Surface Conformal/Quasi-conformal Parameterization with Applications

CHOI, Pui Tung

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Mathematics

The Chinese University of Hong Kong
July 2016

Thesis Assessment Committee

Professor CHAN Hon Fu Raymond (Chair)

Professor LUI Lok Ming (Thesis Supervisor)

Professor ZOU Jun (Committee Member)

Professor NG Kwok-po (External Examiner)

Abstract

Surface conformal and quasi-conformal parameterizations are important in computer graphics and medical imaging. In this thesis, we develop efficient and practical algorithms for mesh and point cloud parameterizations with various applications. Firstly, we propose a linear algorithm for the spherical conformal parameterizations and an efficient algorithm called *FLASH* for landmark-aligned spherical optimized conformal mappings of genus-0 closed triangular meshes. The algorithms are applied for the registration of human cortical surfaces, and the shape analysis of the carotid arteries and the hippocampal surfaces in medical imaging. Secondly, we propose two fast disk conformal parameterization algorithms for simply-connected open triangular meshes and apply the parameterizations for texture mapping. Thirdly, we develop a linear algorithm for computing spherical quasi-conformal parameterization and Teichmüller parameterization of genus-0 closed meshes. Fourthly, we develop an iterative scheme for the spherical conformal parameterizations of genus-0 point clouds with applications in meshing and multilevel representation. Fifthly, we propose a novel algorithm called *TEMPO* for the landmark aligned Teichmüller parameterization of disk-type point clouds with theoretical guarantee. The algorithm is applied for developing a dissimilarity metric of point clouds. Experimental results are presented for demonstrating the effectiveness of our proposed algorithms.

Acknowledgement

I wish to express my deepest gratitude to my advisor Prof. Ronald Lui, who introduced me to the beautiful field of computational differential geometry, for his invaluable guidance and support in my research over these years at The Chinese University of Hong Kong. I am also sincerely thankful to Prof. Raymond Chan and Prof. Thomas Au, for their invaluable advice and consistent support. My appreciation is further extended to Prof. Jun Zou and Prof. Michael Ng, for their careful examination of this thesis.

I am very grateful to the faculty, staff and students in the Department of Mathematics at CUHK. Special thanks go to my research teammates, Mr. Jeffery Lam, Ms. Tingwei Meng, Mr. Antonie Chan, Mr. Leon Yam, Mr. Kinder Ho and Mr. Edward Yung, for the useful discussions and the laughter at our office all the time.

I would like to thank my parents and brothers. Without their continuous encouragement, this thesis would not have been possible.

This thesis is dedicated to Mandy, for her love, patience and understanding.

Contents

Chapter 1

Introduction

Surface parameterization refers to the process of finding a bijective correspondence between a surface and a simple parameter domain with the same topology. With the advancement in computer technology in recent decades, the use of surface parameterization has been widespread in various fields. For instance, the study of genus-0 human brain cortical surface can be simplified with the aid of a spherical parameterization, which maps the cortical surface onto the unit sphere. Besides, texture mapping can be easily achieved with the aid of parameterization. Instead of processing the texture on a complicated 3D surface, we can design the texture on a simple parameter domain such as the unit disk, and map the texture onto the surface via the parameterization. These applications reflect the importance of surface parameterization.

To apply the abstract mathematical theories of surface parameterization in real applications, discretizations of the continuous surfaces are necessary. There are two major types of discretizations of surfaces, namely meshes and point clouds. Surface meshes provide accurate approximations of continuous surfaces with the aid of a set of sample data points on the original surface, and a set of piecewise linear polygons such as triangles and quadrilaterals. In contrast to the presence of both the vertices and the connectivity information

in surface meshes, point clouds serves as an alternative representation of continuous surfaces without the connectivity information. Hence, it is more difficult to approximate the differential operators on point clouds.

As surface parameterization aims to simplify the computations on a complicated surface with the aid of a simple parameter domain, the parameterization schemes should preserve certain geometric structures of the original surface. There are three major types of surface parameterizations, namely isometric parameterizations, area-preserving parameterizations and conformal parameterizations. Ideally, isometric parameterizations provide the most accurate result as the metric is preserved, which implies that they are both area and angle preserving. However, it is well known that isometric parameterizations only exist for surfaces with zero Gaussian curvature [20]. This limitation largely hinders the application of isometric parameterizations in real life. On the other hand, area-preserving parameterizations suffer from the occurrence of extreme angular distortions [27]. This serious drawback limits the real applications of area-preserving parameterizations. In contrast to the two abovementioned types of parameterizations, conformal parameterization is more desirable for two reasons. First, the existence of conformal parameterization is theoretically guaranteed. By the uniformization theorem, every simply-connected Riemann surface is conformally equivalent to either the Riemann sphere, the complex plane or the open unit disk. Hence, conformal parameterization is applicable for a wide range of surfaces in practice. Second, since conformal parameterization preserves angles, the local shapes are well-retained.

Parameterization can be categorized into two categories, including landmark-free parameterization and landmark-based parameterization. The existence of landmark-free conformal parameterization is guaranteed by the uniformization

theorem. However, with the presence of landmark constraints, conformal parameterization may not exist. In this case, we can consider the optimized conformal parameterization, which balances the conformality distortion and the landmark mismatch error of the parameterization. Besides, we can consider quasi-conformal parameterization, which is a generalization of conformal parameterization. Even with the presence of landmark constraints, quasi-conformal parameterization exists. Moreover, there exists a special type of quasi-conformal map, called the *Teichmüller map*, that achieves uniform conformality distortion under the prescribed landmark constraints. This thesis aims to develop efficient and accurate algorithms for computing the abovementioned conformal and quasi-conformal parameterizations of surface meshes and point clouds.

The organization of this thesis is as follows. In Chapter 2, we review the related previous works. In Chapter 3, we introduce the mathematical concepts relevant to our works. In Chapter 4, we invent a linear spherical conformal parameterization algorithm and an efficient algorithm called *FLASH* for landmark aligned spherical optimized conformal parameterization for genus-0 closed triangulated meshes. Applications in brain cortical surface registration, hippocampal surface registration and shape analysis of carotid arteries are presented. In Chapter 5, we propose two fast algorithms for the disk conformal parameterization for simply-connected open triangulated meshes. In Chapter 6, we propose a linear algorithm for spherical quasi-conformal parameterizations of genus-0 triangulated meshes. In Chapter 7, we develop a fast algorithm for spherical conformal parameterization of genus-0 point clouds with applications in meshing and multilevel representation. In Chapter 8, we invent an algorithm called *TEMPO* for computing the landmark aligned Teichmüller parameterization of disk-type point clouds with landmark

constraints. Applications in point cloud shape analysis are explored. This thesis is concluded in Chapter 9.

Much of the work in this thesis appears in the following publications and preprints:

(i) P. T. Choi, K. C. Lam, and L. M. Lui, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces.* SIAM Journal on Imaging Sciences, Volume 8, Issue 1, pp. 67–94, 2015.

(ii) P. T. Choi and L. M. Lui, *Fast disk conformal parameterization of simply-connected open surfaces.* Journal of Scientific Computing, Volume 65, Issue 3, pp. 1065–1090, 2015.

(iii) P. T. Choi and L. M. Lui, *A linear algorithm for disk conformal parameterization of simply-connected open surfaces.* Preprint, arXiv:1508.00396, 2015.

(iv) G. P.-T. Choi, K. T. Ho, and L. M. Lui, *Spherical conformal parameterization of genus-0 point clouds for meshing.* Preprint, arXiv:1508.07569, 2015.

(v) T. W. Meng, G. P.-T. Choi, and L. M. Lui, *TEMPO: Feature-endowed Teichmüller extremal mappings of point clouds.* Preprint, arXiv:1511.06624, 2015.

Chapter 6 and a portion of the material in Chapter 4 are previously unpublished.

Chapter 2

Previous work

In this chapter, we review the previous works on conformal and quasi-conformal parameterizations. Surveys of the pre-existing parameterization algorithms can be found in [27, 29, 76, 39].

2.1 Conformal parameterization of genus-0 closed meshes

The study of the discretization of conformal maps on meshes originated from [71, 22]. In [71], Pinkall and Polthier introduced a discretization of the Dirichlet energy using a cotangent weight. Eck et al. [22] proposed a discrete harmonic map on meshes. For genus-0 closed meshes, numerous spherical conformal parameterization algorithms have been developed. Angenent et al. [9, 36] proposed a spherical conformal parameterization scheme by linearizing the Laplace equation. Hurdal and Stephenson [40] used circle packing for computing the spherical conformal parameterization. Gu and Yau [31, 33] performed an optimization in the tangent space of the unit sphere using the gradient descent method for the parameterization. Lai et al. [47] proposed an algorithm for folding-free spherical conformal maps.

To obtain a more accurate one-to-one correspondence between surfaces,

Algorithms	Landmark constraints	Bijectivity	Complexity
Finite Element Approximation of Conformal Mapping [9, 36]	No	No	Linear
Spherical Conformal Mapping [33]	No	No	Nonlinear
Landmark Constrained Surface Conformal Mapping [58, 78]	Yes	No	Nonlinear
Folding-Free Global Conformal Mapping [47]	No	Yes	Nonlinear

Table 2.1: Several previous works on parameterization of genus-0 closed meshes.

landmark constraints are introduced in computing the spherical parameterizations. Gu et al. [33] proposed to apply a Möbius transformation for aligning the landmarks on $\mathbb{S}^2$. In [58, 78], Lui et al. introduced a nonlinear algorithm for computing landmark-matching optimized conformal parameterization by minimizing a combined energy which balances the Dirichlet energy and the landmark mismatch energy. Table 2.1 lists several existing parameterization schemes and their properties.

2.2 Conformal parameterization of simply-connected open meshes

For the conformal parameterizations of simply-connected open meshes, a number of algorithms have been established (see Table 2.2). Floater [24] introduced the shape-preserving parameterizations for disk-type triangulated meshes. Hormann and Greiner [38] proposed the Most Isometric Parameterization of Surfaces (MIPS) method. Sheffer and De Sturler [74] proposed the Angle Based Flattening (ABF) method by considering the angular distortion. In [75], Sheffer et al. introduced an improved algorithm called the

Methods	Boundary	Bijectivity	Complexity
Shape-preserving [24]	Fixed	Yes	Linear
MIPS [38]	Free	Yes	Nonlinear
ABF/ABF++ [74, 75]	Free	Local (no flips)	Nonlinear
LSCM/DNCP [51, 18]	Free	No	Linear
Mean-value [28]	Fixed	Yes	Linear
Circle patterns [45]	Free	Local (no flips)	Nonlinear
Spectral conformal [69]	Free	No	Linear
Double covering [43]	Free	No	Nonlinear
Discrete Ricci flow [44]	Fixed	Yes	Nonlinear
IDRF [85]	Fixed	Yes	Nonlinear
Yamabe Riemann map [64]	Fixed	Yes	Nonlinear
Holomorphic 1-form [31]	Fixed	Yes	Nonlinear

Table 2.2: The existing conformal parameterization algorithms for simply-connected open meshes.

ABF++ method. In [51], Lévy et al. proposed the Least Squares Conformal Maps (LSCM) for simply-connected open meshes. Desbrun et al. [18] introduced the Discrete, Natural Conformal Parameterization (DNCP). LSCM and DNCP were later shown to be equivalent. In [28], Floater proposed a parameterization method using a generalization of barycentric coordinates. Kharevych et al. [45] introduced an algorithm based on circle patterns. Gu and Yau [31] proposed an algorithm using holomorphic 1-forms and holomorphic differentials. In [64], Luo developed the combinatorial Yamabe flow for conformal parameterizations. Jin et al. [43] proposed a disk conformal parameterization algorithm using the double covering technique [32]. Later, Mullen et al. [69] developed the spectral conformal parameterizations. In [44], Jin et al. proposed an algorithm using discrete surface Ricci flow. In [85], Yang et al. generalized the discrete Ricci flow to improve the computation of conformal parameterizations.

2.3 Quasi-conformal parameterization of surface meshes

In recent years, a few methods for quasi-conformal parameterizations of surface meshes have been developed. Weber et al. [80] presented an algorithm for computing extremal quasi-conformal mappings for simply-connected open meshes using holomorphic quadratic differentials. In [54], Lipman introduced bounded distortion mappings for triangular meshes with boundary. Aigerman and Lipman [8] developed an algorithm for computing bounded distortion mappings in 3D. The algorithm can be applied for parameterizing meshes onto the 2D plane or polycubes. In [62], Lui et al. proposed an iterative algorithm for computing Teichmüller maps of simply-connected open meshes. The convergence of the algorithm has been proved in [63].

2.4 Parameterization of point clouds

In the last decade, numerous studies have been devoted to point cloud parameterizations. In [25, 26], Floater and Reimers proposed the meshless parameterization method. In [88], Zwicker and Gotsmann presented the spherical embedding method for point clouds. In [10, 11, 12], Azariadis and Sapidis proposed a scheme for orthogonally projecting a point cloud onto the dynamic base surface. In [77], Tewari *et al.* proposed the doubly-periodic global parameterization of point cloud sampled from a genus-1 closed surface onto the plane. Wang *et al.* [79] mapped a genus-0 point cloud onto its circumscribed sphere and then an octahedron, and finally unfolded the octahedron onto the plane. In [87], Zhang *et al.* presented an as-rigid-as-possible parameterization approach for disk-type point clouds via a local flattening step and a rigid alignment. Meng *et al.* [66] proposed a point cloud

parameterization scheme using neural network.

For conformal/quasi-conformal parameterization of point clouds, only a few works have been reported. Guo *et al.* [35] developed a conformal parameterization algorithm using Riemann surface theory and Hodge theory. In [52], Liang *et al.* constructed spherical conformal mappings of genus-0 point clouds by adopting the harmonic energy minimization algorithm in [47]. In [67], Meng and Lui developed the theory of computational quasi-conformal geometry on point clouds. In particular, they proposed an algorithm for the quasi-conformal parameterizations of disk-type point clouds.

Chapter 3

Mathematical background

In this chapter, we introduce conformal/quasi-conformal maps and some other mathematical concepts relevant to our work. For more details, readers are referred to [72, 73, 30, 23, 42, 61].

3.1 Conformal maps

Before rigorously define conformal maps, we first introduce the concept of Riemann surfaces.

Definition 3.1.1 (Riemann surface)**.** *Let $\mathcal{M}$ be a manifold. An atlas of $\mathcal{M}$ is said to be* conformal *if all its transition maps are biholomorphic. $\mathcal{M}$ is said to be a* Riemann surface *if it is equipped with a conformal structure, that is, the maximal conformal atlas.*

Then, we give the definition of the pull-back metric as follows.

Definition 3.1.2 (Pull-back metric)**.** *Let $\mathcal{M}$, $\mathcal{N}$ be two Riemann surfaces with local coordinate systems $r_1(x^1, x^2)$ and $r_2(x^1, x^2)$, where $r_1, r_2 : \mathbb{R}^2 \to \mathbb{R}^3$ are vector-valued functions. The first fundamental forms of $\mathcal{M}$ and $\mathcal{N}$ are respectively defined by*

$$ds^2_{\mathcal{M}} = \sum_{i,j} g_{ij} dx^i dx^j \quad \textit{and} \quad ds^2_{\mathcal{N}} = \sum_{i,j} \tilde{g}_{ij} dx^i dx^j, \tag{3.1}$$

where $g_{ij} = \left\langle \dfrac{\partial r_1}{\partial x^i}, \dfrac{\partial r_1}{\partial x^j} \right\rangle$, $\tilde{g}_{ij} = \left\langle \dfrac{\partial r_2}{\partial x^i}, \dfrac{\partial r_2}{\partial x^j} \right\rangle$.

Consider $f : \mathcal{M} \to \mathcal{N}$. *In local coordinate systems,* f *can be regarded as* $f : \mathbb{R}^2 \to \mathbb{R}^2$, *with* $f(x^1, x^2) = (f^1(x^1, x^2), f^2(x^1, x^2))$. *The* pull-back metric $f^* ds^2_{\mathcal{N}}$ *defined on* $\mathcal{M}$, *induced by* f *and* $ds^2_{\mathcal{N}}$, *is the metric*

$$f^* ds^2_{\mathcal{N}} = \sum_{m,n} \left(\sum_{i,j} \tilde{g}_{ij}(f(x^1, x^2)) \frac{\partial f^m}{\partial x^i} \frac{\partial f^n}{\partial x^j} \right) dx^m dx^n. \tag{3.2}$$

With the notion of the pull-back metric, we can now define conformal maps between Riemann surfaces.

Definition 3.1.3 (Conformal maps)**.** *Let* $\mathcal{M}$ *and* $\mathcal{N}$ *be Riemann surfaces. A map* $f : \mathcal{M} \to \mathcal{N}$ *is said to be* conformal *if there exists a positive scalar function* $\lambda(x^1, x^2)$, *called the conformal factor, such that*

$$f^* ds^2_{\mathcal{N}} = \lambda ds^2_{\mathcal{M}}. \tag{3.3}$$

Consequently, every conformal map preserves angles and hence the infinitesimal shapes of the surface.

Theorem 3.1.4 (Uniformization of Riemann surfaces)**.** *Every simply connected Riemann surface* $\mathcal{M}$ *is conformally equivalent to exactly one of the following three domains:*

(i) the Riemann sphere,

(ii) the complex plane,

(iii) the open unit disk.

In other words, there exists a conformal map between $\mathcal{M}$ and exactly one of the three domains.

3.2 Harmonic maps

Next, we introduce the concept of harmonic maps.

Definition 3.2.1 (Harmonic maps)**.** *The Dirichlet energy for a map* $f : \mathcal{M} \to \mathcal{N}$ *is defined as*

$$E(f) = \frac{1}{2} \int_{\mathcal{M}} |\nabla f|^2 dv_{\mathcal{M}}. \tag{3.4}$$

In the space of mappings, the critical points of $E(f)$ *are called harmonic maps.*

On triangular meshes, the discrete version of the Dirichlet energy is given by

$$E(f) = \frac{1}{2} \sum_{[u,v] \in K} k_{uv} ||f(u) - f(v)||^2. \tag{3.5}$$

Here $k_{uv} = \cot \alpha + \cot \beta$, where α, β are the angles opposite to the edge $[u, v]$.

The discretization of the Laplacian is given by

$$\Delta f = \sum_{[u,v] \in K} k_{uv}(f(u) - f(v)). \tag{3.6}$$

For genus-0 closed surfaces, conformal maps are equivalent to harmonic maps [42]. Hence, the problem of finding a conformal map $f : \mathcal{M} \to \mathcal{N}$ between two genus-0 closed surfaces is equivalent to an energy minimization problem.

3.3 Quasi-conformal maps

In this section, we introduce the concept of quasi-conformal maps, a generalization of conformal maps, and the related mathematical properties of them.

Definition 3.3.1 (Quasi-conformal maps)**.** *A map $f : \mathbb{C} \to \mathbb{C}$ is called a* quasi-conformal(QC) map *if it satisfies the Beltrami equation*

$$\frac{\partial f}{\partial \overline{z}} = \mu(z)\frac{\partial f}{\partial z} \tag{3.7}$$

for some complex-valued function μ satisfying $||\mu||_\infty < 1$ and $\frac{\partial f}{\partial z}$ is non-vanishing almost everywhere. Here, the complex partial derivatives are defined by

$$\frac{\partial f}{\partial z} := \frac{1}{2}\left(\frac{\partial f}{\partial x} - i\frac{\partial f}{\partial y}\right) \quad \text{and} \quad \frac{\partial f}{\partial \overline{z}} := \frac{1}{2}\left(\frac{\partial f}{\partial x} + i\frac{\partial f}{\partial y}\right). \tag{3.8}$$

μ is called the *Beltrami coefficient* of the quasi-conformal map f. f is conformal around a small neighborhood of p if and only if $\mu(p) = 0$, as Equation (3.7) becomes the Cauchy-Riemann equation in this situation. Hence, the Beltrami coefficient μ is closely related to the conformality distortion of f.

Besides, Beltrami coefficients are also related to the bijectivity of their associated quasi-conformal maps, as explained by the following theorem.

Theorem 3.3.2. *If $f : \mathbb{C} \to \mathbb{C}$ is a C^1 map satisfying $\|\mu_f\|_\infty < 1$, then f is bijective.*

In addition, the *maximal quasi-conformal dilation* of f is given by

$$K = \frac{1 + ||\mu||_\infty}{1 - ||\mu||_\infty}. \tag{3.9}$$

A geometrical illustration of quasi-conformal maps is shown in Figure 3.1.

Now, we consider the composition of two quasi-conformal maps. The Beltrami coefficient associated with the composition map can be expressed in terms of the Beltrami coefficients of the two quasi-conformal maps.

Theorem 3.3.3. *Let $f : \Omega \subset \mathbb{C} \to f(\Omega)$ and $g : f(\Omega) \to \mathbb{C}$ be two quasi-conformal maps. The Beltrami coefficient of $g \circ f$ is explicitly given by*

$$\mu_{g\circ f} = \frac{\mu_f + (\overline{f_z}/f_z)(\mu_g \circ f)}{1 + (\overline{f_z}/f_z)\overline{\mu_f}(\mu_g \circ f)}. \tag{3.10}$$

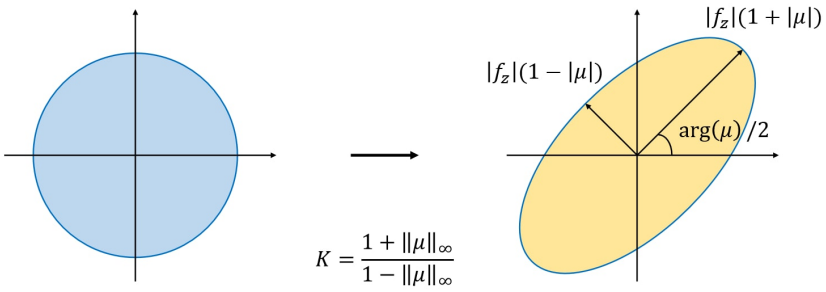

Figure 3.1: An illustration of quasi-conformal maps.

Conversely, given complex function, a quasi-conformal map can also be computed. More specifically, given a Beltrami coefficient $\mu : \mathbb{C} \to \mathbb{C}$ with $\|\mu\|_\infty < 1$, there exists a quasi-conformal map satisfying the Beltrami equation (3.7) in the distribution sense [30].

Let μ be a given Beltrami coefficient associated with a quasi-conformal map $f : \mathbb{C} \to \mathbb{C}$. Denote $f = u + iv$ and $\mu = \rho + i\tau$. It follows from the Beltrami Equation (3.7) that each pair of the partial derivatives v_x, v_y and u_x, u_y can be expressed as linear combinations of the other [61]:

$$\begin{aligned} v_y &= \alpha_1 u_x + \alpha_2 u_y; \\ -v_x &= \alpha_2 u_x + \alpha_3 u_y, \end{aligned} \quad \text{and} \quad \begin{aligned} -u_y &= \alpha_1 v_x + \alpha_2 v_y; \\ u_x &= \alpha_2 v_x + \alpha_3 v_y, \end{aligned} \tag{3.11}$$

where $\alpha_1 = \frac{(\rho-1)^2+\tau^2}{1-\rho^2-\tau^2}; \alpha_2 = -\frac{2\tau}{1-\rho^2-\tau^2}; \alpha_3 = \frac{(1+\rho)^2+\tau^2}{1-\rho^2-\tau^2}$. Since $\nabla \cdot \begin{pmatrix} -v_y \\ v_x \end{pmatrix} = 0$ and $\nabla \cdot \begin{pmatrix} -u_y \\ u_x \end{pmatrix} = 0$, f can be obtained by solving the following equations

$$\nabla \cdot \left(A \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right) = 0 \quad \text{and} \quad \nabla \cdot \left(A \begin{pmatrix} v_x \\ v_y \end{pmatrix} \right) = 0 \tag{3.12}$$

where $A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_3 \end{pmatrix}$. Equation (3.12) is called the *generalized Laplace equation*. Consequently, to find an optimal quasi-conformal map, it suffices to find an optimal complex-valued function.

Quasi-conformal maps can also be defined between two Riemann surfaces

$\mathcal{M}$ and $\mathcal{N}$ by introducing the concept of Beltrami differentials. A *Beltrami differential* $\mu(z)\frac{\overline{dz}}{dz}$ on $\mathcal{M}$ is an assignment to each chart (U_α, ϕ_α) of an L^∞ complex-valued function μ_α defined on the local parameter z_α such that $\mu_\alpha(z_\alpha)\frac{\overline{dz_\alpha}}{dz_\alpha} = \mu_\beta(z_\beta)\frac{\overline{dz_\beta}}{dz_\beta}$ on the domain also covered by another chart (U_β, ψ_β), where $\frac{dz_\beta}{dz_\alpha} = \frac{d}{dz_\alpha}\phi_{\alpha\beta}$ and $\phi_{\alpha\beta} = \phi_\beta \circ \phi_\alpha^{-1}$. An orientation preserving diffeomorphism $f : \mathcal{M} \to \mathcal{N}$ is said to be *quasi-conformal* associated with $\mu(z)\frac{d\overline{z}}{dz}$ if for any chart (U_α, ϕ_α) on $\mathcal{M}$ and any chart (U_β, ψ_β) on $\mathcal{N}$, the composition map $f_{\alpha\beta} := \psi_\beta \circ f \circ f_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha\frac{\overline{dz_\alpha}}{dz_\alpha}$.

In the discrete case, the Beltrami coefficients can be approximated on every triangular face. Let $f : K_1 \to K_2$ be a quasi-conformal map between two triangulated meshes K_1, K_2, and let T_1, T_2 be two corresponding faces on K_1, K_2 respectively. Suppose $T_1 = [a_1 + i\ b_1, a_2 + i\ b_2, a_3 + i\ b_3]$ and $T_2 = [w_1, w_2, w_3]$, where $a_i, b_i \in \mathbb{R}$ for all i. The Beltrami coefficient of f is approximated on T_1 by

$$\mu_f(T_1) = \frac{\frac{1}{2}(D_x + i\ D_y)\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}}{\frac{1}{2}(D_x - i\ D_y)\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}}, \tag{3.13}$$

where

$$D_x = \frac{1}{2Area(T_1)}\begin{pmatrix} b_3 - b_2 \\ b_1 - b_3 \\ b_2 - b_1 \end{pmatrix}^t \quad \text{and} \quad D_y = -\frac{1}{2Area(T_1)}\begin{pmatrix} a_3 - a_2 \\ a_1 - a_3 \\ a_2 - a_1 \end{pmatrix}^t. \tag{3.14}$$

Similarly, $\alpha_1, \alpha_2, \alpha_3$ in Equation (3.11) can be discretized. Ultimately, the elliptic PDEs (3.12) can be discretized into sparse symmetric positive definite linear systems as described in [61].

3.4 Teichmüller extremal maps

In the space of all quasi-conformal maps, there is a special class of maps called the Teichmüller maps. Intuitively, Teichmüller maps produce uniform conformality distortions over the whole domain. The rigorous definition of the Teichmüller maps is given below.

Definition 3.4.1 (Teichmüller map)**.** *Let $f : \mathcal{M} \to \mathcal{N}$ be a quasi-conformal map between two Riemann surfaces $\mathcal{M}$ and $\mathcal{N}$. f is said to be a* Teichmüller map (T-map) *associated with the quadratic differential $q = \varphi dz^2$ where $\varphi : \mathcal{M} \to \mathbb{C}$ is a holomorphic function, if its associated Beltrami coefficient is of the form*

$$\mu(f) = k\frac{\overline{\varphi}}{|\varphi|} \tag{3.15}$$

for some constant $k < 1$ and quadratic differential $q \neq 0$ with $||q||_1 = \int_{S_1} |\varphi| < \infty$.

It follows that a Teichmüller map is a quasi-conformal map whose Beltrami coefficient has a constant norm.

Furthermore, Teichmüller maps are closely related to a class of quasi-conformal maps called extremal maps, which are defined as follows.

Definition 3.4.2 (Extremal map)**.** *Let $f : \mathcal{M} \to \mathcal{N}$ be a quasi-conformal map. f is* extremal *if for any quasi-conformal map $h : \mathcal{M} \to \mathcal{N}$ isotopic to f relative to the boundary, we have*

$$K(f) \leq K(h) \tag{3.16}$$

where $K(f)$ is the maximal quasi-conformal dilation of f. f is uniquely extremal *if the inequality (3.16) is strict whenever $h \neq f$.*

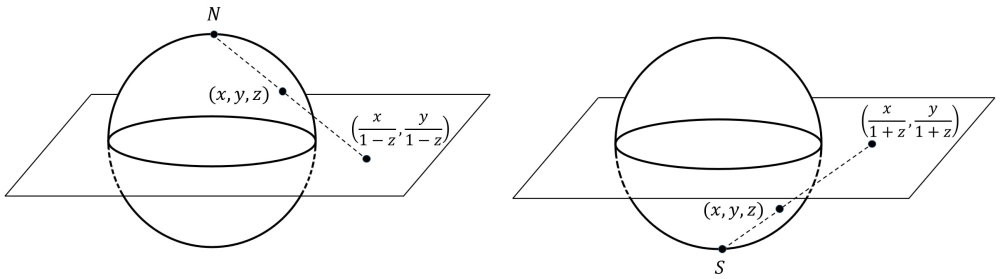

Figure 3.2: Illustrations of the stereographic projection (left) and the south-pole stereographic projection (right).

The following theorem explains the relationship between Teichmüller maps and extremal maps.

Theorem 3.4.3 (Landmark-matching Teichmüller map of D)**.** *Let $g : \partial D \to \partial D$ be an orientation-preserving diffeomorphism of ∂D, where D is the unit disk. Suppose $g'(e^{i\theta}) \neq 0$ and $g''(e^{i\theta})$ is bounded. Let $\{p_i\}_{i=1}^n \in D$ and $\{q_i\}_{i=1}^n \in D$ be two sets of interior landmark constraints with p_i corresponding to q_i for all i. Then, there exists a unique Teichmüller map $f : (D, \{p_i\}_{i=1}^n) \to (D, \{q_i\}_{i=1}^n)$ matching the interior landmarks, which is the unique extremal extension of g to D.*

In other words, Teichmüller maps satisfying the assumption in Theorem 3.4.3 minimize the maximal quasi-conformal dilation over all quasi-conformal maps.

3.5 Stereographic projection

A typical example of conformal maps on $\mathbb{S}^2$ is the stereographic projection. In this section, we introduce the stereographic projection and some of its variations.

Definition 3.5.1 (Stereographic projection)**.** *The* stereographic projection

is a conformal map $P_N : \mathbb{S}^2 \to \overline{\mathbb{C}}$ *with*

$$P_N(x, y, z) = \frac{x}{1-z} + i\frac{y}{1-z}. \tag{3.17}$$

The inverse stereographic projection *is a conformal map* $P_N^{-1} : \overline{\mathbb{C}} \to \mathbb{S}^2$ *with*

$$P_N^{-1}(x+iy) = \left(\frac{2x}{1+x^2+y^2}, \frac{2y}{1+x^2+y^2}, \frac{-1+x^2+y^2}{1+x^2+y^2}\right). \tag{3.18}$$

Note that the stereographic projection maps $\mathbb{S}^2$ onto $\overline{\mathbb{C}}$ with respect to the north pole. Analogously, we can define the south-pole stereographic projection.

Definition 3.5.2 (South-pole stereographic projection)**.** *The* south-pole stereographic projection $P_S : \mathbb{S}^2 \to \overline{\mathbb{C}}$ *is defined by*

$$P_S(x, y, z) = \frac{x}{1+z} + i\frac{y}{1+z}. \tag{3.19}$$

The inverse south-pole stereographic projection *is the map* $P_S^{-1} : \overline{\mathbb{C}} \to \mathbb{S}^2$ *with*

$$P_S^{-1}(x+iy) = \left(\frac{2x}{1+x^2+y^2}, \frac{2y}{1+x^2+y^2}, \frac{1-x^2-y^2}{1+x^2+y^2}\right). \tag{3.20}$$

The stereographic projection and the south-pole stereographic projection are illustrated in Figure 3.2.

3.6 Point cloud and local system

A *point cloud* $P = \{z_1, z_2, \ldots, z_n\} \subset \mathbb{R}^3$ is a set of sample points representing a Riemann surface $\mathcal{M}$. Because of the absence of the connectivity information, we cannot apply the face-based discretizations, which have been developed for meshes, on point clouds. To develop computational schemes on point clouds, we construct a local coordinate system for P on each point z_s and approximate the derivatives.

We first define an atlas (U_s, ϕ_s) for each point z_s, where U_s is an open cover and ϕ_s is the associated local coordinate function. U_s is formed using the collection of the neighboring points of z_s, denoted by $\mathcal{N}(z_s)$. To compute the neighborhood, we apply the *k-Nearest-Neighbors (k-NN)* algorithm. The *k-nearest neighborhood* $\mathcal{N}^k(z_s)$ of z_s is the collection of the k distinct elements in P (including z_s) closest from z_s in the sense of Euclidean distance. The KD-tree implementation by Lin [96] can be applied for k-NN. We denote $\mathcal{N}^k(z_s) = \{z_s^1, z_s^2, \ldots, z_s^k\}$ with $z_s^1 = z_s$. Then, by defining the normal vector at each point as the z-axis and using the principal component analysis (PCA) method [37] for z_s, we obtain three vectors $\{e_s^1, e_s^2, e_s^3\}$ which form an orthonormal basis of $\mathbb{R}^3$.

Then, we project $\mathcal{N}^k(z_s)$ to the plane spanned by $\{e_s^1, e_s^2\}$ by $\hat{z}_s^i = z_s^i - \langle z_s^i - z_s, e_s^3 \rangle e_s^3,\ i = 1, 2, \ldots, k$. With the projection $\hat{\mathcal{N}}^k(z_s) = \{\hat{z}_s^1, \hat{z}_s^2, \ldots, \hat{z}_s^k\}$ and the local coordinates $\{(x_s^1, y_s^1), (x_s^2, y_s^2), \ldots, (x_s^k, y_s^k)\}$, where $x_s^i = \langle z_s^i - z_s, e_s^1 \rangle$ and $y_s^i = \langle z_s^i - z_s, e_s^2 \rangle$ for $i = 1, 2, \ldots, k$, we can define $\phi_s : \mathcal{N}_s \to \mathbb{R}^2$ by $\phi_s(z_s^i) = (x_s^i, y_s^i)$. Also, the neighborhood $\mathcal{N}(z_s)$ can be regarded as a graph of its projection $\hat{\mathcal{N}}(z_s)$, that is, $z_s^i = z_s + x_s^i e_s^1 + y_s^i e_s^2 + f_s(x_s^i, y_s^i) e_s^3$.

Chapter 4

Spherical optimized conformal parameterization of genus-0 closed meshes with landmark constraints

4.1 Introduction

In this chapter, we present our two proposed algorithms in [1]. The first algorithm computes the spherical conformal parameterizations of genus-0 closed meshes in linear time, and the second algorithm efficiently computes a landmark-aligned optimized conformal parameterization of genus-0 closed meshes. The entire framework is abbreviated as *FLASH*, which stands for **F**ast **L**andmark **A**ligned **S**pherical **H**armonic parameterization.

For the parameterization of genus-0 closed surfaces, it is common to use the unit sphere $\mathbb{S}^2$ as the parameter domain. The problem of computing the spherical conformal parameterizations has been widely studied. Since harmonic maps are equivalent to conformal maps for genus-0 closed surfaces, the conventional approach in [33] considers the minimization problem of the Dirichlet energy on $\mathbb{S}^2$, which makes the computation nonlinear and hence time-consuming. The first proposed algorithm aims to accelerate the com-

putation of the spherical conformal parameterization. On the other hand, landmark aligned parameterization is important for computing landmark-matching registration between surfaces. With the presence of landmark constraints, spherical conformal parameterizations may not exist. Nevertheless, we can consider an optimized conformal parameterization that balances the conformality and the landmark mismatch error. The conventional approach in [78, 58] computes the optimized conformal parameterization directly on $\mathbb{S}^2$, which results in long computational time. Also, the bijectivity of the parameterization is lost under larger deformations or multiple landmark constraints. Flips or overlaps are observed especially near the landmark curves. The second proposed algorithm aims to overcome the mentioned drawbacks.

With the landmark aligned optimized conformal parameterization, we can easily obtain landmark-matching registration between genus-0 closed surfaces. Various applications of our algorithms in medical shape analysis are presented in the last section of this chapter.

4.2 Contributions

The contributions of our work are highlighted as follows:

(i) We invent a linear algorithm for computing the spherical conformal parameterization of genus-0 closed meshes. The computational time of our algorithm is significantly shorter than the existing methods.

(ii) The conformality distortion of the spherical conformal parameterization is further reduced by the use of quasi-conformal theory.

(iii) We propose a fast algorithm for computing landmark-aligned spherical conformal parameterizations.

(iv) The bijectivity of the parameterizations is theoretically supported by quasi-conformal theory.

4.3 Our proposed *FLASH* method

In this section, we describe our proposed *FLASH* method in details.

Suppose S_1 and S_2 are two genus-0 closed surfaces, respectively with landmark curves $\{p_i \in S_1\}_{i=1}^n$ and $\{q_i \in S_2\}_{i=1}^n$. The ultimate goal is to register S_1 and S_2 using an optimized conformal map with the landmark curves consistently matched. To achieve this goal, we first compute the spherical conformal parameterizations of S_1 and S_2 and denote the results by $\phi_1 : S_1 \to \mathbb{S}^2$ and $\phi_2 : S_2 \to \mathbb{S}^2$ respectively. Then, the registration problem can be simplified as finding an optimized conformal map $\tilde{f} : \mathbb{S}^2 \to \mathbb{S}^2$ of $\mathbb{S}^2$ such that $\tilde{f}(\phi_1(p_i)) = \phi_2(q_i)$ for $i = 1, 2, ..., n$. With $\tilde{f}$, the final registration can be obtained by $f := {\phi_2}^{-1} \circ \tilde{f} \circ \phi_1$.

