a distributed manner, as shown by Algorithm 7. Given the boundary conditions $f_0 = 0$ and $f_i = 1, i \neq 0$, The initial and final states of a domain are shown in Figure 7.2.

7.4.1.2 Gradient Vector Field

Given an arbitrary smooth function g defined on $\mathcal{N}$, we can approximate the gradient ∇g using M . For a triangle $\triangle p_1 p_2 p_3$, let $g_i = g(p_i)$, $1 \leq i \leq 3$, be the function value defined for each vertex. Let $\overrightarrow{p_i p_{i+1}}$ be the vector from p_i to p_{i+1} (the modulo-3 operator is applied to the subscript index). The gradient ∇g inside triangle $p_1 p_2 p_3$ is given by:

$$\nabla g|_{\triangle p_1 p_2 p_3} = \sum_{i=1}^3 r(\overrightarrow{p_{i+1} p_{i+2}}) g(p_i),$$

Input: The Dirichlet boundary condition f_i . For each $p \in V$, $\tilde{E}(p)$, the step length δ , the stopping tolerance ε
Output: A harmonic function $f : M \rightarrow \mathbb{R}$

- 1 Initialize $f(p) \leftarrow f_i, p \in \gamma_i$; otherwise $f(p) \leftarrow 0$
- 2 For every interior vertex p periodically:
- 3 **if** $|\Delta f(p)| > \varepsilon$ **then**
- 4 $\Delta f(p) = f(p) - \sum_{(p,q) \in \tilde{E}(p)} \omega_{pq} f(q)$
- 5 $f(p) \leftarrow f(p) - \delta \Delta f(p)$
- 6 Broadcast $f(p)$ to all $q : (p, q) \in \tilde{E}(p)$
- 7 **end**

Algorithm 7: Computing harmonic function.

Figure 7.2: Computing harmonic function.

with $r(\vec{v})$ being the counter-clockwise rotation of vector $\vec{v}$ in 90 degree.

Given a vertex p , let $Nb(p)$ denote the triangles adjacent to p . We first compute the gradient for each triangle in $Nb(p)$ and then compute the gradient on p by weighted sum:

$$\nabla g(p) = \frac{\sum_{t \in Nb(p)} s(t) \nabla g|_t}{\sum_{t \in Nb(p)} s(t)},$$

where $s(t)$ measures the area of triangle t .

7.4.1.3 Tracing Integral curves

Suppose $\mathbf{F}$ is a 2D vector field defined on M , i.e., $\mathbf{F} : M \rightarrow \mathbb{R}^2$, and $\mathbf{x}(\tau)$ a parametric curve. Then $\mathbf{x}(\tau)$ is an integral curve of $\mathbf{F}$ if it satisfies $\mathbf{x}'(\tau) = \mathbf{F}(\mathbf{x}(\tau))$. In-

tuitively speaking, the integral curve is a curve such that the tangent vector to the curve at any point p along the curve is precisely the vector $\mathbf{F}(p)$. Given a triangle $\triangle p_1 p_2 p_3$ and an arbitrary point $q \in \triangle p_1 p_2 p_3$, we can compute the integral curve in a localized manner as shown in Algorithm 8, where $Barycentric(q | \triangle p_1 p_2 p_3)$

Input: A triangle $t = \triangle p_1 p_2 p_3$, the vector $\mathbf{F}(p_i)$, $i = 1, 2, 3$, a point $q \in t$, the step length δ
Output: The integral curve $\gamma \in t$ passing through q and following the vector field $\mathbf{F}$

```

1 while  $q \in t$  do
2    $(\lambda_1, \lambda_2, \lambda_3) = Barycentric(q | \triangle p_1 p_2 p_3)$ 
3    $\mathbf{F}(q) = \sum_{i=1}^3 \lambda_i \mathbf{F}(p_i)$ 
4    $q \leftarrow q + \delta * \mathbf{F}(q)$ 
5 end
```

Algorithm 8: Tracing inside a triangle.

returns the barycentric coordinates of q with respect to t . Note that only one of the vertices of t needs to perform the computation, which involves pure arithmetic operations. The collective outcome of tracing a gradient vector field (i.e., integral curves and iso-value curves) in a WSN is shown in Figure 7.3.

Figure 7.3: The integral curves are perpendicular to the iso-value curves.

7.4.1.4 Annulus Embedding of 2-connected domain

Assume M is a 2-connected domain $\partial M = \gamma_0 \cup \gamma_1$, and $f : M \rightarrow \mathbb{R}$ is a harmonic function, $\Delta f(p) = 0$, $p \notin \partial M$, and $f|_{\gamma_0} = 0$ and $f|_{\gamma_1} = 1$. We have a characterization of the gradient vector field:

Proposition 1 *Let $\mathbf{F} = \nabla f$ be the gradient vector field. Then $\mathbf{F}$ has the following properties:*

- *Any integration curve $\mathbf{x}$ is an open curve with two ending points on γ_0 and γ_1 respectively.*
- *Any two integration curves $\mathbf{x}_1$ and $\mathbf{x}_2$ do not intersect.*
- *For any point $p \in M$, there is a unique integration curve $\mathbf{x}$ passing through p .*

Proof: Note f is smooth function, its gradient vector fields are curl-free. Thus, no integral curve can form a loop inside M . Furthermore, the function f is harmonic and there is no critical points (where the gradient vanishes) inside M . Thus, the function value is strictly monotonic along the integral curve. Note that all points on the same boundary curve have the same function value, so the ending points of each integral curve must be on different boundary curve.

Then we show that two integral curves do not intersect. Assume two integral curves $\mathbf{x}_1 \in M$ and $\mathbf{x}_2 \in M$ intersect at a point p . Then p is a critical point and the gradient ∇f vanishes at p . Since f is harmonic, the maximum and minimum must be on the boundaries. Therefore the Hessian matrix at p has negative eigenvalue values. Suppose $f(p) = s$, then according to Morse theory, the homotopy types of the level sets $f^{-1}(s - \varepsilon)$ and $f^{-1}(s + \varepsilon)$ will be different, where ε is a small positive value. At all the interior critical points, the Hessian matrices have negative eigenvalues, the homotopy type of the level sets will be changed. Therefore, the homotopy type of γ_0 is different from that of γ_1 . This contradicts the given condition that M is 2-connected. Therefore $\mathbf{x}_1$ and $\mathbf{x}_2$ have no intersection points anywhere. $\square$

According to these properties, we can naturally embed the 2-connected domain to an annulus in a conformal (or angle-preserving) manner. Firstly, we calculate all integral curves from outer boundary to the inner boundary, then we map the inner boundary to an

inner circle of annulus. Finally we map each integral curve to corresponding radius on annulus. For the discrete case (where a triangular mesh M is concerned), we only need to calculate the integral curve from each vertex to the inner boundary. Once getting the integral curve, we can determine the vertex location on the annulus by polar coordinates (ρ, θ) , where ρ is determined by the harmonic function value f and θ is determined by the end point of integral curve on the inner boundary.

Figure 7.4: Propagative embedding from a 2-connected triangle mesh into an annulus.

Since for each vertex, f is known and we only need to compute θ in polar coordinates, the embedding can be accomplished in a distributed and efficient way, by tracing part of the integral curve (rather than the whole curve from vertex to inner boundary). Basically, a tracing can be stopped when it meets an edge whose f at two endpoints are both larger than the f of the starting vertex. Then the stopping position on the edge is recorded, and the θ of this position (which is in turn the θ of the starting vertex) is obtained by linear interpolation. As a vertex with lower f always obtain its θ from some vertices that have bigger f , a distributed iteration may start from the vertices on the inner boundary, whose f is highest, and gradually propagate to the outer boundary. We can always obtain the θ for all vertices after the iteration reaches the outer boundary. This distributed

embedding procedure is illustrated in Figure 7.4 and is shown by Algorithm 9, where $l(p, q)$ (line 15) returns the length of edge (p, q) .

<p>Input: A 2-connected triangle mesh $M \in \mathbb{R}^2$ with two boundaries $\partial M = \gamma_0 \cup \gamma_1$</p> <p>Output: A conformal map ϕ that maps M to an annulus, i.e., $\phi(p) = (\rho_p, \theta_p)$, where (ρ_p, θ_p) is the polar coordinate.</p> <ol style="list-style-type: none"> 1 Compute a harmonic function $f : M \rightarrow \mathbb{R}$ with Dirichlet boundary condition $f _{\gamma_i} = i$, $i = 0, 1$, by using Algorithm 1 2 Choose a vertex $p_0 \in \gamma_1$ and set its $\theta_{p_0} = 0$ 3 Compute $\ell = \int_{\gamma_1} dl$, the length of γ_1 4 if vertex $p \in \gamma_1$ then <ol style="list-style-type: none"> 5 $l_p =$ length from p to p_0 on γ_1 in counter clockwise direction 6 $\theta_p = \frac{l_p}{\ell}$ 7 else <ol style="list-style-type: none"> 8 Apply Algorithm 2 to trace the integral curve from p to an edge (a, b), s.t. $f(a) > f(p)$ and $f(b) > f(p)$ 9 Record the stopping position s_p 10 Initialize the θ of p to NaN 11 end 12 For every interior vertex p: 13 Periodically check all $(p, q) \in \tilde{E}(p)$ for which at least one stopping position $s_{\bar{p}}$ is recorded 14 if $\theta_{\bar{p}} = NaN$, $\theta_p \neq NaN$, $\theta_q \neq NaN$ then <ol style="list-style-type: none"> 15 $\theta_{\bar{p}} = \theta_p \frac{l(p, s_{\bar{p}})}{l(p, q)} + \theta_q \frac{l(s_{\bar{p}}, q)}{l(p, q)}$ 16 Update vertex $\bar{p}$ with $\theta_{\bar{p}}$ 17 end 18 return $(\rho_p = f(p), \theta_p)$

Algorithm 9: Annulus embedding.

The first part of the algorithm (lines 2 to 6) needs distributed coordinations along the inner boundary, we refer to Sec. 7.3.1.3 for details. The second part involves curve tracing (lines 7 to 11) and θ value propagation (lines 12 to 17). It can be shown that none of them require transmissions beyond a single hop, so the algorithm is by default localized.