In the following subsections, we introduce our proposed methods for

1. accelerating the computation of the spherical conformal parameterization;

2. accelerating the computation of the landmark aligned optimized conformal parameterization;

3. enforcing the bijectivity of the landmark aligned optimized conformal parameterization.

4.3.1 Linear spherical conformal parameterization

In this subsection, we propose a linear spherical conformal parameterization algorithm. Recall that the desired parameterization $\phi_i : S_i \to \mathbb{S}^2 (i = 1, 2)$ can

be obtained by solving the following Laplace equation on S_i:

$$\Delta_{S_i}^T \phi_i = 0, \tag{4.1}$$

where $\Delta_{S_i}^T \phi_i$ is the tangential component of $\Delta_{S_i} \phi_i$ on the tangent plane of $\mathbb{S}^2$, subject to the constraints: (i) $||\phi_i|| = 1$ and (ii) $\phi_i(p_1^i) = q_1^i$, $\phi_i(p_2^i) = q_2^i$ and $\phi_i(p_3^i) = q_3^i$ and $p_1^i \in S_i$, $p_2^i \in S_i$, $p_3^i \in S_i$, $q_1^i \in \mathbb{S}^2$, $q_2^i \in \mathbb{S}^2$, $q_3^i \in \mathbb{S}^2$. Since the operator $\Delta_{S_i}^T$ is non-linear, solving Equation (4.1) with the above constraints is expensive.

To accelerate the computation, Angenent et al. [9, 36] linearized the problem on the complex plane. Consider removing a point p_1^i from S_i. Then, $S_i \setminus \{p_1^i\}$ is topologically equivalent to $\mathbb{S}^2 \setminus \{\phi_i(p_1^i)\}$, and hence it can be mapped conformally to $\mathbb{C}$ (here, p_i^1 is mapped to ∞). In other words, there exists a conformal parameterization $\varphi_i : S_i \setminus \{p_1^i\} \cong \mathbb{C} \to \mathbb{C}$, such that $\lim_{x \to p_1^i} \varphi(x) = \infty$. φ_i can be computed by solving the Laplace equation

$$\Delta_{S_i} \varphi_i = 0, \tag{4.2}$$

with three-point correspondences: $\varphi_i(a_1^i) = b_1^i$, $\varphi_i(a_2^i) = b_2^i$ and $\varphi_i(a_3^i) = b_3^i$, where a_1^i, a_2^i, a_3^i, b_1^i, b_2^i, $b_3^i \in \mathbb{C}$. Here $\Delta_{S_i}^T = \Delta_{S_i}$ since the target domain is now $\mathbb{C}$.

Unlike the original problem (4.1), the new problem (4.2) is linear as Δ_{S_i} is linear and the nonlinear constraint (i) in the problem (4.1) is removed. Also, in the discrete case, Δ_{S_i} can be discretized as a symmetric positive definite matrix. Therefore, Equation (4.2) can be efficiently solved by the conjugate gradient method. A spherical parameterization $\phi_i : S_i \to \mathbb{S}^2$ can be given by $P_N^{-1} \circ \varphi_i$, where P_N^{-1} denotes the inverse stereographic projection.

Despite the fast computation, the method proposed by Angenent et al. [9, 36] has a critical drawback. Since the region near the puncture is

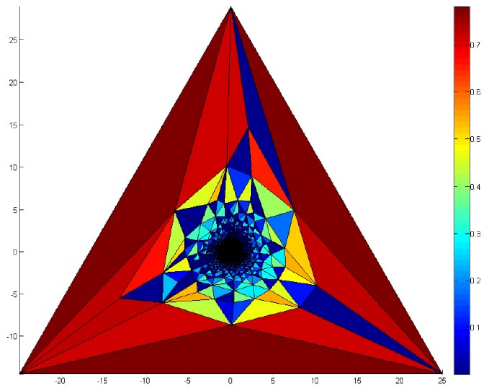

Figure 4.1: An illustration of how the conformality distortions are introduced near the puncture. Here, the outer region of the big triangle corresponds to the region near the puncture of the mesh. The colormap represents the norm of the Beltrami coefficient of the map from the big triangle to the original mesh. The red color on the triangular faces in the outer region of the big triangle indicates the large conformality distortions introduced near the puncture.

significantly enlarged when it is mapped to $\mathbb{C}$, severe conformality distortions are induced near the puncture p_1^i (or a triangular face in the discrete case). Figure 4.1 illustrates the distortions.

To improve the conformality near the puncture, we propose to compose the above result with a quasi-conformal map $g_i : \mathbb{S}^2 \to \mathbb{S}^2$. Intuitively, composing two maps with the same Beltrami coefficient cancels out the conformality distortion. It is noteworthy that although the genus-0 surfaces S_i are not topologically equivalent to $\mathbb{C}$, all the computations are performed on the punctured surfaces. Therefore, in practice, we only need to handle Beltrami coefficients but not Beltrami differentials. By an abuse of terminology, in the following discussion, we continue explaining our method with the term Beltrami coefficients. Let μ be the Beltrami coefficient of ϕ_i^{-1}, where $\phi_i : S_i \to \mathbb{S}^2$ is obtained by solving Equation (4.2). Since serious conformality distortions are induced near the north pole (or equivalently, at $\infty \in \bar{\mathbb{C}}$), $\mu(z)$

is large if z is close to ∞. In contrary, the conformality distortion near the south pole is negligible. Hence, $\mu(z) \approx 0$ when z is close to 0. We proceed to find a quasi-conformal map $g : \mathbb{S}^2 \to \mathbb{S}^2$ with the same Beltrami coefficient μ. The composition map $\widetilde{\phi_i} := g \circ \phi_i^{-1} : S_i \to \mathbb{S}^2$ is conformal, as explained by the following theorem.

Theorem 4.3.1. *Let $f : M_1 \to M_2$ and $g : M_2 \to M_3$ be quasi-conformal maps. Denote the Beltrami coefficient of f^{-1} and g by $\mu_{f^{-1}}$ and μ_g respectively Suppose $\mu_{f^{-1}} = \mu_g$. Then, the Beltrami coefficient of $g \circ f$ is equal to 0 and hence $g \circ f : M_1 \to M_3$ is conformal.*

Proof. Note that $\mu_{f^{-1}} \circ f = -(f_z/|f_z|)^2 \mu_f$. Since $\mu_{f^{-1}} = \mu_g$, we have

$$\mu_f + \frac{\overline{f_z}}{f_z}(\mu_g \circ f) = \mu_f + \frac{\overline{f_z}}{f_z}(\mu_{f^{-1}} \circ f) = \mu_f + \frac{\overline{f_z}}{f_z}(-\frac{f_z}{\overline{f_z}})\mu_f = 0. \tag{4.3}$$

Hence, by Equation (3.10),

$$\mu_{g \circ f} = \frac{\mu_f + \overline{f_z}/f_z(\mu_g \circ f)}{1 + \overline{f_z}/f_z \overline{\mu_f}(\mu_g \circ f)} = 0. \tag{4.4}$$

This shows that $g \circ f$ is conformal. □

The computation of the quasi-conformal map g_i with a given Beltrami coefficient can be achieved by the Linear Beltrami Solver (LBS) [61]. It is noteworthy that the original algorithm in [61] only works for rectangular domains, and the computation is linear. The method is extended hyperbolic domains in [81], but the periodic boundary conditions on the hyperbolic domains results in a non-linear computational time. In this work, we apply the linear algorithm for the spherical case. Note that if the elliptic PDEs are solved directly on the surfaces or the spheres, the computation is non-linear. Instead, we use the stereographic projection to transform the problem domain from the spherical domain to a big triangle on $\mathbb{C}$. This linearizes the problem

for the spherical case, although distortions are induced near the pole under the stereographic projection. To alleviate this issue, we invent a North Pole-South Pole iterative scheme to correct the conformality distortion on the spherical parameterizations.

Our iterative scheme has two major stages. In the first stage, we apply the spherical parameterization algorithm in [9, 36]. In particular, in Equation (4.2), we set $[b_1^i, b_2^i, b_3^i]$ as be a big triangle on $\mathbb{C}$ containing the origin that shares the same angle structure as the north pole triangle $[a_1^i, a_2^i, a_3^i]$. The second stage aims to fix the conformality distortion near the north pole. We first project $\mathbb{S}^2$ conformally onto $\overline{\mathbb{C}}$ using the south pole stereographic projection P_S, so that $(0, 0, -1)$ is mapped to ∞. Then, it remains to find a quasi-conformal map $\tilde{g}_i : \mathbb{C} \to \mathbb{C}$ with Beltrami coefficient $\widetilde{\mu}$, where $\widetilde{\mu}$ is the Beltrami coefficient of $(P_S \circ \phi_i)^{-1}$. The required quasi-conformal map g_i is given by $g_i := P_S^{-1} \circ \widetilde{g}_i \circ P_S$. However, the south pole stereographic projection will again introduce undesirable conformality distortions near the south pole. Nevertheless, we observe that $\mu \approx 0$ near $(0, 0, -1) \in \mathbb{S}^2$. Therefore, we can fix a small region Ω_S enclosing the south pole, which corresponds to the outermost region Ω_C in $\mathbb{C}$ after being projected onto the 2D plane using P_S. In other words, we look for a quasi-conformal map $\tilde{g}_i$ by solving Equation (3.12) such that

$$\begin{aligned} \tilde{g}_i|_{\Omega_C} &= \mathbf{id}|_{\Omega_C}; \\ \mu(\tilde{g}_i)|_{\mathbb{C}\setminus\Omega_C} &= \widetilde{\mu}_{\mathbb{C}\setminus\Omega_C} \end{aligned} \tag{4.5}$$

where $\mu(\tilde{g}_i)$ is the Beltrami coefficient of $\tilde{g}_i$.

By this treatment, the originally negligible conformality distortion of the resulting spherical parameterization $g_i \circ \phi_i$ near the south pole remains unchanged. On the other hand, the severe conformality distortions near the north pole can be significantly improved as guaranteed by Theorem 4.3.1.

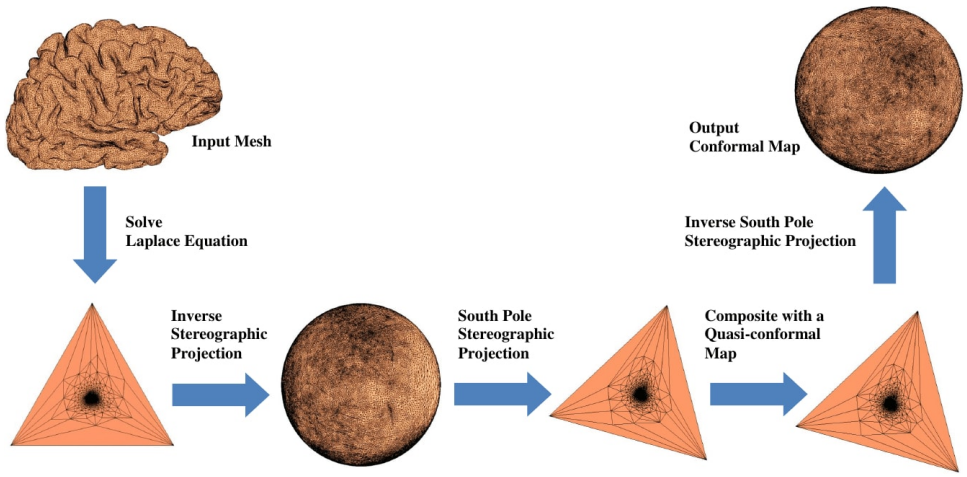

Figure 4.2: An illustration of our proposed linear spherical conformal parameterization algorithm.

The abovementioned procedures are summarized in Figure 4.2.

In addition, despite the conformality distortions, another serious problem of the spherical parameterization is the distribution of the vertices on $\mathbb{S}^2$. If the big triangle $[b_1^i, b_2^i, b_3^i]$ in Equation (4.2) is too large/small, the vertices will concentrate near the north/south pole under the inverse stereographic projection. This affects the quality of the spherical parameterization. To achieve a balanced distribution, we propose to find an optimal big triangle size with the aid of the following theorem.

Theorem 4.3.2. *Let T_1 and T_2 be two triangles of $\mathbb{C}$. The product of the perimeters of T_1 and $P_S(P_N^{-1}(T_2))$ is invariant under arbitrary scaling of T_1 and T_2.*

Proof. Denote the vertices of T_1, T_2 by $\{a_i\}_{i=1}^3, \{b_i\}_{i=1}^3$ respectively. Note that

$$P_S(P_N^{-1}(X+iY)) = \frac{-\frac{2X}{1+X^2+Y^2}}{1+\frac{-1+X^2+Y^2}{1+X^2+Y^2}} + i\frac{\frac{2Y}{1+X^2+Y^2}}{1+\frac{-1+X^2+Y^2}{1+X^2+Y^2}} = \frac{-X}{X^2+Y^2} + i\frac{Y}{X^2+Y^2}. \tag{4.6}$$

Hence the perimeter of T_1 is given by $p(T_1) = \sum_{1\le i<j\le 3} |a_i - a_j|$ and

$p(P_S(P_N^{-1}(T_2))) = \sum_{1\le i<j\le 3} |\frac{-\overline{b_i}}{|b_i|^2} - \frac{-\overline{b_j}}{|b_j|^2}|$. Now suppose T_1 and T_2 are scaled by a factor k. Then

$$p(kT_1) \times p(P_S(P_N^{-1}(kT_2))) = (\sum_{1\le i<j\le 3} |ka_i - ka_j|)(\sum_{1\le i<j\le 3} |\frac{-\overline{kb_i}}{|kb_i|^2} - \frac{-\overline{kb_j}}{|kb_j|^2}|)$$
$$= (k\sum_{1\le i<j\le 3} |a_i - a_j|)(\frac{1}{k}\sum_{1\le i<j\le 3} |\frac{-\overline{b_i}}{|b_i|^2} - \frac{-\overline{b_j}}{|b_j|^2}|) = p(T_1) \times p(P_S(P_N^{-1}(T_2))) \tag{4.7}$$

This shows that the product is an invariance. □

Now we project the sphere onto the complex plane using the stereographic projection, and apply Theorem 4.3.2 on the big triangle T and the innermost triangle t on $\mathbb{C}$, which correspond to the north pole triangle and the south pole triangle on $\mathbb{S}^2$. Given an arbitrary big triangle of size s, we define the *optimal big triangle size* by $s \times \frac{\sqrt{\text{perimeter}(T)\times\text{perimeter}(P_S(P_N^{-1}(t)))}}{\text{perimeter}(T)}$. By the scaling factor, the two triangles T and $P_S(P_N^{-1}(t))$ have the same perimeter. This indicates that the two polar triangles are of similar sizes. As a result, the points will not concentrate at any of the poles. This additional balancing scheme is performed after solving Equation (4.2) in our algorithm.

In the discrete case, let K be a triangulated mesh representing a surface S. It is noteworthy that since all steps in our algorithm, including solving the Laplace equation on $\mathbb{C}$, the stereographic projections, the reconstruction of quasi-conformal maps on $\mathbb{C}$, and the balancing scheme are linear, the entire spherical conformal parameterization algorithm is linear. The implementation of our linear algorithm for computing a spherical conformal parameterization $\phi : K \to \mathbb{S}^2$ is summarized in Algorithm 1.

Algorithm 1: Linear spherical conformal parameterization of genus-0 closed surfaces

Input: A genus-0 closed surface mesh K.
Output: A spherical conformal parameterization $\phi : K \to \mathbb{S}^2$.

1 Remove a triangular face $T_k = [v_{k_1}, v_{k_2}, v_{k_3}]$ on the brain mesh K. Choose a big triangle $[b_1, b_2, b_3]$ on $\mathbb{C}$ that shares the same angle structure with $T_k = [v_{j_1}, v_{j_2}, v_{j_3}]$. In practice, we choose the most regular triangular face of the mesh. This is because an irregular boundary big triangle may affect the conformality of the result of step 2;

2 A conformal parameterization $\varphi : K \setminus \{T_k\} \to \mathbb{C}$ can be obtained by solving the sparse linear system

$$\begin{cases} \sum_{[u,v]\in K} k_{uv}(\varphi(u) - \varphi(v)) = 0 & \text{if } u \neq v_{j_1}, v_{j_2}, v_{j_3} \\ \varphi(v_{j_t}) = b_t & \text{if } t = 1, 2, 3 \end{cases}.$$

Here $k_{uv} = \cot\alpha + \cot\beta$, where α, β are the two angles opposite to the edge $[u, v]$ spanned by the vertices u, v. Obtain the spherical mesh by the inverse stereographic projection P_N^{-1};

3 Balance the distribution of the spherical parameterization using Theorem 4.3.2;

4 Project the sphere onto $\mathbb{C}$ using the south pole stereographic projection P_S. Denote the result by D;

5 Compute the Beltrami coefficient μ of the map $f : D \to K$;

6 Compute $g = \mathbf{LBS}(\mu)$ with a small region around the south pole fixed;

7 Project the plane to $\mathbb{S}^2$ by the inverse south pole stereographic projection P_S^{-1};

8 Denote the final result by $\phi := P_S^{-1} \circ g \circ P_S \circ P_N^{-1} \circ \varphi$;

4.3.2 Landmark aligned optimized conformal parameterization

With the aid of our proposed linear spherical conformal parameterization algorithm, we can compute a landmark aligned optimized conformal map between the spheres and ultimately obtain a landmark-matching registration between the two surfaces S_1 and S_2. Suppose the spherical conformal parameterizations are given by $\phi_1 : S_1 \to \mathbb{S}^2$ and $\phi_2 : S_2 \to \mathbb{S}^2$. Denote the two spheres $\phi_1(S_1)$ and $\phi_2(S_2)$ by B_1 and B_2 respectively. To consistently align the pre-

scribed landmarks, we look for an optimized conformal map $\widetilde{\phi}_1 : S_1 \to \mathbb{S}^2$ that satisfies $\widetilde{\phi}_1(p_i) = \phi_2(q_i)$ for $i = 1, 2, ..., n$. A landmark-matching registration between S_1 and S_2 can then be obtained by $f := \phi_2^{-1} \circ \widetilde{\phi}_1$.

Note that the landmark curves are usually mapped to inconsistent locations on B_1 and B_2, and handling large deformations is challenging. Hence, it is undesirable to directly compute the landmark-aligned optimized conformal map between the spheres. In [33], Gu and Yau proposed to apply a suitable Möbius transformation that minimizes a landmark mismatch energy to match the landmark curves. Let $T = \frac{az+b}{cz+d}$ with $ad - bc = 1$ be a Möbius transformation of $\mathbb{S}^2$. Here $\mathbb{S}^2$ is identified with $\overline{\mathbb{C}}$ through the stereographic projection. Assume that the landmark curves are represented by discrete landmark points. The landmark mismatch energy in [33] is defined by

$$E_{MT}(a, b, c, d) = \sum_{i=1}^{n} |T(\phi_1(p_i)) - \phi_2(q_i)|^2 \tag{4.8}$$

By adding an extra constraint on T that it maps ∞ to ∞, the energy (4.8) can be simplified as:

$$\widetilde{E}_{MT}(a, b) = \sum_{i=1}^{n} g(\phi_1(p_i))|a(\phi_1(p_i)) + b - \phi_2(q_i)|^2 \tag{4.9}$$

where $g(z) = \frac{4}{1+z\bar{z}}$. It remains to find $a, b \in \mathbb{C}$ such that the energy (4.9) is minimized. More details can be found in [33].

Since Möbius transformations are conformal, the landmarks cannot be exactly aligned in most cases using the above method. Nevertheless, we adopt this method as an initial landmark-matching step. Denote the sphere obtained under the optimal Möbius transformation by $\widetilde{B}_1 := T(B_1)$. Based on $\widetilde{B}_1$, we further align the landmarks by finding an optimized conformal map that balances the conformality and the landmark mismatch energy.

In [58, 78], Lui et al. proposed the following combined energy for this

task:

$$E_{combined}(\widetilde{\phi}_1) = \int |\Delta^T_{\mathbb{S}^2}\widetilde{\phi}_1|^2 + \lambda \int |\widetilde{\phi}_1(\phi_1(p_i)) - \phi_2(q_i)|^2 \tag{4.10}$$

where λ is a weighting factor. If $\lambda = 0$, the combined energy is the same as the Dirichlet energy. If λ is large, the resulting map will be with small landmark mismatch energy but the conformality will be lost. In [58, 78], this optimization problem is solved using the gradient descent method, which iteratively finds a vector field in the tangent space of the sphere. Since $\Delta^T_{\mathbb{S}^2}$ is a nonlinear operator on $\mathbb{S}^2$, the computation is time-consuming.

Now, we propose a linear algorithm to solve the abovementioned optimization problem. The key idea is to formulate the problem on $\overline{\mathbb{C}}$ using the stereographic projection in order to linearize the problem. Firstly, by the stereographic projection, $\widetilde{B}_1 \setminus \Omega$ and $B_2 \setminus \Omega$ can be conformally mapped to C_1 and C_2 in $\mathbb{C}$ respectively, where Ω is a small region enclosing $(0, 0, 1) \in \mathbb{S}^2$. Let $\varphi : C_1 \to \mathbb{C}$ be the landmark aligned optimized conformal map, and $\{\tilde{p}_i\}_{i=1}^n$ and $\{\tilde{q}\}_{i=1}^n$ be the corresponding landmark curves on C_1 and C_2 respectively. Given three-point correspondences $\varphi(a_1) = a_1$, $\varphi(a_2) = a_2$ and $\varphi(a_3) = a_3$ on C_1, φ can be computed by solving the following equation:

$$\Delta_{\mathbb{R}^2}\varphi + \lambda\delta_E(\varphi - \tilde{q}_i) = 0 \tag{4.11}$$

where $\delta_E(w)$ is the smooth approximation of the characteristic function

$$\chi_E(w) = \begin{cases} 1 & \text{if } w \text{ is on } \tilde{p}_i \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

Since $\Delta_{\mathbb{R}^2}$ is a linear differential operator, the computation becomes linear.

In the discrete case, suppose K_1, K_2 are two triangulated meshes representing S_1 and S_2, with landmark points $\{p_i\}_{i=1}^n$ and $\{q_i\}_{i=1}^n$ respectively. The discrete form of the combined energy is

$$E_{combined}(\varphi) = \sum_{[u,v]\in K_1} k_{uv}||\varphi(u) - \varphi(v)||^2 + \lambda \sum_{i=1}^{n} ||\varphi(p_i) - q_i||^2. \tag{4.13}$$

Hence, Equation (4.11) can be discretized as a sparse linear system

$$\Delta\varphi = \sum_{[u,v]\in K_1} k_{uv}(\varphi(u) - \varphi(v)) + \lambda \sum_{u\in K_1} (\varphi(u) - q_i)\chi_{K_1}(u), \tag{4.14}$$

which can be efficiently solved by the conjugate gradient method. The algorithm is summarized in Algorithm 2.

Algorithm 2: Landmark aligned optimized conformal map

Input: Two genus-0 closed surface meshes S_1, S_2 and the weight λ.
Output: A landmark aligned optimized conformal map φ on the complex plane.

1 Apply the linear spherical conformal parameterization on S_1, S_2 and obtain two spherical meshes B_1, B_2;
2 Apply a Möbius transformation f to B_1 of the form $f(z) = az + b$ which minimizes the landmark mismatch energy $\sum_{i=1}^{n} |f(z_i) - w_i|^2$, where $\{z_i \in \mathbb{C}\}$, $\{w_i \in \mathbb{C}\}$ are the landmarks of B_1, B_2 under the stereographic projection [33]. Denote the resulting spherical mesh by $\widetilde{B}_1$;
3 Project $\widetilde{B}_1, B_2$ onto $\mathbb{C}$ by the stereographic projection P_N;
4 Obtain the landmark constrained optimized conformal map φ on $\mathbb{C}$ by solving the sparse linear system

$$\begin{cases} \sum_{[u,v]\in K} k_{uv}(\varphi(u) - \varphi(v)) + \lambda(\varphi(u) - q_i) = 0 & \text{if } u = p_i, i = 1, 2, ..., n \\ \sum_{[u,v]\in K} k_{uv}(\varphi(u) - \varphi(v)) = 0 & \text{if } u \notin \{p_1, p_2, ..., p_n, j_1, j_2, j_3\} \\ \varphi(v_{j_t}) = v_{j_t} & \text{if } t = 1, 2, 3. \end{cases} \tag{4.15}$$

Here $\{p_i\}_{i=1}^n$, $\{q_i\}_{i=1}^n$ are the landmarks of the meshes on $\mathbb{C}$;

4.3.3 Preservation of bijectivity via Beltrami coefficients

Note that the landmark-aligned optimized conformal map obtained using the combined energy may not be a diffeomorphism. In this subsection, we propose an iterative algorithm to enforce the bijectivity of the map using quasi-conformal theories. By Theorem 3.3.2, f is a diffeomorphism if and only if $\|\mu_f\|_\infty < 1$, where μ_f is the Beltrami coefficient of f. Hence, by

adjusting the norm of the Beltrami coefficients, the diffeomorphic property of the corresponding map can be easily achieved.

In [81], Lui and Wen proposed an iterative algorithm for obtaining a bijective registration using the Beltrami coefficients of the surface map. The key step of the method is to smoothen the Beltrami coefficient by minimizing the following energy

$$\int |\nabla \mu|^2 + \frac{\lambda}{2} \int |\mu - \mu_n|^2. \tag{4.16}$$

However, this energy is not suitable for our case as our problem is on a big triangle on $\mathbb{C}$. More specifically, the outermost region of the big triangle is coarse while the innermost region is extremely dense. Hence, a slight change of the Beltrami coefficients at the outermost region results in an undesirably large change at the innermost region. Consequently, the southern hemisphere of the spherical parameterization will have large distortions. Hence, we have to limit the change of the Beltrami coefficient at the outermost region. To solve this problem, we propose to use the geodesic distance between a point and $(0, 0, 1)$ as a weighting factor for controlling the change of the Beltrami coefficient in the outermost region.

More explicitly, let $\widetilde{\phi}_1^0 : \mathbb{C} \setminus \Omega \to \mathbb{C}$ be the landmark-aligned optimized conformal parameterization obtained from the last subsection. Denote the Beltrami coefficient of $\widetilde{\phi}_1^0$ by ν^0. First, we chop down $|\nu^0|$ by:

$$|\widetilde{\nu}(z)| = \min\{|\nu^0(z)|, 1 - \epsilon\} \tag{4.17}$$

for all $z \in \mathbb{C} \setminus \Omega$. Here, ϵ is a small positive number (usually chosen as 0.01). We then look for a smooth approximation μ_{smooth} of $\widetilde{\nu}$ by:

$$\mu^0_{smooth} = \mathbf{argmin}_\mu \int (|\nabla \mu|^2 + |\mu - \widetilde{\nu}|^2 + \frac{1}{D}|\mu|^2) \tag{4.18}$$

where $D : \mathbb{C} \setminus \Omega \to \mathbb{R}^+$ and $D(p)$ is defined as the geodesic distance between $(0, 0, 1)$ and the image of $p \in \mathbb{C} \setminus \Omega$ under the inverse stereographic projection

P_N^{-1}. Here, $|\nabla\mu|^2$ measures the smoothness of the Beltrami coefficient, $|\mu-\widetilde{\nu}|^2$ measures the difference between the original and the new Beltrami coefficient, and D gives a weight for controlling the changes of the Beltrami coefficient at the outermost region (which corresponds to the northernmost region on $\mathbb{S}^2$).

After computing μ^0_{smooth}, we apply the Linear Beltrami Solver (LBS) [61] to reconstruct the quasi-conformal map g^0 with the corresponding Beltrami coefficient μ^0_{smooth} and the landmark constraints. Denote the Beltrami coefficient of g^0 by μ_{lm}. Let $\mu^0 = \mu^0_{smooth} + t(\mu_{lm} - \mu^0_{smooth})$, where $t \in [0,1]$ is the landmark matching factor. We then compute a quasi-conformal map f^0 with Beltrami coefficient μ^0 using LBS. This step balances the smoothness and landmark-matching property of the quasi-conformal map f^0. We then set ν^1 as the Beltrami coefficient of f^0 and set $\widetilde{\phi}_1^1 = f^0$. We continue the iteration until the resulting map $\widetilde{\phi}_1^n$ becomes a diffeomorphism.

In summary, the proposed algorithm for enforcing the bijectivity is described as follows. Suppose ν^n and $\widetilde{\phi}_1^n$ are obtained at the n^{th} iteration, ν^{n+1} and $\widetilde{\phi}_1^{n+1}$ can be obtained as follows:

$$\begin{aligned} f^n &= \mathbf{LBS}(\mu^n); \\ \nu^{n+1} &= \mu(f^n); \\ \widetilde{\phi}_1^{n+1} &= f^{n+1}. \end{aligned} \tag{4.19}$$

In the discrete case, denote the area of the triangle T by $A(T)$. We propose to approximate D by the area of the triangular faces. The reason is that under the stereographic projection, the triangular faces closer to the north pole will be mapped to larger triangles on $\bar{\mathbb{C}}$. Hence, $A(T)$ is proportional to $1/D(T)$. The minimizer μ^0_{smooth} of the optimization problem (4.18) can be obtained by solving the following equation:

$$\Delta\mu^0_{smooth} + (1 + A(T))\mu^0_{smooth} = \nu. \tag{4.20}$$

After a bijective map $\widetilde{\phi}_1 : \mathbb{C} \setminus \Omega \to \mathbb{C} \setminus \Omega$ is obtained, a bijective landmark-aligned spherical parameterization of S_1 can be achieved by $\phi_1^* := P_N^{-1} \circ \widetilde{\phi}_1 \circ \phi_1$. Again, one drawback of this algorithm is that there are large conformality distortions around the north pole. Thus, the Beltrami coefficient near the north pole is large. In contrary, the Beltrami coefficient of the southernmost region is close to 0. To correct the distortions at the northernmost part, we again composite the map with a quasi-conformal map. Firstly, we apply the south pole stereographic projection P_S to map $\mathbb{S}^2$ to $\overline{\mathbb{C}}$. Then we compute the Beltrami coefficient μ of the map from the plane to the original surface. We fix the landmark positions and also the region in $\mathbb{C}$ that corresponds to the southernmost region, and then apply LBS [61] to reconstruct the quasi-conformal map using the same Beltrami coefficient μ. By Theorem 4.3.1, the conformality distortions are corrected. Also, this further step does not affect the landmark-matching property of the parameterization as the landmark points are kept fixed. The algorithm we introduced above is summarized in Algorithm 3.

This complete our proposed *FLASH* framework for computing landmark aligned spherical optimized conformal parameterization. The procedure of *FLASH* is summarized in Figure 4.3.

4.4 Experimental results

In this section, we demonstrate the effectiveness of our proposed algorithms. Several models are adopted from Aim@Shape shape repository [89] for the linear spherical conformal parameterization algorithm. Also, we perform experiments on 38 pairs of right hemispheric brain cortical surfaces reconstructed from MRI images of older adult subjects, aged from 60 to 96, using the open source reconstruction software FreeSurfer. Half of the subjects are

Algorithm 3: Preservation of bijectivity via Beltrami coefficients

Input: Two meshes S_1, S_2, the spherical conformal parameterization $\phi_1 : S_1 \to \widetilde{B}_1$ and the optimized conformal map $\varphi : P_N(\widetilde{B}_1) \to \mathbb{C}$.

Output: A bijective landmark aligned spherical optimized conformal parameterization.

Compute the Beltrami coefficient ν of the map $\psi : \varphi(P_N(\widetilde{B}_1)) \to S_1$;
repeat
  Compute

$$\mu_{smooth} := \mathbf{argmin}_{\mu} \int (|\nabla \mu|^2 + |\mu - \nu|^2 + A(T)|\mu|^2)$$

  where $A(T)$ is the area of the triangular face T on the plane. This can be done by solving $\Delta\mu + (1 + A(T))\mu = \nu$;
  For every triangular face T, chop down the norm of $\mu_{smooth}(T)$ by $|\mu_{smooth}(T)| := \min\{|\mu_{smooth}(T)|, 0.99\}$;
  Compute $g = \mathbf{LBS}(\mu_{smooth})$ with landmarks fixed;
  Compute the Beltrami coefficient μ_{lm} of g;
  Define $\mu := \mu_{smooth} + t(\mu_{lm} - \mu_{smooth})$, where $t \in [0, 1]$ is the landmark matching factor;
  For every triangular face T, chop down the norm of $\mu(T)$ by $|\mu(T)| := \min\{|\mu(T)|, 0.99\}$;
  Compute $f = \mathbf{LBS}(\mu)$;
  Update ν by the Beltrami coefficient of f;
until *The resulting map is bijective*;
Obtain the spherical mesh using the inverse stereographic projection;
Project the sphere onto $\bar{\mathbb{C}}$ through the south pole stereographic projection P_S. Denote the result as D;
Compute the Beltrami coefficient μ_D of the map $f_D : D \to K_1$;
Fixing a small region near the south pole and the landmarks, compute $g_D = \mathbf{LBS}(\mu_D)$;
Project the plane to $\mathbb{S}^2$ by the inverse south pole stereographic projection P_S^{-1};
Obtain the parameterization $P_S^{-1} \circ g_D \circ P_S \circ P_N^{-1} \circ f \circ \varphi \circ P_N \circ \phi_1$;

characterized as non-demented and half are with dementia of the Alzheimer's type. The MRI images are freely available on the Open Access Series of Imaging Studies (OASIS) [65]. Each reconstructed cortical surface is of about

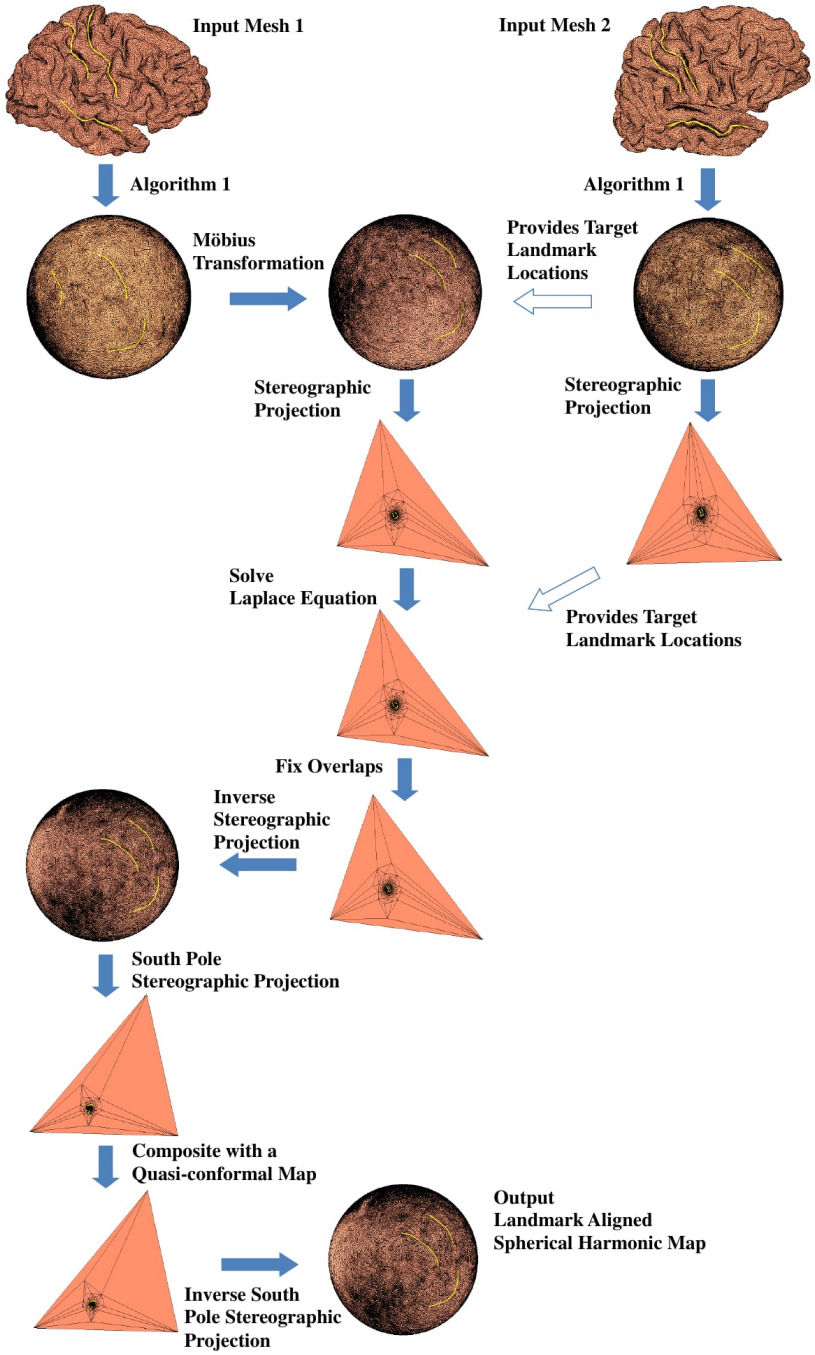

Figure 4.3: Illustration of the FLASH algorithm.

45k vertices and 90k faces. We first compare our proposed linear spherical conformal parameterization algorithm with the existing algorithms in [33] and

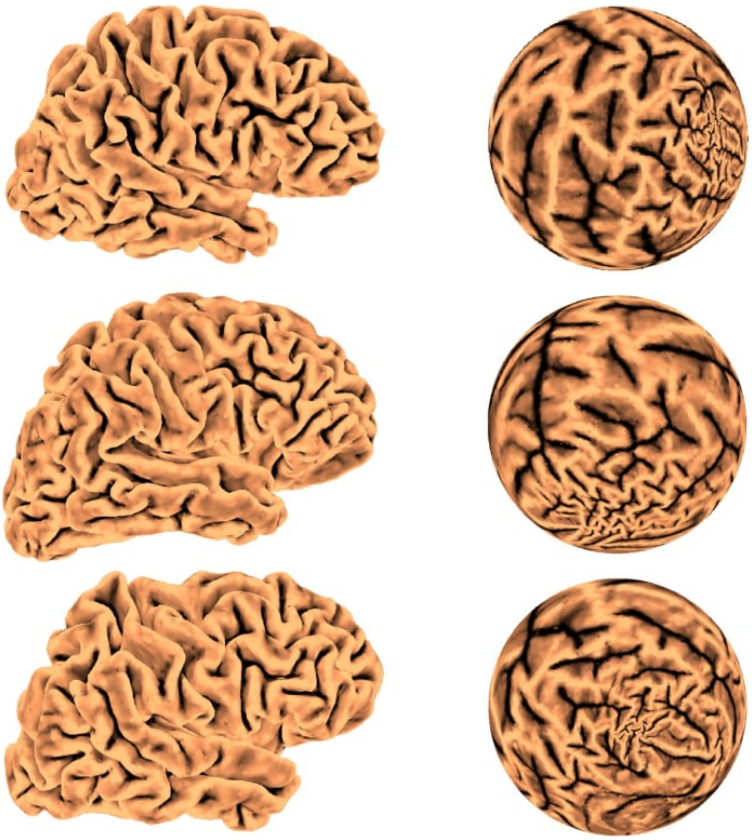

Figure 4.4: Spherical conformal parameterizations using our proposed linear algorithm.

[9, 36]. Then, for landmark aligned parameterization, we compare the performance of *FLASH* with the algorithms in [58], [33] and [59]. All experiments are performed on a PC with a 3.20 GHz CPU.

4.4.1 Spherical conformal parameterization

We first test our proposed linear algorithm for spherical conformal parameterization. Figure 4.4 shows three of the cortical surfaces and their parameterizations obtained by our proposed linear algorithm. Figure 4.5 shows two more meshes and their spherical conformal parameterizations.

To evaluate our proposed algorithm, we compare the results with two

Figure 4.5: Two meshes and their spherical conformal parameterizations obtained by our proposed linear algorithm.

existing algorithms in [33] and [9, 36]. We first compare the conformality distortions of the parameterizations obtained by the algorithm in [9, 36] with those by our proposed linear algorithm. We define the conformality distortion index (CDI) on each triangular face by $(|\alpha-\alpha'|+|\beta-\beta'|+|\gamma-\gamma'|)/2\pi$, where α, β, γ are the three angles of the resulting triangular face and α', β', γ' are those on the original triangular face. The CDI measures the changes of the angles on each triangular face under the parameterization. As shown in Figure 4.6, the method in [9, 36] produces severe conformality distortions near the north pole. On the contrary, our proposed linear algorithm effectively corrects the conformality distortions. Figure 4.7 further illustrates the improvement of the conformality using our proposed algorithm.

We then highlight the bijectivity of our proposed algorithm. Figure 4.8 shows the mesh of a dinosaur with a sharp tail and the spherical conformal parameterizations obtained by [33] and our proposed algorithm. It can be observed that our algorithm guarantees bijectivity while the method in [33] cannot.

Table 4.1 lists the statistics of our proposed method, the algorithms proposed in [33] and [9, 36]. The parameters we used to run the algorithm in [33] are: time step $\delta t = 0.1$, Tuette embedding energy threshold = 0.001 and

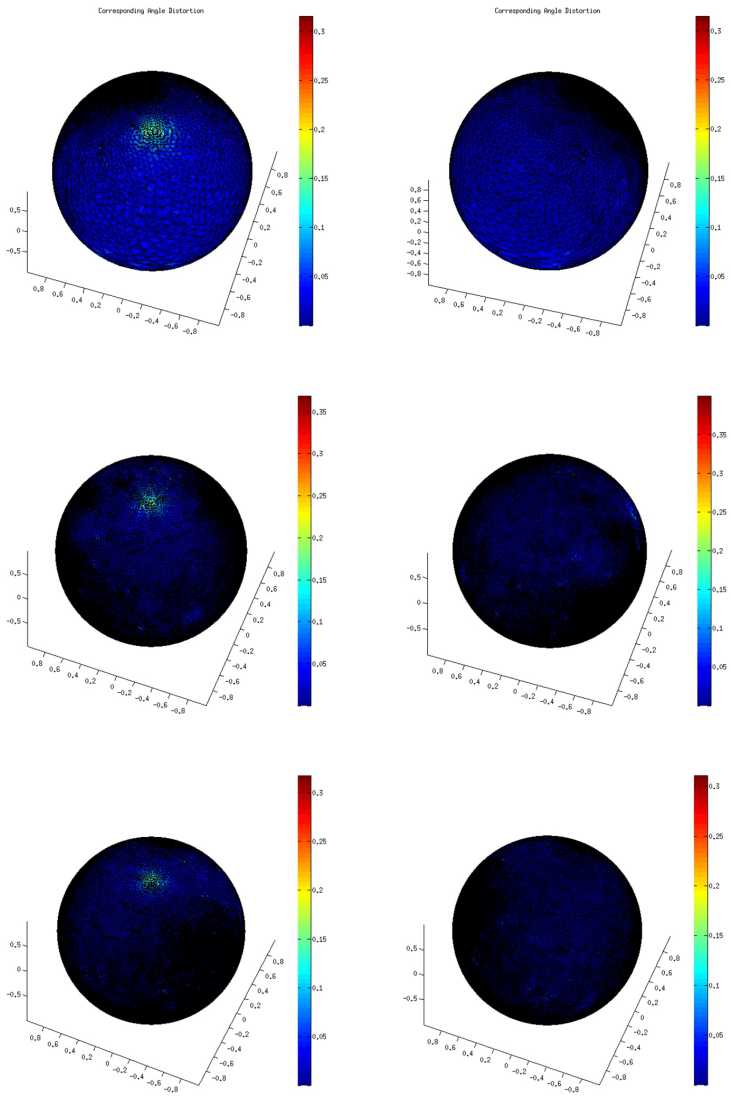

Figure 4.6: Comparisons between the spherical conformal parameterizations by the algorithm in [9, 36] and our proposed linear algorithm. Each row represents the results of one brain sample. The spheres are colored with the conformality distortion index (CDI). Conformality distortions are induced near the north pole using the algorithm in [9, 36]. The distortions are corrected using our proposed linear algorithm. Left: The results of [9, 36]. Right: The results of our proposed linear algorithm.

Dirichlet energy threshold = 0.00001. It is noteworthy that our algorithm takes only about 1 second on average, which is over 5000 times faster than

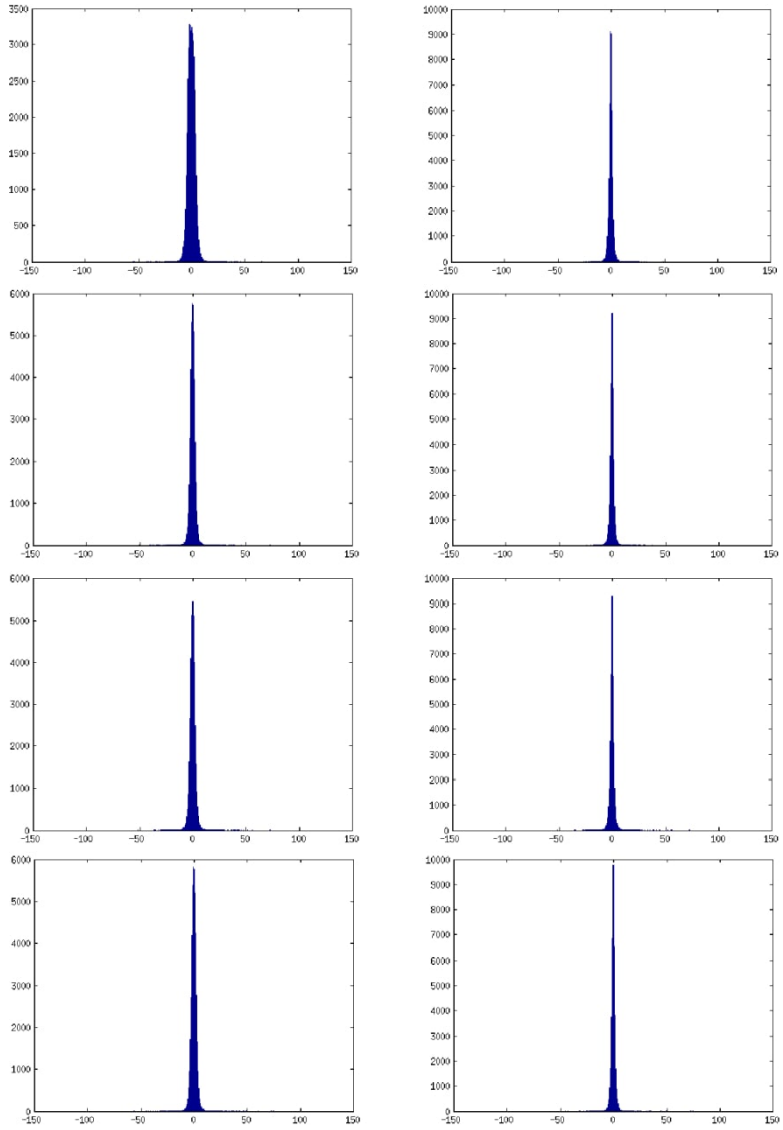

Figure 4.7: Histograms of angular distortions under the parameterizations obtained by the algorithm in [9, 36] and our proposed linear algorithm. Each row represents the results of one brain sample. Left: The results obtained by [9, 36]. Right: The results obtained by our proposed linear algorithm.

the existing algorithm in [33]. Also, with comparable speed as [9, 36], our algorithm improves the conformality by 30% on average. These experimental results reflect the advantages of our proposed linear algorithm.

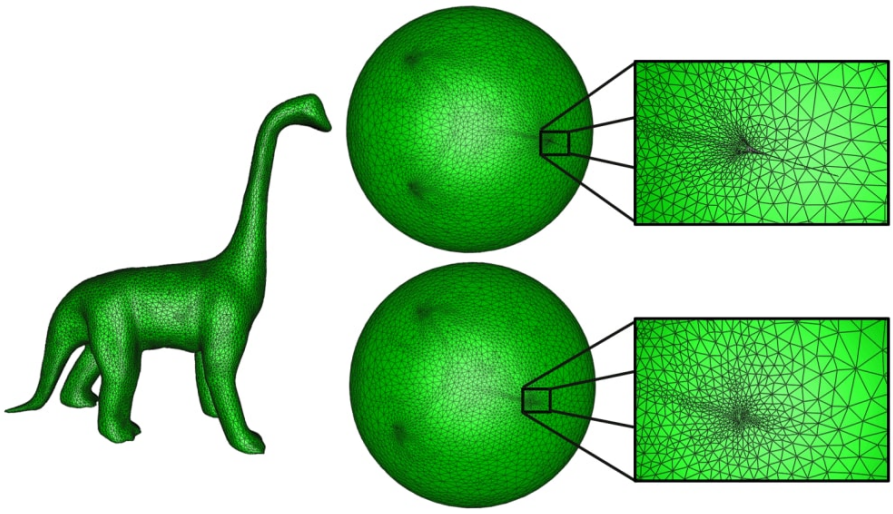

Figure 4.8: A genus-0 closed mesh of a dinosaur with a sharp tail and two parameterization results. Top right: The result obtained by [33]. Bottom right: The result of our proposed linear algorithm. It can be observed that the conventional approach produces a result with overlaps while our result is bijective.

Surface	Time spent (seconds) / mean(CDI)		
	Algorithm [33]	Algorithm [9, 36]	Our linear algorithm
Brain 18	6394.50/0.0241	0.4367/0.0128	0.9707/0.0107
Brain 21	6838.46/0.0240	0.4431/0.0216	1.0194/0.0105
Brain 36	5984.28/0.0237	0.4548/0.0142	0.9846/0.0101
Brain 16	7197.37/0.0256	0.4507/0.0149	0.9849/0.0109
Brain 14	7476.53/0.0248	0.4783/0.0148	1.0664/0.0103

Table 4.1: The statistics of different spherical conformal parameterization algorithms.

4.4.2 Landmark aligned spherical optimized conformal parameterization for brain surfaces

Next, we perform experiments to test our proposed *FLASH* algorithm for computing the landmark aligned spherical optimized conformal parameterizations (with $\lambda = 3$). On each cortical surface, six landmark curves, including Central Sulcus (CS), Postcentral Sulcus (PostCS), Superior Frontal Sulcus (SFS),

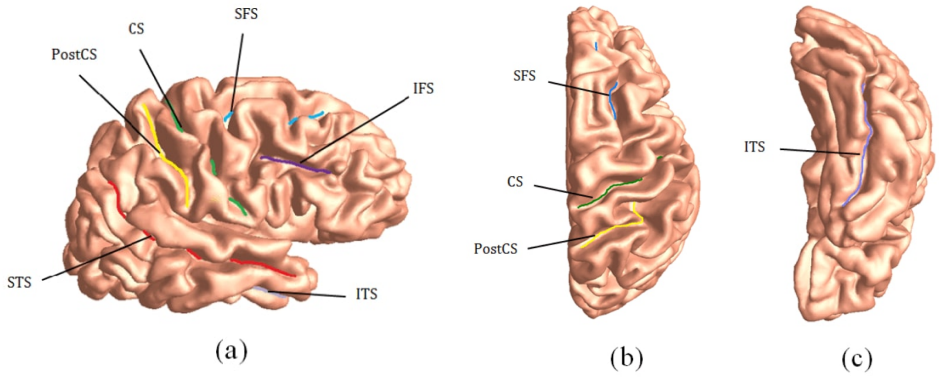

Figure 4.9: The six sulcal landmarks selected on a right hemispheric brain cortical surface. (a): the lateral view. (b): the top view. (c): the bottom view.

Inferior Frontal Sulcus (IFS), Superior Temporal Sulcus (STS) and Inferior Temporal Sulcus (ITS), were manually labeled. Figure 4.9 shows the selected sulci. The landmark aligned spherical parameterizations are shown in Figure 4.10. It can be observed that the landmarks are consistently aligned. Using the landmark aligned spherical parameterization, a landmark-matching registration between the brain surfaces is easily obtained. Figure 4.11 illustrates the difference between landmark-free registration and the landmark-matching optimized conformal registration obtained using *FLASH*.

Next, we compare the performance of our proposed landmark aligned spherical optimized conformal algorithm *FLASH* with the algorithm proposed in [58] in three different aspects.

Firstly, we compare the bijectivity of the parameterizations. Figure 4.12 shows the registration results obtained from *FLASH* and the algorithm in [58]. As shown in the figure, foldings can be found in the result by [58]. On the contrary, The result obtained by *FLASH* is free of foldings or flips. Table 4.2 lists the number of foldings of the parameterizations obtained by *FLASH* and the algorithm in [58].

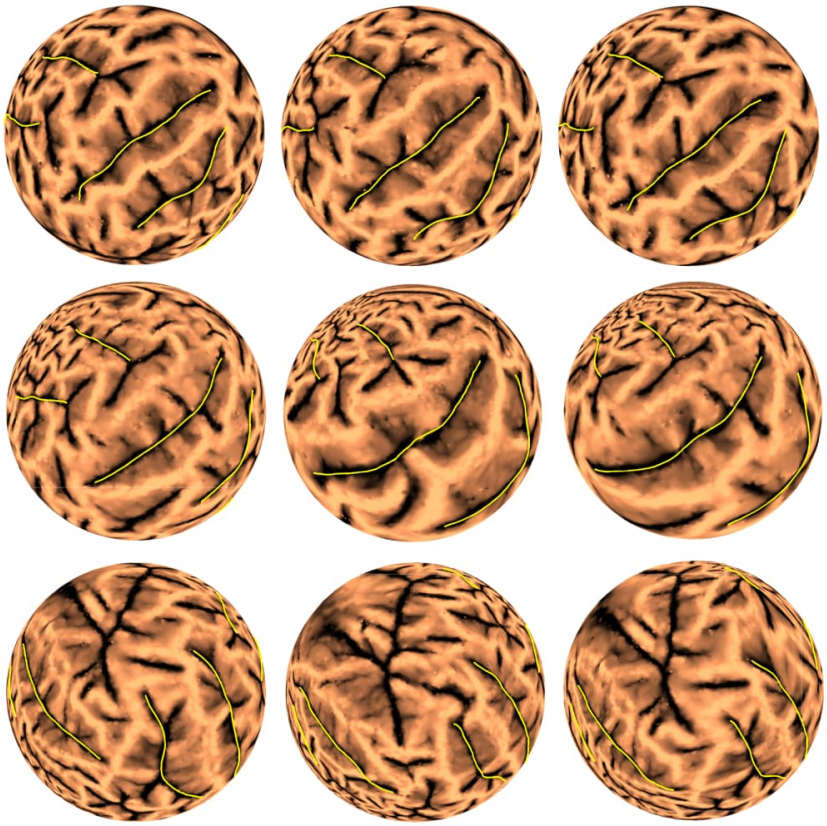

Figure 4.10: Landmark aligned spherical optimized conformal parameterizations using *FLASH*. One set of experimental result is shown in each row. The spherical conformal parameterizations of the source brains and the target brains are shown respectively in the first and the second column. The resulting landmark-aligned parameterizations (with $\lambda = 3$) are given in the third column. The landmark curves are highlighted in yellow.

Secondly, we compare the angular distortions under the parameterizations. Figure 4.13 shows the angular distortions of the results obtained by [58] and *FLASH*. The high similarity of the angular distortion histograms suggests that the global conformality distortion of *FLASH* is comparable to (or even better than) that of the conventional algorithm in [58].

Thirdly, we compare the computational time of the algorithms. The computational time required by *FLASH* and the algorithm proposed in [58] is listed in Table 4.2. Note that for fair comparison, when computing the

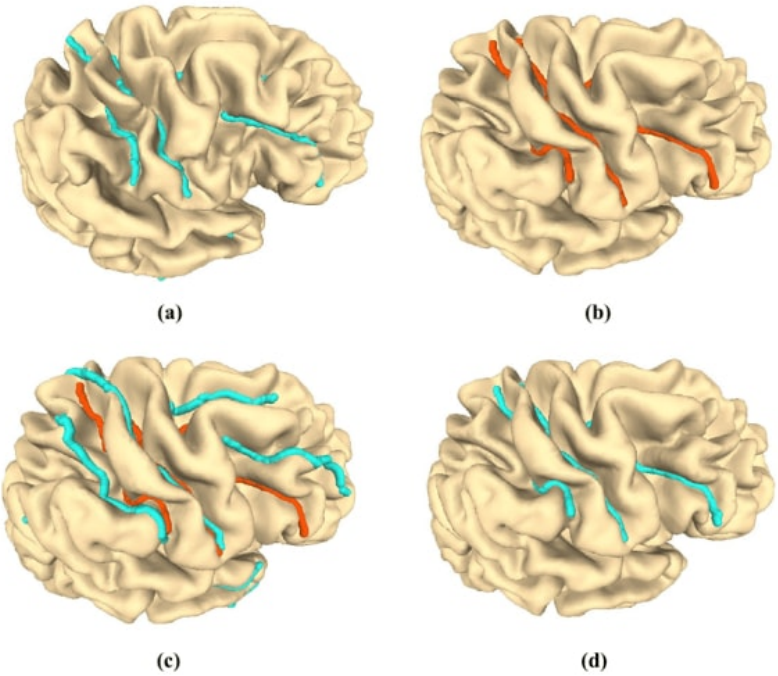

Figure 4.11: Two cortical surfaces and the registration results. (a) and (b) respectively show the source and the target surface. (c) shows the landmark-free conformal registration. One can observe that the landmark curves are not matched. (d) shows the registration obtained by *FLASH*. The landmarks are consistently aligned.

Surfaces	Algorithm in [58]			*FLASH*		
	Time spent for LM matching (s)	No. of flips	mean$\|\mu\|$	Time spent (s)	No. of flips	mean$\|\mu\|$
Brain 16 and 18	2585	24	0.0755	15.34	0	0.0669
Brain 21 and 35	2921	52	0.0647	16.95	0	0.0696
Brain 13 and 14	2449	133	0.0715	16.44	0	0.0799
Brain 32 and 36	2893	117	0.1047	9.28	0	0.1036
Brain 10 and 8	2837	231	0.0781	16.97	0	0.0822

Table 4.2: The statistics of the landmark aligned optimized conformal parameterizations by the algorithm [58] and our *FLASH* algorithm.

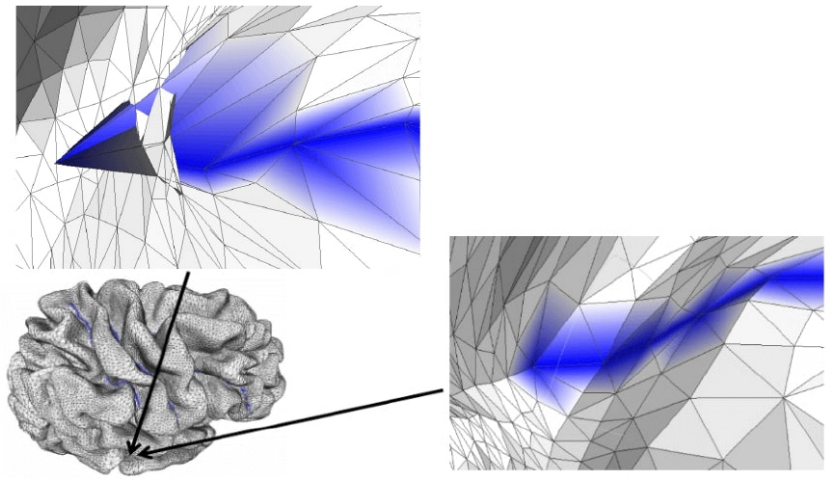

Figure 4.12: A comparison between the registration results of *FLASH* and the algorithm in [58]. Foldings are observed near a landmark curve (blue) in the result of [58] while our *FLASH* algorithm preserves the bijectivity of the registration. Top left: The result of [58]. Bottom right: The result of *FLASH*.

Algorithms	Time spent (s)	No. of overlaps	CDI	Landmark mismatch energy
Möbius transformation [33]	13591.88	0	0.0256	2718.19
Shape based landmark matching [59]	368.25	7	0.0663	0
FLASH ($\lambda = 3$)	5.1455	0	0.0205	113.70

Table 4.3: Comparison with other algorithms for landmark aligned cortical registration.

landmark aligned spherical parameterization using [58], we use our linear spherical conformal parameterization as the initialization. In practice, the conventional method in [58] initializes with a Gauss map. Thus, it would probably take much longer time to converge. The time required for landmark matching using [58], excluding the spherical conformal parameterizations of the brain surfaces, is over 40 minutes. On the contrary, the time required

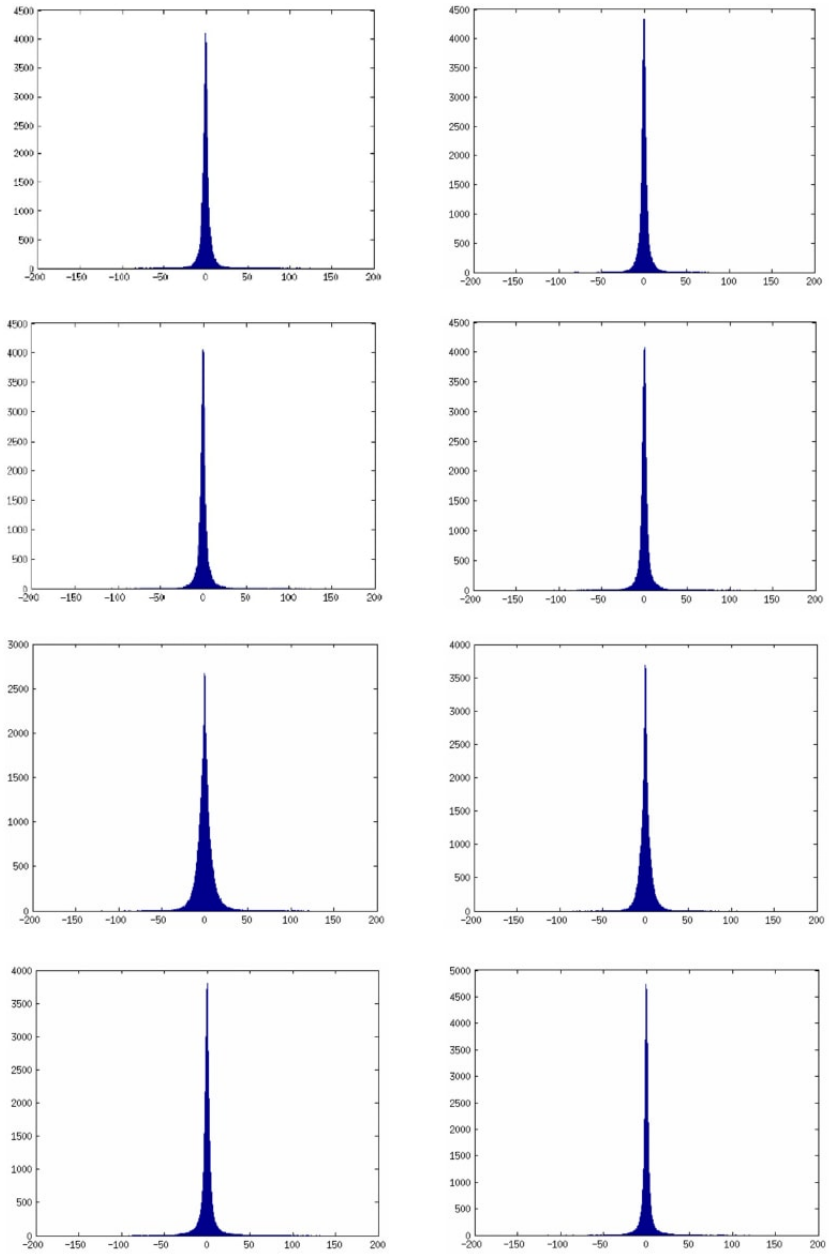

Figure 4.13: Histograms of angular distortions under the landmark aligned optimized conformal parameterizations obtained by the algorithm in [58] and those by *FLASH*. Each row represents one experiment. Left: The results of [58]. Right: The results of *FLASH*.

by *FLASH* (including the linear spherical conformal parameterizations of the brain surfaces) is usually less than 20 seconds. Our *FLASH* algorithm

accelerates the computation of landmark aligned spherical optimized conformal parameterization by 100 times.

Finally, we compare our proposed *FLASH* algorithm with two other models in [33] and [59] for landmark-aligned registration. PostCS, ITS and IFS are used as landmark constraints. Table 4.3 shows the statistics. Despite the small conformality distortions, the method of [33] suffers from slow computation and large landmark mismatch energy. For the method in [59], although exact landmark matching can be enforced, the conformality distortion is large and the bijectivity is not guaranteed. Our proposed *FLASH* algorithm outperforms the two mentioned algorithms.

4.5 Further applications

Besides cortical surface registration, our proposed *FLASH* algorithm can be further applied in a number of medical shape analysis problems. In this section, we outline two applications of our *FLASH* algorithm in medical shape analysis.

4.5.1 Shape analysis of carotid arteries

In our ongoing work [7], we aim to compare the thickness of the plaque of patients with carotid atherosclerosis before and after treatments in order to understand the effectiveness of the treatments. For each patient, 3D lumen and wall surfaces of the internal carotid artery (ICA), external carotid artery (ECA) and common carotid artery (CCA) at 2 time points are scanned. Figure 4.14 shows a carotid artery mesh. The vessel thickness of every carotid artery is measured. To perform the comparison of the thicknesses of different carotid arteries, we need to establish a 1-1 correspondence between the carotid arteries so that we can effectively study the intra-variation in thickness at

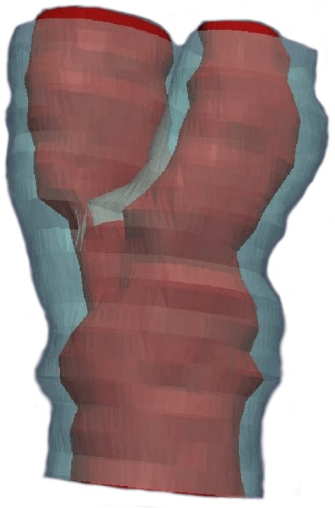

Figure 4.14: The lumen and the wall surface of a carotid artery.

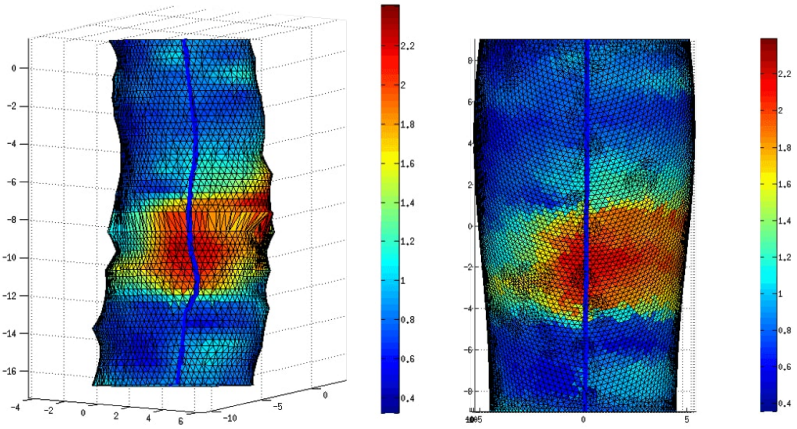

Figure 4.15: Mapping a carotid artery onto a template surface using *FLASH* with landmark constraints. Left: The side view of the carotid artery with a bulb line. Right: The side view of the mapping result with the corresponding bulb line constraint. The figures are colored with the vessel thickness data.

different time points of a single patient, as well as the inter-variation in thickness of different patients.

The major difficulty arises from the varying structures of the carotid arteries. To overcome the difficulty, we can consider mapping all carotid arteries onto a common template surface. Then the thickness data of all

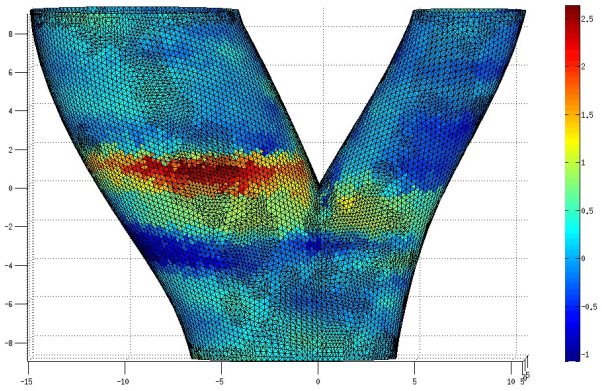

Figure 4.16: Visualizing the thickness difference of a carotid artery at two time points on a template surface.

carotid arteries can be represented on the common template surface, and this enables us to systematically perform the evaluation of the variations in thickness of the patients.

To accurately map each carotid artery surface onto a common template surface, landmark constraints of certain important features are needed. In this case, we use the artery bifurcation and two carotid bulb lines as the landmark constraints. This ensures that all carotid surfaces are consistently mapped onto the common template surface. Also, it is desirable that the local shapes of the carotid arteries are well-retained on the template surface. The above factors suggest the use of our *FLASH* algorithm. For the computation of the landmark constrained optimized conformal mappings, we apply the *FLASH* algorithm with a further projection step so as to enforce the longitudinal correspondences of the surfaces.

Figure 4.15 shows the mapping result obtained by our approach. The simple geometry of the template surface enables us to systematically analyze the regions with medical significance. In Figure 4.16, the thickness difference of a carotid artery at two time points is computed on the template surface. The

regions with a significant change in vessel thickness can be easily visualized.

4.5.2 Shape analysis of hippocampal surfaces

The hippocampus is a subcortical structure of the human brain which is closely related to the Alzheimer disease. Numerous studies have been devoted to the local shape analysis of the hippocampal surfaces. In particular, establishing a 1-1 correspondence between two hippocampal surfaces is challenging.

As the local shapes are important in analyzing the hippocampal surfaces, the registration between two hippocampal surfaces should preserve them. Hence, the conformality distortion of the registration should be as low as possible. Also, the distribution of the registration result should be sufficiently even. In other words, if we divide the hippocampal surfaces into several regions, every pair of corresponding regions should be roughly matched under the registration. The use of landmark constraints can help fulfilling this requirement. Therefore, our *FLASH* algorithm for landmark aligned optimized conformal maps can be applied.

First, we compute the Laplace-Beltrami operators of the two surfaces. Then, by computing the eigenfunctions of the LB operators, we can obtain numerous eigenloops on the two hippocampal surfaces. The eigenloops effectively divide the hippocampal surfaces into several parts. To ensure the accuracy of the registration, the eigenloops are used as landmark constraints in our *FLASH* algorithm.

Figure 4.17 shows the registration result of two hippocampal surfaces. It can be observed that the landmark curves are roughly matched on a common domain. With the registration result, systematic analyses of the difference between the two hippocampal surfaces become possible.

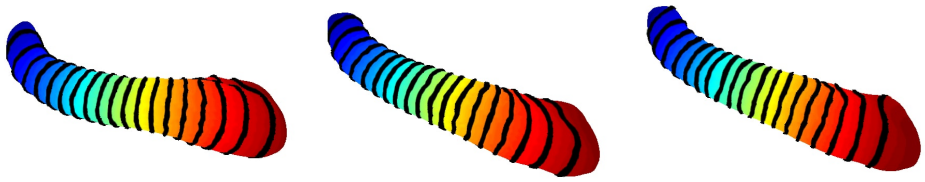

Figure 4.17: Registering two hippocampal surfaces using *FLASH*. Left: The source surface. Middle: The target surface. Right: The registration result. The landmark curves are highlighted.

Chapter 5

Disk conformal parameterization of simply-connected open meshes

5.1 Introduction

In this chapter, we present our two proposed algorithms in [2, 3] for efficiently computing the disk conformal parameterization of simply-connected open meshes.

Conformal parameterization of disk-type surfaces has been widely studied by numerous research groups. The two major challenges of the computation are the efficiency and the accuracy. Firstly, most of the existing algorithms are nonlinear and hence inefficient. This hinders the practical use of the parameterization especially when a large number of surfaces are involved. Secondly, the conformality distortion is far from negligible in the discrete case. The distortion affects the accuracy of the parameterizations. Our proposed algorithms aim to overcome these two problems.

5.2 Contributions

The two disk conformal parameterization algorithms we invent are with the following advantages:

(i) Our proposed algorithms are highly efficient. The first algorithm requires only a few iterations, and the second proposed algorithm is linear. The algorithms significantly accelerate the computation of disk conformal parameterization.

(ii) The conformality of the disk parameterizations obtained by our algorithms is largely improved when compared with the pre-existing methods.

(iii) The bijectivity of the disk conformal parameterizations is theoretically supported by quasi-conformal theory.

5.3 Two-step iterative scheme for disk conformal parameterization

In this section, we describe our proposed method in [2] for the disk conformal parameterization of a simply-connected open surface M. The disk conformal parameterization is achieved with the aid of an efficient iterative algorithm. In [1], the North Pole-South Pole iterative scheme was introduced for computing the spherical conformal parameterization of genus-0 closed surfaces in linear time. The main idea of the iterative scheme is to improve the conformality distortions near the north pole and the south pole of the spherical parameterizations step by step. In the "north pole" step, a genus-0 closed surface is mapped to $\mathbb{S}^2$ using an efficient but not highly accurate method. The conformality distortion near the south pole of the sphere is

small while the distortion of the northernmost region is large. After that, in the "south pole" step, the conformality distortion near the north pole is corrected, with the southernmost region kept fixed. In other words, to achieve a globally conformal parameterization, one can try to ensure the conformality of one part first, and then obtain the conformality of the other part in the second step, with the aid of the conformal part obtained before. Motivated by this idea, we introduce a two-step iterative scheme for disk conformal parameterizations of simply-connected open surfaces. By an abuse of terminology, the two steps are called the "north pole" step and the "south pole" step. The names does not imply that the actual geometric poles of the unit disk are used. Table 5.1 highlights the features of and the comparisons between the linear spherical conformal parameterization [1] and our proposed method.

Our proposed method consists of three stages: 1) initialization, 2) "north pole" step, 3) "south pole" iteration. The three stages will be described in the following subsections.

5.3.1 Initialization: The discrete harmonic map

In the first stage, we compute an initial map for the disk parameterizations. In [34], Gu and Yau described a simple method to compute a disk harmonic map $f : M \to \mathbb{D}$ by solving the following Laplace equation:

$$\begin{cases} \Delta_M f(u) = 0 & \text{if } u \in M \setminus \partial M \\ f|_{\partial M} = g \end{cases} \tag{5.1}$$

where $g : \partial M \to \partial \mathbb{D}$ is given by the arc-length parameterization.

In the discrete case, the surface M is represented by a triangulated mesh. The Laplace equation $\Delta_M f(u) = 0$ in Equation (5.1) is a sparse symmetric positive definite linear system. The arc-length parameterization boundary

Features	Linear Spherical Conformal Parameterization [1]	Our two-step iterative scheme
Topology	Spherical	Disk
"North pole" step	Use the stereographic projection and work on $\mathbb{C}$	Use the Cayley transform and work on the upper half plane
"South pole" step	South pole stereographic projection and compute a QC map	Reflection along the unit circle with iterations
Boundary adjustment	No	Yes
Output	Unit Sphere	Unit disk
Bijectivity	Yes	Yes

Table 5.1: Features of the Fast Spherical Conformal Parameterization [1] and our two-step iterative scheme in [2].

constraint is computed using the ratio of the edge lengths:

$$f(v_i) = (\cos\theta_i, \sin\theta_i), \tag{5.2}$$

where $l_{[v_i,v_{i+1}]}$ denotes the length of the edge $[v_i, v_{i+1}]$ and

$$\begin{cases} s := \sum_{i=0}^{n-1} l_{[v_i,v_{i+1}]} \\ s_i := \sum_{j=0}^{i-1} l_{[v_j,v_{j+1}]} \\ \theta_i := 2\pi \dfrac{s_i}{s}. \end{cases} \tag{5.3}$$

Hence, the Laplace Equation (5.1) can be efficiently solved. Note that in general, the harmonic map with Dirichlet boundary condition is not conformal. Nevertheless, the disk map computed can serve as an efficient initialization. The conformality distortions of the interior region and the boundary region of the disk will be corrected by the two-step iterative scheme described below.