7.4.1.5 Extension to 3D

We have shown computing discrete harmonic function and its gradient on triangular mesh embedded in $\mathbb{R}^2$. This machinery can be easily generalized to a tetrahedral mesh in $\mathbb{R}^3$. Let $M = (V, \tilde{E}, F, T)$ be a tetrahedral mesh, where T is the set of tetrahedral. Computing harmonic function on tetrahedral mesh can also be done by Algorithm 7; the outcome is illustrated by Figure 7.5(a)–(c).

Figure 7.5: The cut views of the harmonic function on the 3D network domain with a void (a) – (c). Integral curves in a network domain (d) and in the embedding domain, i.e., a solid ball with a spherical hole in the center (e).

Computing the gradient in tetrahedral mesh is similar to the 2D counterpart. Given a tetrahedron $p_1p_2p_3p_4$, let $\vec{v}_i$, $1 \leq i \leq 4$, be a vector, which is perpendicular to triangle $\triangle p_{i+1}p_{i+2}p_{i+3}$ and points outward of the tetrahedron. The magnitude $\|\vec{v}_i\|$ equals the area of triangle $\triangle p_{i+1}p_{i+2}p_{i+3}$. Then, the gradient ∇g inside tetrahedron $p_1p_2p_3p_4$ is

given by:

$$\nabla g|_{p_1 p_2 p_3 p_4} = \sum_{i=1}^4 \vec{v}_i g(p_i).$$

Computing the gradient for each vertex is also similar to the 2D case except that tetrahedron volume is used in the weighted sum.

Figure 7.6: Distributed parametrization of a 2D WSN. Each iteration of the algorithm consists of three steps, since the given 2D WSN contains three holes. Initially, all holes are filled virtually. In each step, we open one hole to get a 2-connected domain, and then apply Algorithm 3 to parameterize it to an annulus. The original network domain, the intermediate results, and the final parametrization after two iterations are shown from left to right.

When embedding a 2-connected domain in 2D to an annulus, we need to map the inner boundary to a circle (lines 2-6 in Algorithm 9). Let $D_r = \{x \in \mathbb{R}^3 | r \leq \|x\|_2 \leq 1\}$ be the 3D embedding domain, then we need to map the inner boundary (surface) γ_1 to a sphere with radius r . This can be tackled by using Algorithm 7 to solve a harmonic function, so it is again localized and involves only arithmetic operations. Details are omitted for brevity. We also need to compute another harmonic function $f : M \rightarrow \mathbb{R}$ with Dirichlet boundary condition $f|_{\gamma_0} = 0$ and $f|_{\gamma_1} = 1$. Then the gradient vector field ∇f has exactly the same properties as its 2D counterpart (see Proposition 1). Therefore, similar to Algorithm 9, we can embed M to D_r by tracing the integral curves, as shown by Figure 7.5(d) and (e).

7.4.2 Distributed Parametrization

With the set of algorithms presented in Sec. 7.4.1, we can map a network domain with one hole to an annulus. We now present Algorithm 10 that parameterizes a network domain with many holes, by iterating the single-hole map (Algorithm 9) among all the holes. We also illustrate the algorithm in Figure 7.6 and the outcome for a 3D domain in Figure 7.7.

Input: A domain M with multiply boundaries
Output: A bijective parametrization ϕ map M such that all boundaries become convex

```

1 Fill all holes virtually
2 while Some hole boundary  $\gamma_i$  is not convex do
3   |   Open the  $i$ -th hole virtually
4   |   Perform Algorithm 9 in terms of  $\gamma_i$ 
5 end
```

Algorithm 10: Iterative parametrization of a domain with multiply genus-0 boundaries.

The procedures of filling and opening holes virtually (lines 1 and 3) simply require an inner boundary vertex to identify a corresponding vertex on the boundary and to bridge, with virtual edges, all the vertex pairs or to cut the virtual edges. These can be achieved with the support of a boundary detection service [127]. The convexity test in line 2 can be performed in a localized manner. In particular, for each vertex $p \in \gamma_i$, its neighborhood $N(p)$ in $\mathcal{G}$ should contain a subset $\bar{N}(p) \subset \gamma_i$. Let α_p be the angle (or solid angle in 3D) determined by $\{p\} \cup \bar{N}(p)$, then the convexity test returns true of $\alpha_p < \pi$ (or $\alpha_p < 2\pi$ in 3D). In general, we have

Proposition 2 *Our iterative parametrization algorithm terminates with all boundaries being convex.*

Remark: Here is a brief discussion about the correctness this proposition:

2D WSN. Consider a 2D multiply connected network domain $\mathcal{N} \in \mathbb{R}^2$ with boundaries $\partial\mathcal{N} = \gamma_0 - \gamma_1 - \dots - \gamma_l$, where γ_0 is the outer boundary and γ_i , $1 \leq i \leq l$, are the hole boundaries.

Each iteration of the parametrization algorithm contains l steps: in the first step, we conformally map γ_0 to the unit circle and γ_1 to a concentric circle. Let $\phi_1 : \mathcal{N} \rightarrow \mathbb{D}$ be the conformal map of the first step, where $\mathbb{D}$ is the unit disk. Then in the i -th step, $i > 1$, we conformally map $\phi_{i-1} \circ \dots \circ \phi_1(S)$ to the unit disk $\mathbb{D}$ such that γ_0 is mapped to the unit circle and γ_i to a concentric circle, i.e., $\phi_i : \phi_{i-1} \circ \dots \circ \phi_1(\mathcal{N}) \rightarrow \mathbb{D}$. Note that $\phi_1(\gamma_1)$ is a circle. After the map, $\phi_2 \circ \phi_1(\gamma_1)$ is still close to a circle. Intuitively, all the boundaries are getting rounder and rounder in the iterations, and eventually become circles. As each map ϕ_i is conformal, Henrici's theorem (see [40], pp.502-505) on complex analysis guarantees the convergence of our method.

3D WSN. Similar to planar case, each time we map an inner void to the center spherical void, the map can be extended to the volume with its reflection with respect to the boundary surface of the inner void. Eventually, the complement of the union of reflected volumes is a Cantor set, with zero measure. The harmonic map can be extended to the whole volume with all voids filled. By using the Poisson integral formula for harmonic maps [97], it can be shown that the images of the inner boundary surfaces are spherical.

In fact, for 2D WSNs, we can use circularity as the termination condition, such that the algorithm terminates with all boundaries being circular. We omit the details for brevity.

7.5 Tunable Greedy Routing

With the parametrization procedure applied to a WSN, we are now ready to present our routing protocol under the resulting virtual coordinates. In this section, we first prove,

Figure 7.7: Parametrization of a 3D WSN, where the central void is mapped to a sphere and the others are mapped to the convex polyhedron.

in Sec. 7.5.1, that stateless greedy routing never fails under the virtual coordinates computed through the distributed parametrization. Then we present an augmented greedy routing in Sec. 7.5.2: it has a tunable greedy objective that leverages on the harmonic field computed during the parametrization phase (see Sec. 7.4.1.1), and can hence avoid concentrating the routing load on the hole boundaries.

7.5.1 Guaranteed Delivery

A stateless greedy routing, as defined in Sec. 7.3.1.1 needs the locations of the destination d , the current forwarding node u , and its one-hop neighbors $N(u)$. The greedy forwarding decision at each hop u in general takes the following form:

$$\hat{v} = \arg \min_{v \in N(u)} \text{dist}(v, d), \quad (7.2)$$

where $\text{dist}(v, d)$ returns the Euclidian distance between v and d , and $\hat{v}$ indicates the next hop. Here the *greedy objective* is $\text{dist}(v, d)$ and the *decision oracle* returns $\hat{v}$ that minimizes the objective. Due to the existence of holes in a WSN, oracle (7.2) may return $\{u\}$ and thus lead to a routing failure. To cope with this issue, we add another criterion

in case of failure:

$$\hat{v} = \arg \min_{v \in N(u)} (\vec{d} - \vec{v}) \cdot \vec{d}, \quad (7.3)$$

where the vector is defined with the source node s as the origin. As we have shown in Sec. 7.4.2, GRIP's distributed parametrization results in virtual coordinates under which all the boundaries are convex. Now we show that stateless greedy routing never fails.

Proposition 3 *If a network, under certain virtual coordinates, have all its boundaries convex, our stateless greedy routing always guarantees data delivery.*

Proof: The correctness is obvious if oracle (7.2) never fails, so we only need to show how (7.3) may overcome the situation where (7.2) fails. The only possibility that (7.2) fails is that the local angle (or solid angle for 3D) become close to π (or 2π for 3D) [24]. However, this also indicates that the routing path has hit a boundary [30], while a boundary is now convex according to PROPOSITION 2. The convexity further suggests that, for any node u on the boundary, there exists $v \in N(u)$ such that $(\vec{d} - \vec{v}) \cdot \vec{d} < (\vec{d} - \vec{u}) \cdot \vec{d}$. Therefore, the greedy objective (7.3) always drives the routing path to make progress along $\vec{d}$, hence away from the starting point u_1 , as shown in Figure 7.8. Consequently, no routing loop can be produced by (7.3). As soon as (7.3) brings the routing path sufficiently far from the node where (7.2) fails (e.g., to node v in Figure 7.8), oracle (7.2) will return successfully and hence the protocol will switches back to (7.2). The existence of such a “switching” node can be proven by contradiction. Assume it does not exist, then one can trivially show that d is in the hole, a contradiction. $\square$

For each $u \in V$ not on a boundary, there always exists a $v \in N(u)$ such that $d(v, d) < d(u, d)$ under the virtual coordinate, therefore,

Corollary 1 *The decision oracle in (7.2) only needs to search within $N^+(u) \subset N(u)$, such that $d(v, d) < d(u, d), \forall v \in N^+(u)$.*

Figure 7.8: Greedy forward decision (7.3) always succeeds and is loop-free on a convex boundary.

7.5.2 Tuning Greedy Objective

Although stateless greedy routing never fails under GRIP’s virtual coordinates, many routing paths could be concentrated along the inner boundaries. This unfortunate phenomenon is a direct consequence of the decision oracle (7.2): as distance is the only greedy objective, many routing paths are forced to turn away from the line segment between source and destination only upon “hitting” a hole boundary.