5.3.2 "North pole" step: Improvement of conformality on the upper half plane

To improve the conformality of the initial disk harmonic map f, our strategy is to compose f with a specific quasi-conformal map $g : \mathbb{D} \to \mathbb{D}$ as described in Theorem 4.3.1. More explicitly, if the Beltrami coefficients of f^{-1} and g, denoted by $\mu_{f^{-1}}$ and μ_g, are equal, then the composition map $g \circ f$ is conformal. The computation of g can be achieved using the Linear Beltrami Solver (LBS) [61], together with a suitable boundary condition.

However, since the boundary of $f(M)$ is given by the arc-length parameterization, currently we do not have any optimal boundary correspondence $g|_{\partial\mathbb{D}}$ for the computation of g. Finding such optimal correspondence is highly nonlinear. To solve this problem, we first transform the current domain to another domain, such that the problem can be linearized. More specifically, we apply the Cayley transform to map the unit disk $\mathbb{D}$ onto the upper half plane. The Cayley transform is a bijective conformal map $W : \mathbb{D} \to \mathbb{H} = \{x + iy | y \geq 0; x, y \in \mathbb{R}\}$ such that

$$W(z) = i\frac{1+z}{1-z}. \tag{5.4}$$

Using the Cayley transform, our problem is simplified to finding a quasi-conformal map $h : \mathbb{H} \to \mathbb{H}$ with $\mu_h = \mu_{(W \circ f)^{-1}}$. The composition map $h \circ W \circ f : M \to \mathbb{H}$ will then be conformal as suggested by Theorem 4.3.1.

Note that under the Cayley transform, the boundary of the disk is mapped onto the real axis $y = 0$. Hence, to enforce a circular boundary of the disk parameterization, we only need to enforce that h maps the real axis to the real axis. Equivalently, we only need to restrict $v = 0$ on $\partial\mathbb{H}$ while solving equation (3.12) and do not set any restriction on u. The desired map h can be computed by solving two separate elliptic equations for u and v.

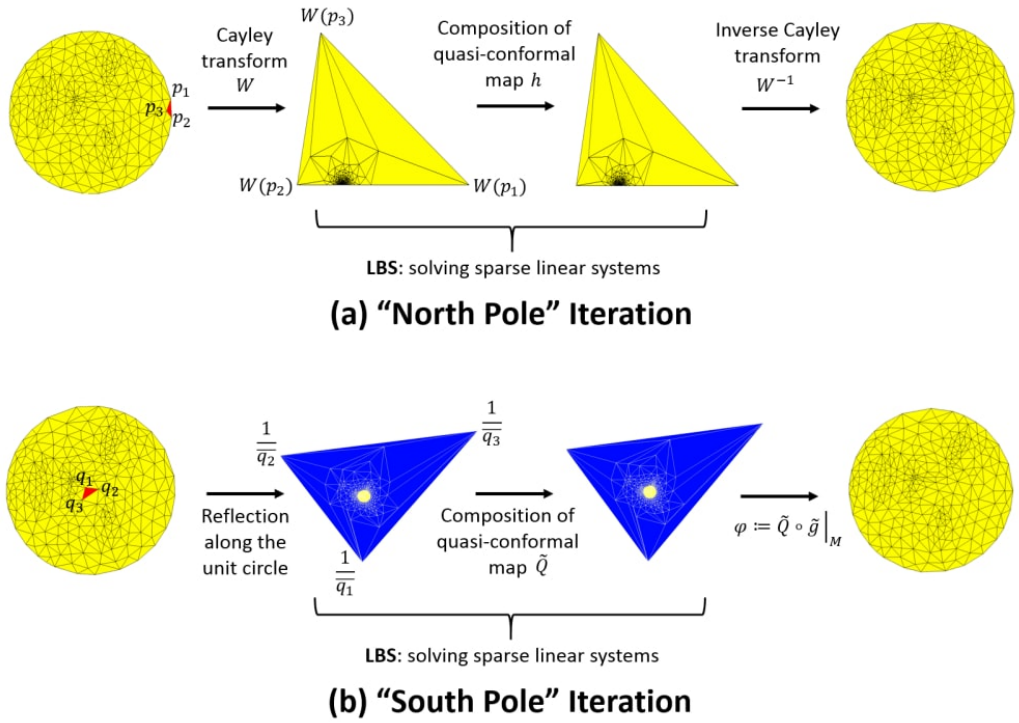

Figure 5.1: An illustration of (a) the "north pole" step and (b) the "south pole" step.

In the discrete case, the initial parameterization f maps M onto $\mathbb{D}$. In the abovementioned "north pole" step, the Cayley transform W maps $\mathbb{D}$ to a big triangle in $\mathbb{H}$. Denote the three vertices of the big triangle by $W(p_1)$, $W(p_2)$ and $W(p_3)$, where $[p_1, p_2, p_3]$ is the triangular face of $\mathbb{D}$ enclosing the point $z = 1 \in \mathbb{D}$ (see Figure 5.1(a)). To compute the desired quasi-conformal map h, we solve Equation (3.12) with the three vertices fixed, and that the vertices on the real axis stay on the real axis. In other words, we solve the following equation

$$\begin{cases} h = \mathbf{LBS}(\mu_{(W \circ f)^{-1}}) \\ h(W(p_i)) = W(p_i) \text{ for } i = 1, 2, 3 \\ \mathbf{Im}(h(W(z))) = 0 \text{ for any } z \in \partial\mathbb{D}. \end{cases} \tag{5.5}$$

The above equation can be formulated as two sparse symmetric positive

linear systems for the two coordinates. The systems can be solved efficiently using the conjugate gradient method.

After that, we map $\mathbb{H}$ (or the big triangle in the discrete case) back to the unit disk using the inverse Cayley transform

$$W^{-1}(z) = \frac{z-i}{z+i}, \tag{5.6}$$

which is conformal. As we enforce the boundary vertices to stay on the real axis under the map h, the corresponding boundary vertices under the composition map

$$g := W^{-1} \circ h \circ W \circ f \tag{5.7}$$

will still be on $\partial\mathbb{D}$ and hence the circular boundary condition is preserved.

In summary, the conformality distortion of the inner region of $\mathbb{D}$ is significantly improved with the aid of the quasi-conformal map h. Also, the additional freedom on the boundary vertices slightly alleviates the conformality distortion near the disk boundary. Hence, the composition map g is a disk parameterization with improved conformality. A geometric illustration of the proposed "north pole" step is shown in Figure 5.1(a). The conformality near the disk boundary is further improved in the "south pole" step.

5.3.3 "South pole" step: Correction of boundary conformality distortion via reflection

In the "south pole" step, we aim to correct the conformality distortion near the disk boundary.

In the "north pole-south pole" iterative scheme [1], the "south pole" step improves the conformality near the north pole, with the southernmost region, which is the most conformal part, fixed as the boundary constraints. We extend this idea to the case of disk conformal map. Recall that the

conformality of the inner region of the disk is significantly improved by the "north pole" step as described in subsection 5.3.2. Thus, the innermost region of the disk can serve as an appropriate boundary constraint for computing an accurate disk conformal map. our strategy is to fix this least distorted region and correct the conformality distortion near the boundary using a quasi-conformal map.

Similar to the last subsection 5.3.2, we make use of the LBS [61] to construct a quasi-conformal map. However, it is noteworthy that the composition of a quasi-conformal map can only significantly reduce the distortion at the inner region but not near the boundary. Hence, the idea in the last subsection cannot be directly applied in the "south pole" step. To solve this problem, we consider enlarging the domain so that boundary region of $\mathbb{D}$ becomes the inner region of a much bigger domain. If the boundary region of the enlarged domain is appropriate, then the conformality of the whole unit disk can be improved using the composition of QC maps.

To construct such appropriate enlarged domain, we conformally reflect the unit disk along the circular boundary. In other words, we construct a copy of the points on $\mathbb{D} \setminus \partial\mathbb{D}$ outside $\mathbb{D}$ by the correspondence

$$z \in \mathbb{D} \setminus \partial\mathbb{D} \longleftrightarrow \frac{1}{\bar{z}} \in \overline{\mathbb{C}} \setminus \mathbb{D}, \tag{5.8}$$

Now, the new domain becomes the whole complex plane $\overline{\mathbb{C}}$. In the discrete case, the new domain is a big triangular domain, with the three outermost vertices being the reflected vertices of the innermost triangle on $\mathbb{D}$ that contains the origin. The whole unit disk is now located at the inner region of the new big triangle (see Figure 5.1b). It remains to perform the composition of QC maps on the enlarged domain.

To achieve this task, Lui et al. [60] introduced the following extension of

a diffeomorphism on $\mathbb{D}$ to $\overline{\mathbb{C}}$ through reflection.

Theorem 5.3.1. *Let $f : \mathbb{D} \to \mathbb{D}$ be a diffeomorphism of the unit disk fixing 0 and 1 and satisfying the Beltrami equation $f_{\bar{z}} = \mu_f f_z$ with μ_f defined on $\mathbb{D}$. Then an extension of f from $\mathbb{D}$ to $\overline{\mathbb{C}}$ given by*

$$\tilde{f}(z) = \begin{cases} f(z) & \text{if } |z| \leq 1 \\ \frac{1}{\overline{f(1/\bar{z})}} & \text{if } |z| > 1 \end{cases} \tag{5.9}$$

satisfies the Beltrami Equation $\tilde{f}_{\bar{z}} = \tilde{\mu}_{\tilde{f}} \tilde{f}_z$ on $\overline{\mathbb{C}}$, where

$$\tilde{\mu}_{\tilde{f}}(z) = \begin{cases} \mu_f(z) & \text{if } |z| \leq 1 \\ \frac{z^2}{\bar{z}^2} \overline{\mu_f(1/\bar{z})} & \text{if } |z| > 1. \end{cases} \tag{5.10}$$

Proof. See [60]. □ □

Using Equation (5.9), we extend the map g in Equation (5.7) to the extended map $\tilde{g} : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ defined by

$$\tilde{g}(z) = \begin{cases} g(z) & \text{if } z \in \mathbb{D} \\ \frac{1}{\overline{g(1/\bar{z})}} & \text{if } z \in \overline{\mathbb{C}} \setminus \mathbb{D}. \end{cases} \tag{5.11}$$

Since the conformality distortion of the innermost region of $\mathbb{D}$ is small, the corresponding outermost region of the new domain is also with negligible conformality distortion. Hence, we apply the LBS [61] to compose the map $\tilde{g}$ by a quasi-conformal map $\widetilde{Q} : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ with the Beltrami coefficient $\tilde{\mu}_{\tilde{g}^{-1}}$, fixing the outermost region of the new domain:

$$\begin{cases} \widetilde{Q} = \mathbf{LBS}(\tilde{\mu}_{\tilde{g}^{-1}}) \\ \widetilde{Q}(z) = z \text{ for } |z| \gg 1. \end{cases} \tag{5.12}$$

The conformality of $\tilde{g}$ is significantly improved as suggested by Theorem 4.3.1. In particular, the conformality distortion near the boundary of the original unit disk will be alleviated. Besides, by Theorem 5.3.1, the region corresponding to $\mathbb{D}$ is exactly mapped onto $\mathbb{D}$ under the map $\widetilde{Q}$. Hence, the

boundary of the region is exactly circular in the continuous case. This results in a disk conformal parameterization

$$\varphi := \widetilde{Q} \circ \widetilde{g}|_M, \tag{5.13}$$

with improved conformality at both the boundary region and the inner region. Moreover, we have $\|\mu_\varphi\|_\infty < 1$. By Theorem 3.3.2, φ is bijective.

In the discrete case, denote the big triangular domain obtained under the reflection by Ω. As the Beltrami coefficient is piecewise constant on every triangular face T in $\mathbb{D}$, Equation (5.10) cannot be directly applied for computing the Beltrami coefficient $\tilde{\mu}_{\widetilde{g}^{-1}}(\widetilde{T})$ on the reflected triangular faces $\widetilde{T}$ on $\Omega \setminus \mathbb{D}$. Instead, we approximate $\tilde{\mu}_{\widetilde{g}^{-1}}(\widetilde{T})$ by

$$\tilde{\mu}_{\widetilde{g}^{-1}}(\widetilde{T}) = \frac{(\overline{z_1}^2/z_1^2 + \overline{z_2}^2/z_2^2 + \overline{z_3}^2/z_3^2)}{3} \overline{\mu_{g^{-1}}(T)}, \tag{5.14}$$

where $T = [z_1, z_2, z_3]$. This approximation unavoidably introduces numerical errors. Hence, the boundary of $\mathbb{D}$ may not be transformed to a perfect circle after the "south pole" step. To solve this problem, we normalize the boundary vertices by

$$z \mapsto \frac{z}{|z|}. \tag{5.15}$$

Then, we repeat the "south pole" step until convergence. The detailed implementation of our proposed method is described in Algorithm 4.

5.4 Linear disk conformal parameterization algorithm

In this section, we present our proposed linear disk conformal parameterizations algorithm in [3]. Given a simply-connected open surface M, we aim to efficiently compute a conformal map $f : M \to \mathbb{D}$, where $\mathbb{D}$ denotes the open unit disk.

Algorithm 4: Two-step iterative scheme for disk conformal parameterization

Input: A simply-connected open mesh M, an energy threshold ϵ.
Output: A bijective disk conformal parameterization $\varphi : M \to \mathbb{D}$.

1 Denote the boundary of M as $\partial M = [v_0, v_1, ..., v_n]$. Compute the edge lengths $l_{[v_i,v_{i+1}]}$ for $i = 0, 1, ..., n$, where $v_{n+1} := v_0$;
2 Obtain an initial disk parameterization $f : M \to \mathbb{D}$ by

$$\begin{cases} \sum_{v \in N(u)} k_{uv}(f(u) - f(v)) = 0 & \text{if } u \notin \partial M \\ f(v_i) = (\cos\theta_i, \sin\theta_i) & \text{if } u = v_i \in \partial M \end{cases}$$

where $s := \sum_{i=0}^{n-1} l_{[v_i,v_{i+1}]}$, $s_i := \sum_{j=0}^{i-1} l_{[v_j,v_{j+1}]}$ and $\theta_i := 2\pi s_i/s$;
3 Apply the Cayley transform $W : \mathbb{D} \to \mathbb{H}$ defined by $W(z) = i\frac{1+z}{1-z}$;
4 Compute the Beltrami coefficient $\mu_{(W\circ f)^{-1}}$ of the map $(W \circ f)^{-1}$;
5 Compute the quasi-conformal map $h = \mathbf{LBS}(\mu_{(W\circ f)^{-1}})$ with the boundary vertices $(W \circ f)(v_i)$ restricted on the real axis;
6 Project the upper half plane to the unit disk by the inverse Cayley transform $W^{-1}(z) = \frac{z-i}{z+i}$. Denote $\varphi := W^{-1} \circ h \circ W \circ f$;
7 **repeat**
8 Update ν by the Beltrami coefficient $\mu_{\varphi^{-1}}$ of the map φ^{-1};
9 By reflection, extend φ^{-1} and $\mu_{\varphi^{-1}}$ on $\mathbb{D}$ to $\widetilde{\varphi}^{-1}$ and $\tilde{\mu}_{\widetilde{\varphi}^{-1}}$ on a big triangular domain B using Equation (5.9) and Equation (5.10). For each face $T = [z_1, z_2, z_3]$ on $\mathbb{D}$, define $\tilde{\mu}_{\widetilde{\varphi}^{-1}}(\widetilde{T}) = \frac{(\overline{z_1}^2/z_1^2 + \overline{z_2}^2/z_2^2 + \overline{z_3}^2/z_3^2)}{3}\overline{\mu_{\varphi^{-1}}(T)}$;
10 Compute the quasi-conformal map $\widetilde{Q} = \mathbf{LBS}(\tilde{\mu}_{\widetilde{\varphi}^{-1}})$ with the outermost vertices of B fixed;
11 Update φ by the restriction $\widetilde{Q} \circ \widetilde{\varphi}|_M$;
12 Project the boundary of $\varphi(M)$ onto the unit circle;
13 **until** $mean(|\mu_{\varphi^{-1}}|) - mean(|\nu|) < \epsilon$;

5.4.1 Finding an initial map via double covering

We tackle the problem using a simple double covering technique suggested in [32]. In the following, we introduce the double covering technique. The technique aims to transform the simply-connected open surface M to a genus-0 closed surface $\widetilde{M}$. First, we duplicate M and change its orientation. Denote

the new copy by M'. Then, we identify the boundaries of M and M':

$$\partial M \longleftrightarrow \partial M'. \tag{5.16}$$

By the above identification, M and M' are glued along the two boundaries. Note that we do not identify the interior parts of M and M'. As a result, a closed surface $\widetilde{M}$ is formed. Since M and M' are simply-connected, $\widetilde{M}$ is a genus-0 closed surface. More explicitly, denote the Gaussian curvature and geodesic curvature by K and κ_g respectively. Assume that we have slightly modified the boundary parts of M and M' so that $\widetilde{M}$ is smooth. Then by the Gauss-Bonnet theorem, we have

$$\int_M K dA + \int_{\partial M} \kappa_g ds = 2\pi\chi(M) = 2\pi \tag{5.17}$$

and

$$\int_{M'} K dA + \int_{\partial M'} \kappa_g ds = 2\pi\chi(M') = 2\pi. \tag{5.18}$$

Hence, we have

$$\begin{aligned} 2\pi\chi(\widetilde{M}) &= \int_{\widetilde{M}} K dA \\ &= \int_M K dA + \int_{M'} K dA \\ &= \left(2\pi - \int_{\partial M} \kappa_g ds\right) + \left(2\pi - \int_{\partial M'} \kappa_g ds\right) \\ &= 4\pi - \int_{\partial M} \kappa_g ds + \int_{\partial M} \kappa_g ds \\ &= 4\pi. \end{aligned} \tag{5.19}$$

Therefore, $\widetilde{M}$ has Euler characteristic $\chi(\widetilde{M}) = 2$, which implies that it is a genus-0 closed surface.

The abovementioned technique is applicable in the discrete case. Let $K = (V, E, F)$ be a simply-connected open triangulated mesh with the vertex set V, the edge set E and the triangular face set F. Each face in F can be

represented as an ordered triple $[u, v, w]$ where u, v, w are three vertices. Let v, e, f be the number of vertices, edges and faces of K respectively. Denote the boundary vertices of K by $\{w_i\}_{i=1}^{r}$, and let $K' = (V', E', F')$ be a duplication of K. The abovementioned notations are naturally extended to K'. By Euler's formula, we have

$$v - e + f = v' - e' + f' = 1. \tag{5.20}$$

To reverse the orientation of K', we rearrange the order of the vertices of each face in F' from $[u, v, w]$ to $[u, w, v]$. Then, to glue the two surfaces K and K', we identify the boundary vertices:

$$w_i \longleftrightarrow w_i' \text{ for all } i = 1, 2, \cdots, r. \tag{5.21}$$

Now, denote the number of vertices, edges and faces of the glued mesh $\widetilde{K}$ by $\tilde{v}$, $\tilde{e}$ and $\tilde{f}$ respectively. We have

$$\begin{aligned} & \tilde{v} - \tilde{e} + \tilde{f} \\ = \; & (v + v' - r) - (e + e' - r) + (f + f') \\ = \; & v + v' - e - e' + f + f' \\ = \; & 2, \end{aligned} \tag{5.22}$$

Hence, $\widetilde{M}$ is a genus-0 closed mesh. The implementation of the double covering procedure is described in Algorithm 5.

After obtaining $\widetilde{M}$ by double covering, we look for a conformal map that maps $\widetilde{M}$ to the unit sphere. The linear spherical conformal parameterization algorithm in [1] is applied because of its efficiency and accuracy. It is noteworthy that the combination of double covering and the linear spherical conformal parameterization algorithm in [1] is particularly advantageous. Because of the symmetry of the double covered surface, half of the entries in the coefficient matrix of the discretization of the Laplace Equation (4.2) are duplicated. Therefore, although we have doubled the problem scale with the

Algorithm 5: The double covering technique in our initial step.

Input: A simply-connected open triangulated mesh $K = (V, E, F)$.
Output: A genus-0 closed mesh $\widetilde{K} = (\widetilde{V}, \widetilde{E}, \widetilde{F})$.

1 Duplicate K and denote the copy by $K' = (V', E', F')$;
2 Change the order of the vertices of each face in F' from $[u, v, w]$ to $[u, w, v]$;
3 Replace the boundary vertices w_i' by w_i in E' and F' for $i = 1, 2, \cdots, r$;
4 Set $\widetilde{V} = V \cup V' \setminus \{w_i'\}_{i=1}^{r}$;
5 Set $\widetilde{E} = E \cup E'$;
6 Set $\widetilde{F} = E \cup E'$;

double covering technique, we only need to compute half of the entries in the coefficient matrix and hence we can save half of the computational cost.

After finding a spherical conformal map $\tilde{f} : \widetilde{M} \to \mathbb{S}^2$, note that by symmetry, we can divide $\mathbb{S}^2$ into two parts, each of which exactly corresponds to one of M and M'. Since our goal is to find a disk conformal map $f : M \to \mathbb{D}$, we focus on only one of the two parts. Now, we apply a Möbius transformation on $\mathbb{S}^2$ so that the two parts are exactly the northern and southern hemispheres of $\mathbb{S}^2$. Then, we can apply the stereographic projection P_N to the southern hemisphere. This gives us an open unit disk $\mathbb{D}$. Since the Möbius transformation and the stereographic projection are both conformal, the combination of the above steps results in a conformal map $f : M \to \mathbb{D}$.

In the discrete case, due to irregular triangulations of the meshes and the conformality distortions of the spherical map, the boundary is usually different from a perfect circle. In other words, the planar region $R := f(M)$ we obtained after applying the stereographic projection may not be a unit disk. To solve this problem, we need one further step to enforce the circular boundary. At the same time, we need to maintain the low conformality distortions and preserve the bijectivity of the parameterization.

5.4.2 Enforcing the circular boundary to achieve a bijective disk conformal parameterization

To control the conformality distortion and the bijectivity, our strategy is to normalize the boundary and then compose the map with a quasi-conformal map. We first normalize the boundary of the region R by

$$v \mapsto \frac{v}{|v|} \tag{5.23}$$

for all $v \in \partial R$. Denote the normalized region by $\widetilde{R}$. Since the vertices near ∂R may be very dense, a direct normalization of the boundary may cause overlaps as well as conformality distortions. To eliminate the overlaps and the distortions of $\widetilde{R}$, we apply the Linear Beltrami Solver (LBS) [61]. More specifically, denote the Beltrami coefficient of the mapping $g : \widetilde{R} \to M$ from the normalized planar region to the original surface by μ. We compute a quasi-conformal map $h : \widetilde{R} \to \mathbb{D}$ by

$$h = \mathbf{LBS}(\mu) \tag{5.24}$$

with the normalized circular boundary constraint $h(v) = v$ for all $v \in \partial\widetilde{R}$.

By Theorem 4.3.1, $h \circ g^{-1}$ is a conformal parameterization from the original surface M to the unit disk $\mathbb{D}$. Also, since the Beltrami coefficient of $h \circ g^{-1}$ is with supremum norm less than 1, the bijectivity of $h \circ g^{-1}$ is guaranteed by Theorem 3.3.2. The complete implementation of our linear disk conformal parameterization algorithm is described in Algorithm 6.

5.5 Experimental Results

5.5.1 Two-step iterative scheme

In this section, we demonstrate the effectiveness of our proposed two-step iterative scheme in Section 5.3 using various meshes. The meshes are freely

Algorithm 6: Linear disk conformal parameterization algorithm

Input: A simply-connected open mesh M.
Output: A bijective disk conformal parameterization $f : M \to \mathbb{D}$.

1 Double cover M and obtain a genus-0 closed mesh $\widetilde{M}$;
2 Apply the linear spherical conformal parameterization [1] on $\widetilde{M}$ and obtain the parameter sphere S;
3 Apply a Möbius transformation on S so that the original surface M corresponds to the southern hemisphere of S;
4 Apply the stereographic projection for the southern hemisphere of S and obtain a planar region R;
5 Normalize the boundary of R by

$$v \mapsto \frac{v}{|v|} \tag{5.25}$$

for all $v \in \partial R$ and denote the normalized region by $\widetilde{R}$;
6 Compute the Beltrami coefficient of the map $g : \widetilde{R} \to M$ and denote it by μ;
7 Apply the Linear Beltrami Solver [61] to obtain a map $h : \widetilde{R} \to \mathbb{D}$

$$h = \mathbf{LBS}(\mu) \tag{5.26}$$

with the boundary constraint $h(v) = v$ for all $v \in \partial\widetilde{R}$. The composition map $f = h \circ g^{-1}$ is the desired disk conformal parameterization;

adopted from the AIM@SHAPE Shape Repository [89]. The algorithm is implemented in MATLAB on Windows 7 platform. All experiments are performed on a PC with a 3.40 GHz CPU. The error threshold in Algorithm 4 is set to be $\epsilon = 10^{-5}$.

Figure 5.2, Figure 5.3 and Figure 5.4 show three meshes and the disk conformal parameterizations obtained by our proposed two-step iterative scheme. The histograms of the norms of the Beltrami coefficients are given in Figure 5.5. It can be observed that the peaks of the norms are close to 0, which implies that the conformality distortions are negligible. Besides, as suggested by the energy plots in Figure 5.6, our proposed method converges

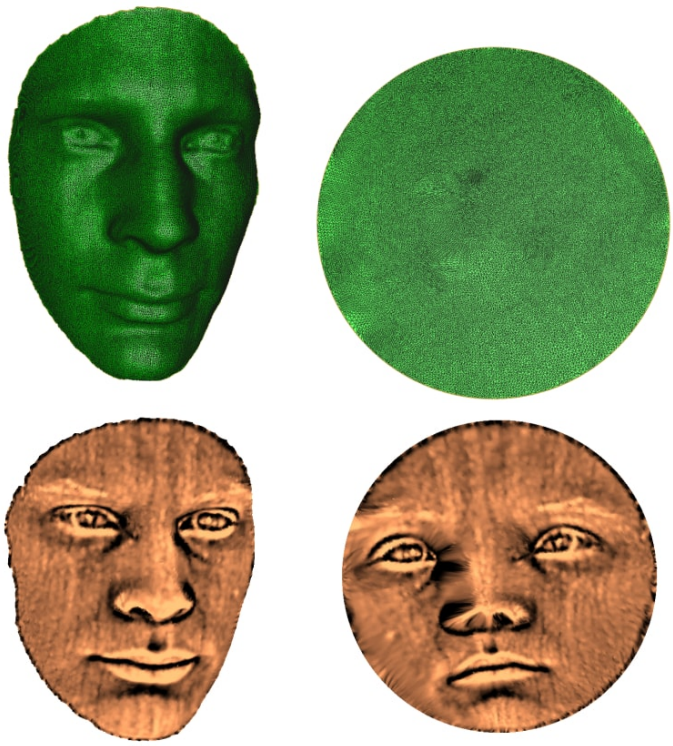

Figure 5.2: A human face and its disk conformal parameterization using our proposed two-step iterative scheme. Top: the triangulations. Bottom: the mean curvature texture maps.

shortly.

To quantitatively assess the quality of our proposed two-step iterative scheme, we consider three different factors including the computational time, the mean of the norm of the Beltrami coefficients, and the standard deviation. We compare our two-step iterative scheme with four state-of-the-art algorithms that guarantee bijectivity and enforce a circular boundary. The algorithms are the discrete Ricci flow (RF) algorithm [44], the inversive distance Ricci flow (IDRF) algorithm [85], the Yamabe Riemann map algorithm [64] and the holomorphic 1-form algorithm [31]. The statistics of the performance of our proposed two-step iterative scheme and the four algorithms are listed in

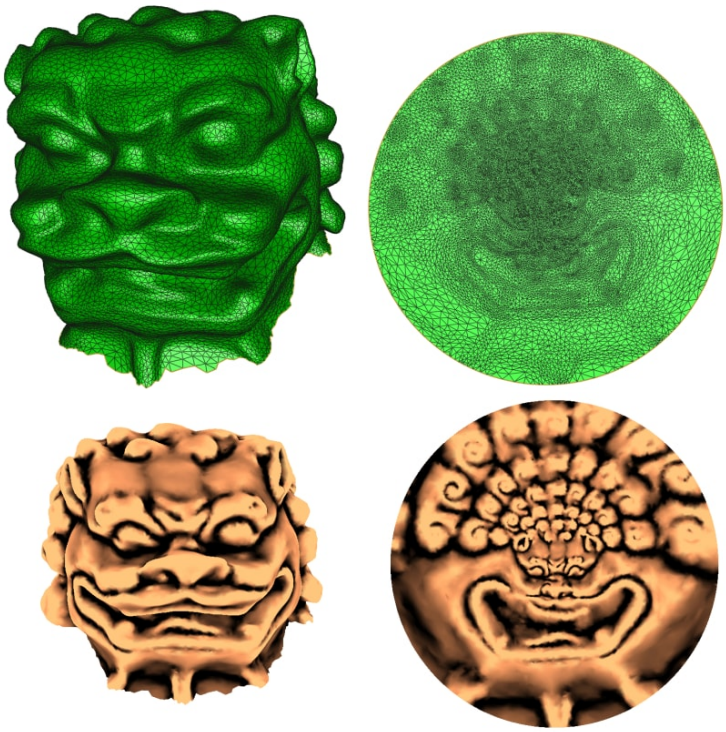

Figure 5.3: A Chinese lion head and its disk conformal parameterization using our two-step iterative scheme. Top: the triangulations. Bottom: the mean curvature texture maps.

Table 5.2.

As shown in Table 5.2, our proposed two-step iterative scheme is highly efficient and accurate. For a 3D mesh with 100k triangular faces, the time spent is usually less than 10 seconds. Also, our method works for meshes with irregular shapes and bad triangulations, such as the example shown in Figure 5.7. As a remark, in all our experiments, the resulting disk conformal parameterizations are bijective. In other words, the parameterizations do not contain any flips or overlaps.

For a more detailed comparison, the computational time of our two-step iterative scheme is respectively 60%, 75%, 80%, and 90% shorter than that

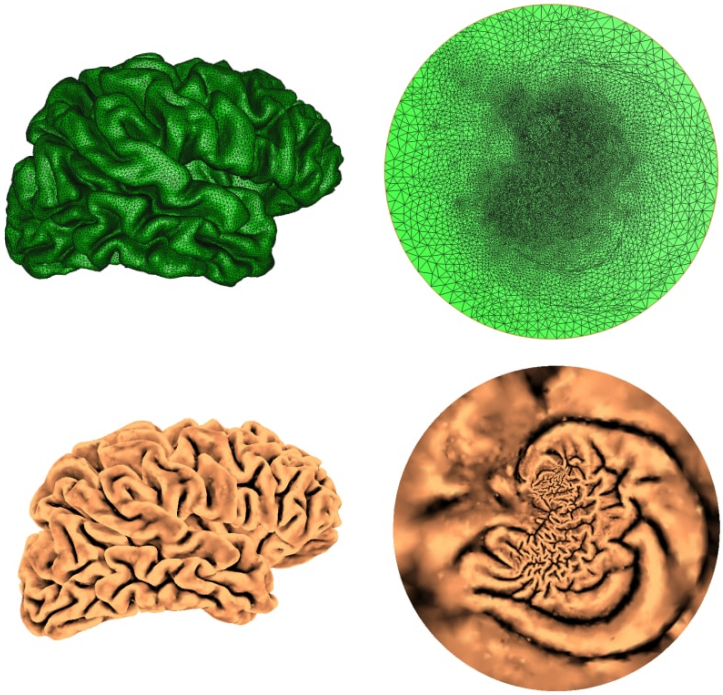

Figure 5.4: A human brain and its disk conformal parameterization using our two-step iterative scheme. Top: the triangulations. Bottom: the mean curvature texture maps.

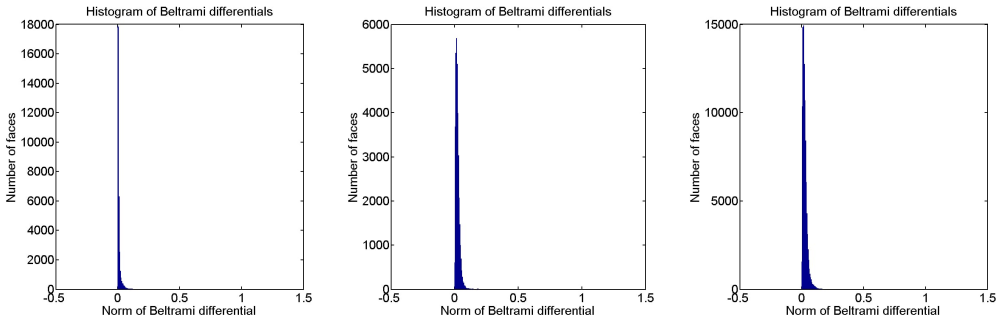

Figure 5.5: Histograms of the norm of Beltrami coefficients $|\mu|$ of our proposed two-step iterative scheme for a human face mesh, a Chinese lion head mesh and a human brain mesh. Left: Human face. Middle: Chinese lion head. Right: Human brain.

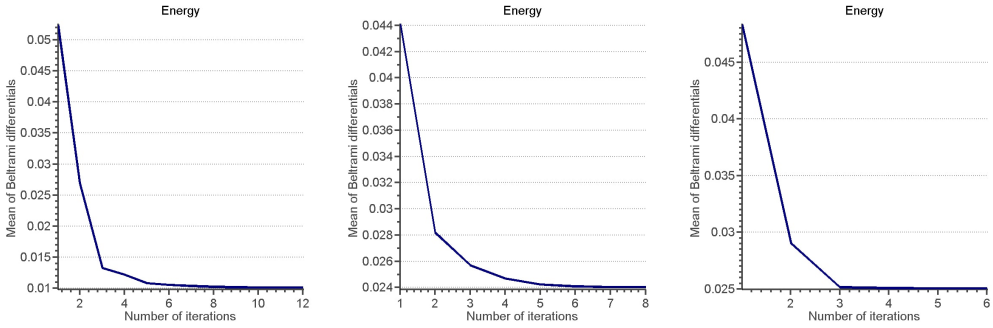

Figure 5.6: Energy plots of mean($|\mu|$) of our two-step iterative scheme for a human face mesh, a Chinese lion head mesh and a human brain mesh. Left: Human face. Middle: Chinese lion head. Right: Human brain.

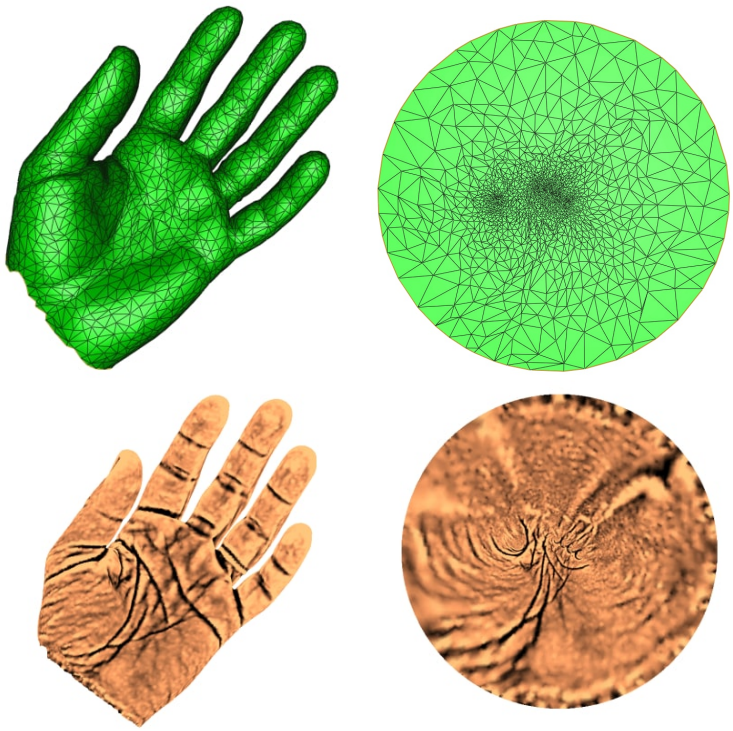

Figure 5.7: A hand mesh and its disk conformal parameterization using our two-step iterative scheme. Top: the triangulations. Bottom: the mean curvature texture maps.

of the discrete Ricci Flow [44], the Inversive Distance Ricci Flow algorithm [85], the Yamabe Riemann map algorithm [64] and the holomorphic 1-form

Surfaces	No. of faces	Our Method	RF [44]	IDRF [85]
		Time (seconds) / mean($\|\mu\|$) / sd($\|\mu\|$)		
Human face	49982	6.86/0.0101/0.0284	13.01/0.2421/0.1563	fail
Sophie	41587	4.88/0.0056/0.0083	23.86/0.1476/0.0792	29.43/0.0057/0.0086
Max Planck	99515	6.44/0.0102/0.0109	21.54/0.1431/0.0818	30.86/0.0103/0.0106
Mask	62467	8.63/0.0043/0.0051	13.46/0.2379/0.1425	fail
Nicolo da Uzzano	50042	5.71/0.0136/0.0314	10.34/0.2992/0.1434	fail
Julius Caesar	433956	72.69/0.0032/0.0100	108.72/0.1033/0.0689	173.55/0.0033/0.0094
Bimba	48469	2.61/0.0217/0.0254	10.07/0.2947/0.1430	fail
Human brain	96811	4.90/0.0250/0.0217	22.87/0.1861/0.1007	32.30/0.0249/0.0220
Hand	105860	6.89/0.0194/0.0168	35.38/0.0550/0.0281	39.63/0.0211/0.0212
Chinese lion	34421	2.23/0.0240/0.0271	8.03/0.2029/0.1024	10.11/0.0238/0.0265
Lion vase	98925	4.19/0.0236/0.0257	33.45/0.3687/0.1726	fail

Surfaces	No. of faces	Our Method	Yamabe Riemann map [64]	Holomorphic 1-form [31]
		Time (seconds) / mean($\|\mu\|$) / sd($\|\mu\|$)		
Human face	49982	6.86/0.0101/0.0284	fail	52.12/0.0111/0.0292
Sophie	41587	4.88/0.0056/0.0083	34.32/0.0057/0.0085	57.36/0.0058/0.0083
Max Planck	99515	6.44/0.0102/0.0109	34.34/0.0103/0.0106	92.74/0.0103/0.0109
Mask	62467	8.63/0.0043/0.0051	fail	fail
Nicolo da Uzzano	50042	5.71/0.0136/0.0314	fail	50.97/0.0143/0.0275
Julius Caesar	433956	72.69/0.0032/0.0100	175.01/0.0033/0.0094	462.63/0.0033/0.0096
Bimba	48469	2.61/0.0217/0.0254	fail	56.12/0.0230/0.0250
Human brain	96811	4.90/0.0250/0.0217	31.52/0.0249/0.0220	89.88/0.0251/0.0217
Hand	105860	6.89/0.0194/0.0168	44.25/0.0211/0.0212	104.04/0.0224/0.0269
Chinese lion	34421	2.23/0.0240/0.0271	14.82/0.0238/0.0265	40.43/0.0244/0.0271
Lion vase	98925	4.19/0.0236/0.0257	fail	137.93/0.0271/0.0282

Table 5.2: Performance of our proposed two-step iterative method and four state-of-the-art algorithms.

algorithm [31] on average. The conformality of our proposed method is at least comparable to (and often better than) that of the four algorithms. More importantly, the four algorithms fail for some meshes (for instance, the lion vase mesh shown in Figure 5.9) while our proposed algorithm is applicable for a wider range of meshes.