Our solution is to slightly change the greedy objective for non-boundary nodes by taking into another “coordinate” into account. One byproduct of the distributed parametrization is a scalar field³ given by the harmonic function $f(\cdot)$. Specifically, every node $u \in V$ has a scalar value $f(u) \in [0, 1]$ that indicates its “distance” towards some inner boundary: the larger the $f(u)$, the closer u is to that inner boundary. We define a new decision

³The vector field will be discarded after the parametrization, as it is used to assist the parametrization and is of no use to a greedy routing.

oracle by using $f(u)$:

$$\hat{v} = \arg \min_{v \in N^+(u)} (1 + \lambda |\Delta f_{vd}|) \text{dist}(v, d), \quad (7.4)$$

where $\Delta f_{vd} = f(v) - f(d)$. As $\hat{v} \in N^+(u)$, the distance toward d keeps reducing and thus the correctness of this routing algorithm is guaranteed. The rationale behind the greedy objective $(1 + \lambda |\Delta f_{v,d}|) \text{dist}(v, d)$ is that, as a routing path whose harmonic values deviate from $f(d)$ will be penalized, routing paths tend to be bent by *isosurfaces* rather than by inner boundaries. Consequently, the routing paths are evenly distributed instead of being concentrated at the inner boundaries. By tuning the weight λ , we may choose a proper tradeoff between energy efficiency and load balancing: a larger value of λ biases towards tracing the isosurfaces rather than the line from source to destination, resulting in better load balancing but longer routing paths.

In order to better adapt to the geometry of the network domain, we may tune λ online, as shown by Algorithm 11. Basically, the $\lambda_{\hat{v}}$ for $\hat{v}$ (the next hop) is tuned by the current

```

1  $\hat{v} = \arg \min_{v \in N^+(u)} (1 + \lambda_u |\Delta f_{vd}|) \text{dist}(v, d)$ 
2 if  $\Delta f_{\hat{v}u} = f(\hat{v}) - f(u) > 0$  then
3   |  $\lambda_{\hat{v}} = \text{incr}(\lambda_u)$ 
4 else
5   |  $\lambda_{\hat{v}} = \text{decr}(\lambda_u)$ 
6 end
7 Piggyback  $\lambda_{\hat{v}}$  with the packet to be forwarded to  $\hat{v}$ 

```

Algorithm 11: Online Greedy Objective Tuning.

forwarding node u according to $\Delta f_{\hat{v}u}$. On one hand, λ is increased to penalize a positive $\Delta f_{\hat{v}u}$, as the routing path tends to approach some inner boundary. On the other hand, λ is decreased to encourage a negative $\Delta f_{\hat{v}u}$, as it either helps to improve load balancing or allows the routing path to “jump” from one isosurface to another; this later jumping effect is necessary to maintain a reasonable routing stretch when the isosurface of the source is not connected (in topological sense) to that of the destination.

7.6 Discussions

As GRIP's virtual coordinates are obtained by embedding the network domain, they are very insensitive to changes in network topology. Nodes may join or leave the network, or they may move within the network domain. As far as no new holes are created, such changes in network topology will not cause another distributed parametrization procedure. Nodes that are affected by such changes only need to locally reconstruct the RDG and interpolate their virtual coordinates from their neighbors. As demonstrated by Figure 7.9, the cost to maintain the virtual coordinates is often negligible. On the contrary,

Figure 7.9: Changes in network topology due to node joining (left) and leaving (right) only have local influences. The dashed circle represents the transmission range of a node.

the approach that embeds the network topology [53], though appears to be lightweight during the parametrization phase, is very sensitive to changes in the topology: virtually any slight changes can lead to the revamp of the embedding.

Though discrete surface Ricci flow [95] is an effective technique to parameterize multiply connected surfaces by manipulating the target curvature, GRIP is superior to [95] in two aspects: First, the distributed version of the 2D discrete Ricci flow method is computationally expensive for sensor nodes, as trigonometric functions and square root are involved. GRIP, in sharp contrast, only entails simple arithmetic functions, and it

is thus much more efficient. Second, although the continuous version of Ricci flow for 3-manifolds has been widely studied, defining the discrete Ricci flow on tetrahedral meshes is still an open problem. GRIP is based on tracing integral curves, which is well defined in $\mathbb{R}^n$. We will compare their complexity in Sec. 7.7.1.

One potential drawback of our GRIP framework is that its 3D part has not been able to take into account the irregularities in network connectivity. In 2D, such irregularities are captured by extending UDG to Quasi UDG, and parametrization algorithms relying on triangulation can be adapted to Quasi UDG [95]. However, there is no extension to 3D UBG yet, and it is an open problem that what impact could be brought to triangulation by such an extension.

7.7 Simulations and Experiments

In this section, we evaluate the performance of GRIP through simulations. We have implemented GRIP in TOSSIM [60]. However, as TOSSIM has difficulty in simulating large scale WSNs (e.g., tens of thousands of nodes), we have also developed a high level simulator (which neglects the MAC effects) for those WSNs.

We evaluate the distributed parametrization phase and the routing phase separately. We first compare GRIP's parametrization with the Ricci flow based approach [95] in Sec. 7.7.1. Then we evaluate the performance of different greedy routing schemes under the virtual coordinates in Sec. 7.7.2.

7.7.1 Parametrization Efficiency

We compare GRIP's parametrization with the distributed Ricci flow method [95] in terms of time complexity. As discussed in Sec. 7.6, the latter only works for 2D meshes

for now, so the comparison is done in nine 2D WSNs with different sizes. As the Ricci flow method has not been implemented under the TinyOS framework, we only compare these two algorithms in a simulated environment. This performance comparison is shown in Figure 7.10. Ricci flow differs from GRIP's parametrization in that it is

Figure 7.10: CPU time comparison between GRIP's parametrization and Ricci flow. $l_1 = l_4 = l_7 = 2$, $l_2 = l_5 = l_8 = 3$, and $l_3 = l_6 = l_9 = 4$, where l_i is the number of holes in the i -th 2D WSN.

able to deal with all holes in a WSN simultaneously. However, as Ricci flow involves trigonometric function and square root function, and it usually requires many iterations to terminate, the overall performance is much worse.

7.7.2 Routing Performance

We evaluate the greedy routing performance in this section. The three routing protocols concerned in this paper are:

- (i) Pure Greedy Routing: with decision oracle (7.2). This is also the method used in [95]

Figure 7.11: Comparing three greedy routing schemes in 2D WSNs.

- (ii) Offline Tunable Greedy Routing: with decision oracle (7.4) and λ is set offline.
- (iii) Online Tunable Greedy Routing: with decision oracle (7.4) and λ is tuned online using Algorithm 11.

As proven in Sec. 7.5.1, all these routing schemes always succeed, and this is also what we observed in our experiments. Therefore, we compare them with respect to other metrics, namely, *load balancing*, *routing path stretch*, and *total energy consumption*. We assume that, as soon as a routing path goes through a node, the load (defined as energy consumption) taken by this node is increased by one unit. Therefore, given a certain (large) number of routing paths, the load balancing is verified by checking the maximum load taken by a node, whereas the total energy consumption is the sum of the load for all nodes.

The features of the WSNs we worked on are listed in Table 7.1. The first 8 cases are 2D WSNs, while the remaining are 3D WSNs (with almost the same features but different shapes). Before comparing these WSNs against the three metrics, we first use Net1 to illustrate the behavior of the three routing schemes in Figure 7.12. The three routing paths between two arbitrary nodes are drawn in both the network domain and embedding domain, with the harmonic field as the background. Though they all succeed while facing non-convex holes, pure greedy scheme is “attracted” by the inner boundaries,

Table 7.1: Simulated WSNs

s/n	# of nodes	# of holes	s/n	# of nodes	# of holes
1	9299	3	2	1089	2
3	10433	3	4	9654	4
5	8358	2	6	9146	4
7	8750	2	8	2628	2
9	99999	2	10	99996	2
11	99997	2	12	99995	2

Figure 7.12: Comparing different greedy routing schemes. While the pure greedy scheme (gray curve) sticks to the inner boundary, the two tunable schemes (purple and black curves) obviously tend to follow the isosurfaces, and the online tunable scheme (black curve) appears to achieve the most balanced performance.

while the online tunable scheme makes the best tradeoff between energy efficiency and load balancing.

The load balancing effects of the three routing schemes are shown in Figure 7.11(a) for the 2D WSNs and in Figure 7.13(a) for the 3D cases. The straightforward observation is that, while the two tunable routing schemes have rather similar performance, the maximum load of the pure greedy routing is often much higher (it can be up to 3.8 times of those of the other two schemes). We also compare the load balancing effects of the pure greedy and the online tunable schemes in Figure 7.14. Obviously, in both 2D and 3D WSNs, the online tunable scheme achieves almost perfect load balancing,

Figure 7.13: Comparing three greedy routing schemes in 3D WSNs.

whereas the pure greedy scheme generates many hot spots all over the network domain, in particular around the inner boundaries.

Of course, the load balancing effects of the two tunable schemes come at a cost: longer routing paths and hence higher total energy consumption for a whole WSN. However,

Figure 7.14: Illustration of the load balancing effect in two WSNs. Cut views are applied to reveal the loads inside the 3D WSN.

as shown in Figure 7.11(b)(c) (2D WSNs) and in Figure 7.13(b)(c) (3D WSNs), such a cost is worthy of trading for the significant improvement in load balancing. Especially for the online tunable scheme, only up to 30% of the cost is sacrificed in path length or in total energy consumption.

Chapter 8

Generalized Escher-Droste Effect

In this chapter, we use the planar conformal maps as image editing tools and bring it to the Escher-Droste effect, that greatly increases the flexibility of this effect. And inspired by the spirit, which regards the Droste effect and a conformal map from original image to basic element of tiling pattern, we also extend Escher-Droste effect to 3D space and develop a simple algorithm to create interesting visual effects from general 3D models.