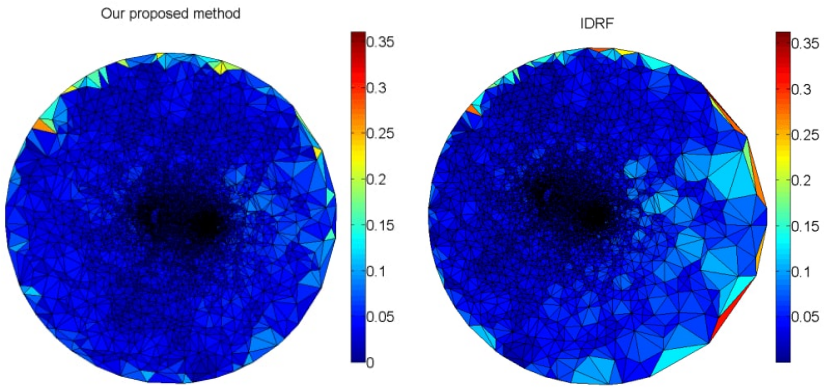

Figure 5.8: Comparison of the norm of the Beltrami coefficients between our two-step iterative scheme and the IDRF algorithm [85]. Each triangular face is colored with the norm of the approximated Beltrami coefficients on it. Left: our two-step iterative scheme. Right: the IDRF algorithm [85].

Figure 5.9: A lion vase mesh and its disk conformal parameterization using our two-step iterative scheme. The features of the lion vase mesh, such as the circular patterns around the boundary and the texture of the hair, are well-preserved.

In addition, Figure 5.10 highlights another difference between our proposed method and the IDRF algorithm [85]. For the four abovementioned algorithms, one triangular face has to be punctured at the beginning and filled at the end. The conformality distortion at the region near the puncture is undesirably large. On the contrary, the parameterization obtained by our proposed

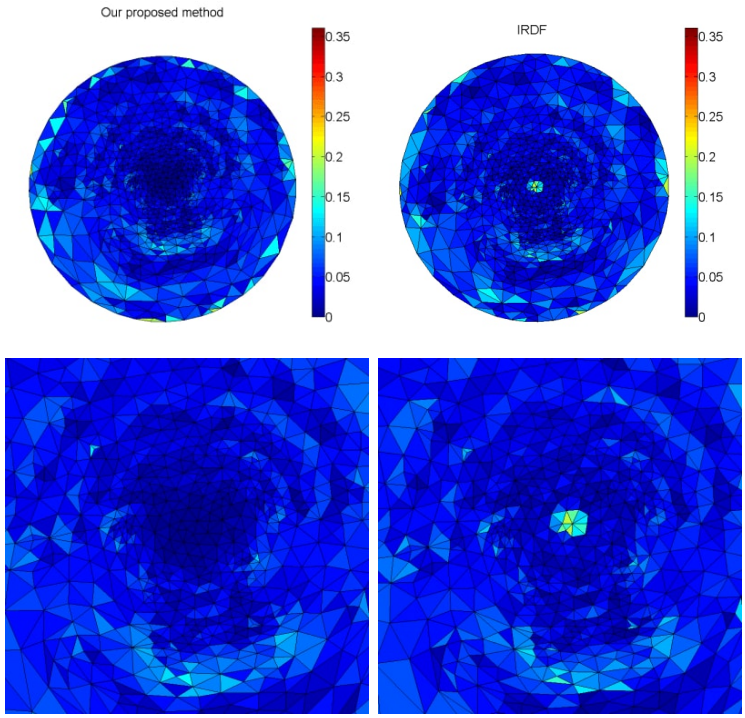

Figure 5.10: Comparison between the disk parameterizations obtained by our proposed method and by the IDRF algorithm [85]. The colormaps show the norm of the Beltrami coefficients on each triangular face. Top: the disk parameterizations obtained by our proposed method (left) and by the IDRF algorithm [85] (right). Bottom: The zoom in of the center of the disks.

method is of small conformality distortion and is free of such unnaturally distorted regions. This again demonstrates the advantage of our proposed two-step iterative scheme.

Besides, we compare our two-step iterative scheme with two other bijective and linear algorithms, namely the shape-preserving parameterization method [24] and the mean-value parameterization method [28]. As shown in Table 5.3, our proposed algorithm is more efficient than both of the two methods. It is also noteworthy that the conformality of our method is significantly better. The comparisons demonstrate the effectiveness of our two-step iterative scheme.

Surfaces	No. of faces	Our Method	Shape-preserving [24]	Mean-value [28]
		Time (seconds) / mean($\|\mu\|$) / sd($\|\mu\|$)		
Human face	49982	6.86/0.0101/0.0284	8.19/0.0818/0.0692	8.81/0.0722/0.0659
Sophie	41587	4.88/0.0056/0.0083	6.58/0.0449/0.0411	6.25/0.0470/0.0393
Max Planck	99515	6.44/0.0102/0.0109	32.28/0.0786/0.0459	29.92/0.0413/0.0260
Mask	62467	8.63/0.0043/0.0051	12.15/0.1154/0.0259	13.26/0.1132/0.0237
Nicolo da Uzzano	50042	5.71/0.0136/0.0314	9.07/0.1306/0.0852	9.08/0.1144/0.0800
Julius Caesar	433956	72.69/0.0032/0.0100	203.55/0.1332/0.0362	221.69/0.1323/0.0359
Bimba	48469	2.61/0.0217/0.0254	10.60/0.1271/0.0724	10.51/0.0722/0.0448
Human brain	96811	4.90/0.0250/0.0217	27.48/0.0950/0.0452	27.79/0.0656/0.0316
Hand	105860	6.89/0.0194/0.0168	37.65/0.0798/0.0431	39.29/0.0552/0.0371
Chinese lion	34421	2.23/0.0240/0.0271	5.32/0.1083/0.0590	5.74/0.0721/0.0450
Lion vase	98925	4.19/0.0236/0.0257	27.26/0.3892/0.2480	30.29/0.2496/0.1610

Table 5.3: Performance of our proposed method in [2], the shape-preserving parameterization [24] and the mean-value parameterization [28].

Finally, we demonstrate the bijectivity of our proposed algorithm. We consider two examples that the initial disk harmonic maps are not bijective. The examples are shown in Figure 5.11 and Figure 5.12. It can be observed that our two-step iterative scheme can correct the overlaps and ultimately result in a bijective disk conformal parameterization.

5.5.2 Linear disk conformal parameterization algorithm

In this section, we demonstrate the effectiveness of our proposed linear disk conformal parameterization algorithm in Section 5.4 using a number of 3D simply-connected open meshes. The meshes are freely adopted from the AIM@SHAPE Shape Repository [89], the Stanford 3D Scanning Repository [94] and the Benchmark for 3D Mesh Segmentation [92]. Our proposed linear algorithm is implemented in MATLAB. The sparse linear systems are solved using the backslash operator (\) in MATLAB. All experiments are performed on a PC with a 3.40 GHz quad core CPU and 16 GB RAM.

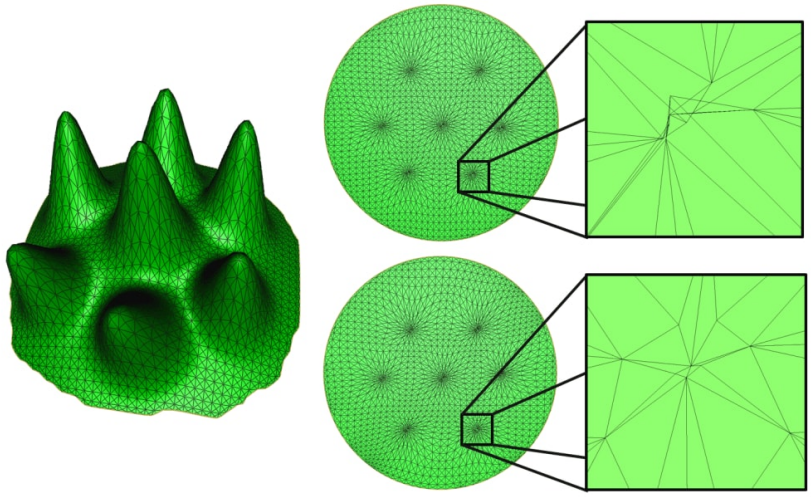

Figure 5.11: A comparison between the disk parameterizations of a simply-connected open surface with multiple peaks obtained by the disk harmonic map and by our two-step iterative scheme. It is observed that there are foldings near the peak in the disk harmonic map, while the final result by our proposed method is bijective. Left: the input surface. Top right: the initial disk harmonic map and the zoom in of a peak of the disk. Bottom right: the final parameterization result obtained by our proposed algorithm and the zoom in.

Figure 5.13, 5.14 and 5.15 show three meshes and the disk conformal parameterizations obtained by our proposed linear algorithm. The local shapes of original models are well-preserved under the disk parameterizations. This shows the conformality of our algorithm. Also, as shown in Figure 5.16, the angle differences highly concentrate at 0, which indicates that our proposed linear algorithm produces negligible conformality distortions. Besides, our proposed linear algorithm takes only about 1 second for meshes with moderate size, and less than half a minute for extremely dense meshes.

To quantitatively assess the performance, we compare our proposed linear algorithm with two other methods, including the holomorphic 1-form method [31] and two-step iterative approach in [2]. The holomorphic 1-form method

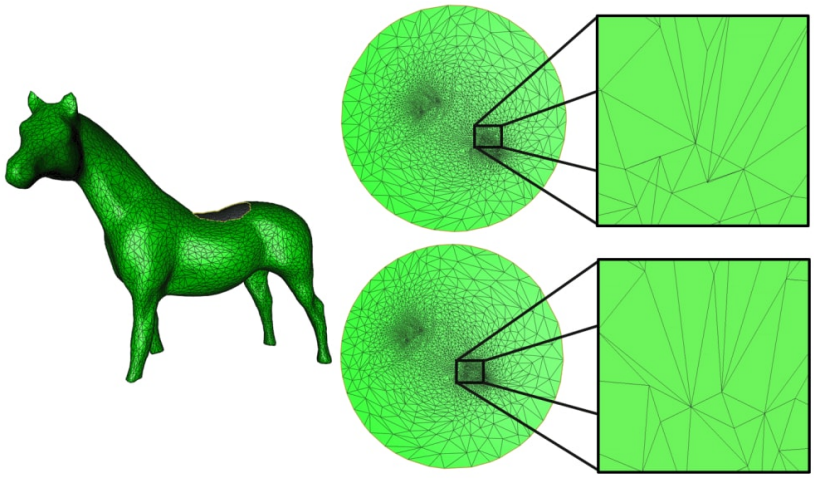

Figure 5.12: A comparison between the disk parameterizations of a horse surface obtained by the disk harmonic map and by our two-step iterative scheme. The foldings near the legs in the initial disk harmonic map are corrected by our proposed algorithm. Left: the input surface. Top right: the initial disk harmonic map and the zoom in of a leg in the disk harmonic map. Bottom right: the final parameterization result obtained by our proposed algorithm and the zoom in.

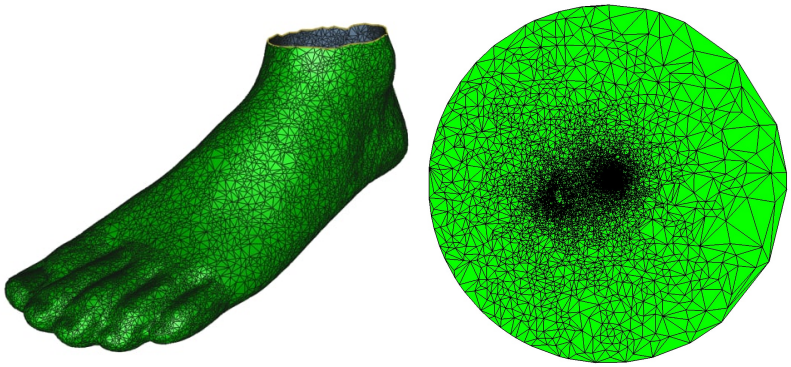

Figure 5.13: A simply-connected open human foot model and the disk conformal parameterization by our proposed linear algorithm.

is available in the RiemannMapper Toolkit [93], which is written in C++, and the two-step iterative approach in [2] is implemented in MATLAB, as described in the last subsection. The error thresholds in both the holomorphic

Figure 5.14: A simply-connected open statue model and the disk conformal parameterization by our proposed linear algorithm, colored with the mean curvature of the model.

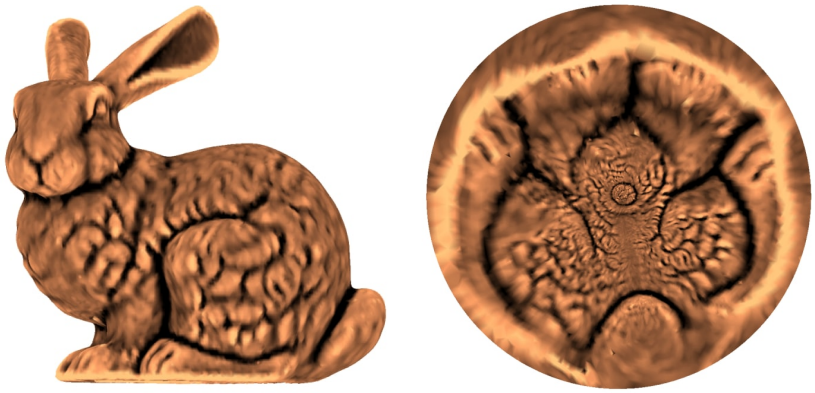

Figure 5.15: A simply-connected open Stanford bunny model and the disk conformal parameterization by our proposed linear algorithm, colored with the mean curvature of the model.

1-form method and the two-step iterative approach in [2] are set to be $\epsilon = 10^{-5}$.

The performances of the algorithms are recorded in Table 5.4. It is noteworthy that the angular distortion of our proposed linear method is comparable and sometimes better than the two state-of-the-art algorithms [31, 2]. Moreover, our linear method significantly accelerates the computation.

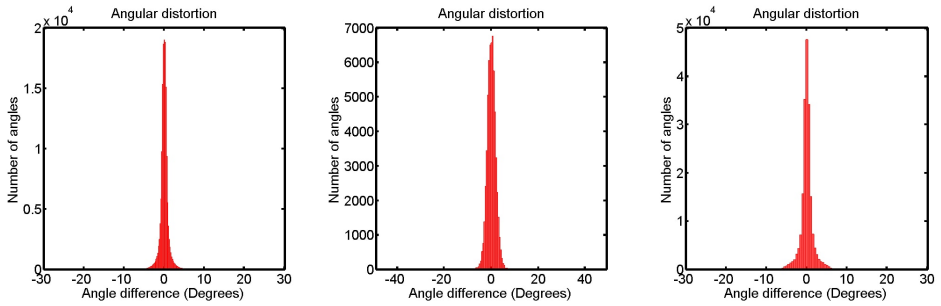

Figure 5.16: The histograms of the angular distortion of our linear algorithm. Here the angular distortion is defined by the difference (in degrees) between an angle in the final parameterization and its corresponding angle on the original mesh. Left: The result of statue. Middle: The result of foot. Right: The result of Stanford bunny.

Surfaces	No. of faces	Holomorphic 1-form [31]	Fast disk map [2]	Linear disk map [3]
		Time (s) / Mean($\|distortion\|$) (degrees) / SD($\|distortion\|$) (degrees)		
Horse	9K	fail	0.76 / 4.58 / 5.45	0.18 / 4.60 / 5.49
T-shirt	14K	18.64 / 1.38 / 3.26	2.09 / 1.34 / 3.25	0.34 / 1.35 / 3.26
Foot	20K	11.73 / 1.40 / 1.24	1.75 / 1.42 / 1.22	0.47 / 1.42 / 1.22
Chinese lion	30K	29.87 / 1.44 / 2.05	2.70 / 1.42 / 2.04	0.92 / 1.42 / 2.05
Sophie	40K	28.29 / 0.36 / 0.61	5.87 / 0.34 / 0.60	1.31 / 0.35 / 0.60
Bimba	50K	28.04 / 1.29 / 1.78	2.22 / 1.22 / 1.74	1.32 / 1.22 / 1.75
Human face	50K	28.45 / 0.55 / 1.84	4.61 / 0.53 / 1.82	1.40 / 0.53 / 1.83
Niccolo da Uzzano	50K	29.49 / 0.78 / 1.73	7.95 / 0.75 / 1.75	1.34 / 0.76 / 1.74
Mask	60K	fail	7.08 / 0.25 / 0.33	1.93 / 0.25 / 0.33
Bunny	70K	40.14 / 1.08 / 1.80	4.18 / 1.08 / 1.79	1.99 / 1.08 / 1.79
Brain	100K	58.49 / 1.46 / 1.59	6.81 / 1.46 / 1.59	2.73 / 1.46 / 1.59
Lion vase	100K	93.64 / 1.44 / 1.91	5.34 / 1.27 / 1.75	2.64 / 1.27 / 1.76
Max Planck	100K	75.92 / 0.61 / 0.80	6.58 / 0.61 / 0.80	2.88 / 0.61 / 0.80
Hand	110K	63.90 / 1.40 / 1.99	7.51 / 1.21 / 1.31	3.30 / 1.21 / 1.31
Igea	270K	173.60 / 0.40 / 0.72	54.63 / 0.40 / 0.71	9.47 / 0.40 / 0.71
Julius Caesar	430K	295.63 / 0.21 / 0.67	65.54 / 0.20 / 0.67	19.51 / 0.20 / 0.67

Table 5.4: The performance of three state-of-the-art algorithms. Here the distortion refers to the angular difference (in degrees) between an angle in the final parameterization and its corresponding angle on the original mesh.

Specifically, our linear algorithm is about 20 times faster than the holomorphic 1-form method [31] on average. Also, the computational time of our linear

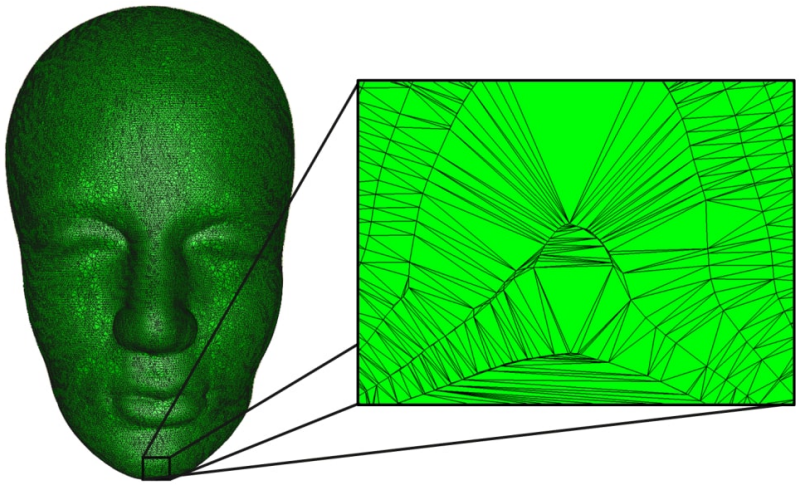

Figure 5.17: A mask model with a highly irregular triangulation. Left: The mask model. Right: A zoom-in of the model.

algorithm is 60% less than that of our two-step iterative approach in [2] on average. As a remark, our proposed linear algorithm results in folding-free parameterization results in all experiments.

After demonstrating the advantages of our proposed linear algorithm over the two aforementioned approaches, we compare our algorithm with two other linear bijective disk conformal parameterization algorithms in [24, 28]. These parameterizations are implemented in the GoTools C++ libraries [90]. The performances of the algorithms are recorded in Table 5.5. It can be observed that the average absolute angular distortion of our proposed linear algorithm is 80% lower than that of the shape-preserving method [24] and 75% lower than that of the mean-value method [28]. Also, our method is about 7 times faster than both of the two algorithms. Hence, our proposed linear algorithm outperforms the two linear existing approaches in terms of both the conformality and the efficiency.

Our linear method is highly robust to irregular triangulations. Figure 5.17

Surfaces	No. of faces	Shape-preserving [24]	Mean-value [28]	Our linear method [3]
		Time (s) / Mean(|*distortion*|) (degrees) / SD(|*distortion*|) (degrees)		
Horse	9K	0.97 / 18.52 / 16.51	1.21 / 12.48 / 12.02	0.18 / 4.60 / 5.49
T-shirt	14K	1.29 / 14.72 / 12.98	1.21 / 14.59 / 12.84	0.34 / 1.35 / 3.26
Foot	20K	2.92 / 3.13 / 2.46	3.13 / 2.15 / 1.84	0.47 / 1.42 / 1.22
Chinese lion	30K	5.47 / 6.48 / 5.26	5.90 / 4.31 / 3.86	0.92 / 1.42 / 2.05
Sophie	40K	6.70 / 2.75 / 3.16	6.40 / 2.87 / 3.08	1.31 / 0.35 / 0.60
Bimba	50K	10.65 / 7.33 / 6.28	10.73 / 4.14 / 3.75	1.32 / 1.22 / 1.75
Human face	50K	8.13 / 4.73 / 5.29	8.99 / 4.18 / 4.98	1.40 / 0.53 / 1.83
Niccolo da Uzzano	50K	9.03 / 7.38 / 6.93	9.24 / 6.46 / 6.42	1.34 / 0.76 / 1.74
Mask	60K	12.12 / 6.54 / 3.96	13.43 / 6.39 / 3.81	1.93 / 0.25 / 0.33
Bunny	70K	17.73 / 10.61 / 13.68	19.39 / 5.91 / 8.03	1.99 / 1.08 / 1.79
Brain	100K	27.41 / 5.61 / 4.23	28.40 / 3.87 / 2.93	2.73 / 1.46 / 1.59
Lion vase	100K	27.53 / 21.31 / 19.84	30.87 / 13.40 / 12.46	2.64 / 1.27 / 1.76
Max Planck	100K	32.52 / 4.77 / 3.95	30.15 / 2.50 / 2.17	2.88 / 0.61 / 0.80
Hand	110K	38.55 / 5.03 / 3.93	39.92 / 3.47 / 3.14	3.30 / 1.21 / 1.31
Igea	270K	129.07 / 8.70 / 5.77	137.41 / 5.13 / 3.35	9.47 / 0.40 / 0.71
Julius Caesar	430K	202.09 / 8.19 / 4.79	228.23 / 8.14 / 4.76	19.51 / 0.20 / 0.67

Table 5.5: The performance of our proposed linear algorithm, the shape-preserving parameterization [24] and the mean-value parameterization [28].

shows a simply-connected mask model with a very irregular triangulation. Sharp and irregular triangles can be easily observed. The holomorphic 1-form method [31] fails for this model while our linear method succeeds (please refer to Table 5.4). This demonstrates the robustness of our proposed linear algorithm.

As a final remark, we discuss the possibility of further accelerating the computation of our proposed linear algorithm. Recall that in our proposed linear algorithm, we need to compute the spherical conformal parameterization of the double covered surface using the linear algorithm in [1]. The algorithm in [1] consists of two major steps, namely a North-pole step and a South-pole step. The North-pole step results in a spherical parameterization such that the southernmost region is with negligible conformality distortion. Then, the South-pole step aims to correct the conformality distortion in the northernmost

Surfaces	No. of faces	Our linear method [3]	Our linear method [3] without the south-pole step in [1]
		Time (s) / Mean(|*distortion*|) (degrees) / SD(|*distortion*|) (degrees)	
Horse	9K	0.18 / 4.60 / 5.49	0.11 / 4.63 / 5.50
T-shirt	14K	0.34 / 1.35 / 3.26	0.22 / 1.37 / 3.27
Foot	20K	0.47 / 1.42 / 1.22	0.31 / 1.43 / 1.23
Chinese lion	30K	0.92 / 1.42 / 2.05	0.56 / 1.43 / 2.05
Sophie	40K	1.31 / 0.35 / 0.60	0.81 / 0.37 / 0.62
Bimba	50K	1.32 / 1.22 / 1.75	0.75 / 1.23 / 1.76
Human face	50K	1.40 / 0.53 / 1.83	0.83 / 0.55 / 1.84
Niccolo da Uzzano	50K	1.34 / 0.76 / 1.74	0.78 / 0.77 / 1.74
Mask	60K	1.93 / 0.25 / 0.33	1.22 / 0.28 / 0.37
Bunny	70K	1.99 / 1.08 / 1.79	1.14 / 1.09 / 1.79
Brain	100K	2.73 / 1.46 / 1.59	1.55 / 1.49 / 1.60
Lion vase	100K	2.64 / 1.27 / 1.76	1.45 / 1.28 / 1.76
Max Planck	100K	2.88 / 0.61 / 0.80	1.54 / 0.63 / 0.80
Hand	110K	3.30 / 1.21 / 1.31	1.74 / 1.22 / 1.31
Igea	270K	9.47 / 0.40 / 0.71	5.22 / 0.54 / 0.74
Julius Caesar	430K	19.51 / 0.20 / 0.67	12.41 / 0.21 / 0.68

Table 5.6: The performance of the current version of our linear disk conformal parameterization algorithm and a possible improved version of it without the "south-pole" step in [1], provided a suitable choice of the boundary triangle in Equation (4.2).

region. In our case, since we are only interested in half of the glued surface, the South-pole step may be skipped as we can take the Southern hemisphere obtained by the north-pole step as our result.

The conformality distortion in the North-pole step in [1] is primarily caused by the choice of the boundary triangle $[a_1, a_2, a_3]$. If the chosen boundary triangle and its neighboring triangular faces are regular enough, then the conformality distortion of the southernmost region after the North-pole step is already satisfactory. In this case, the South-pole step is not needed. In other words, with a well-chosen boundary triangle $[a_1, a_2, a_3]$ in Equation (4.2), half of the computational time in computing the spherical conformal parameterization can be saved.

Table 5.6 records the performance of the current version of our proposed linear algorithm and the version without the South-pole step in [1], under a

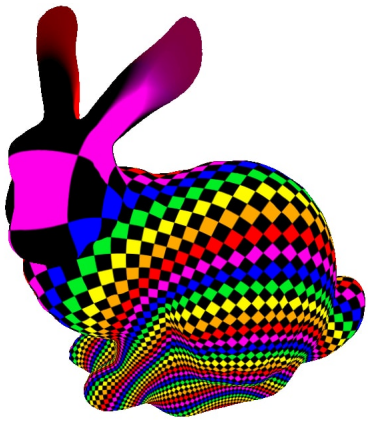

Figure 5.18: The Stanford bunny model with a rainbow checkerboard texture mapped onto it using our linear disk conformal parameterization algorithm.

suitable choice of the boundary triangle $[a_1, a_2, a_3]$ in Equation (4.2). The conformality distortion of the two versions are highly similar, while the version without the South-pole step reduces the computational time by 40% on average. However, it should be remarked that the version without the South-pole step requires a suitably chosen boundary triangle $[a_1, a_2, a_3]$ for [1], while the current version of our method is fully automatic and independent of the choice of $[a_1, a_2, a_3]$. Hence, the current version of our proposed linear algorithm is more suitable for real applications until the new version is fully analyzed.

5.6 Applications

With our developed disk conformal parameterization algorithms, the texture mapping problem on simply-connected open meshes can be efficiently solved. Suppose we are given a simply-connected open mesh and a planar texture image. To map the texture onto the mesh, we first compute the disk conformal parameterizations using our proposed algorithms. Then, the texture can be

Figure 5.19: A T-shirt model with a flower pattern design mapped onto it using our linear disk conformal parameterization algorithm.

mapped onto the unit disk, and finally the inverse of the disk conformal parameterization maps the texture back onto the original mesh.

In Figure 5.18, we consider mapping a rainbow checkerboard texture onto a simply-connected bunny mesh using our proposed linear algorithm. It can be easily observed that the orthogonal checkerboard structure is well preserved under the our proposed parameterization scheme because of the angle-preserving property of conformal maps.

Using this method, planar clothing designs can be easily visualized in 3D. Figure 5.19 shows a T-shirt model with a planar flower pattern design mapped onto it. The patterns are well preserved on the 3D T-shirt model. This approach provides a more comprehensive and realistic preview of the clothes for both the designers and customers. For instance, with the real-time computation of the disk conformal parameterization, online customers can virtually try on clothes in virtual dressing rooms after selecting 2D images of the clothes.

Chapter 6

Spherical quasi-conformal parameterization of genus-0 closed meshes

6.1 Introduction

In this chapter, we propose a linear algorithm for the spherical quasi-conformal parameterization of genus-0 closed meshes [6]. To the best of the author's knowledge, this is the first work on the computation of spherical quasi-conformal parameterizations.

Conformal parameterization of genus-0 closed meshes has been widely studied, and we have proposed a linear algorithm in Chapter 4 for the computation. Also, various quasi-conformal parameterization algorithms have been developed for planar domains and simply-connected open meshes by different researchers. However, the study of quasi-conformal parameterization on spherical meshes is limited. Given a user-defined quasi-conformality distortion, we aim to develop a linear algorithm for computing the spherical quasi-conformal parameterization with the prescribed distortion.

6.2 Contributions

Our proposed spherical quasi-conformal parameterization algorithm is with the following advantages:

1. *Linearity*: Our proposed algorithm is linear and hence highly efficient in practice.

2. *Bijectivity*: The bijectivity of the resulting parameterization is guaranteed by quasi-conformal theory.

3. *Accuracy*: Our algorithm can accurately compute a spherical parameterization with the prescribed distortion.

6.3 Our proposed method

Note that quasi-conformal maps are flexible and not unique in general. Therefore, it is desirable to have an algorithm for computing a spherical quasi-conformal parameterization based on an user-defined quasi-conformal distortion. The user-defined distortion can be freely changed to fit into different applications.

6.3.1 Quasi-conformal dilation

To achieve this goal, we first need to choose an appropriate candidate for user-defined distortion. It is desirable to have a quantity that accurately reflects the quasi-conformality and is easy to compute. For spherical conformal maps, we use the angle difference between the three angles of a triangular face on the input mesh and those of the face on the sphere as a measure of the conformality. The map is with good conformality on the face if the three angle differences are all close to 0, or equivalently, if the mean of the

absolute angle differences is close to 0. However, the measurement is not appropriate for the case of spherical quasi-conformal maps. For instance, under the shear mapping $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + \lambda y \\ y \end{pmatrix}$, the three angle differences are highly different from each other and none of them can accurately represent the quasi-conformality. Hence, instead of the angles, it is desirable to have the user-defined distortion defined on every triangular face of the input mesh.

In the following, we consider the *dilation* on every triangular face as the measurement of quasi-conformality. Mathematically, let $f : \mathbb{C} \to \mathbb{C}$ be a quasi-conformal map. The *dilation* of f at a point z is defined by

$$K_f(z) = \frac{1 + |\mu_f(z)|}{1 - |\mu_f(z)|}, \tag{6.1}$$

where μ_f is the Beltrami coefficient of f. Geometrically, the dilation is the ratio of the length of the axes shown in Figure 3.1 under the quasi-conformal map f.

The dilation of f is related to the maximal quasi-conformal dilation K in Equation (3.9). More specifically, we have

$$K = \sup_z K_f(z). \tag{6.2}$$

The map f is said to be *p-quasiconformal* if the maximal quasi-conformal dilation is bounded above by p. In other words, every infinitesimal circle is mapped to an infinitesimal ellipse with eccentricity at most p. In particular, a conformal map is a 1-quasiconformal map.

An important property about the maximal dilation of composition of quasi-conformal mappings is as follows.

Proposition 6.3.1. *If $f : \Omega_1 \to \Omega_2$ is a K_1-quasiconformal map and $g : \Omega_2 \to \Omega_3$ is a K_2-quasiconformal map, then $g \circ f$ is a K_1K_2-quasiconformal map.*

In the discrete case, since the Beltrami coefficients are approximated on every triangular face as described in Equation (3.13), it is natural to define the dilation on every face. We have the following discretization:

Definition 6.3.2 (Discrete dilation)**.** *Let $f : M_1 \to M_2$ be a quasi-conformal map between two triangulated meshes M_1, M_2 on $\mathbb{C}$. For every triangular face T of M_1, the discrete dilation of f on T is defined by*

$$K_f(T) = \frac{1 + |\mu_f(T)|}{1 - |\mu_f(T)|}, \tag{6.3}$$

where $\mu_f(T)$ is the Beltrami coefficient of f approximated on T.

Moreover, the measurement of the dilation can be naturally extended to quasi-conformal maps between meshes in $\mathbb{R}^3$.

Definition 6.3.3 (Discrete dilation in $\mathbb{R}^3$)**.** *Let $f : M_1 \to M_2$ be a quasi-conformal map between two triangulated meshes M_1, M_2 in $\mathbb{R}^3$, and let T_1, T_2 be two corresponding triangular faces on K_1, K_2 respectively. Let $\phi_i : T_i \to \mathbb{C}$ be an isometric embedding of T_i onto $\mathbb{C}$, where $i = 1, 2$. The discrete dilation of f on T_1 is defined by*

$$K_{\tilde{f}}(\phi_1(T_1)), \tag{6.4}$$

where $\tilde{f} : \phi_1(T_1) \to \phi_2(T_2)$ is a quasi-conformal map on $\mathbb{C}$.

Note that the above definition is well-defined because only the norm of the Beltrami coefficients is considered. With the above concepts, we are ready to introduce our proposed spherical quasi-conformal parameterization algorithm for a genus-0 closed triangulated mesh M and a user-defined quasi-conformal dilation $K \geq 1$ defined on every face.

6.3.2 Initial map

We first compute a spherical conformal parameterization $f : M \to \mathbb{S}^2$ as an initialization. Among all existing algorithms for computing the spherical conformal parameterization, we use the linear spherical conformal parameterization algorithm in [1] for three reasons. Firstly, the algorithm is linear and hence the computation is highly efficient. Secondly, the algorithm in [1] achieves the best conformality when compared with the pre-existing approaches. The conformality of the initial spherical map is important in the subsequent steps. Thirdly, the algorithm in [1] results in a bijective spherical parameterization. The bijectivity is also crucial for the computation in the remaining steps.

6.3.3 Optimally project the sphere onto the complex plane

After obtaining the initial spherical parameterization, we choose a triangular face $T = [v_1, v_2, v_3]$ on $f(M)$ such that T and its neighboring triangular faces are the most regular. Then, we apply a rotation ψ on $f(M)$ such that the centroid of T lies on the positive z-axis, followed by the stereographic projection P_N.

The regularity of T and its neighboring faces is important because of the stereographic projection P_N. When applying the stereographic projection, the north pole $(0, 0, 1)$ is mapped to ∞ on the extended complex plane, and the northernmost region on $\mathbb{S}^2$ is mapped to the outermost region on the plane. In particular, T is mapped to a big triangle on the plane. Now, denote the geodesic between v_i and v_j on $\mathbb{S}^2$ by $g_{v_i v_j}$. Note that $g_{v_i v_j}$ is a circular arc on $\mathbb{S}^2$, while the edge $e_{v_i v_j}$ connecting v_i and v_j on M is an Euclidean straight line. On $\mathbb{S}^2$, this discrepancy between $g_{v_i v_j}$ and $e_{v_i v_j}$ may not be very

large. However, under the stereographic projection, this discrepancy between $P_N(g_{v_iv_j})$ and the Euclidean straight line $e_{P_N(v_i)P_N(v_j)}$ becomes serious.

In the continuous case, under the stereographic projection, all other vertices are mapped to the interior of the region enclosed by $g_{v_1v_2}$, $g_{v_2v_3}$ and $g_{v_3v_1}$. However, in the discrete case, if T and its neighboring faces are not regular enough, some vertices may be mapped outside the Euclidean triangle $[P_N(v_1), P_N(v_2), P_N(v_3)]$. The outlying vertices causes computational difficulty in the following step, in which only the three vertices $P_N(v_1), P_N(v_2), P_N(v_3)$ are involved in the boundary constraints. Hence, a suitable choice of T is necessary.

6.3.4 Achieving the quasi-conformality

By the stereographic projection, the chosen triangular face T is mapped to a big triangle on $\mathbb{C}$. Next, we compose the map with a quasi-conformal map h that satisfies the prescribed dilation.

To compute a quasi-conformal map using LBS [61], 3 point boundary constraints of the outermost triangular face T are required. Moreover, the boundary constraints must be set optimally, otherwise the prescribed quasi-conformality cannot be achieved.