8.1 Introduction

The Escher-Droste effect is a special type of recursive picture. An image exhibiting the Droste effect depicts a smaller version of itself in a place where a similar picture would realistically be expected to appear. This smaller version then depicts an even smaller version of itself in the same place, and so on. An image exhibiting the Droste effect is invariant under special transformations. Figure 8.2(a) is the image from a Dutch chocolate box made in 1904, where the whole image is iteratively shrunk to the cover of the box. Therefore, the image is invariant under the uniform scaling. Figure 8.2(b) is a computer-generated Droste image, in which the content is repeated along a spiral. Figure 8.2(c) is the famous M. C. Escher's *Print Gallery*, which demonstrates a much more

Figure 8.1: Generalized Escher-Droste effect. (a) *The Persistence of Memory*, Salvador Dali, 1931. (b) Droste effect on the clock. (c) Close-up view of the clock. (d) Another Droste effect.

complicated Droste effect. The mysterious mathematical structure of this image has been revealed in [16], which shows the image is invariant under a special combination of rotation and scaling, or a complex scaling. The complex scaling $z \rightarrow \sigma z$, where σ is a complex number, and $\sigma \in \mathbb{C}$ is an analytic function, which is conformal, namely, angle preserving. Locally, conformal map is just a scaling, so it preserves local shapes. This shape-preserving property is prominent in Escher's *Print Gallery*. In the following discussion, we use **Droste effect** to refer images invariant under conventional uniform scaling, which maps circles to circles, like Figure 8.2(a); and **Escher effect** for images under complex scaling, which maps circles to spirals, like Figure 8.2(b).

The self-invariant maps for conventional Escher-Droste images are limited to the combinations of complex scalings and translations. This requires inner frame and the outer frame only differs by rotation and scaling, which are rectangles or circles in most cases. This drawback greatly confines the application scope of the method in practice.

In this work, we generalize Escher-Droste effect, such that the self-invariant maps are

Figure 8.2: An image exhibiting the Droste effect depicts a smaller version of the image within itself in a recursive manner. (a) Dutch chocolate box in 1904. (b) A computer-generated Droste image. (Courtesy of webdesignerdepot.com) (c) Escher's *Print Gallery*. (d) The revealed structure shows that the visitor is both the guy looking at the picture and the guy in the picture.

arbitrary planar conformal maps, instead of just complex scalings and translations. The novel Escher-Droste effect greatly increases the application scope and the flexibility. For example, we convert the famous painting *The Persistence of Memory* by Salvador Dali to an Escher-Droste image (see Figure 8.1). The distorted surface of a clock is treated as the inner frame. A conformal map shrinks the whole image into the clock surface iteratively while the conventional approach can not do that due to the irregular boundary of the clock. The flexibility of our algorithm allows artists to release their creativity and imagination power to create fascinating Escher-Droste images.

The specific contributions of this chapter includes:

- Generalizing Escher-Droste effect, such that the self-invariant maps are general conformal maps, increasing the flexibility and the scope of the method.
- More generally, introducing planar conformal maps as image editing tools, which can locally preserve shapes, and globally deform the whole image. The method can be applied for general image warping applications.
- Extending Droste effect to 3D space and developing an algorithm to create 3D Escher-Droste effect from general 3D models.

8.2 Related Work on Escher-Droste Effect

Although Escher did not have a deep understanding of mathematics, his artistic vision incorporated a surprising mathematical consistency, that make it possible to re-produce his paintings using computer programs. [48] introduced the Escherization problem that aims to find an Escher-like tiling of the plane from tiles that resemble a user-supplied goal shape. They presented a solution for isohedral tilings. Later on, [49] generalized the algorithm to dihedral case that allows tiling with two distinct shapes. [116] developed an interactive program that allows the users to design the Escher-like tiles on sphere. [96] developed a mathematical theory which captures some of the underlying principles of Escher's impossible world. The physical models for Escher's impossible figures were created by Elber [23] and Lipson [67]. [81] presented an interactive system that allows the users to easily specify the edge visibility information that is used to define impossible figures. Inspired by how modeling artists construct physical 3D models to produce a valid 2D view of an impossible figure, [110] presented an optimization algorithm that allows an efficient computation and rendering new views of impossible figures at interactive rates.

The mathematical structure of Escher’s *Print Gallery* is thoroughly analyzed in the work of de Smit and Lenstra Jr. in [16], which lifts the transformation from images to their covering spaces by complex logarithm, and shows the lifted map is a simple complex scaling. Based on Smit and Lenstra’s idea, Leys developed a logarithmic image transformation algorithm to generate Droste effect on general images [63].

Figure 8.3: Escher-Droste effect of an office photo.

8.3 Traditional Escher-Droste Effect

8.3.1 Mathematical Structure for Escher-Droste Effect

We explain the fundamental concepts of Escher-Droste effect using an office photo as shown in Figure 8.3, which contains a rectangular photo frame in the center of the image. The frame has the same aspect ratio of the image.

Complex Scaling Invariant The images with Escher-Droste effects are invariant under a special complex scaling transformation $\phi_\sigma : \mathbb{C} \rightarrow \mathbb{C}$,

$$\phi_\sigma(z - z_f) = \sigma(z - z_f),$$

where z_f is the fixed point $\phi_\sigma(z_f) = z_f$. If σ is a real number, then ϕ_σ is a simple scaling transformation, as shown in frame (a), which maps concentric circles to concentric circles; if σ is a general complex number, then ϕ_σ maps spirals to spirals.

In details, Figure 8.3(a) is a rectangle D_0 with a boundary γ_0 on the complex plane $\mathbb{C}$. D_1 is a smaller rectangle inside D_0 with a boundary γ_1 . Furthermore, D_k is a rectangle contained in D_{k-1} with a boundary γ_k . We denote the annulus $A_k = D_k - D_{k+1}$. There exists a simple scaling $\phi : D_0 \rightarrow D_1$, $\phi(z) = sz, z \in \mathbb{C}$, where $0 < s < 1$ is a real number. The map ϕ shrinks D_0 to D_1 . In fact, ϕ maps D_k to D_{k+1} , thus $D_{k+1} = \phi(D_k) = \phi^k(D_0)$. Similarly, one can define $D_{-k} = \phi^{-k}(D_0)$. Then $D_{-\infty} = \mathbb{C}$. There exists a unique *fixed point* p , such that $\phi(p) = p$. In the current image, the fixed point is the origin.

The image with Escher effect in Figure 8.3(b) looks more complicated, but it is also invariant under the conformal transformation, $\phi_\sigma(z) = \sigma z$, where σ is a complex number. If σ is a real number, the frames look like concentric circles, otherwise, the frames look like spirals. There is a unique fixed point p invariant under ϕ_σ .

Torus Structure We use $\mathbb{C}^*$ to denote $\mathbb{C} - \{z_f\}$. The group generated by ϕ_σ is denoted as $\langle \sigma \rangle$. The quotient space $\mathbb{C}^* / \langle \sigma \rangle$ is a torus.

Intuitively, in Figure 8.3(a), the scaling transformation group is $\langle \sigma \rangle$, where σ is a real number, $0 < \sigma < 1$. If we glue the two boundaries γ_k, γ_{k+1} of an annulus A_k by ϕ_s , then A_k becomes a torus.

The complex exponential map $\exp : \mathbb{C} \rightarrow \mathbb{C}^* / \langle s \rangle$ induces a surjective conformal map. Therefore, $\mathbb{C}$ is the universal covering space of $\mathbb{C}^* / \langle s \rangle$. The kernel $L_\sigma = 2\pi i\mathbb{Z} + \log \sigma \mathbb{Z}$, which is a lattice can be treated as the fundamental group of $\mathbb{C}^* / \langle \sigma \rangle$. The red rectangle in frame (c) shows one fundamental domain of the torus.

Map Between Tori Given two images with Escher-Dorste effect, suppose their invariant transformation are $\langle \sigma_k \rangle$, the corresponding fixed points are z_f^k , $k = 1, 2$, we would like to find a conformal map between the tori $\phi : \mathbb{C}_1^* / \langle \sigma_1 \rangle \rightarrow \mathbb{C}_2^* / \langle \sigma_2 \rangle$, where $\mathbb{C}_k^*$ is $\mathbb{C} - \{z_f^k\}$. Instead of considering the map between the tori, we lift it to the universal covering space. Define $\tau_k : \mathbb{C}_k^* / \langle \sigma_k \rangle \rightarrow \mathbb{C}$ as the complex logarithm $\tau_k = \log(z - z_f^k)$. Then the lift of ϕ on $\mathbb{C}$ is simply a complex scaling $z \rightarrow \beta z$, which maps the lattice of the first torus L_{σ_1} to the lattice of the second torus L_{σ_2} , namely

$$\beta \log \sigma_1 \mathbb{Z} + \beta 2\pi i \mathbb{Z} = \log \sigma_2 \mathbb{Z} + 2\pi i \mathbb{Z}.$$

8.3.2 Conventional Escher-Droste Algorithm

Conventionally, the input image has an inner frame, which has the same aspect ratio of the image itself. We denote the exterior frame of the image itself as D_0 , the inner frame as D_1 .

8.3.2.1 Compute the Invariant Complex Scaling

Let z_0 be the center of D_0 , z_1 be the center of D_1 , s be the scaling between D_1 and D_0 , θ be the offset angle between the two frames. Then the complex scaling coefficient is given by $\sigma = se^{i\theta}$, the fixed point satisfies the equation $\sigma(z_0 - z_f) = z_1 - z_f$, therefore $z_f = \frac{\sigma z_0 - z_1}{\sigma - 1}$. The invariant transformation ϕ_σ is $z \rightarrow \sigma(z - z_f) + z_f$. The lattice is generated by the two complex numbers $L_\sigma = \{\log \sigma, 2\pi i\}$.

8.3.2.2 Compute the Lifted Map

We compute the coefficient β of the lifted map in the following two steps.

Compute Lattice Generators We compute a different set of generators of the lattice.

We choose an integer matrix

$$T := \begin{pmatrix} a & b \\ c & d \end{pmatrix}, a, b, d, c \in \mathbb{Z},$$

such that $ad - bc > 0$, which is called the transformation matrix. Define a lattice transformation induced by T ,

$$TL_\sigma = (a \log \sigma + b2\pi i)F + (c \log \sigma + d2\pi i)F.$$

By the construction rule, TL_σ is a subset of L_σ . If T is invertible, then $TL_\sigma = L_\sigma$. In Figure 8.3(c), the red rectangle shows one period of the lattice L_σ , the blue parallelogram shows one period of the lattice TL_σ , it is obvious that the two lattices are identical. In this case, the transformation matrix is given by

$$T = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

Figure 8.4 shows the Escher effects induced by different transformation matrices.

Normalize the Lattice We normalize the transformed lattice to make the second generator $c \log \sigma + d2\pi i$ to be $2\pi i$,

$$\beta = \frac{2\pi i}{c \log \sigma + d2\pi i}.$$

Therefore the map ϕ is given in the following formula

$$\phi(z) = \exp(\beta \log(z - z_f)) + z_f = (z - z_f)^\beta + z_f.$$

In Figure 8.3(d), the blue parallelogram is normalized, such that the longer edge is align with the imaginary axis, with length 2π .

(a) $T = \text{diag}(2, 1)$ (b) $T = \text{diag}(1, 2)$

Figure 8.4: Escher effects with different transformation matrices.

(a) Input Mesh (b) ω_1 (c) ω_2 (d) ω (e) Canonical Annulus

Figure 8.5: Pipeline for computing discrete conformal map by holomorphic 1-form.

8.4 Generalized Escher-Droste Algorithm

We aim at generalizing the invariant transformation from simple complex scaling function $z \rightarrow \sigma z$ to general conformal maps, such that the interior frame can be an *arbitrarily irregular shape*.

8.4.1 Droste Effect

Select Region of Interest As shown in Figure 8.6, the first step is to choose the region of interest by specifying two loops on the image, γ_0 , γ_1 , where γ_1 is inside γ_0 . The boundary curves γ_0 and γ_1 are represented as chain code. The region bounded by γ_0 is D_0 , the region bounded by γ_1 is D_1 . Taking all the pixels on the user-specified curves as constraints, our program computes a conforming constrained Delaunay triangulation using [102] to convert the image to a triangle mesh (see Figure 8.6). After selecting the region of interest, we get a topological annulus $A_0 = D_0 - D_1$.

Figure 8.6: Given the input image (*Sting Caused by the Flight of a Bee*, Salvador Dali, 1944), the user specifies the region of interest by sketching two closed curves (in yellow). The inner curve is the mouth of the left tiger. Then, our program automatically generates a conforming constrained Delaunay triangulation.

Compute Conformal Maps Compute a discrete Riemann map $\phi_0 : D_0 \rightarrow \mathbb{D}$ and $\phi_1 : D_1 \rightarrow \mathbb{D}$ using the Ricci flow and holomorphic 1-form, where $\mathbb{D}$ is the unit disk (see Figure 8.7 (a) and (b)). If it is necessary, we compose a Möbius transformation with ϕ_1 to fit the boundary constraints. The conformal map which transforms D_0 to D_1 is given by $\phi = \phi_1^{-1} \circ \phi_0$. Figure 8.8 (b) shows the conformal map ϕ .

Generate Droste Effect using Texture Map The image with Droste effect can be constructed iteratively by ϕ . Basically,

$$D_0 = A_0 \cup A_1 \cup A_2 \cdots A_n \cdots,$$

where $A_k = \phi^k(A_0)$. By applying ϕ iteratively onto A_0 , and merging the images together, we can generate the desired Droste effect.

In practice, this can be easily achieved by texture map. We use the original image on

Figure 8.7: Conformal maps. The first row shows the triangulation of the region of interest in the original image, and the second row shows the conformal parameterization results.

Figure 8.8: Droste effect for region of interest with irregular boundaries. (a) The Droste effect. (b) The triangulation of the repeated region in the tiger mouth. (c) The close-up view of the tiger mouth.

D_0 as the initial texture T_0 . The texture coordinates are set on the triangle corners of mesh D_0 , instead of the vertices. Each corner is represented as a tuple (f, v) , where f is a face, v is a vertex. For each corner (f, v) in A_0 , its UV coordinates are equal to the coordinates of the vertex v ; for each corner (f, v) in D_1 , its UV coordinates is given by $\phi^{-1}(v)$. The texture map result is denoted as T_1 . We use T_1 to update T_0 to generate another texture map image T_2 . By repeating this procedure, we can generate T_n from T_{n-1} using texture map. The Droste effect is given by T_∞ , which is invariant under the conformal map ϕ . Because ϕ is a compressive map, there exists a unique fixed point z_f , $z_f \in \bigcap_{k=0}^{\infty} \phi^k(D_0)$. Figure 8.8 shows the final Droste effect.

8.4.2 Escher Effect

The Escher effect is generated by converting irregular invariant maps to simple complex scalings. Similar to Sec. 8.4.1, we use $A_0 = D_0 - D_1$ to denote the topological annulus. Then we conformally map A_0 to a canonical annulus using [125], such that the exterior boundary γ_0 is mapped to the unit circle, the interior boundary γ_1 is mapped to a concentric circle with radius $0 < s < 1$. Figure 8.7(c) shows the result.

Let us still denote the image of the conformal map as A_0 . Define $\sigma = se^{i\theta}$, where θ is chosen freely, then $\phi_\sigma(z) = \sigma z$ is the complex scaling with the fixed point 0. We scale A_0 by σ to fill the complex plane, $\mathbb{C}^* := \bigcup_{k=-\infty}^{\infty} \phi_\sigma^k(A_0)$, which is the image with Escher effect.

In the design process, it is very common that the fixed point of the complex scaling is at the general location instead of the origin (see Figure 8.9). To obtain this, we can use Möbius map to transform the canonical annulus A_0 , such that the inner circle is mapped to another circle with center z_1 and radius r_1 . Then $s = r_1$, the fixed point is $z_f = \frac{z_1}{1-s}$. Then the invariant transformation is given by $\phi_\sigma(z) = s(z - z_f) + z_f$. The images of $\phi_\sigma^k(A_0)$ tiles the whole plane.

Note that $\mathbb{C}^*/\langle\sigma\rangle$ is a torus. By using the conventional algorithm described in Section 8.3, we can generate images with Escher effect (see Figure 8.10).

8.5 3D Droste Effect

The Escher-Droste effect for 2D image can be viewed as a *conformal map* from the region of interest in the input image to the tiling like annulus, which can be further conformally mapped to a rectangle and then be seamlessly tessellated in the complex

Figure 8.9: Möbius transformation is able to map the fixed point from the origin (see (a)) to arbitrary location (see (b)).

Figure 8.10: Escher effects with (row 2) and without (row 1) Möbius transformation.

plane $\mathbb{C}$. However, due to the lack of conformal map in $\mathbb{R}^3$, we have to abandon the conformal property in the 3D Escher-Droste effect.

Let M denote a closed 3D mesh and $S(r, \mathbf{c})$ be a sphere of radius r centered at $\mathbf{c}$. Translate M so that the center of interest (the fixed point of the 3D Escher-Droste effect) coincides with the origin $\mathbf{O}$. We enclose M into a sphere $S_0 = S(r_0, \mathbf{O})$, and place another smaller sphere $S_1 = S(r_1, \mathbf{O})$ inside M , i.e., the input model M is in the layered space of $L_0 = S_0 - S_1$ (see Figure 8.11(a)).

For every point $p = (x, y, z) \in L_0$ inside the layered volume L_0 , we compute its spherical

coordinate (r, ϕ, θ)

$$x = r \cos \phi \sin \theta, y = r \sin \phi \cos \theta, z = r \cos \theta,$$

where $r \in [r_1, r_0]$, $\phi \in [0, 2\pi)$ and $\theta \in [0, \pi]$. Using spherical coordinates, we map the layered space L_0 into the cuboid volume C_0 (see Figure 8.11(b)). Note that top and bottom faces of the cuboid volume C_0 can be glued seamlessly.

Let $R(\text{axis}, \alpha)$ be a 3D rotation of angle α around the principal axis and $T(\cdot, \cdot, \cdot)$ the 3D translation. Then, we rotate the cuboid volume along the θ axis with the user-specified angle δ (see Figure 8.11(c)).

Next, we duplicate the rotated cuboid and glue them with the $\theta - \phi$ faces, which forms a cuboid of infinite length (along the ϕ axis, see Figure 8.11(d)), i.e.,

$$\bigcup_{k=0}^{\infty} R(\phi - \text{axis}, \delta) T(0, 2k\pi, 0) C_0$$

Finally, by evaluating the surface along the infinite cuboid and converting the spherical coordinates back to the Euclidean coordinates, we can get the 3D Droste effect (see Figure 8.11(e)).

8.6 Experimental Results

We implemented our 2D and 3D Escher-Droste algorithms in C++ and conducted experiments on a PC with 2.33GHz Dual processor and 4GB RAM. For the 2D case, the bottle neck is the conformal map step, which is equivalent to solve a sparse linear system. We adopt the TAUCS solver¹ in our current implementation. The performance of conformal map depends on the complexity of the input mesh. Taking *The Persistence of*

¹<http://www.tau.ac.il/~stoledo/taucs/>

Figure 8.11: The algorithmic pipeline for 3D Escher-Droste effect. (a) Enclose the input mesh M into a layered space of concentric spheres. (b) Map the layered space to a cuboid volume by using spherical coordinates. (c) Rotate the cuboid volume around the θ axis. (d) The rotated cuboid volume can be duplicated and then glued along the ϕ -axis, which forms a cuboid of infinite length. For better visualization, only three copies are rendered in this figure. (e) By evaluating the surface along the infinite cuboid and converting the spherical coordinates back to the Euclidean coordinates, we can get the 3D Escher-Droste effect.

Memory as an example (Figure 8.1), the triangle mesh has about $50k$ vertices and $130k$ faces. The conformal map takes about 30 seconds. Since the 3D case does not require any numerical solver, all the computations are done in real-time.