To set a suitable boundary condition, we explicitly compute the image of T under the prescribed dilation $K(T)$. Denote $T = [x_1 + iy_1, x_2 + iy_2, x_3 + iy_3]$. By Equation (6.3), we define the Beltrami coefficient $\mu(T)$ on the triangular face T by

$$\mu(T) = \frac{K(T) - 1}{K(T) + 1}. \tag{6.5}$$

Note that the argument of $\mu(T)$ is set to be 0 without loss of generality.

Since h is piecewise linear, we have

$$h|_T \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} a_T x_i + b_T y_i + r_T \\ c_T x_i + d_T y_i + s_T \end{pmatrix} \tag{6.6}$$

for $i = 1, 2, 3$, where $a_T, b_T, c_T, d_T, r_T, s_T$ are to be determined.

Without loss of generality, we can assume that $h|_T \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ and $h|_T \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$.

Also, by Equation (3.11), we have

$$\begin{aligned} d_T &= \alpha_1 a_T + \alpha_2 b_T; \\ -c_T &= \alpha_2 a_T + \alpha_3 b_T, \end{aligned} \tag{6.7}$$

where

$$\alpha_1 = \frac{(\rho_T - 1)^2 + \tau_T^2}{1 - \rho_T^2 - \tau_T^2}; \quad \alpha_2 = -\frac{2\tau_T}{1 - \rho_T^2 - \tau_T^2}; \quad \alpha_3 = \frac{(1 + \rho_T)^2 + \tau_T^2}{1 - \rho_T^2 - \tau_T^2}. \tag{6.8}$$

Here, $\rho(T)$ and $\tau(T)$ are respectively the real part and the imaginary part of $\mu(T)$. By our construction of $\mu(T)$ introduced before, we have $\rho(T) = \frac{K(T)-1}{K(T)+1}$ and $\tau(T) = 0$. A direct calculation yields

$$\begin{cases} \alpha_1 = -\frac{1}{K(T)} \\ \alpha_2 = 0 \\ \alpha_3 = K(T). \end{cases} \tag{6.9}$$

Altogether, $a_T, b_T, c_T, d_T, r_T, s_T$ can be explicitly solved by the following linear system:

$$\begin{pmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ x_2 & y_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_2 & y_2 & 0 & 1 \\ \frac{1}{K(T)} & 0 & 0 & -1 & 0 & 0 \\ 0 & K(T) & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_T \\ b_T \\ c_T \\ d_T \\ r_T \\ s_T \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ 0 \\ 0 \end{pmatrix}. \tag{6.10}$$

The existence and uniqueness of $(a_T, b_T, c_T, d_T, r_T, s_T)$ is guaranteed by the following proposition.

Proposition 6.3.4. *The matrix in Equation (6.10) is nonsingular.*

Proof. Denote the matrix in Equation (6.10) by A. By a direct calculation, we have

$$\det(A) = -K(T)(x_1 - x_2)^2 - \frac{1}{K(T)}(y_1 - y_2)^2. \tag{6.11}$$

Since T is non-degenerate, we have $(x_1, y_1) \neq (x_2, y_2)$. Also, note that $K \geq 1$. It follows that $\det(A) \neq 0$. □

After obtaining $a_T, b_T, c_T, d_T, r_T, s_T$, we can explicitly compute $h|_T \begin{pmatrix} x_3 \\ y_3 \end{pmatrix}$ using Equation (6.6). The above computations give us the desired boundary condition for $h(x_1 + iy_2), h(x_1 + iy_2)$ and $h(x_3 + iy_3)$ of the triangular face T.

With the above boundary conditions, we apply the Linear Beltrami Solver (LBS) [61] for computing a quasi-conformal map h that satisfies the prescribed quasi-conformal distortion. More specifically, by Equation (6.3), we have

$$|\mu(F)| = \frac{K(F) - 1}{K(F) + 1} \tag{6.12}$$

for all triangular faces F. We apply LBS with μ and the boundary constraints on T, obtaining the quasi-conformal map h. It is noteworthy that since $\|\mu\|_\infty < 1$, Theorem 3.3.2 guarantees the bijectivity of the map h.

Since T may be severely distorted by the prescribed distortion, the origin may no longer be located inside T under the quasi-conformal map h. In this case, the resulting parameterization obtained by the inverse stereographic projection P_S^{-1} may not be a sphere but only a portion of it. To overcome this problem, we perform a translation on $\mathbb{C}$ so that the centroid of the whole domain is at the origin. This ensures that T will be the northernmost triangular face under P_S^{-1}.

Now, the desired quasi-conformality distortion is achieved. However, as we have fixed two vertices of T in computing the boundary constraints, the size of

the whole triangular domain may not be optimal. More specifically, if the size of T is too large, most vertices will be mapped to the northern hemisphere by P_S^{-1}. On the other hand, if the size of T is too small, most vertices will be mapped to the southern hemisphere by P_S^{-1}. To achieve an optimal distribution on the spherical parameterization, we apply the balancing scheme in the linear spherical conformal parameterization algorithm [1]. Based on Theorem 4.3.2, the balancing scheme ensures that T and the innermost triangle t on $\mathbb{C}$ will be mapped to two triangles with similar size on the unit sphere under P_S^{-1}. This completes our task of computing a spherical quasi-conformal parameterization with prescribed quasi-conformality distortion.

It is noteworthy that our proposed algorithm is linear as all steps are linear. Hence, our algorithm is highly efficient in practice. Also, the desired quasi-conformality of the spherical parameterization is guaranteed by Theorem 6.3.1. Since the initial spherical map, the rotation and the stereographic projections are all conformal maps (i.e. 1-quasiconformal maps) and h is K-quasiconformal, the composition of the maps is also K-quasiconformal. The implementation of our linear spherical quasi-conformal (LSQC) parameterization algorithm is described in Algorithm 7.

6.4 Experimental Results

In this section, we demonstrate the effectiveness of our proposed linear spherical quasi-conformal parameterization algorithm. Various genus-0 closed triangulated meshes are adopted from the AIM@SHAPE shape repository [89] for testing our algorithm. Our entire algorithm is implemented in MATLAB. All experiments are performed on a PC with an Intel(R) Core(TM) i5-3470 CPU @3.20 GHz processor and 8.00 GB RAM.

Figure 6.1 shows a genus-0 closed brain mesh and the spherical quasi-

Algorithm 7: Linear spherical quasi-conformal (LSQC) parameterization

Input: A simply-connected closed triangular mesh M, a user-defined quasi-conformal dilation $K \geq 1$ defined on every face.

Output: A bijective spherical quasi-conformal parameterization $\varphi : M \to \mathbb{S}^2$.

1 Compute a spherical conformal parameterization $f : M \to \mathbb{S}^2$ using the linear algorithm in [1];
2 Choose a triangular face T on $f(M)$ as described in Section 6.3.3;
3 Apply a rotation ψ on $f(M)$ such that the centroid of T lies on the positive z-axis;
4 Apply the stereographic projection P_N on $\psi(f(M)$;
5 Compute a quasi-conformal map $h : P_N(\psi(f(M))) \to \mathbb{C}$ with the prescribed distortion, and an appropriate boundary condition of the big triangle T;
6 Perform a translation so that the centroid of the whole domain is at the origin;
7 Apply the balancing scheme in the linear algorithm [1];
8 Apply the inverse stereographic projection P_N^{-1} and denote the overall result by φ;

conformal parameterization obtained by our linear algorithm. It can be observed that the resulting quasi-conformal distortion closely resembles the desired quasi-conformal distortion. Another example is shown in Figure 6.2. In this example, we consider a discontinuous dilation as the target quasi-conformal distortion. Even with the discontinuity, the spherical quasi-conformal parameterization obtained can satisfy the desired distortion. It can be observed that the circles on the input mesh are transformed to two types of ellipses on the spherical quasi-conformal parameterization. Also, two sharp peaks can be observed in the histogram of the resulting dilation plot.

Then, we apply our algorithm for computing spherical Teichmüller parameterization of genus-0 closed meshes. Note that Teichmüller maps are with uniform conformality distortions. Hence, to compute a spherical Teichmüller

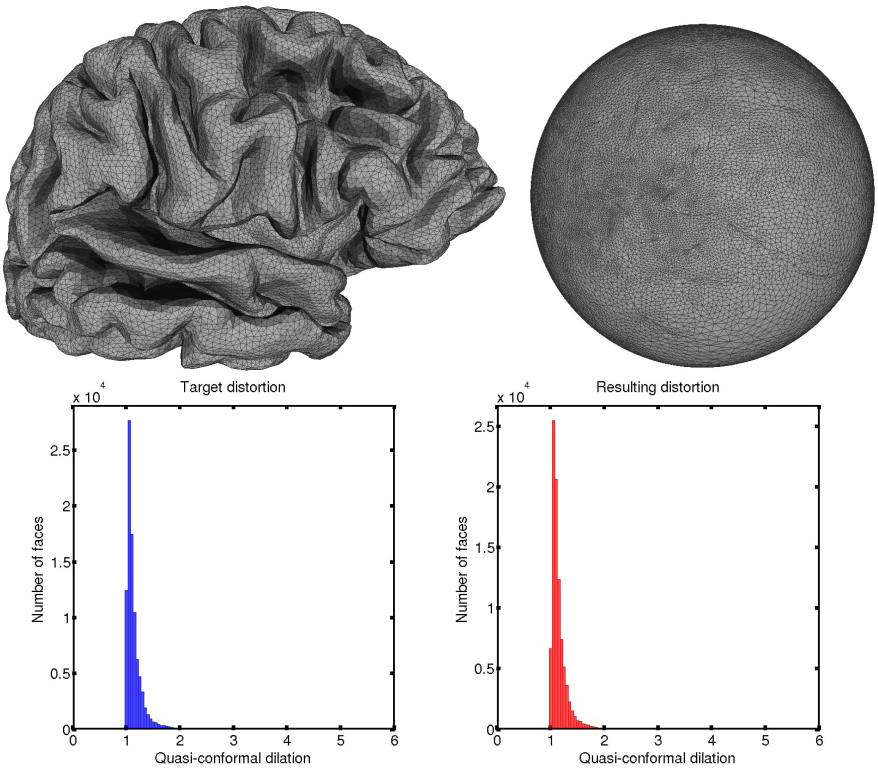

Figure 6.1: A brain and the spherical quasi-conformal parameterization obtained by our algorithm. Top left: the input surface. Top right: the spherical parameterization. Bottom left: The target quasi-conformal distortion. Bottom right: The resulting quasi-conformal distortion of the parameterization.

parameterization, we set the target dilation as a constant. Figure 6.3 and Figure 6.4 show two examples of the spherical Teichmüller parameterizations obtained by our algorithm. It is noteworthy that even for the highly convoluted spiral model, the resulting dilations significantly concentrate at the desired constant. The uniform dilation can also be observed from the triangular faces on the spherical parameterizations. This implies that our algorithm can effectively produce the spherical Teichmüller parameterizations.

Table 6.1 records the performance of our proposed linear spherical quasi-conformal parameterization algorithm. Because of the linearity of our algorithm, the computations finish within a few seconds even for very dense

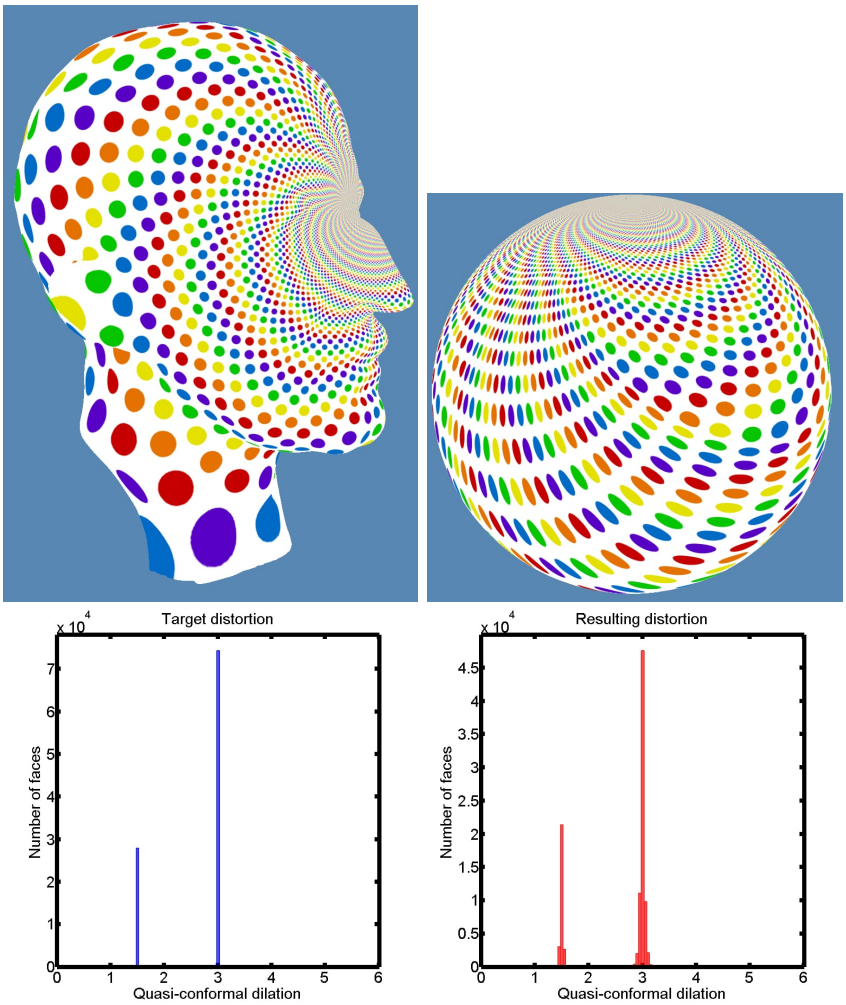

Figure 6.2: A Max Planck model with a circle pattern and the spherical quasi-conformal parameterization obtained by our algorithm. Top left: the input surface. Top right: the spherical parameterization. Bottom left: The target quasi-conformal distortion. Bottom right: The resulting quasi-conformal distortion of the parameterization. The user-defined synthetic distortion is achieved on the spherical parameterization.

meshes. Also, in all examples, the resulting quasi-conformal distortion is highly close to the target distortion. This reflects the accuracy of our proposed algorithm. Besides, the absence of extreme values in the resulting dilation distribution implies that the Beltrami coefficient is with sup norm $\ll 1$. Hence, the resulting parameterizations are bijective.

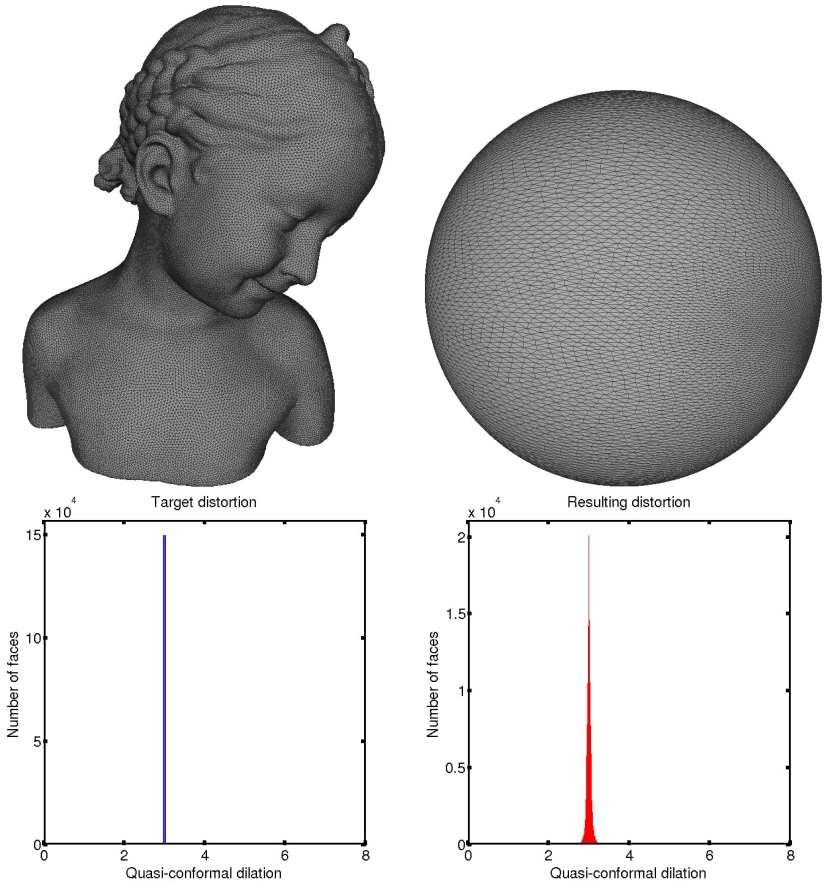

Figure 6.3: A bimba model and the spherical Teichmüller parameterization obtained by our proposed algorithm. Top left: the input surface. Top right: the spherical parameterization. Bottom left: The target quasi-conformal distortion. Bottom right: The resulting quasi-conformal distortion of the parameterization.

Surfaces	# of faces	Time (s)	Target dilation		Resulting dilation	
			Mean	SD	Mean	SD
Max Planck	102212	1.8867	2.5887	0.6692	2.5896	0.6687
Brain 1	91124	1.9399	1.1496	0.2486	1.1643	0.2319
Brain 2	92210	2.0185	1.2149	0.3021	1.2246	0.3030
Lion	100000	2.0651	1.2174	0.2180	1.2246	0.2228
Spiral	96538	1.7577	4.0000	0.0000	4.0079	0.2552
Bimba	149524	3.8332	3.0000	0.0000	3.0005	0.0649

Table 6.1: The performance of our linear spherical quasi-conformal parameterization algorithm.

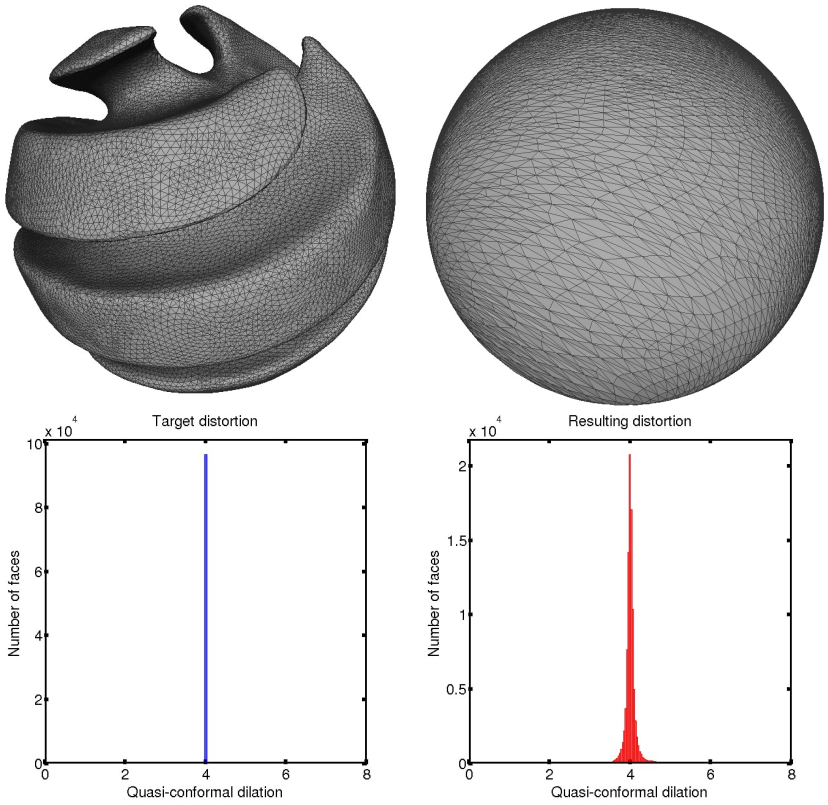

Figure 6.4: A spiral and the spherical Teichmüller parameterization obtained by our algorithm. Top left: the input surface. Top right: the spherical parameterization. Bottom left: The target quasi-conformal distortion. Bottom right: The resulting quasi-conformal distortion of the parameterization.

Chapter 7

Spherical conformal parameterization of genus-0 point clouds

7.1 Introduction

In this chapter, we present our proposed algorithm in [4] for the spherical conformal parameterization of genus-0 point clouds.

Recall that the linear spherical conformal parameterization algorithm [1] is developed for genus-0 triangulated meshes. In this work, we aim to develop an analogous algorithm for point clouds with spherical topology. Note that the Laplace-Beltrami (LB) operator is involved in Equation (4.2). We first propose a new weight function for accurately approximating the LB operators on point clouds. with the accurate approximation, Equation (4.2) can be solved on point clouds. Also, we replace the "south pole" step of the algorithm in [1], which involves computing quasi-conformal maps, by an iterative scheme called the *North-South reiteration*. Furthermore, we introduce a balancing scheme for achieving an even distribution of the spherical parameterization. Using our proposed parameterization algorithm, we can easily generate high quality triangulations and quadrangulations on genus-0 point clouds with

guarantee. An application in constructing multilevel representations of the point clouds is also explained.

7.2 Contributions

The contributions of our work are highlighted as follows.

(i) We develop an improved approximation of the Laplace-Beltrami operator using the moving least square method [49, 52, 53] together with a new Gaussian-type weight function.

(ii) We propose an iterative algorithm for computing the spherical conformal parameterizations of genus-0 point clouds. The distribution of the spherical parameterization is improved using our proposed balancing scheme.

(iii) Our proposed algorithm is effective. The algorithm completes within a few minutes and can handle highly convoluted point clouds.

(iv) Our proposed parameterization algorithm can be applied for meshing genus-0 point clouds. The conformality of the parameterization ensures the good quality of the resulting triangular/quad meshes. The topology of the meshes is guaranteed to be spherical.

(v) Multilevel representations of genus-0 point clouds can be easily constructed using our proposed algorithm.

7.3 Our proposed method

In this section, we explain our proposed framework for computing the spherical conformal parameterizations of genus-0 point clouds. Let $P = \{z_1, z_2, \ldots, z_n\}$

Weight	Formula of $w(d)$
Constant weight	$w(d) = 1$
Exponential weight	$w(d) = \exp\left(-\frac{d^2}{h^2}\right)$
Inverse of squared distance weight	$w(d) = \frac{1}{d^2 + \epsilon^2}$
Wendland weight [82, 83]	$w(d) = \left(1 - \frac{d}{D}\right)^4 \left(\frac{4d}{D} + 1\right)$
Special weight [52]	$w(d) = \begin{cases} 1 & \text{if } d = 0 \\ \frac{1}{k} & \text{if } d \neq 0 \end{cases}$

Table 7.1: Some common weighting functions for the moving least-square approximation.

be a point cloud sampled from a genus-0 closed surface $\mathcal{M}$. we aim to find a point cloud mapping $\tilde{f} : P \to \mathbb{S}^2$ which effectively resembles the conformal map $f : \mathcal{M} \to \mathbb{S}^2$.

7.3.1 Approximation of the Laplace-Beltrami operator

First, we approximate the Laplace-Beltrami operator in Equation (4.2) on P. To achieve this, we need to approximate the derivatives on P. For simplicity, we only discuss the approximation on the patch $\mathcal{N}(z_s)$ of a point $z_s \in P$. Recall that $\mathcal{N}(z_s)$ can be regarded as a graph of its projection $\hat{\mathcal{N}}(z_s)$, that is, $z_s^i = z_s + x_s^i e_s^1 + y_s^i e_s^2 + f_s(x_s^i, y_s^i) e_s^3$. Now, we need to select a set of basic functions $\{f_s^1, f_s^2, \ldots, f_s^m\}$ as a basis and write $f_s(x, y) \approx \sum_{i=1}^m c_i f_s^i(x, y)$, where $\{c_i\}_{i=1}^m$ are to be determined. In our work, since the LB operator involves second derivatives, polynomials with at least second order are necessary. Therefore, we set $m = 6$ and use $\{1, x, y, x^2, xy, y^2\}$ as the basis.

Now, it suffices to determine the coefficients $\{c_i\}_{i=1}^m$. To achieve this goal, we apply the *moving least-square (MLS) method* [55, 50, 70, 16, 52, 53], which has been widely used in computing point cloud mappings. The MLS method

aims to minimize

$$\sum_{i=1}^{n} w_i \left(\sum_{j=1}^{m} c_j f_s^j(x_i, y_i) - f_s(x_i, y_i) \right)^2 \tag{7.1}$$

where $w_i = w(\|z_i - z_s\|)$ for some weighting function $w : \mathbb{R} \to \mathbb{R}$. It is noteworthy that the choice of the weight function w significantly affects the accuracy of the approximation. Some weighting functions are listed in Table 7.1.

In this work, we propose a new weight function to approximating the LB operator. Consider the approximation at a point z_s. It is conceivable that the information provided by the points closer to z_s are more reliable than that of the points far away from z_s. This motivates us to use a Gaussian-type weight function, which concentrates at z_s:

$$\begin{cases} w_s = w(0) = 1 \\ w_i = w(\|z_i - z_s\|) = \dfrac{1}{k} \exp\left(-\dfrac{\sqrt{k}}{h^2} \|z_i - z_s\|^2 \right) \quad \text{for all } i \neq s, \end{cases} \tag{7.2}$$

where h is the maximum distance from z_s in $\mathcal{N}^k(z_s)$.

Then, we solve the minimization problem (7.1). Denote $f_{s,i}^j = f_s^j(x_i, y_i)$ and $f_{s,i} = f_s(x_i, y_i)$. Let $\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}$, $\vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$, $\vec{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}$, and $\vec{b} = \begin{pmatrix} f_{s,1} \\ f_{s,2} \\ \vdots \\ f_{s,n} \end{pmatrix}$. The minimization problem in (7.1) can be written as

$$\min_{c \in \mathbb{R}^n} \left\langle \vec{D}(\vec{A}c - \vec{b}), \vec{A}c - \vec{b} \right\rangle. \tag{7.3}$$

The above problem can be solved using the least-square method:

$$\vec{A}^T \vec{D} \vec{A} \vec{c} = \vec{A}^T \vec{D} \vec{b}. \tag{7.4}$$

Also, we can approximate any function u defined on $\mathcal{N}(z)$ by a combination of $\{f_s^1, f_s^2, \ldots, f_s^m\}$:

$$u = f_s(x, y) \approx \sum_{i=1}^{m} \hat{c}_i f_s^i(x, y). \tag{7.5}$$

Similarly, the coefficients $\hat{c}_i$ can be determined. Let $\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}$, $\vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$, $\vec{\hat{c}} = \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_m \end{pmatrix}$, and $\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$. $\hat{c}_i$ can be obtained by solving

$$\vec{A}^T \vec{D} \vec{A} \vec{\hat{c}} = \vec{A}^T \vec{D} \vec{u}. \tag{7.6}$$

As we know the explicit formula of the derivatives of every f_s^i, we can compute the approximated derivatives of u, for instance,

$$\frac{\partial u}{\partial x} = \sum_{i=1}^{m} \vec{c}_i \frac{\partial f_s^i}{\partial x} \quad \text{and} \quad \frac{\partial u}{\partial y} = \sum_{i=1}^{m} \vec{c}_i \frac{\partial f_s^i}{\partial y}. \tag{7.7}$$

Then, for any smooth real-valued function u on $\mathcal{N}(z_s)$, the LB operator of u is given by

$$\Delta u(z) = \frac{1}{W} \sum_{i,j=1}^{2} \partial_i (g^{ij} W \partial_j (u(z))), \tag{7.8}$$

where $z \in \mathcal{N}(z_s)$, (g_{ij}) is the surface metric at z, $W = \sqrt{det(g_{ij})}$, and $(g^{ij}) = (g_{ij})^{-1}$.

Since $z_s^i = (x_s^i, y_s^i, f_s(x_s^i, y_s^i))$ and $\mathcal{N}(z_s)$ is a graph of $\hat{\mathcal{N}}(z_s)$, we have

$$(g_{ij}) = \begin{pmatrix} 1 + (f_s)_x^2 & (f_s)_x (f_s)_y \\ (f_s)_x (f_s)_y & 1 + (f_s)_y^2 \end{pmatrix} \quad \text{and} \quad (g^{ij}) = \frac{1}{W^2} \begin{pmatrix} 1 + (f_s)_y^2 & -(f_s)_x (f_s)_y \\ -(f_s)_x (f_s)_y & 1 + (f_s)_x^2 \end{pmatrix}, \tag{7.9}$$

where $W = \sqrt{1 + (f_s)_x^2 + (f_s)_y^2}$.

The first order partial derivatives of f_s can be obtained using Equation (7.7). Since we have a closed form of Δu and the LB operator is second order, by differentiating Equation (7.8), we have

$$\Delta u(z_s) = \alpha_1 \frac{\partial u}{\partial x}(z_s) + \alpha_2 \frac{\partial u}{\partial y}(z_s) + \alpha_3 \frac{\partial^2 u}{\partial x^2}(z_s) + \alpha_4 \frac{\partial^2 u}{\partial x \partial y}(z_s) + \alpha_5 \frac{\partial^2 u}{\partial y^2}(z_s) \quad (7.10)$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ are coefficients which depend on partial derivatives of f_s. This completes our approximation for the LB operator.

7.3.2 Spherical conformal parameterization of genus-0 point clouds

We are now ready to introduce our proposed spherical conformal parameterization algorithm of genus-0 point clouds.

Denote the approximated LB operator in Equation (4.2) on P by Δ_{PC}. A map $\phi : P \to \mathbb{C}$ can then be solved using

$$\Delta_{PC}\phi = 0 \quad (7.11)$$

subject to the constraints $\phi(a_i) = b_i$ for $i = 1, 2, 3$, where $a_i, b_i \in \mathbb{C}$. In the case of triangular meshes, a_1, a_2, a_3 are chosen to be the three vertices of the most regular triangular face on the triangular mesh [1], where the regularity of a triangle $[a_1, a_2, a_3]$ is defined by

$$\text{Regularity}[a_1, a_2, a_3] = \left|\alpha - \frac{\pi}{3}\right| + \left|\beta - \frac{\pi}{3}\right| + \left|\gamma - \frac{\pi}{3}\right|. \quad (7.12)$$

Here, α, β and γ are the three angles in the triangle $[a_1, a_2, a_3]$. However, since point clouds do not contain any connectivity information, we do not have the concepts of face and angle and hence the above definition of regularity is not applicable.

To overcome this problem, we consider forming a triple $[z_s, z_s^i, z_s^j]$ using z_s and any two other points z_s^i and z_s^j in $\mathcal{N}(z)$, where $i \neq j$. Then, we choose

the three points a_1, a_2, a_3 in the constraint of Equation (7.11) by considering

$$\min_{s,i,j} \text{ Regularity}[z_s, z_s^i, z_s^j] \tag{7.13}$$

over all triples $[z_s, z_s^i, z_s^j]$. With the chosen boundary constraint, we solve Equation (7.11) and apply the inverse stereographic projection P_N^{-1} on $\phi(P)$ to obtain a spherical point cloud.

As suggested in [1], the spherical map obtained suffers from a serious drawback that the conformality distortion around the north pole is significant. In [1], the distortion is corrected by the "south pole" step, which involves the approximation of Beltrami coefficients on triangular faces and Theorem 3.3.3. Since we do not have a proper discretization of the Beltrami coefficients on point clouds that satisfies the composition formula (3.10), it is necessary to replace the "south pole" step in [1] by a new approach applicable on point clouds.

In our "south pole" step for point clouds, we propose to correct the conformality distortion around the north pole by solely using the Laplace-Beltrami operator. First, we apply the south-pole stereographic projection P_S. Note that the northernmost and the southernmost regions of $\mathbb{S}^2$ are respectively mapped to the innermost and the outermost regions on $\mathbb{C}$. Hence, the conformality distortion at the outermost region is negligible while that at the innermost part is serious. With the outermost region as the boundary constraints, we re-solve the Laplace equation to obtain $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \to \mathbb{C}$ as described below:

$$\Delta_{PC}\psi = 0 \tag{7.14}$$

subject to the boundary constraints $\psi(x) = x$ for all points x at the outermost low-distortion region. Because of the conformality at the outermost region, Equation (7.14) gives us a map ψ with improved conformality at the innermost

region. Then, we apply the inverse south-pole stereographic projection P_S^{-1} and obtain a composition map

$$\widetilde{f} = P_S^{-1} \circ \psi \circ P_S \circ P_N^{-1} \circ \phi. \tag{7.15}$$

With this strategy, we can improve the conformality of the spherical parameterization without introducing the concept of Beltrami coefficients and the related properties on point clouds. It is noteworthy that the underlying principle of our proposed approach and the "south pole" step in [1] is equivalent. The equivalence follows from the theorem below.

Theorem 7.3.1. *Let $(S_1, \sigma|dz|^2)$ and $(S_2, \rho|dw|^2)$ be two Riemann surfaces, and let μ be a prescribed Beltrami differential on S_1. Then, the map solved by Equation (3.12) is a harmonic map between $(S_1, |dz + \mu d\bar{z}|^2)$ and $(S_2, \rho|dw|^2)$. Hence, solving the Laplace equation (7.14) is equivalent to solving the generalized Laplace equation (3.12).*

Proof. Denote the coordinates of S_1 with respect to the distorted metric $|dz+\mu d\bar{z}|^2$ by ζ. The harmonic map between $(S_1, |dz+\mu d\bar{z}|^2)$ and $(S_2, \rho|dw|^2)$ is a critical point of the energy

$$E_{harm}(h) = \int_{S_1} \rho(h(\zeta))(|h_\zeta|^2 + |h_{\bar{\zeta}}|^2) dxdy. \tag{7.16}$$

On the other hand, from the Beltrami equation (3.7), we can easily see that the solution to Equation (3.12) is the critical point of the following energy

$$E_{QC}(f) = \int_{S_1} \rho(f(z))(|f_{\bar{z}} - \mu f_z|^2) dxdy. \tag{7.17}$$

It is shown in [63] that the two energy functionals have the same set of critical points. Hence, solving the two equations are equivalent. □

Thus, under appropriate boundary conditions in Equation (3.12) and Equation (7.14), both of our proposed approach and the "south pole" step in [1] are guaranteed to produce a conformal map in the continuous case.

However, in the discrete case, the accuracy of our approach is more dependent on the boundary constraints than that of the "south pole" step. A small error in the boundary constraints may slightly affect the map obtained by Equation (7.14). Hence, we introduce an iterative scheme called the *North-South (N-S) reiterations* to further enhance the parameterization result until it becomes stable.

In each N-S reiteration, we first project the previous spherical parameterization result onto $\mathbb{C}$ using P_N. Next, we compute a harmonic map $\widetilde{\phi} : (P_N \circ \widetilde{f})(P) \to \mathbb{C}$ by solving the Laplace equation

$$\Delta_{PC}\widetilde{\phi} = 0 \tag{7.18}$$

with the boundary constraints $\widetilde{\phi}(x) = x$ for the outermost $r\%$ of the points on $\mathbb{C}$. Here, $r\%$ of the points are used as boundary constraints in order to avoid ill-poseness in the numerical computations of the large matrix equation. After obtaining $\widetilde{\phi}$, we apply P_N^{-1} and then the south-pole stereographic projection P_S. We then compute another harmonic map $\widetilde{\psi} : (P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f})(P) \to \mathbb{C}$ by solving the Laplace equation

$$\Delta_{PC}\widetilde{\psi} = 0 \tag{7.19}$$

with the boundary constraints $\widetilde{\psi}(x) = x$ for the outermost $r\%$ of the points on $\mathbb{C}$. Denote the updated spherical parameterization by

$$P_S^{-1} \circ \widetilde{\psi} \circ P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f}. \tag{7.20}$$

We decide whether to continue the reiterations or stop by checking the difference between $P_S^{-1} \circ \widetilde{\psi} \circ P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f}$ and $\widetilde{f}$. A negligible difference

implies that the parameterization is stable and hence we can stop. If the difference is not negligible, we repeat the N-S reiteration. In our experiment, we set $r = 10$. Our proposed spherical conformal parameterization algorithm for genus-0 point clouds is summarized in Algorithm 8.

Algorithm 8: Spherical conformal parameterization algorithm for genus-0 point clouds

Input: A genus-0 point cloud P.
Output: A spherical conformal parameterization $f : P \to \mathbb{S}^2$.

Approximate the LB operator Δ_{PC} on P;
Find the most regular triple by solving problem (7.13);
Obtain $\phi : P \to \overline{\mathbb{C}}$ by solving the Laplace equation (7.11);
Apply the inverse stereographic projection P_N^{-1} on $\phi(P)$;
Apply the south-pole stereographic projection P_S on $(P_N^{-1} \circ \phi)(P)$;
Solve the Laplace equation (7.14) for $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \to \mathbb{C}$;
Apply the inverse south-pole stereographic projection P_S^{-1} and denote the composition of the maps by $f = P_S^{-1} \circ \psi \circ P_S \circ P_N^{-1} \circ \phi$;
repeat
    Update $\widetilde{f}$ by f;
    Solve the Laplace equation (7.18) for $\widetilde{\phi} : (P_N \circ \widetilde{f})(P) \to \mathbb{C}$;
    Solve the Laplace equation (7.19) for $\widetilde{\psi} : (P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f})(P) \to \mathbb{C}$;
    Update f by $P_S^{-1} \circ \widetilde{\psi} \circ P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f}$;
until $mean(\|f(p_i) - \widetilde{f}(p_i)\|^2) < \epsilon$;

The explanation of our proposed N-S reiteration scheme is as follows. Define the *N-S Dirichlet energy* by

$$\tilde{E}(f) = \frac{1}{2}\left(E(P_N(f)) + E(P_S(f))\right), \tag{7.21}$$

where $E(f)$ is the Dirichlet energy. Obviously, $\tilde{E}$ is minimized if and only if $E(P_N(f))$ and $E(P_S(f))$ are minimized, which implies that $E(f)$ is minimized and f is conformal. Therefore, to find a conformal map f, we consider minimizing $\tilde{E}(f)$. More specifically, we aim to minimize both $E(P_N(f))$ and

$E(P_S(f))$. Note that solving the Laplace equations (7.18) and Equation (7.19) helps minimizing two energies. Moreover, since Equation (7.18) and Equation (7.19) are linear, the computation is highly efficient. Furthermore, since both the north-pole step and the south-pole step are involved in each N-S reiteration, the errors induced by the stereographic projections can be corrected. This explains the motivation and advantage of our proposed N-S reiteration scheme.