Figure 8.13 demonstrates the Escher-Droste effect of a Chinese ink and wash painting: *Nine Goldfish*, where the inner curve γ_1 is the boundary of a stone (see (a)). By removing its interior part, we map the remaining image to an annulus (see (b)), which leads the Droste effect (see (c)). The logarithm of (c) is shown in (d). Various Droste effects can be obtained by defining complex exponential maps on different periods. There are several examples, such as 1) clockwise: the clockwise exponential map of the fundamental domain $\Gamma(a, a+b)$ is computed (see (e)); 2) counter-clockwise: the counter-clockwise exponential map of the fundamental domain $\Gamma(a, a+b)$ is computed (see (f)); 3) double layer: the whole circle domain in (b) is reflected into the inner hole, then the resulting circle domain is a covering space of a topological torus. It is a double covering along the original inner circle and exterior circle. The exterior boundary becomes a smaller inner circle. The fundamental domain $\Gamma(2a, b)$ is selected for clockwise exponential

Figure 8.12: 3D Droste effect.

map (see (g)); 4) double period: the fundamental domain $\Gamma(a, a + 2b)$ is selected for the clockwise exponential map (see (h)).

Figure 8.14 shows our method applied to *The Golden Mask of King Tutankhamun*, where the inner curve γ_1 is the boundary of lips. Due to the degeneracy of the bounded region, the conventional Escher-Droste algorithm does not work. Our algorithm, in contrast, is very flexible and supports γ_0 and γ_1 with arbitrary shapes (as long as they do not self-intersect).

Figure 8.15 shows Vincent van Gogh's self-portrait, where we use Möbius transformation to adjust the position of the inner frame. Figure 8.12 demonstrates Droste effects for 3D models.

Figure 8.13: Escher-Droste effect of *Nine Goldfish*, where γ_1 is the boundary of a stone. The Droste effect is shown in (c) and various Escher effects are shown in (e)-(h).

Figure 8.14: Escher effect of *King Tutankhamun's Golden Mask*, where the inner curve γ_1 is the boundary of lips. Note that the conventional Escher-Droste algorithm does not work for this case, since the region bounded by the inner curve γ_1 is degenerate.

Figure 8.15: Given Vincent van Gogh's self-portrait, we mark his ear as the region of interest (see (a)). The Droste effect is shown in (b). And various Escher effects are shown in (c)-(f).

Figure 8.16: Escher-Droste effect of *The Scream*, Edvard Munch, 1893. The irregular hole is cut along the boundary of open mouth.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

In this thesis, we make the encouraging progress for conformal map in both theorem and applications, we first propose the way to compute the extremal quasi-conformal map and then we apply the traditional conformal map and the extremal quasi-conformal map to various of different applications like medical image, sensor network routing and Escher-Droste effect.

In chapter 4, we proposed a simple yet effective technique to compute Teichmüller map on surfaces of non-trivial topology. Taking two surfaces of the same topology and an initial map (not necessarily a bijection), our method iteratively improves the map quality. For each iteration, we compute the map on the local chart and then diffuse it through the whole mesh. Our method is linear, local, easy to implement and has a very natural parallel structure. Furthermore, our method does not require the global parametrization. These features distinguish our method from the existing techniques [109, 70], which are either non-linear or require global parametrization. The experimental results on both synthetic and real-world datasets of various topological types demonstrate the efficacy of our method.

In Chapter 5, we present a rigorous algorithm to register brainstem surfaces for the disease analysis of Adolescent Idiopathic Scoliosis. The basic idea is to extract four consistent features, which describe the global geometry of the brainstem, to guide the surface registration. Using the Ricci Flow method, brainstem surfaces are parameterized conformally onto quadrilaterally-faced hexahedron, which naturally induces the feature landmark-matching brainstem registration. Our registration algorithm provides a registration between brainstem surfaces with exact matching of the extracted feature curves. A shape energy can then be defined to measure the local shape variations between different brainstem surfaces. Experiments on real brainstem surfaces of 15 normal subjects and 15 AIS patients show the efficacy of our proposed algorithms to register and detect local shape differences between different brainstem surfaces. The proposed algorithm can potentially be used to study factors that influence shape changes of brainstems in AIS.

In Chapter 6, we presents an effective and practical solution to register the genus-3 vestibular systems. Our method first extracts six geodesic landmarks for the VS, which are stable, intrinsic, and insensitive to the VSs resolution and tessellation. Moreover, they are highly consistent regardless of the disease and normal groups. The detected geodesic landmarks partition the VS into three patches, a topological annulus and two topological disks. For each pair of patches of the AIS subject and the control, we compute a bijective map using the holomorphic 1-form and extremal quasi-conformal map techniques. With a carefully designed boundary condition, the three individual maps can be glued in a seamless manner so that the resulting registration is a homeomorphism with exact landmark matching. Our method is robust, efficient and fully automatic, which takes only a few seconds to complete on a PC. The experimental results demonstrate the efficacy of our method for registration of the vestibular system.

In Chapter 7, we have presented the GRIP (Greedy Routing through dIstributed Parametrization), a greedy routing algorithm with guaranteed delivery in 2D/3D WSNs. The key part of the greedy routing algorithm is the distributed algorithm we proposed, it is modified from the algorithm to compute the holomorphic 1-form, where we change the second step of computing the Hodge star to trace along the integral curve first. Based on this novel distributed parametrization algorithm, we can embed a complicated 2D/3D network domain to a simple parametric domain, in which each hole is mapped to a circle or convex domain. This property favors greedy routing and allows it to have guaranteed delivery.

In Chapter 8, we generalize classical Escher-Droste effect, such that the self-invariant maps of the images include general conformal maps. We introduce the planar conformal maps as image editing tools, which can locally preserve shapes, and globally deform the whole image. Compared to the conventional techniques, our new algorithm allows the region of interest having arbitrary shape. With Möbius transformation, the user can set the fixed point at arbitrary location inside. These features greatly increase the flexibility and the scope of the method. We also extend the Escher-Droste effect to 3D space and develop a simple algorithm to create interesting visual effects from general 3D models. We demonstrate our method to a wide range of 2D images and 3D models. The experimental results demonstrate that the generalized Escher-Droste algorithm can generate visually pleasing visual effects.

9.2 Future Work

Although the conformal map is powerful and produces impressive results in a wide range of applications, it is far from perfect. There are several future works to explore for the conformal map.

9.2.1 Conformal Map on Broken and Noisy Mesh

Figure 9.1: The face captured by 3D camera will have noise and holes.

In real life, most 3D surface meshes captured by the 3D camera are not as clean as the model we have tested, as demonstrated in Figure 9.1. On this noisy and broken surface, discrete mathematical definitions such as the Gaussian curvature are hard to define (if the mesh is neither triangle soup nor non-manifold). Moreover, these surfaces have a lot of topological noise, such as small holes and topological shortcuts. Hence, calculating the conformal map for such surfaces is a real challenge. The traditional method to process these models requires a cumbersome way to clean and repair the model, which is quite undesirable.

Some recent research efforts have aimed to calculate the meaningful geometry information on broken meshes. Campen and Kobbelt [77] pioneer the algorithm to compute meaningful geodesic distances on broken meshes while Lipman [66] define the biharmonic distance which is globally smooth, shape-aware, isometry invariant, and insensitive to noise and small topology changes. Although some work has already been done, how to compute the conformal map remains a persistent problem. However, with a basic

tool like geodesic distance field on a broken mesh, we can continue to move the conformal map forward on noisy surfaces. This work can be very practical and meaningful and will have a broad range of applications.

9.2.2 Distortion Control

Conformal map can minimize the angle distortion; however, we also need to consider the area distortion in some applications such as picture morphing.

Figure 9.2: Picture morphing uses conformal map, although conformal map can keep local feature isometry, the area distortion still causes serious visible defects.

Controlling the compromise between angle distortion and the area distortion remains a problem. One popular way to do that is to put some singularities for conformal parametrization, whereas looking for such singularities is still a challenge. One of the heuristic ways is to choose a place with high Gauss curvature to put the singularity, since for most cases, the place with high Gauss curvature has higher area distortion. However,

there are no theoretic guarantees for this method. Finding the singularities to control the compromise between angle distortion and area distortion will be an interesting problem.

Another promising approach with theoretic guarantee for this problem is that we define the maximal bound of the area distortion first, then we compute an extremal quasi-conformal map under the bounded area distortion, which is similar to the extremal quasi-conformal map we have computed for the landmark registration on high genus model.

9.2.3 More Applications for Medical Images

The conformal map has many other applications for medical images due to its desirable properties. We can see that the algorithm which is proposed in Chapter 5 and Chapter 6 can be naturally extended to the registration of other anatomical structures, such as hippocampus, cerebellum and the brain vessel.

Conformal registration for the Willis's circle

Figure 9.3: The Willis's circle (courtesy of Wikipedia) and its MRI model.

The Willis's circle is one of the important brain vessel which supplies blood to the brain and surrounding structures. It has a very special structure, and its main component is the genus one circle, with some small branches connecting to the main circle, as shown in Figure 9.3. We can detect the salient feature line on the main circle first and match the

respective vessel by the extremal quasi-conformal map with exact landmark matching. We believe that the resulting registration could provide valuable analysis for medical research on the diseases about the brain vessel.

References

- [1] L. V. Ahlfors. *Conformal invariants: topics in geometric function theory*. American Mathematical Soc., 1973. 13
- [2] M. Alexa. Merging polyhedral shapes with scattered features. In *The Visual Computer*, volume 16, pages 26–37, 2000. 19
- [3] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. Conformal geometry and brain flattening. In *MICCAI*, pages 271–278, 1999. 14
- [4] R. T. B. Fischl, M.I. Sereno and A. Dale. High-resolution intersubject averaging and a coordinate system for the cortical surface. In *Human Brain Mapping*, 8, pages 272–284, 1999. 68
- [5] M. F. Beg, M. I. Miller, A. Trounev, and L. Younes. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, 2005. 91
- [6] A. Belyaev, E. Anoshkina, and T. Kunil. Ridges: ravines, and singularities. In *Topological Modeling for Visualization*, 13, pages 375–383, 1997. 69
- [7] H. Birkholz. Shape-preserving parametrization of genus 0 surfaces. In *Proceedings of Winter Conference on Computer Graphics (WSCG)*, 2004. 20
- [8] A. I. Bobenko and B. A. Sprinborn. Variational principles for circle patterns and koebe's theorem. *Transactions of the American Mathematical Society*, 356(2):659–689, 2003. 22
- [9] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. In *Proc. of the 3rd ACM DialM*, 1999. 94, 96
- [10] P. Bowers and M. Hurdal. Planar conformal mappings of piecewise flat surfaces. In *Visualization and Mathematics III*, pages 3–34, 2003. 18, 22
- [11] P. L. Bowers and K. Stephenson. Uniformizing dessins and belyi maps via circle packing. 1997. 18
- [12] N. N. Byl and J. M. Gray. Complex balance reactions in different sensory conditions: adolescents with and without idiopathic scoliosis. *Journal of Orthopaedic Research*, 11(2):215–227, 1993. 79

REFERENCES

- [13] B. Chow and F. Luo. Combinatorial ricci flows on surfaces. In *JDG*, 63(1), pages 97–129, 2003. 22, 73
- [14] C. Collins and K. Stephenson. A circle packing algorithm. In *Computational Geometry: Theory and Applications*, pages 233–256, 2003. 18, 22
- [15] P. Daripa. A fast algorithm to solve the beltrami equation with applications to quasiconformal mappings. *J. Comput. Phys.*, 106(2):355–365, 1993. 23
- [16] B. de Smit and H. W. L. Jr. The mathematical structure of escher’s print gallery. *Notices of the AMS*, 50(4). 124, 127
- [17] Y. C. de Verdiere. On a new graph invariant and a criterion for planarity. In *Graph structure Theory/Contemporary Mathematics*, pages 137–147, 1990. 19
- [18] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. *ACM Transactions on Graphics*, pages 45:1–45:9, 2008. 81
- [19] R. J. Duffin. Basic properties of discrete analytic functions. *Duke Math. J.*, 23(2):189–370, 1956. 14
- [20] R. J. Duffin. Distributed and lumped networks. *J. Math. Mech.*, pages 793–826, 1959. 14
- [21] S. Durocher, D. Kirkpatrick, and L. Narayanan. Guaranteed Delivery in Three-Dimensional Ad Hoc Wireless Networks. In *Proc. of the 9th ICDCN*, 2008. 94, 96
- [22] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *ACM Trans. Graph*, pages 173–182, 1995. 14, 16
- [23] G. Elber. Escher for real. <http://www.cs.technion.ac.il/~gershon/EscherForReal>, 2002. 126
- [24] Q. Fang, J. Gao, and L. Guibas. Locating and Bypassing Holes in Sensor Networks. In *Proc. of the 23rd IEEE INFOCOM*, 2004. 94, 113
- [25] H. M. Farkas and I. Kra. *Riemann surfaces*. Graduate Texts in Mathematics. 14
- [26] M. Floater. Parameterization and smooth approximation of surface triangulations. In *Computer Aided Geometric Design*, volume 14, pages 231–250, 1997. 14, 16
- [27] M. S. Floater. Mean value coordinates. In *Computer Aided Geometric Design*, volume 20 of 1, pages 19–27, 2003. 16
- [28] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, pages 157–186. Springer, 2005. x, 13, 17
- [29] R. Flury and R. Wattenhofer. Randomized 3D Geographic Routing. In *Proc. of the 27th IEEE INFOCOM*, 2008. 94, 96

REFERENCES

- [30] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric Spanner for Routing in Mobile Networks. In *Proc. of the 14th ACM MobiHoc*, 2003. 99, 100, 113
- [31] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *ACM SIGGRAPH*, pages 358–364, 2003. 18, 19
- [32] X. Gu, Y. Wang, T. Chan, P. Thompson, and S.-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. In *TMI*, 23(8), pages 949–958, 2004. x, xi, 18, 19, 20, 21, 67, 91
- [33] X. Gu and S.-T. Yau. Computing conformal structures of surfaces. In *Commun. Inform. Syst.*, volume 2, pages 121–146, 2002. 14, 15, 19, 20
- [34] X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Global Conformal Parameterization*, pages 127–137, 2003. 21, 85
- [35] X. Gu and S.-T. Yau. *Computational conformal geometry*. Advanced lectures in mathematics. International Press, 2008. x, 2, 3, 5, 6, 7, 8, 25
- [36] R. Guo. Local rigidity of inversive distance circle packing. *Transactions of the American Mathematical Society*, 363:4757–4776, 2011. 22
- [37] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. In *Trans. Visual. Comput. Graphics*, volume 6, page 181189, 2000. 14, 18, 19
- [38] R. S. Hamilton. Three manifolds with positive ricci curvature. *Journal of Differential Geometry*, 17:255–306, 1982. 22, 36
- [39] R. S. Hamilton. The ricci flow on surfaces. *Mathematics and general relativity*, 71:237–262, 1988. 22
- [40] P. Henrici. *Applide and Computational Complex Analysis, Discrete Fourier Analysis, Cauchy Integrals, Construction of Conformal Maps, Univalent Functions*, volume 3. Wiley-Interscience, 1993. 111
- [41] K. Hormann, G. Greiner, and S. Campagna. Hierarchical parametrization of triangulated surfaces. In *Vision, Modeling, and Visualization*, pages 219–226, 1999. 17
- [42] M. Hurdal, P. Bowers, K. Stephenson, D. Sumners, K. Rehm, K. Schaper, and D. Rottenberg. Quasiconformally flat mapping the human cerebellum. In *MICCAI*, pages 279–286, 1999. 18
- [43] M. K. Hurdal and K. Stephenson. Discrete conformal methods for cortical brain flattening. In *NeuroImage*, 45, pages 86–98, 2009. 67
- [44] M. V. J. Glaunès and M. Miller. Landmark matching via large deformation diffeomorphisms on the sphere. In *J. Maths. Imaging and Vision*, 20, pages 179–200, 2004. 67
- [45] M. M. D. Joseph S. B. Mitchell and C. H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, 1987. 54

REFERENCES

- [46] S. C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Transaction on Image Processing*, 9(8):1357–1370, 2000. 91
- [47] G. G. K. Hormann. Mips: An efficient global parameterization method. In *Curve and Surface Design*, 2000. 17
- [48] C. S. Kaplan and D. H. Salesin. Escherization. In *SIGGRAPH '00*, pages 499–510, 2000. 126
- [49] C. S. Kaplan and D. H. Salesin. Dihedral escherization. In *GI '04*, pages 255–262, 2004. 126
- [50] B. Karp and H. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. of the 6th ACM MobiCom*, 2000. 94, 96
- [51] L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.*, 25(2):412–438, 2006. 15
- [52] S. P. Kharevych Liliya, Springborn Boris. Discrete conformal mappings via circle patterns. In *ACM Transactions on Graphics*, 25(2), 2006. 18, 22
- [53] R. Kleinberg. Geographic Routing Using Hyperbolic Space. In *Proc. of the 26th IEEE INFOCOM*, 2007. 94, 95, 96, 116
- [54] L. P. Kobbelt, J. Vorsatz, U. Labisk, and H.-P. Seidel. A shrink-wrapping approach to remeshing polygonal surfaces. In *Proceedings of Eurographics*, pages 358–364, 2003. 19
- [55] G. Kos and T. Varady. Parameterizing complex triangular meshes. In *Curve and Surface Design*, pages 265–274, 2003. 16
- [56] R. Lai, Z. Wen, W. Yin, X. Gu, and L. Lui. Fast and accurate algorithms for harmonic energy minimization on genus-0 surfaces. *Journal of Scientific Computing*, 2013. 20
- [57] Y. Lee, H. S. Kim, and S. Lee. Mesh parameterization with a virtual boundary. In *Computers and Graphics*, volume 26(5), pages 677–686, 2002. 16
- [58] T. Leighton and A. Moitra. Some Results on Greedy Embeddings in Metric Spaces. In *Proc. of the 49th IEEE FOCS*, 2008. 96
- [59] A. Leow, C.L.Yu, S. Lee, S. Huang, H. Protas, R. Nicolson, K.M.Hayashi, A. Toga, and P. Thompson. Brain structural mapping using a novel hybrid implicit/explicit framework based on the level-set method. In *NeuroImage*, 24(3), pages 910–927, 2005. 67, 68
- [60] P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *Proc. of the 1st ACM SenSys*, 2003. 117
- [61] B. Levy and J.-L. Mallet. Non-distorted texture mapping for sheared triangulated meshes. In *ACM SIGGRAPH*, pages 343–352, 1998. 14
- [62] B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Comput. Graphics (Proc. SIGGRAPH 02)*, 2002. 14, 16

REFERENCES

- [63] J. Leys. The mathematics behind the droste effect. <http://www.josleys.com/>, 2008. 127
- [64] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris. A Scalable Location Service for Geographic Ad-Hoc Routing. In *Proc. of the 7th ACM MobiCom*, 2001. 100
- [65] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, and H. Qin. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Trans. Vis. Comput. Graph.*, 14(4):805–819, 2008. 23, 64
- [66] Y. Lipman, R. Rustamov, and T. Funkhouser. Biharmonic distance. *ACM Transactions on Graphics*, 29(3), June 2010. 144
- [67] A. Lipson. Andrew Lipson’s LEGO page. <http://www.andrewlipson.com/lego.htm>, 2002. 126
- [68] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. J. Gortler. A local/global approach to mesh parameterization. *Computer Graphics Forum*, 27(5):1495–1504, 2008. 17
- [69] L. Lovasz and A. Schrijver. On the null space of a colin de verdere matrix. In *Annales de l’Institut Fourier*, volume 40, pages 1017–1026, 1999. 19
- [70] L. M. Lui, K. C. Lam, S.-T. Yau, and X. Gu. Teichmüller extremal mapping and its applications to landmark matching registration. *CoRR(2012)*, abs/1211.2569, 2012. 15, 24, 48, 51, 52, 57, 63, 64, 141
- [71] L. M. Lui, T. W. Wong, P. Thompson, T. Chan, X. Gu, and S.-T. Yau. Shape-based diffeomorphic registration on hippocampal surfaces using beltrami holomorphic flow. In *Proceedings of MICCAI ’10: Part II*, pages 323–330, 2010. 91
- [72] F. Luo. Combinatorial yamabe flow on surfaces. *Commun. Contemp. Math.*, 6:765C780, 2004. 22
- [73] P. A. M. Desbrun, M. Meyer. Intrinsic parametrizations of surface meshes. In *Proc. Eurographics*, pages 210–218, 2002. 16
- [74] F. L. M. Jin, J. Kim and X. Gu. Discrete surface ricci flow. In *TVCG*, 14, pages 1030–1043, 2008. 70
- [75] F. L. X. G. M. Jin, J. Kim. Discrete surface ricci flow: theory and applications. In *Mathematics of Surfaces*, XII, 2007. 15, 22, 56
- [76] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *ACM SIGGRAPH*, pages 27–34, 1993. 14
- [77] L. K. Marcel Campen. Walking on broken mesh: Defect-tolerant geodesic distances and parameterizations. In *Comput. Graph. Forum*, volume 30(2), page 623632, 2011. 144
- [78] N. Miller. Cause and natural history of adolescent idiopathic scoliosis. In *Orthop Clin North Am.*, 30, pages 343–352, 1999. 65