Then, we explain the validity of the boundary constraints. Note that for numerical stability, we fix $r\%$ of the points, instead of 3 points, in solving Equation (7.18) and Equation (7.19). Fixing the extra points do not largely affect the accuracy of the result, as explained by the following theorem:

Theorem 7.3.2 (Beltrami holomorphic flow on $\overline{\mathbb{C}}$ [30])**.** *There exists a 1-1 correspondence between the set of quasi-conformal diffeomorphisms of* $\overline{\mathbb{C}}$ *that fix* $0, 1, \infty$ *and the set of smooth complex-valued functions* μ *on* $\overline{\mathbb{C}}$ *with* $\|\mu\|_\infty = k < 1$*. Moreover, the solution* f^μ *to the Beltrami equation (3.7) depends holomorphically on* μ*. Let* $\{\mu(t)\}$ *be a family of Beltrami coefficients depending on a parameter* t*. Suppose that* $\mu(t)$ *can be written in the form*

$$\mu(t)(z) = \mu(z) + t\nu(z) + t\epsilon(t)(z) \tag{7.22}$$

for $z \in \mathbb{C}$*, with suitable* μ *in the unit ball of* $C^\infty(\mathbb{C}), \nu, \epsilon(t) \in L^\infty(\mathbb{C})$ *such that* $\|\epsilon(t)\|_\infty \to 0$ *as* $t \to 0$*. Then, for all* $w \in \mathbb{C}$*,*

$$f^{\mu(t)}(w) = f^\mu(w) + tV(f^\mu, \nu)(w) + o(|t|) \tag{7.23}$$

locally uniformly on $\mathbb{C}$ *as* $t \to 0$*, where*

$$V(f^\mu, \nu)(w) = -\frac{f^\mu(w)(f^\mu(w)-1)}{\pi} \int_{\mathbb{C}} \frac{\nu(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z)-1)(f^\mu(z)-f^\mu(w))} dxdy. \tag{7.24}$$

When we solve Equation (7.18) or Equation (7.19), since the conformality distortion of the outermost region is negligible, ν is compactly supported around the origin. Thus, from Equation (7.24), the points located farther away from the origin are associated with a smaller flow V, as the denominators in the integral at those points are larger. Consequently, the outermost points will remain almost unchanged in each iteration, while the innermost points will be adjusted and the conformality there is improved. Hence, extra boundary constraints can be used for enhancing the numerical stability.

Finally, we remark that our proposed algorithm can also serve as an algorithm for the spherical conformal parameterizations of genus-0 closed triangular meshes, since the LB operator can be easily constructed on meshes.

7.3.3 Improving the distribution of the spherical parameterization

In [1], Theorem 4.3.2 computes an even distribution of the spherical parameterization for meshes. Under the stereographic projection, the outermost triangle T and the innermost triangle t on $\mathbb{C}$ are considered. A scaling is applied so that $P_N^{-1}(T)$ and $P_N^{-1}(t)$ are with the same perimeters on $\mathbb{S}^2$, which leads to an even distribution.

For genus-0 point clouds, we cannot define the outermost triangle T and the innermost triangle t because of the absence of the mesh structure. Hence, the above method is not applicable. Nevertheless, we can extend Theorem 4.3.2 for genus-0 point clouds by considering two sets of points instead of two triangles on $\mathbb{C}$.

Theorem 7.3.3. *Let $\{u_i\}_{i=0}^m$ and $\{v_j\}_{j=0}^n$ be two sets of points on $\mathbb{C}$. Then*

$$\left(\sum_{i=1}^{m}\|\lambda u_i-\lambda u_0\|\right)\left(\sum_{j=1}^{n}\left\|P_S(P_N^{-1}(\lambda v_j))-P_S(P_N^{-1}(\lambda v_0))\right\|\right)$$
$$=\left(\sum_{i=1}^{m}\|u_i-u_0\|\right)\left(\sum_{j=1}^{n}\left\|P_S(P_N^{-1}(v_j))-P_S(P_N^{-1}(v_0))\right\|\right)$$

for any scaling factor $\lambda\neq 0$. In other words, the above product is an invariance under arbitrary scaling.

Proof. The proof is similar as the one of Theorem 4.3.2. For any $z=x+iy$, we have

$$\begin{aligned}P_S(P_N^{-1}(z)) &= P_S(P_N^{-1}(x+iy))\\ &= \frac{-\frac{2x}{1+x^2+y^2}}{1+\frac{-1+x^2+y^2}{1+x^2+y^2}}+i\frac{\frac{2y}{1+x^2+y^2}}{1+\frac{-1+x^2+y^2}{1+x^2+y^2}}\\ &= \frac{-x}{x^2+y^2}+i\frac{y}{x^2+y^2}=\frac{-Re(z)}{|z|^2}+i\frac{Im(z)}{|z|^2}.\end{aligned}\tag{7.25}$$

For any scaling factor $\lambda\neq 0$, we have

$$\begin{aligned}&\left(\sum_{i=1}^{m}\|\lambda u_i-\lambda u_0\|\right)\left(\sum_{j=1}^{n}\left\|P_S(P_N^{-1}(\lambda v_j))-P_S(P_N^{-1}(\lambda v_0))\right\|\right)\\ &=\left(\sum_{i=1}^{m}\|\lambda u_i-\lambda u_0\|\right)\left(\sum_{j=1}^{n}\left\|\frac{-Re(\lambda v_j)}{|\lambda v_j|^2}+i\frac{Im(\lambda v_j)}{|\lambda v_j|^2}-\frac{-Re(\lambda v_0)}{|\lambda v_0|^2}+i\frac{Im(\lambda v_0)}{|\lambda v_0|^2}\right\|\right)\\ &=\left(\lambda\sum_{i=1}^{m}\|u_i-u_0\|\right)\left(\frac{\lambda}{\lambda^2}\sum_{j=1}^{n}\left\|\frac{-Re(v_j)}{|v_j|^2}+i\frac{Im(v_j)}{|v_j|^2}-\frac{-Re(v_0)}{|v_0|^2}+i\frac{Im(v_0)}{|v_0|^2}\right\|\right)\\ &=\left(\sum_{i=1}^{m}\|u_i-u_0\|\right)\left(\sum_{j=1}^{n}\left\|P_S(P_N^{-1}(v_j))-P_S(P_N^{-1}(v_0))\right\|\right).\end{aligned}\tag{7.26}$$

□

Now, to obtain an even distribution on $\mathbb{S}^2$, we propose to use the average distance between the poles on $\mathbb{S}^2$ and their k-NN neighborhoods. Let $f: P\rightarrow$

$\mathbb{S}^2$ be the spherical parameterization obtained by Algorithm 8. Denote the northernmost and the southernmost points on $f(P)$ by v_N and v_S respectively. Let $x_N = P_N(v_N)$ and $x_S = P_S(v_S)$. Denote the average distances of x_N and x_S to their k-NN neighborhoods on the corresponding planar domains by d_N and d_S respectively. We have

$$d_p = \text{mean}(\{|P_N(f(z)) - x_N| : z \in \mathcal{N}^k(f^{-1}(v_N))\}) \tag{7.27}$$

and

$$d_s = \text{mean}(\{|P_S(f(z)) - x_S| : z \in \mathcal{N}^k(f^{-1}(v_S))\}). \tag{7.28}$$

Then, we scale the whole planar domain $(P_N \circ f)(P)$ by a factor

$$\lambda = \frac{\sqrt{d_p \times d_s}}{d_p}. \tag{7.29}$$

Now, denote the two updated average distances by $\tilde{d}_p$ and $\tilde{d}_s$. We have

$$\tilde{d}_p = \lambda d_p = \frac{\sqrt{d_p \times d_s}}{d_p} \times d_p = \sqrt{d_p \times d_s}. \tag{7.30}$$

Also, it follows from Theorem 7.3.3 that

$$\tilde{d}_p \times \tilde{d}_s = d_p \times d_s. \tag{7.31}$$

Hence,

$$\tilde{d}_s = d_p \times d_s \times \frac{1}{\tilde{d}_p} = \sqrt{d_p \times d_s}. \tag{7.32}$$

In other words, we have $\tilde{d}_p = \tilde{d}_s$, which implies the distribution at the two poles of $P_N^{-1}(\lambda(P_N(f(P))))$ is now balanced. The balancing scheme is summarized in Algorithm 9. In conclusion, by combining Algorithm 8 and Algorithm 9, we can obtain spherical conformal parameterizations of genus-0 point clouds with an even distribution.

Finally, we outline a possible extension of our proposed parameterization algorithm for a disk-type point cloud P by developing a point cloud double

Algorithm 9: Our proposed balancing scheme for better distribution.

Input: A spherical conformal parameterization $f : P \to \mathbb{S}^2$.

Output: A spherical conformal parameterization with improved distribution.

1 Apply the stereographic projection P_N on $f(P)$;

2 Denote the northernmost and the southernmost points of $f(P)$ by v_N and v_S respectively. Multiply all points in $P_N(f(P))$ by a scaling factor $\lambda = \frac{\sqrt{d_p \times d_s}}{d_p}$, where $d_p = \text{mean}(\{|P_N(f(z)) - x_N| : z \in \mathcal{N}^k(f^{-1}(v_N))\})$ and $d_s = \text{mean}(\{|P_S(f(z)) - x_S| : z \in \mathcal{N}^k(f^{-1}(v_S))\})$;

3 Apply the inverse stereographic projection P_N^{-1} on $\lambda(P_N(f(P)))$;

covering technique analogous to [32, 3] to turn a point cloud with disk topology into a genus-0 point cloud:

Step 1: Approximate the derivatives on P.

Step 2: Duplicate P and denote the copy of it by P'.

Step 3: Define the derivatives on P' using the approximations in Step 1 with orientations reversed.

Step 4: Identify the boundary points of P, P' and obtain a genus-0 point cloud $\widetilde{P}$.

Step 5: Using the derivatives on P and P', create the LB operator on $\widetilde{P}$.

After the above steps, we can apply our proposed spherical conformal parameterization algorithm on $\widetilde{P}$. Applying P_N on a hemisphere of the spherical parameterization yields a conformal parameterization of the disk-type point cloud P.

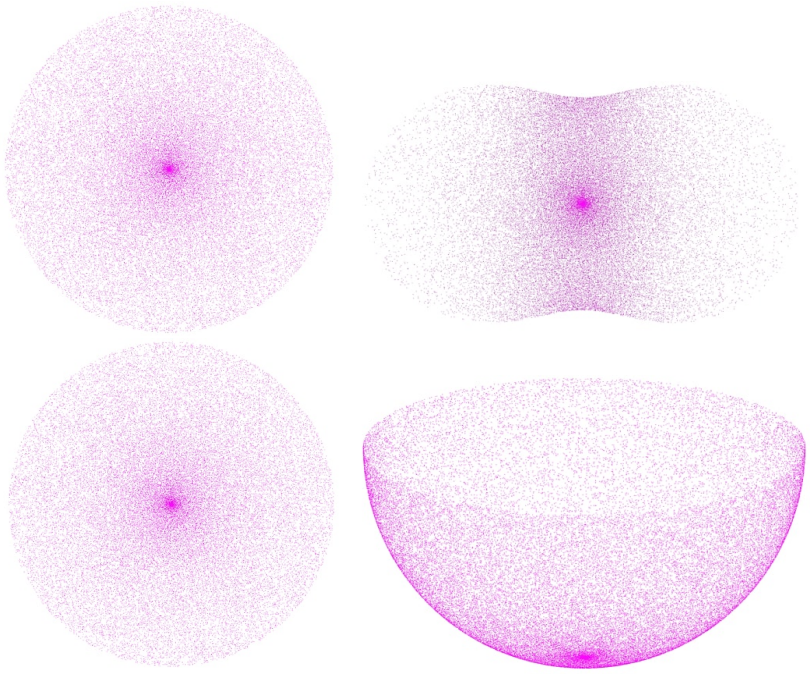

Figure 7.1: Two experiments for evaluating the approximation accuracies of the LB operator. Top: the first experiment. Bottom: the second experiment.

7.4 Experimental results

In this section, we demonstrate the effectiveness of our proposed algorithm. The datasets used in the experiments are adopted from the AIM@SHAPE Shape Repository [89], the Stanford 3D Scanning Repository [94] and the RGB-D Scenes Dataset v.2 [48]. Our proposed algorithm is implemented in MATLAB. To solve the sparse linear systems, the built-in backslash operator (\) in MATLAB is used. All experiments are performed on a PC with an Intel(R) Core(TM) i5-3470 CPU @3.20 GHz processor and 8.00 GB RAM.

7.4.1 Performance of our approximation of the Laplace-Beltrami operator

First, we compare the numerical accuracy of the following approaches for the approximation of the LB operator on point clouds:

Method	maximum position error	average position error
Local mesh method [46]	1.3427	0.0179
MLS with the Wendland weight in [82, 83]	3.3074	0.1696
MLS with the Gaussian weight in [13, 14]	0.5697	0.0114
MLS with the special weight in [52]	0.0427	0.0006
MLS with our proposed weight	0.0245	0.0004

Method	maximum position error	average position error
Local mesh method [46]	1.5148	0.0271
MLS with the Wendland weight in [82, 83]	2.0082	0.0803
MLS with the Gaussian weight in [13, 14]	1.5460	0.0925
MLS with the special weight in [52]	0.0110	0.0001
MLS with our proposed weight	0.0103	0.0002

Table 7.2: The approximation error in the two experiments. Top: the first experiment. Bottom: the second experiment.

1. the local mesh method [46],

2. the MLS method with the Wendland weight in [82, 83],

3. the MLS method with the Gaussian weight in [13, 14],

4. the MLS method with the special weight in [52], and

5. the MLS method with our proposed weight function.

The evaluation of the accuracy is performed in the following way. In each experiment, a point cloud is generated on the unit disk. The point cloud serves as the ground truth result. Then, the point cloud is transformed by a conformal map with an explicit formula. On the transformed point cloud, we apply the above schemes to approximate the LB operator. With the approximated LB operators, we solve the Laplace equation with the circular boundary constraints on the unit disk. Theoretically, the resulting map should

be exactly the same as the original point cloud. By measuring the maximum and average position error between the resulting map and the ground truth result, we can assess the accuracy of the approximations of the LB operator. Two experiments are illustrated in Figure 7.1.

Table 7.2 records the approximation errors obtained in the two experiments. In both experiments, the MLS method with our proposed Gaussian-type weight function produces much more accurate results than the local mesh method [46] and the MLS method with the Wendland weight [82, 83] and the Gaussian weight [13, 14]. Also, the maximum position error of our proposed scheme is about 25% less than that of the MLS method with the special weight [52] on average. The experimental results demonstrate the advantage of our proposed approximation scheme.

7.4.2 Performance of our proposed spherical conformal parameterization

Next, we evaluate the performance of our proposed spherical conformal parameterization algorithm for genus-0 point clouds. In our experiment, we consider $k = 25$ nearest neighbors in approximating the LB operator in Algorithm 8. The stopping threshold for the N-S reiteration scheme is set to be $\epsilon = 0.0001$. Two experiments are shown in Figure 7.2 and Figure 7.3. Our proposed algorithm can also handle convoluted point clouds as shown in Figure 7.4. To have a better visualization of the parameterization, we create a Delaunay triangulation on the spherical parameterization by the spherical Delaunay algorithm. Using the triangulation created, we can define an induced triangulation on the original point cloud.

Using the mesh structures created above, we can quantitatively evaluate the conformality of our spherical parameterization scheme by measuring the

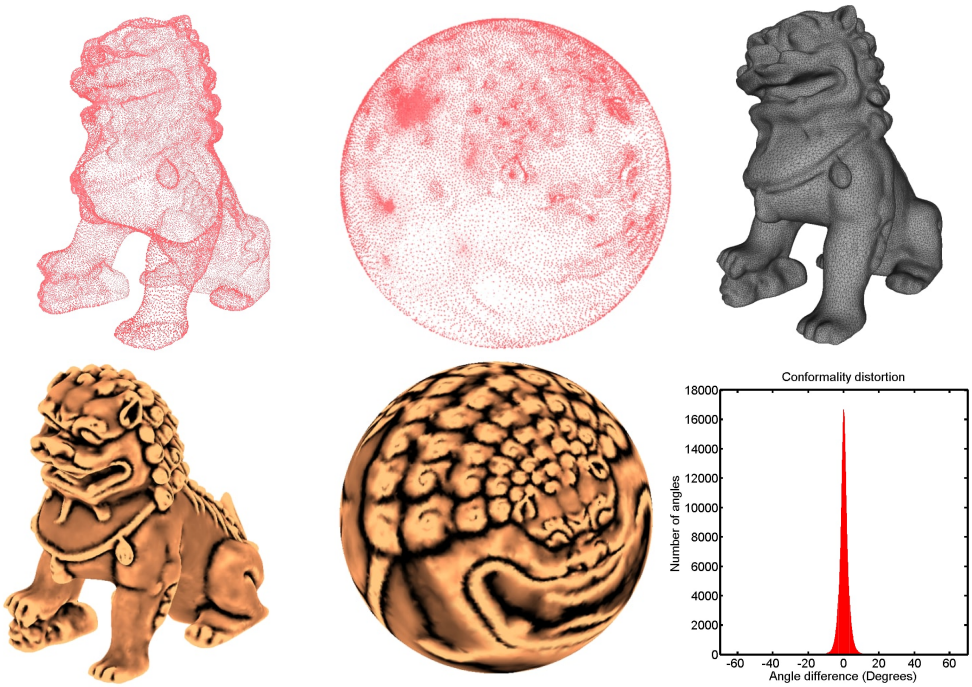

Figure 7.2: Applying our algorithm on a lion point cloud. Top left: A lion point cloud. Top middle: The spherical conformal parameterization. Top right: A triangulation created via the parameterization. Bottom left and middle: The triangulated point cloud and the spherical parameterization colored with the approximated mean curvature at each vertex. Bottom right: The conformality distortion.

angle distortions. In Figure 7.2 and Figure 7.3, the histograms of the angle differences highly concentrate at 0. It can be also observed from the colored figures that the local geometries are well-retained under the parameterizations. The results illustrate the conformality of our proposed algorithm.

Then, we demonstrate the convergence of our proposed North-South reiteration scheme. Several difference plots are shown in Figure 7.5. It can be observed that the parameterization results become stable after a few iterations.

Finally, we compare our proposed spherical conformal parameterization algorithm with the spherical embedding algorithm [88] and the global confor-

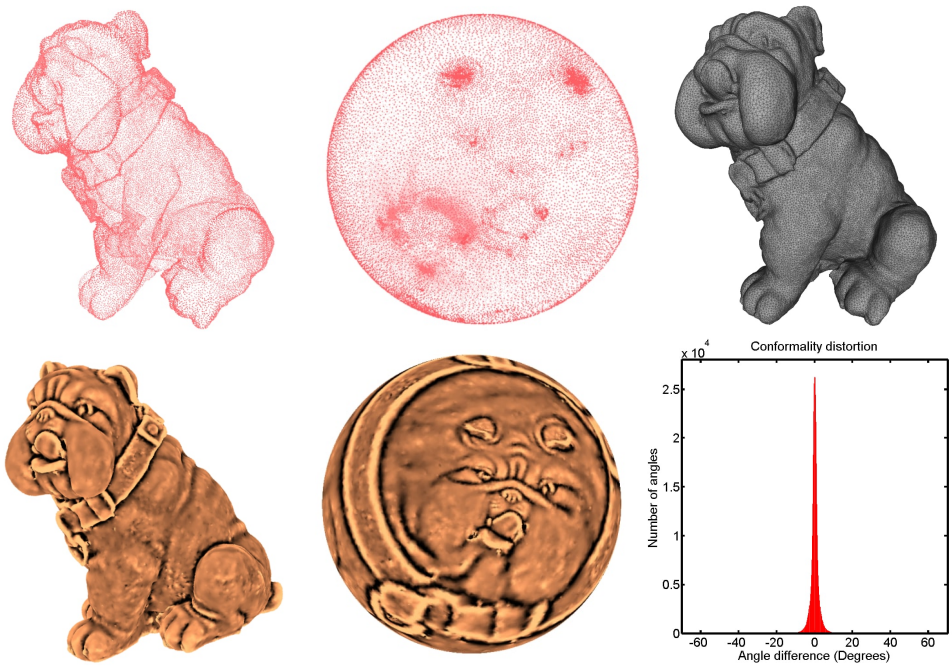

Figure 7.3: Applying our algorithm on a bulldog point cloud. Top left: A bulldog point cloud. Top middle: The spherical conformal parameterization. Top right: A triangulation created via the parameterization. Bottom left and middle: The triangulated point cloud and the spherical parameterization colored with the approximated mean curvature at each vertex. Bottom right: The conformality distortion.

mal map [52]. Table 7.3 records the performances of the three algorithms. It can be observed that our proposed algorithm outperforms the two other algorithms in terms of both the computational time and the conformality. This demonstrates the effectiveness of our spherical conformal parameterization algorithm.

7.5 Applications

Our spherical conformal parameterization algorithm can be applied for creating mesh structures and developing multilevel representations on genus-0 point clouds. In this section, we explain the two applications with numerical

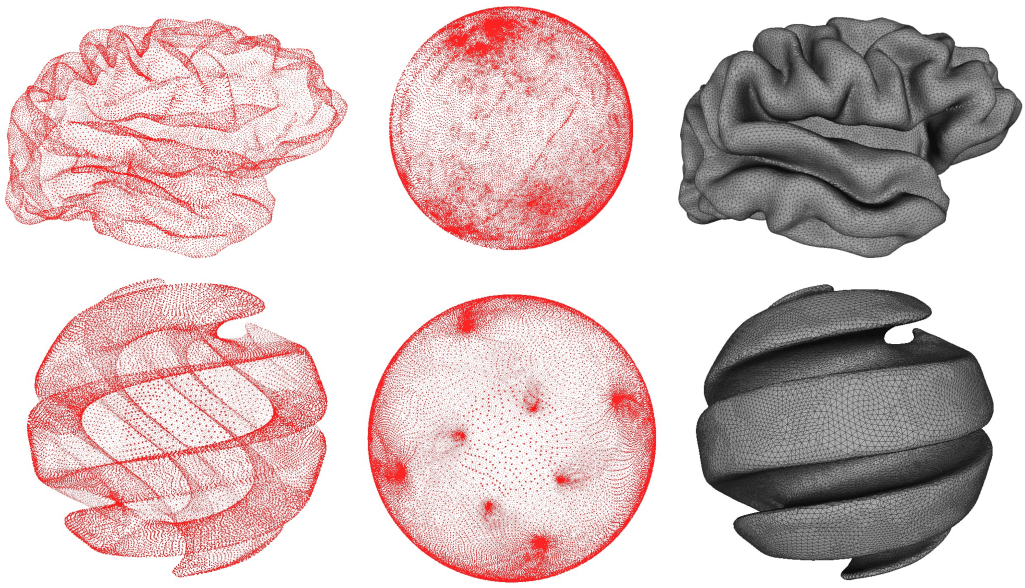

Figure 7.4: Parameterizing a convoluted human brain point cloud and a spiral point cloud. Left: The input point clouds. Middle: The spherical conformal parameterizations. Right: The triangulations created via the parameterization.

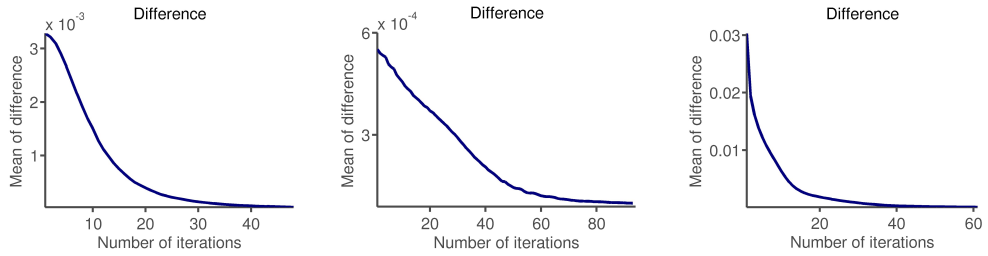

Figure 7.5: Plots of the difference mean($\|f(p_i) - \widetilde{f}(p_i)\|^2$) with the number of iterations in our proposed North-South reiteration scheme. Left: Cereal box. Middle: Hippocampus. Right: Bulldog.

experiments.

7.5.1 Meshing genus-0 point clouds

As mentioned in the last section, mesh structures can be created on genus-0 point clouds by building Delaunay triangulations on their spherical conformal parameterizations obtained by our algorithm. This meshing scheme has two

Point clouds	No. of points	Performance	Our proposed method	Spherical embedding [88]	Global conformal map [52]
Soda Can	6838	Time (s)	8.1768	23.7988	16.7058
		Mean($\|\delta\|$)	0.5902	4.0431	2.3352
		SD($\|\delta\|$)	0.8007	5.2731	1.9803
Hippocampus	10242	Time (s)	13.0919	37.4124	18.7151
		Mean($\|\delta\|$)	1.2855	14.3072	1.3062
		SD($\|\delta\|$)	1.4701	19.6461	1.5100
Max Planck	21530	Time (s)	30.7785	87.0887	40.4214
		Mean($\|\delta\|$)	0.7326	8.6058	1.0792
		SD($\|\delta\|$)	1.0803	14.0857	1.5756
Cereal Box	33061	Time (s)	50.7390	132.1765	
		Mean($\|\delta\|$)	0.6523	12.3573	Fail
		SD($\|\delta\|$)	0.9165	14.0440	
Spiral	48271	Time (s)	114.7057	291.8312	122.8818
		Mean($\|\delta\|$)	0.8580	16.4704	0.9658
		SD($\|\delta\|$)	1.3280	22.5073	1.3135
Brain	48487	Time (s)	115.5802	198.2285	126.7621
		Mean($\|\delta\|$)	1.4266	35.2629	2.2495
		SD($\|\delta\|$)	2.9093	35.7986	2.7430
Bulldog	49797	Time (s)	88.9297	206.9920	113.4447
		Mean($\|\delta\|$)	1.5432	16.2010	1.8700
		SD($\|\delta\|$)	2.9183	21.1544	3.1891
Chinese Lion	50002	Time (s)	95.8935	212.5685	136.3296
		Mean($\|\delta\|$)	1.8474	19.1579	2.4907
		SD($\|\delta\|$)	1.9286	22.7259	2.6207
Bimba	74764	Time (s)	198.6064	360.7178	227.0290
		Mean($\|\delta\|$)	0.6227	18.0340	0.6379
		SD($\|\delta\|$)	0.8129	20.6272	0.7975
Igea	134345	Time (s)	427.7658	731.8661	560.6077
		Mean($\|\delta\|$)	0.7076	5.0853	3.8293
		SD($\|\delta\|$)	1.4273	8.2623	2.9703
Armadillo	172974	Time (s)	676.4106	995.7537	
		Mean($\|\delta\|$)	1.4167	23.2354	Fail
		SD($\|\delta\|$)	1.6855	23.9892	
Lion Vase	256094	Time (s)	1305.9013	1484.7682	1642.9208
		Mean($\|\delta\|$)	2.0920	17.8501	3.6696
		SD($\|\delta\|$)	4.1052	21.9588	5.8502

Table 7.3: Performances of three spherical conformal parameterization methods for genus-0 point clouds. Here, δ denotes the angular distortion between the triangulations on the input point cloud and the parameterization.

advantages. First, note that the Delaunay triangulations built on $\mathbb{S}^2$ do not contain extreme angles. Because of the conformality of our spherical parameterization algorithm, the regular angle structures are well-preserved on the induced triangulations on the original point clouds. Consequently, the meshing results are guaranteed to be almost-Delaunay. the regularity of the triangulations generated is guaranteed by the preservation of the

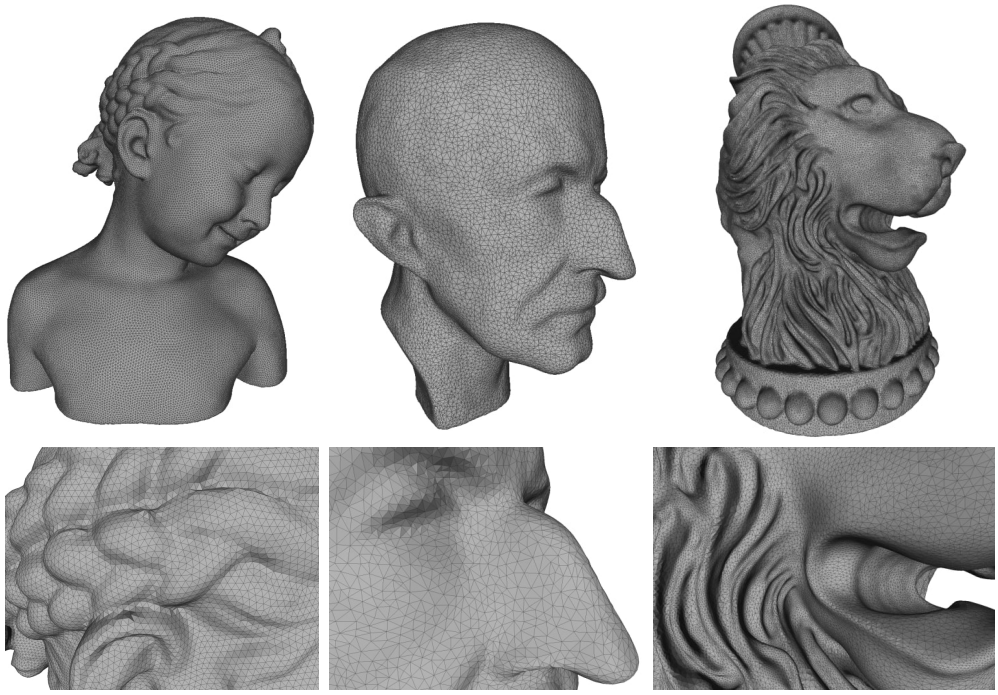

Figure 7.6: Meshes generated by our proposed method and a zoom-in of them. The high quality of the triangulations is attributed to our proposed spherical conformal parameterization algorithm.

angle structures of the Delaunay triangulations computed on the spherical parameterizations. As the angle structures are well retained under the spherical conformal parameterization, a regular triangulation defined on the parameterized point clouds can effectively induce a regular and almost-Delaunay triangulation on the original point clouds. As shown in Figures 7.2, 7.3 7.4 and 7.6, our algorithm can be applied on a large variety of genus-0 point clouds for generating high quality meshes. Second, unlike most of the existing meshing approaches, our proposed scheme is guaranteed to produce genus-0 closed meshes. The guarantee is attributed to the 1-1 correspondence between the input point cloud and the spherical parameterization obtained by our algorithm, as well as the property of Delaunay triangulations. In other words, there are no holes or undesirable artifacts on our meshing results.

Point clouds	Our proposed method	Spherical embedding [88]	Marching cubes [57]	Tight Cocone [19]
Soda Can	0.99	0.93	0.82	0.98
Hippocampus	0.99	0.91	0.82	0.99
Max Planck	0.99	0.93	0.82	0.99
Cereal Box	0.99	0.88	0.81	0.99
Spiral	1.00	0.85	0.82	0.99
Brain	0.99	0.82	0.82	0.99
Bulldog	1.00	0.86	0.82	0.99
Chinese Lion	0.99	0.84	0.82	0.99
Bimba	1.00	0.88	0.81	1.00
Igea	0.97	0.90	0.83	0.97
Armadillo	0.98	0.80	0.82	0.98
Lion Vase	0.99	0.85	0.83	0.99

Table 7.4: The Delaunay ratios of different meshing approaches.

We compare our meshing method with three pre-existing meshing schemes, including the spherical embedding approach [88], the marching cubes algorithm [57] and the Tight Cocone algorithm [19]. In Figure 7.7 , it can be observed that our meshing method and the Tight Cocone algorithm [19] produce high quality triangulations, while the approaches in [88] and [57] generate triangulations with certain sharp and irregular triangles. Also, the mesh generated by the marching cubes algorithm [57] contains holes while our meshing scheme is topology preserving. The ratio assesses the proportion of edges in the resulting triangulations that satisfy the opposite angle sum property $\alpha + \beta \leq \pi$ in a triangulation. A Delaunay ratio exactly equals 1 indicates that the triangulation is Delaunay.

To quantitatively evaluate the "almost-Delaunay" property of our meshing scheme, we recall that every Delaunay triangulation satisfies the opposite angle sum property: For every edge $[u, v]$, the two angles α and β opposite to

$[u, v]$ satisfy

$$\alpha + \beta \leq \pi. \tag{7.33}$$

Now, we define the *Delaunay ratio* of a triangulation by

$$\frac{\#\text{ of edges with opposite angles } \alpha, \beta \text{ s.t. } \alpha + \beta \leq \pi}{\text{Total \# of edges in the triangulation}}. \tag{7.34}$$

Obviously, a higher Delaunay ratio implies that the triangulation is closer to a perfect Delaunay triangulation. Table 7.4 records the Delaunay ratios of the meshing results by the abovementioned approaches. Thanks to the angle-preserving property of our parameterization algorithm, our meshing scheme outperforms the spherical embedding algorithm [88] and the marching cubes algorithm [57]. Also, the Delaunay ratios of our meshing results are comparable to (and sometimes slightly better than) those of the Tight Cocone algorithm [19].

In addition, quadrangulations of genus-0 point clouds can also be achieved with the aid of our spherical conformal parameterization. Given a genus-0 point cloud, we first apply our proposed spherical conformal parameterization algorithm. Then, we interpolate a standard quad mesh onto the spherical parameterization. This effectively induces a quadrangulation of the input point cloud. Two examples are given in Figure 7.8. Again, the high quality of the resulting quadrangulations is attributed to the conformality of our parameterization algorithm. Also, the quad meshes generated are guaranteed to be genus-0.

Besides, our proposed balancing scheme is important in generating admissible meshing results. Figure 7.9 shows the meshes created via our spherical parameterizations with and without the balancing scheme. It can be easily observed that a high quality mesh is generated only with the presence of the balancing scheme.

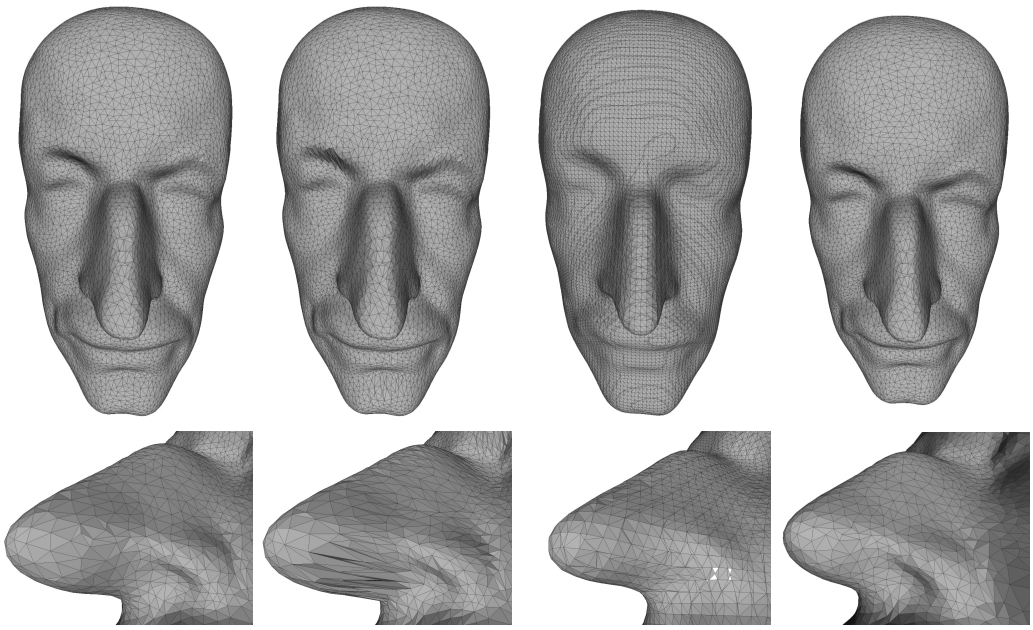

Figure 7.7: The meshing results by different approaches. Left to right: Our meshing scheme, spherical embedding [88], marching cubes [57] and Tight Cocone [19].

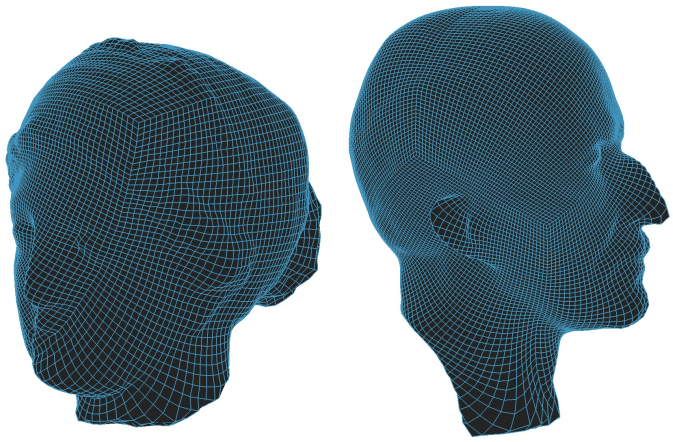

Figure 7.8: Quad mesh generation on point clouds using our spherical parameterization.