REFERENCES

- [79] B. C. V. N. A. Lord, J. Ho and S. Eisenschenk. Simultaneous registration and parcellation of bilateral hippocampal surface pairs for local asymmetry quantification. In *TMI*, 26, pages 471–478, 2007. 68
- [80] S. R. O. Lyttelton, M. Boucher and A. Evans. An unbiased iterative group registration template for cortical surface analysis. In *NeuroImage*, 34, pages 1535–1544, 2007. 68
- [81] S. Owada and J. Fujiki. Dynafusion: a modeling system for interactive impossible objects. In *NPAR '08*, pages 65–68, 2008. 126
- [82] C. Papadimitriou and D. Ratajczak. On a Conjecture Related to Geometric Routing. *Elsevier Theoretical Computer Science*, 344(1):3–14, 2005. 94, 96
- [83] G. Perelman. Finite extinction time for the solutions to the ricci flow on certain three-manifolds. In *Tech. Rep. arXiv.org*, 2003. 22
- [84] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. In *Experimental Mathematics*, volume 2 of *I*, pages 15–36, 1993. 14
- [85] E. Praun and H. Hoppe. Spherical parameterization and remeshing. In *ACM SIGGRAPH*, pages 340–350, 2003. 18
- [86] A. Qiu, L. Youne, and M. I. Miller. Principal component based diffeomorphic surface mapping. *IEEE Trans. Med. Imaging*, 31(2):302–311, 2012. 91
- [87] T. Rad. Aufgabe. In *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 35, pages 41–49, 1926. 73
- [88] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic Routing without Location Informtion. In *Proc. of the 9th ACM MobiCom*, 2003. 96
- [89] N. Ray and B. Levy. Hierarchical least squares conformal maps. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 263?70, 2003. 17
- [90] B. L. Rhaleb Zayer and H.-P. Seidel. Linear angle based parameterization. In *In Proc. ACM/Eurographics Symposium on Geometry Processing*, 2007. 14, 18
- [91] B. Rodin and D. Sullivan. The convergence of circle packings to the riemann mapping. *Journal of Differential Geometry*, 26:349C360, 1987. 22
- [92] S. Saba, I. Yavneh, C. Gotsman, and A. Sheffer. Practical spherical embedding of manifold triangle meshes. In *Proceedings of Shape Modeling International*, 2005. 18
- [93] T. Sahlstrand and J. Lidstrom. Equilibrium factors as predictors of the prognosis in adolescent idiopathic scoliosis. In *Clin Orthop Relat Res.*, 152, pages 232–245, 1980. 65
- [94] P. Sander, S. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. In *Eurographics Workshop on Rendering*, pages 87–100, 2002. 18

REFERENCES

- [95] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. Gu. Greedy Routing with Guaranteed Delivery Using Ricci Flows. In *Proc. of the 8th ACM/IEEE IPSN*, 2009. 94, 95, 97, 102, 116, 117, 118
- [96] G. Savransky, D. Dimerman, and C. Gotsman. Modeling and rendering escher-like impossible scenes. *Computer Graphics Forum*, 18:173–179, 1999. 126
- [97] R. Schoen and S.-T. Yau. Lectures on harmonic maps. In *Cambridge, MA: Harvard Univ.*, 1997. 18, 111
- [98] A. Sheffer and E. de Sturler. Surface parameterization for meshing by triangulation flattening. In *Proceedings of 9th International Meshing Roundtable*, pages 161–172, 2000. 20
- [99] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle-based flattening. In *Eng. with Comput.*, pages 326–337, 2001. 17
- [100] A. Sheffer, B. Levy, M. Mogilnitsky, and A. Bogomyakov. Abf++: Fast and robust angle based flattening. In *ACM Trans. Graph*, 24(2), 2005. 14, 18
- [101] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006. 13
- [102] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. 1996. 131
- [103] M. Simoneau, V. Lamothe, E. Hutin, P. Mercier, N. Teasdale, and J. Blouin. Evidence for cognitive vestibular integration impairment in idiopathic scoliosis patients. *BMC Neurosci*, page 10:102, 2009. 79
- [104] P. U. Springborn Boris, Schroder Peter. Conformal equivalence of triangle meshes. In *ACM Trans. Graph*, 27(3), 2008. 15, 22
- [105] W. P. Thurston. Geometry and topology of three-manifolds. *Princeton lecture notes*, 1976. 22
- [106] W. P. Thurston. The finite riemann mapping theorem. 1985. 22
- [107] W. T. Tutte. How to draw a graph. In *London Mathematical Society*, volume 13, pages 743–768, 1963. 14, 16, 18
- [108] Y. Wang, L. M. Lui, T. F. Chan, and P. M. Thompson. Optimization of brain conformal mapping with landmarks. In *Proceedings of MICCAI '05: Part II*, pages 675–683, 2005. 91
- [109] O. Weber, A. Myles, and D. Zorin. Computing extremal quasiconformal maps. *Comp. Graph. Forum*, 31(5):1679–1689, 2012. xi, 15, 23, 24, 51, 63, 64, 141
- [110] T.-P. Wu, C.-W. Fu, S.-K. Yeung, J. Jia, and C.-K. Tang. Modeling and rendering of impossible figures. *ACM Transactions on Graphics*, 29(2), 2010. 126

REFERENCES

- [111] J. Xia, I. Garcia, Y. He, S. Xin, and G. Patow. Editable polycube mapping for gpu-based subdivision surfaces. In *Proceedings of ACM Symposium on Interactive 3D Graphics and Games*, pages 151–158, 2011. 60
- [112] S. Xin, Y. He, and C.-W. Fu. Efficiently computing exact geodesic loops within finite steps. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):879–889, 2012. xiv, 83
- [113] S. Xin, Y. He, C.-W. Fu, D. Wang, S. Lin, W. C. Chu, J. C. Cheng, X. Gu, and L. M. Lui. Euclidean geodesic loops on high-genus surfaces applied to the morphometry of vestibular systems. In *Proceedings of MICCAI '11: Part II*, pages 384–392, 2011. 80, 82
- [114] S.-Q. Xin and G.-J. Wang. Improving chen and han’s algorithm on the discrete geodesic problem. *ACM Trans. Graph.*, (28) (4) 2009, 2009. 54
- [115] H. Yamabe. On a deformation of riemannian structures on compact manifolds. *Osaka Journal of Mathematics*, 12:21C37, 1960. 36
- [116] J. Yen and C. Séquin. Escher sphere construction kit. In *I3D '01*, pages 95–98, 2001. 126
- [117] B. Yeo, M. Sabuncu, T. Vercauteren, N. Ayache, B. Fischl, and P. Golland. Spherical demons: Fast diffeomorphic landmark-free surface registration. In *TMI*, 29, pages 650–668, 2010. xiii, 72, 78
- [118] F. L. S.-M. H. Yong-Liang Yang, Junho Kim and X. Gu. Optimal surface parameterization using inverse curvature map. In *IEEE Transactions on Visualization and Computer Graphics*, 14(5), pages 1054–1066, 2008. 15
- [119] R. Zayer, C. Rossl, and H. Seidel. Curvilinear spherical parameterization. In *Proceedings of Shape Modeling and Applications*, pages 57–64, 2006. 20
- [120] R. Zayer, C. Rossl, and H.-P. Seidel. Variations on angle based flattening. In *Proceedings of Multiresolution in Geometric Modelling*, pages 285–296, 2003. 14, 15, 18
- [121] W. Zeng, L. Lui, L. Shi, D. Wang, W. Chu, J. Cheng, J. Hua, S. Yau, and X. Gu. Shape analysis of vestibular systems in adolescent idiopathic scoliosis using geodesic spectra. In *MICCAI, Part III, LNCS*, 6363, pages 538–546, 2010. 65, 80
- [122] W. Zeng, L. M. Lui, F. Luo, T. F.-C. Chan, S.-T. Yau, and D. X. Gu. Computing quasiconformal maps using an auxiliary metric and discrete curvature flow. *Numer. Math.*, 121(4):671–703, 2012. 24, 64
- [123] W. Zeng, F. Luo, S.-T. Yau, and X. Gu. Surface quasi-conformal mapping by solving beltrami equations. In *IMA Conference on the Mathematics of Surfaces XIII (IMA'09)*, 2009. xi, 23, 29, 64, 91
- [124] W. Zeng, R. Sarkar, F. Luo, X. Gu, and J. Gao. Resilient Routing for Sensor Networks Using Hyperbolic Embedding of Universal Covering Space. In *Proc. of the 29th IEEE INFOCOM*, 2010. 95, 97, 98

REFERENCES

- [125] W. Zeng, X. Yin, M. Zhang, F. Luo, and X. Gu. Generalized koebe's method for conformal mapping multiply connected domains. In *SPM '09: 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 89–100, 2009. 21, 134
- [126] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. In *ACM Transaction on Graphics*, volume 24(1), pages 1–27, 2005. 16
- [127] H. Zhou, S. Xia, M. Jin, and H. Wu. Localized Algorithm for Precise Boundary Detection in 3D Wireless Networks. In *Proc. of the 30th IEEE ICDCS*, 2010. 100, 110