Before ending this subsection, we show that our meshing scheme is stable under geometrical and topological noises of the input genus-0 point clouds. Given a noisy point cloud, we first apply a Poisson filtering and compute

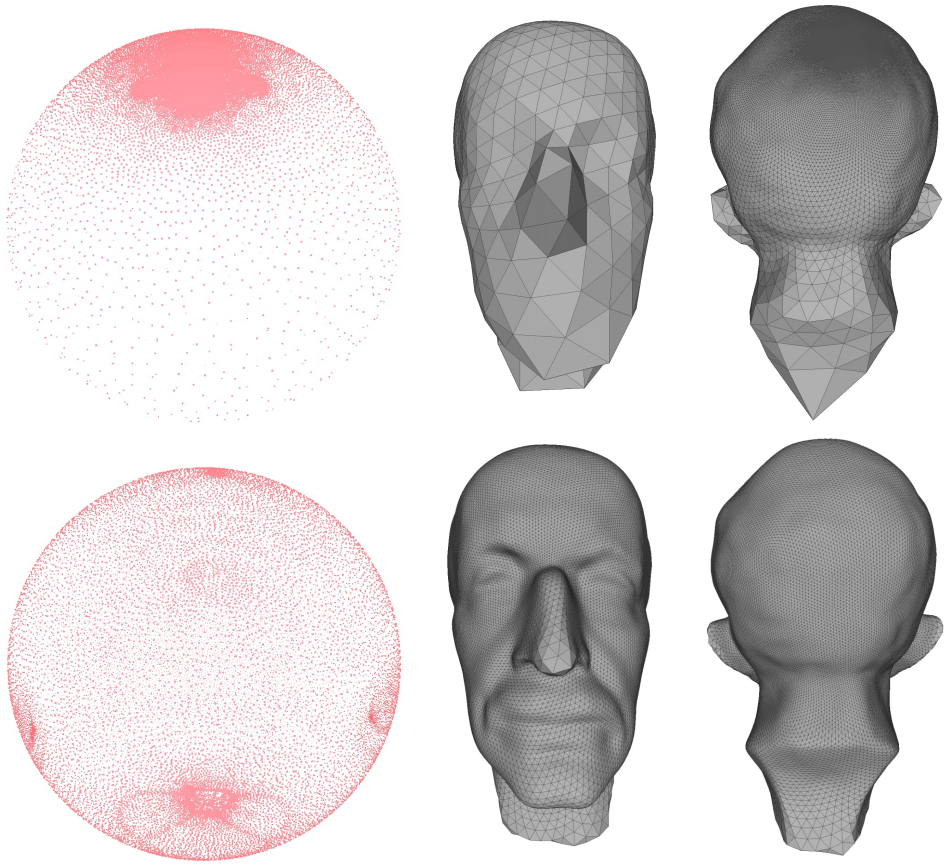

Figure 7.9: The effect of our balancing scheme. Top left: A spherical conformal parameterization without the balancing scheme. Bottom left: A spherical conformal parameterization with the balancing scheme. Middle: The front view of the meshes generated via the parameterizations. Right: The back view.

the spherical conformal parameterization. Then, we can generate a regular triangulation on $\mathbb{S}^2$ and interpolate it back onto the filtered point cloud via the parameterization. This gives us an admissible meshing result. Figure 7.10 shows our meshing result on a synthetic point cloud with 3% uniformly distributed random noise. It can be observed that the mesh generated is with a highly regular triangulation.

It is also possible to construct a faithful triangulated mesh on the geometrically noisy point cloud in Figure 7.10 without applying any filtering

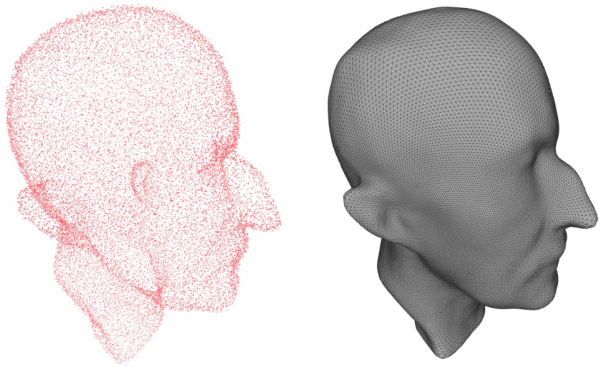

Figure 7.10: Meshing a geometrically noisy point cloud.

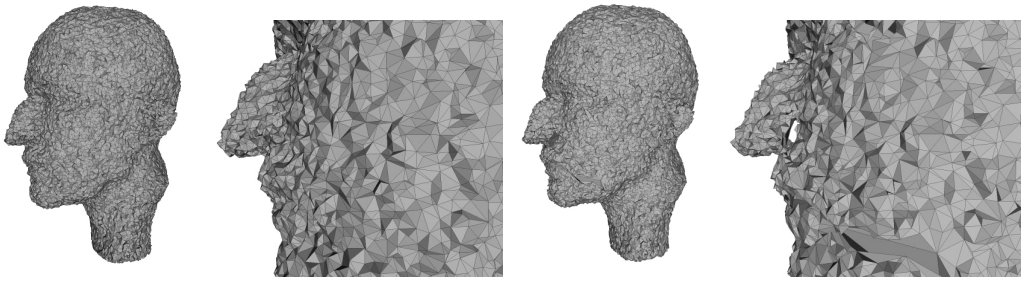

Figure 7.11: Meshing a noisy point cloud without any filtering or subsampling step. Left: Our meshing result with a zoom-in of the nose. Right: The result of the Tight Cocone algorithm [19] with a zoom-in of the nose.

or sampling step. The meshing results of our scheme and the Tight Cocone algorithm [19] are shown in Figure 7.11. It is noteworthy that irregular triangulations and topological holes exist on the meshing result by [19], while our meshing result is regular and topology preserving. This demonstrates the advantage of our meshing scheme over the Tight Cocone algorithm [19].

Besides, point clouds with topological noises are considered. Figure 7.12 shows a point cloud with 1021 randomly created topological holes. Our algorithm effectively produces a high quality genus-0 mesh.

Finally, we test our meshing scheme on real point cloud data adopted from the RGB-D Scenes Dataset v.2 [48]. Figure 7.13 shows a soda can point cloud and a cereal box point cloud, together with our meshing results. The

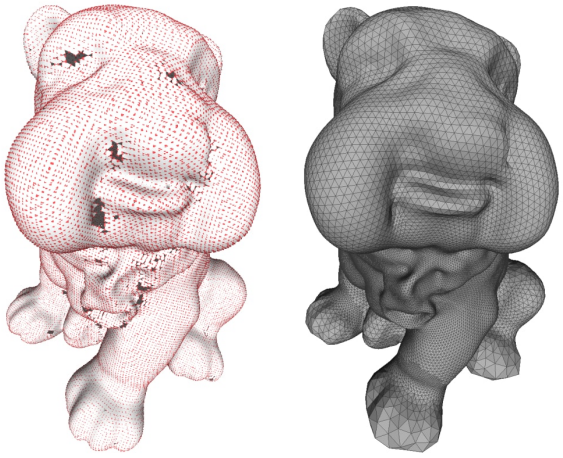

Figure 7.12: Meshing a topologically noisy point cloud.

above experiments demonstrate the effectiveness of our proposed algorithm for meshing genus-0 point clouds.

7.5.2 Multilevel representations of genus-0 point clouds

We can also apply our proposed spherical conformal parameterization algorithm for computing multilevel representations of a genus-0 point cloud. The procedure is outlined below.

Given a genus-0 point cloud, we first compute its spherical conformal parameterization by our proposed algorithm. Next, we create a coarse spherical point cloud. Then, the vertices of the coarse point cloud can be interpolated onto the genus-0 point cloud with the aid of the spherical parameterization. By progressively subdividing the coarse sphere using existing subdivision methods, such as the butterfly subdivision method [21] and the loop subdivision method [56], we can repeat the above steps using the subdivided sphere and obtain multilevel representations of the input point cloud. Figure 7.14 and Figure 7.15 show two examples of multilevel representations generated by our proposed procedure. In our experiments, the loop subdivision method

Figure 7.13: Meshing two real 3D scanned noisy point clouds in the RGB-D Scenes Dataset v.2 [48]. Left: The point clouds. Middle: The meshing results. Right: The meshing results without showing the triangulations.

[56] is applied for generating the subdivisions. The subdivision connectivity of the results can be easily observed. This demonstrates the effectiveness of our algorithm in creating multilevel representations of genus-0 point clouds.

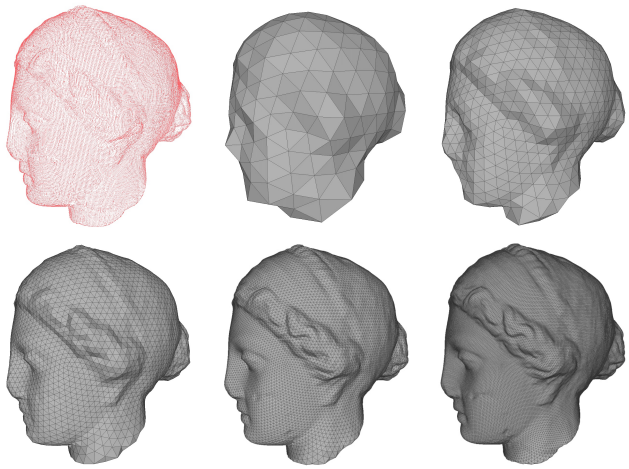

Figure 7.14: Multilevel representations of Igea. Mesh structures are created on the representations for a better visualization. Top left: the Igea point cloud with 134345 points. Top middle to Bottom right: the multilevel representations with 0, 1, 2, 3 and 4 subdivisions. The representations are respectively with 642, 2562, 10242, 40962 and 163842 points.

Figure 7.15: Multilevel representations of lion vase. Mesh structures are created on the representations for a better visualization. Top left: the lion vase point cloud with 256094 points. Top middle to Bottom right: the multilevel representations with 0, 1, 2, 3 and 4 subdivisions. The representations are respectively with 696, 2778, 11106, 44418 and 177666 points.

Chapter 8

Landmark aligned Teichmüller parameterization of disk-type point clouds

8.1 Introduction

In this chapter, we present our proposed algorithm in [5] for computing the landmark aligned Teichmüller parameterization of point clouds with disk topology. To the best of our knowledge, this is the first work on the Teichmüller maps on point clouds.

In the study of point clouds, one challenging topic is the registration between feature-endowed point clouds. An admissible mapping between feature-endowed point clouds should not only fulfill the landmark constraints but also retain the underlying geometric structure as complete as possible. Recall that conformal maps between Riemann surfaces may not exist with the presence of landmark constraints, while landmark-matching Teichmüller maps (T-maps) always exist. Moreover, T-maps are desirable as they ensure uniform conformality distortions on the entire surfaces. This motivates us the use of T-maps in tackling the landmark-matching point cloud registration problem.

In this work, we establish the theory and computation of Teichmüller parameterization on disk-type point clouds. First, we rigorously develop a novel discrete analogue of Teichmüller maps on point clouds. Then, we introduce a hybrid quasi-conformal reconstruction scheme on point clouds with given Beltrami coefficients, and propose an improved approximation scheme for the related differential operators on disk-type point clouds. With the above numerical schemes, we propose an efficient method called *TEMPO*, which stands for *T*eichmüller *E*xtremal *M*appings of *PO*int Clouds, for computing the landmark aligned Teichmüller parameterization of disk-type point cloud. Our proposed algorithm can be applied for point cloud registration and shape analysis. In particular, the Teichmüller parameterizations induce a shape dissimilarity metric, called the *Teichmüller metric*, that can be effectively used in point cloud classification.

8.2 Contributions

The contributions of our work are highlighted below:

1. We develop the concept of PCT-maps, a novel discrete analogue of the continuous Teichmüller maps on point clouds. The relationship between the discrete PCT-maps and the continuous Teichmüller maps is rigorously established.

2. We introduce a new hybrid scheme for reconstructing quasi-conformal maps on point clouds with given Beltrami coefficients.

3. We propose an improved approximation scheme for the differential operators on disk-type point clouds relevant to conformal/qausi-conformal maps.

4. Our proposed *TEMPO* algorithm for point cloud Teichmüller parameterization allows landmark constraints and achieves uniform conformality distortions.

5. Our *TEMPO* algorithm is highly efficient. The computation typically completes within 1 minute for point clouds with around 40K points.

6. The induced Teichmüller metric can be applied as a dissimilarity metric for point cloud classification.

8.3 Some preliminary concepts for quasi-conformal geometry on point clouds

In this section, we introduce some concepts for establishing the theory of quasi-conformal geometry on point clouds. To avoid ambiguity, we use normal characters when introducing the continuous theories and bold characters when introducing the discretization on point clouds.

Definition 8.3.1 (Interior cone condition)**.** *A planar domain $\Omega \subset \mathbb{R}^2$ is said to satisfy* the interior cone condition *with parameter $r > 0$ and $\theta \in (0, \pi/2)$ if for every $x \in \Omega$, there exists a unit vector $d(x)$ such that $C(x, d, \theta, r) \subseteq \Omega$, where*

$$C(x, d, \theta, r) = \{x + ty : y \in \mathbb{S}^1,\, y^T d(x) \geq \cos\theta,\, 0 \leq t \leq r\}. \tag{8.1}$$

Definition 8.3.2 (Quasi-uniform point cloud)**.** *Let P be a point cloud sampled from a planar domain Ω. The* fill distance $h_{P,\Omega}$ *is defined by*

$$h_{P,\Omega} := \sup_{x \in \Omega} \min_{p \in P} \|x - p\|. \tag{8.2}$$

The separation distance q_P *is defined by*

$$q_P := \frac{1}{2} \min_{\substack{p_1, p_2 \in P \\ p_1 \neq p_2}} \|p_1 - p_2\|. \tag{8.3}$$

P is called a quasi-uniform *point cloud with constant $c_{qu} > 0$ if*

$$q_P \leq h_{P,\Omega} \leq c_{qu} q_P. \tag{8.4}$$

With the above definitions, we define the notion of planar point clouds.

Definition 8.3.3 (Planar point cloud)**.** *A* planar point cloud *is a quasi-uniform point cloud $P = \{p_1, p_2, \ldots, p_N\} \subset \Omega$ sampled from a disk-type domain $\Omega \subset \mathbb{R}^2$ which satisfies the interior cone condition.*

Now, let Ω be planar domain satisfying the interior cone condition, and P be a planar point cloud sampled from Ω with fill distance h. Given a function $f : \Omega \to \mathbb{R}$, we apply the Moving Least Square (MLS) method [55, 50, 70, 16, 52, 53] to approximate f locally near every point. Define three mappings $q, q_1, q_2 : \Omega \to \mathbb{R}^6$ respectively by

$$q(x_1, x_2) = [1, x_1, x_2, x_1^2, x_1 x_2, x_2^2]^T, \tag{8.5}$$

$$q_1(x_1, x_2) = \partial_1 q = [0, 1, 0, 2x_1, x_2, 0]^T, \tag{8.6}$$

and

$$q_2(x_1, x_2) = \partial_2 q = [0, 0, 1, 0, x_1, 2x_2]^T. \tag{8.7}$$

Then, the local approximation near x is given by

$$c_x = argmin_c \sum_{i=1}^{N} w(\|x - p_i\|) \left(c^T q(p_i) - f_i\right)^2, \tag{8.8}$$

where w is a weight function compactly supported in $[0, C_\delta h]$ and C_δ is a constant defined in [84]. The solution c_x can also be written as

$$c_x = (Q^T W(x) Q)^{-1} Q^T W(x) F, \tag{8.9}$$

where $W(x)$ is a diagonal matrix with $W_{ii} = w(\|x - p_i\|)$, and Q, F are the matrix forms of q, f respectively. From now on, unless otherwise specified, we

denote the matrix form of a given point cloud mapping by its corresponding capital letter, where the i-th row of the matrix is the function value on p_i. Denote

$$A_x = (Q^T W(x) Q)^{-1} Q^T W(x). \tag{8.10}$$

Then, the value of f at y is approximated by

$$\tilde{f}(y) = q^T(y) A_x F. \tag{8.11}$$

The following estimates for the MLS method have been proved in [68]:

Proposition 8.3.4 (See Corollary 4.10, [68])**.**

$$\|q_j^T(x) A_x\|_1 = O(h^{-1}), \tag{8.12}$$

where h is the fill distance.

Proposition 8.3.5 (See Theorem 4.3, [68])**.**

$$\|\partial_i \partial_j f - (\partial_i \partial_j q^T(y)) A_x f\| = O(h) \tag{8.13}$$

where h is the fill distance.

To establish point cloud quasi-conformal maps, it is necessary to properly define the Beltrami coefficients on point clouds. In this work, we adopt the definition of *discrete diffuse point cloud Beltrami coefficients (PCBC)* in [67]:

Definition 8.3.6 (Discrete diffuse point cloud Beltrami coefficients [67])**.** *Let $P = \{p_i \in \mathbb{R}^2 : i = 1, \cdots, N\}$ sampled from Ω be a planar point cloud, and $\mathbf{f} : P \to \mathbb{R}^2$ be a point cloud mapping. Denote $\mathbf{f} = (\mathbf{u}, \mathbf{v})^T$. The* diffuse point cloud Beltrami coefficient (PCBC) $\tilde{\mu} : \Omega \to \mathbb{C}$ *is defined by*

$$\tilde{\mu}(x) = \frac{\begin{bmatrix} q_1^T(x) & q_2^T(x) \end{bmatrix} \begin{bmatrix} A_x & iA_x \\ iA_x & -A_x \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}}{\begin{bmatrix} q_1^T(x) & q_2^T(x) \end{bmatrix} \begin{bmatrix} A_x & iA_x \\ -iA_x & A_x \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}}, \tag{8.14}$$

where q_1, q_2 and A_x are respectively defined in Equations (8.6), (8.7) and (8.10). In addition, the discrete diffuse PCBC is a complex valued point cloud mapping defined by $\tilde{\boldsymbol{\mu}} = \tilde{\mu}|_P$.

An error estimate of the discrete diffuse PCBC is given below.

Proposition 8.3.7 (See [67], page 9)**.** *Let P be a planar point cloud sampled from Ω with fill distance h, and $\mathbf{f}$ be the point cloud map corresponding to f. Let μ be the Beltrami coefficient associated with f, and $\tilde{\boldsymbol{\mu}}$ be the discrete diffuse PCBC associated with $\mathbf{f}$. Then, there exists a constant $C(f)$ such that if $h \leq C(f)$, the following error bound holds:*

$$|\mu(x) - \tilde{\boldsymbol{\mu}}(x)| \leq C(f)h^2. \tag{8.15}$$

To extend the abovementioned concepts of quasi-conformal point cloud mappings on planar domains to 3D, Meng and Lui [67] introduced the notion of ϵ-conformal parameterization for 3D point clouds, which is analogous to conformal parameterization for 3D surfaces. With this notion, 3D quasi-conformal point cloud mappings can be defined as a composition of ϵ-conformal parameterizations and planar quasi-conformal mappings. For more details, readers are referred to [67].

8.4 Defining Teichmüller mappings on point clouds

In contrast with the continuous Teichmüller theory, the discretization of Teichmüller maps on point clouds has not been developed before. In this section, we establish a novel discretization of the Teichmüller extremal maps on point clouds with theoretical guarantees. We begin with the definition of the discretized T-maps on planar point clouds.

Definition 8.4.1 (Planar point cloud Teichmüller mappings)**.** *Let P be a planar point cloud sampled from Ω. A point cloud mapping $\mathbf{f}$ is called a* planar point cloud Teichmüller map (planar PCT-map) *if its discrete diffuse PCBC $\boldsymbol{\mu}$ has a constant norm $k = |\boldsymbol{\mu}(p)|$ for all $p \in P$ and satisfies*

$$(L_i\boldsymbol{\mu})\overline{\boldsymbol{\mu}(p_i)} \in \mathbb{R} \tag{8.16}$$

for all interior points $p_i \in P$, where L denotes the approximation of the Laplace-Beltrami operator by the MLS method, and L_i denotes the i-th row of L.

We aim to prove that under certain convergence conditions, a planar PCT-map on a planar point cloud converges to a continuous T-map defined on the corresponding underlying domain. We now prove the following lemma.

Lemma 8.4.2. *Let R be a rectangular domain, and $u, v \in C^3(R)$ satisfy $u\Delta v = v\Delta u$ and $u^2 + v^2 = 1$. There exists a real valued function f such that $\phi = f \cdot (u + iv)$ is holomorphic, and $|f| \neq 0$ in R.*

Proof. From the Cauchy-Riemann equation, we have

$$\begin{aligned} f_x u + u_x f &= f_y v + v_y f; \\ f_y u + u_y f &= -f_x v - v_x f. \end{aligned} \tag{8.17}$$

As $u^2 + v^2 = 1$, the above system is equivalent to

$$\begin{aligned} f_x &= G_1 f; \\ f_y &= G_2 f, \end{aligned} \tag{8.18}$$

where

$$\begin{aligned} G_1 &= -uu_x - vv_x - vu_y + uv_y; \\ G_2 &= -uu_y - vv_y - uv_x + vu_x. \end{aligned} \tag{8.19}$$

Suppose $R = [a_1, a_2] \times [b_1, b_2]$. From the first equation above, we have

$$f = \exp\left(\int_{a_1}^{x} G_1(s, y)ds + g(y)\right), \tag{8.20}$$

for some function $g(y)$. Plugging this expression into the second equation, it follows that

$$\int_{a_1}^{x} \partial_y G_1(s, y)ds + g_y(y) = G_2. \tag{8.21}$$

Note that the above equation has a solution if $\partial_y G_1 = \partial_x G_2$. Now, by a direct calculation, we have

$$\partial_y G_1 - \partial_x G_2 = u\Delta v - v\Delta u = 0. \tag{8.22}$$

Hence, the above system has a solution in the form of Equation (8.20), where

$$g = \int_{b_1}^{y} G_2(x, t)dt - \int_{a_1}^{x} (G_1(s, y) - G_1(s, b_1))ds + C. \tag{8.23}$$

Thus, we obtain

$$f = \exp\left(\int_{b_1}^{y} G_2(x, t)dt + \int_{a_1}^{x} G_1(s, b_1)ds + C\right). \tag{8.24}$$

Obviously, $|f| \neq 0$. Furthermore, since $f \cdot (u + iv)$ is continuous and satisfies the Cauchy-Riemann equation, it is holomorphic. □

After proving the above lemma, we establish the consistency between planar PCT-maps and the continuous T-maps.

Proposition 8.4.3. *A sequence of point clouds $\{P_n\}$ is said to be ascending if $P_1 \subset P_2 \subset \cdots P_n \subset P_{n+1} \subset \cdots$. Let $\{P_n\}$ be an ascending sequence of point clouds sampled from a rectangular domain R_1 with fill distance $h_n \to 0$. For all n, let $\mathbf{g}_n$ be a PCT-map defined on P_n with discrete diffuse PCBC $\tilde{\boldsymbol{\mu}}_n$. Assume that there exists a smooth quasi-conformal map $g : R_1 \to R_2$*

with Beltrami coefficient μ such that $\mathbf{g}_n(p)$ converges to $g(p)$ for every point $p \in P := \cup_m P_m$. Let the sup-norm of the error be $\epsilon_n = \|\mathbf{g}_n - g|_{P_n}\|_\infty$ and assume that $\lim_n \epsilon_n h_n^{-2} = 0$. Then g is a continuous Teichmüller map.

Proof. We first show that for all m, $\tilde{\boldsymbol{\mu}}_n(p)$ converges to $\mu(p)$ for all $p \in P_m$.

Let $\tilde{\boldsymbol{\nu}}_n$ be the discrete diffuse PCBC of a point cloud map $g|_{P_n}$. It follows that

$$|\mu(p) - \tilde{\boldsymbol{\mu}}_n(p)| \leq |\mu(p) - \tilde{\boldsymbol{\nu}}_n(p)| + |\tilde{\boldsymbol{\nu}}_n(p) - \tilde{\boldsymbol{\mu}}_n(p)|. \tag{8.25}$$

By Proposition 8.3.7, we have $|\mu(p) - \tilde{\boldsymbol{\nu}}_n(p)| \leq C(g)h_n^2$. Let $\tilde{G}$ and $\tilde{G}_n$ be the vector forms of $g|_{P_n}$ and $\mathbf{g}_n$. Further denote

$$\begin{aligned} D_1 &= \begin{pmatrix} q_1^T(x) & q_2^T(x) \end{pmatrix} \begin{pmatrix} A_x & iA_x \\ iA_x & -A_x \end{pmatrix}, \\ D_2 &= \begin{pmatrix} q_1^T(x) & q_2^T(x) \end{pmatrix} \begin{pmatrix} A_x & iA_x \\ -iA_x & A_x \end{pmatrix}. \end{aligned} \tag{8.26}$$

It follows that

$$\begin{aligned} &|\tilde{\boldsymbol{\nu}}_n(p) - \tilde{\boldsymbol{\mu}}_n(p)| \\ \leq &\left| \frac{D_1(p)\tilde{G}}{D_2(p)\tilde{G}} - \frac{D_1(p)\tilde{G}_n}{D_2(p)\tilde{G}_n} \right| \\ \leq &\frac{\left| D_1(p)\tilde{G}D_2(p)\left(\tilde{G}_n - \tilde{G}\right)\right| + \left| D_2(p)\tilde{G}D_1(p)\left(\tilde{G} - \tilde{G}_n\right)\right|}{\left| D_2(p)\tilde{G} \cdot D_2(p)\tilde{G}_n \right|} \\ \leq &\frac{2\|D_1(p)\|_1 \cdot \|D_2(p)\|_1 \cdot \|\tilde{G}\|_\infty \cdot \|\tilde{G} - \tilde{G}_n\|_\infty}{\left| D_2(p)\tilde{G} \cdot D_2(p)\tilde{G}_n \right|} \\ = &\frac{2O(\epsilon_n h_n^{-2})}{\left| D_2(p)\tilde{G} \cdot D_2(p)\tilde{G}_n \right|} \end{aligned} \tag{8.27}$$

as $\|D_i(p)\|_1 \leq O(h^{-1})$ and $\tilde{G}$ is bounded. Similarly,

$$\begin{aligned} |D_2(p)\tilde{G} - g_z(p)| &= O(h_n^2), \\ |D_2(p)\tilde{G} - D_2(p)\tilde{G}_n| &= O(\epsilon_n h_n^{-1}). \end{aligned} \tag{8.28}$$

Hence, for sufficiently large n,

$$|\mu(p) - \tilde{\boldsymbol{\mu}}_n(p)| \leq O(h_n^2) + |g_z(p)|^{-2} O(\epsilon_n h_n^{-2}) = O(h_n^2 + \epsilon_n h_n^{-2}). \tag{8.29}$$

Therefore, $\tilde{\boldsymbol{\mu}}_n(p)$ converges to $\mu(p)$ for each $p \in P$.

Next, note that $|\tilde{\boldsymbol{\mu}}_n| = k_n$ is a constant for every n. Therefore, $|\mu| = k$ on P and $k = \lim_n k_n$. Also, as $\lim_n h_n = 0$, P is dense in R_1. By the continuity of μ, $|\mu(x)| = k$ for all $x \in R_1$.

Now, we prove that $\mu = k\bar{\varphi}/|\varphi|$ where φ is holomorphic. Assume $k \neq 0$. Fix a point cloud P_n and consider the matrix L, the approximation of the LB operator. By Equation (8.13), for every interior point $p = p_i \in P_n$,

$$\begin{aligned}
&\left|(\Delta\mu(p))\,\overline{\mu(p)} - (L_i\tilde{\boldsymbol{\mu}}_n)\,\overline{\tilde{\boldsymbol{\mu}}_n(p)}\right| \\
\leq &\|\Delta\mu\|_\infty \cdot |\mu(p) - \tilde{\boldsymbol{\mu}}_n(p)| + k_n|\Delta\mu(p) - L_i\tilde{\boldsymbol{\mu}}_n| \\
= &\|\Delta\mu\|_\infty O(h_n^2 + \epsilon_n h_n^{-2}) + k_n O(h_n) \\
= &O(h_n^2 + \epsilon_n h_n^{-2}).
\end{aligned} \tag{8.30}$$

By the definition of PCT-map, we have

$$(L_i\tilde{\boldsymbol{\mu}}_n)\,\overline{\tilde{\boldsymbol{\mu}}_n(p)} \in \mathbb{R} \tag{8.31}$$

for all n. Hence, $(\Delta\mu(p))\,\overline{\mu(p)} \in \mathbb{R}$ if p is an interior point of P_n.

Now, consider an arbitrary point $p \in R_1$. If $p \in P - \partial R_1$, then the above conclusion holds. If $p \in P \cap \partial R_1$ or $p \in R_1 - P$, by the density of P in R_1, and the continuity of μ and the LB operator, we also have the above conclusion. Thus, $(\Delta\mu(p))\,\overline{\mu(p)} \in \mathbb{R}$ for all $p \in R_1$.

Let $\mu = k(\rho + i\tau)$. A direct calculation yields

$$0 = \frac{1}{k^2} Im\left((\Delta\mu)\,\overline{\mu}\right) = \rho\Delta\tau - \tau\Delta\rho. \tag{8.32}$$

By taking $u = \rho$, and $v = -\tau$ in Lemma 8.4.2, it follows that there exists a real-valued function f such that $\varphi = f \cdot (\rho - i\tau)$ is holomorphic. Also, we have $\mu = k\bar{\varphi}/|\varphi|$. We conclude that g is a Teichmüller map. □

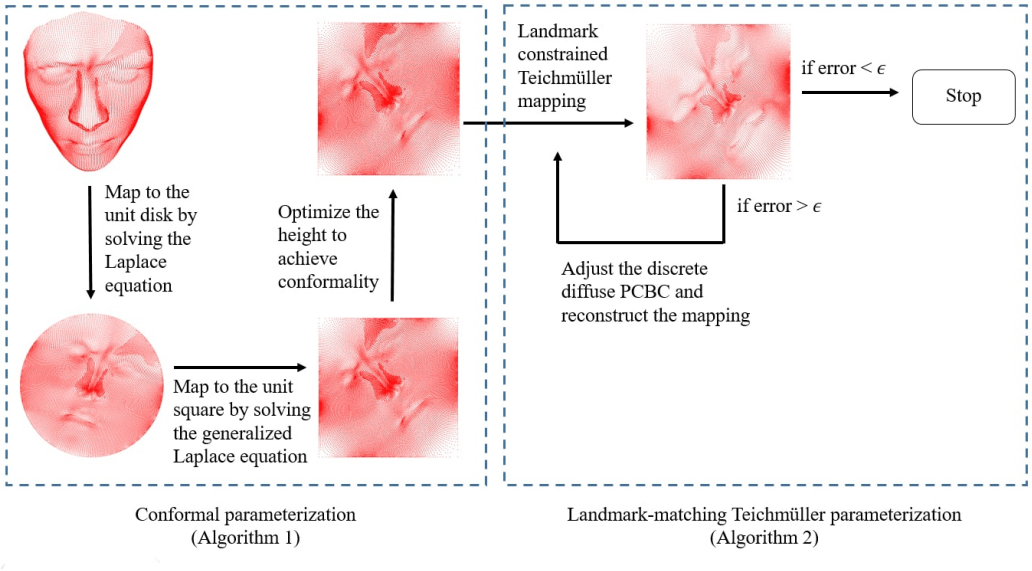

Figure 8.1: The pipeline of our *TEMPO* algorithm.

After establishing the consistency between planar PCT-maps and the continuous T-maps on planar domains, we aim to define T-maps between disk-type point cloud surfaces in 3D.

The generalization can be achieved as follows. Let $\mathbf{f}$ be a point cloud map between two 3D disk-type point clouds P_1 and P_2, and $\boldsymbol{\phi}_i$ be the ϵ-conformal parameterization of P_i. We can define $\mathbf{f}$ to be a *point cloud Teichmüller map (PCT-map)* between P_1 and P_2 if $\boldsymbol{\phi}_2^{-1} \circ \mathbf{f} \circ \boldsymbol{\phi}_1$ is a planar PCT-map.

With this generalization, a property analogous to Proposition 8.4.3 can be easily derived for the PCT-maps between two 3D disk-type point clouds. The details are omitted here.

8.5 Our proposed *TEMPO* method for landmark aligned Teichmüller parameterization

In this section, we propose an efficient algorithm called *TEMPO* for computing landmark aligned Teichmüller parameterizations of disk-type point clouds. Figure 8.1 shows the pipeline of our *TEMPO* algorithm. In each step, certain approximations of the differential operators are necessary. In the following subsections, we first introduce our numerical schemes, and then explain our proposed algorithm in details.

The following three subsections will be covered:

1. A hybrid numerical scheme for computing quasi-conformal maps on disk-type point clouds with a given PCBC.

2. An improved approximation of the related differential operators on point clouds with disk topology.

3. The details of the *TEMPO* algorithm.

8.5.1 A hybrid scheme for computing quasi-conformal mappings on disk-type point clouds with a given PCBC

To compute a quasi-conformal map with a given feasible Beltrami coefficient $\mu = \rho + i\tau$ on triangulated meshes, one method is to solve the linear system (3.11). Another method is the Linear Beltrami Solver (LBS) [61], which aims to solve the generalized Laplace equation (3.12).

However, neither of the above methods work well on point clouds. Specifically, the solution to the system (3.11) is highly sensitive to the input PCBC

approximated on point clouds. If the PCBC is not sufficiently close to a feasible Beltrami coefficient, the system (3.11) does not lead to a quasi-conformal diffeomorphism. Also, the system (3.12) does not result in very accurate solutions. That is, given an input PCBC, the solutions to the system (3.12) may be with a PCBC which is largely different from the input one. Consequently, we cannot directly apply any of the two methods on point clouds. To overcome this hindrance, we propose a hybrid scheme for computing quasi-conformal maps on point clouds by a fusion of the two mentioned methods.

More specifically, we consider solving the following hybrid PDE:

$$\begin{pmatrix} \alpha_1\partial_x + \alpha_2\partial_y & -\partial_y \\ \alpha_2\partial_x + \alpha_3\partial_y & \partial_x \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \gamma \begin{pmatrix} \mathcal{L} & O \\ O & \mathcal{L} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0, \tag{8.33}$$

where $\mathcal{L}$ is the generalized Laplace-Beltrami operator, and γ is a user-defined positive parameter balancing the two methods.

Now, to solve Equation (8.33) on point clouds, we discretize the system (3.11) and denote the discretization by

$$M_1(\boldsymbol{\mu}) \begin{pmatrix} U \\ V \end{pmatrix} = 0. \tag{8.34}$$

Similarly, the system (3.12) is discretized and the discretization is denoted by

$$M_2(\boldsymbol{\mu}) \begin{pmatrix} U \\ V \end{pmatrix} = 0, \tag{8.35}$$

where

$$M_2(\boldsymbol{\mu}) = \begin{pmatrix} M_3(\boldsymbol{\mu}) & O \\ O & M_3(\boldsymbol{\mu}) \end{pmatrix}, \tag{8.36}$$

and $M_3(\boldsymbol{\mu})$ is the approximation of the generalized Laplace-Beltrami operator. Here the discretizations in Equation (8.34) and Equation (8.35) will be explained in Section 8.5.2.

Finally, to compute a quasi-conformal map with a given PCBC $\boldsymbol{\mu}$, we propose to solve the following hybrid system

$$(M_1(\boldsymbol{\mu}) + \gamma M_2(\boldsymbol{\mu})) \begin{pmatrix} U \\ V \end{pmatrix} = 0. \tag{8.37}$$

Since both of the system (3.11) and the system (3.12) are derived from the Beltrami equation (3.7), the $f = (u, v)^T$ in the above hybrid system is theoretically guaranteed to be a solution to the hybrid PDE (8.33).

Moreover, our proposed scheme overcomes the drawbacks of the two approaches. On one hand, the system (8.34) produces accurate results but is unstable. By including the second term M_2 in our hybrid system (8.37), we significantly stabilize the computation. On the other hand, note that the system (8.35) is stable but not accurate, and the first term M_1 in our hybrid system (8.37) increases the accuracy of the computation. With a fusion of the two approaches, we can improve both the accuracy and the stability of the computation of quasi-conformal maps on point clouds.

8.5.2 Discretizations in Equation (8.34) and (8.35)

In this part, we present the numerical schemes for the discretizations in Equation (8.34) and Equation (8.35) on disk-type point clouds.

For the discretization in Equation (8.34), we use a numerical scheme which involves the MLS method. For the discretization in Equation (8.35), we propose a combined scheme which involves both the MLS method and the local mesh method [46]. For the weight function in the MLS method, We adopt the Gaussian weight function in [4]:

$$w(d) = \begin{cases} 1 & \text{if } d = 0 \\ \frac{1}{K}\exp(\frac{-\sqrt{K}d^2}{D^2}) & \text{if } d \neq 0 \end{cases}, \tag{8.38}$$

where K, D are respectively the number of points in the chosen neighborhood and the maximal distance of the neighborhood.

Approximating Equation (8.34) on planar point clouds

We first present the numerical scheme for Equation (8.34) on a point cloud P.

For every $p \in P$, the MLS method constructs an approximating function f_p near p. To satisfy the system (8.34) on p, a direct calculation yields

$$\begin{pmatrix} q_2^T(p)A_p \\ -q_1^T(p)A_p \end{pmatrix} V = A(p) \begin{pmatrix} q_1^T(p)A_p \\ q_2^T(p)A_p \end{pmatrix} U. \tag{8.39}$$

The above system can be rewritten as a linear system $M_1(\boldsymbol{\mu})[U^T, V^T]^T = 0$, where $M_1(\boldsymbol{\mu})$ is a $2N \times 2N$ matrix, with N being the number of points in P. Specifically, for all $i = 1, 2, \cdots, N$, we have

$$\begin{pmatrix} (M_1(\boldsymbol{\mu}))_i \\ (M_1(\boldsymbol{\mu}))_{i+N} \end{pmatrix} = \begin{pmatrix} \alpha_1(p_i)q_1^T(p_i) + \alpha_2(p_i)q_2^T(p_i) & -q_2^T(p_i) \\ \alpha_2(p_i)q_1^T(p_i) + \alpha_3(p_i)q_2^T(p_i) & q_1^T(p_i) \end{pmatrix} \begin{pmatrix} A_{p_i} & O \\ O & A_{p_i} \end{pmatrix}, \tag{8.40}$$

where E_i denotes the i-th row of any matrix E. Now, we consider the following lemma.

Lemma 8.5.1. *Let P be a planar point cloud. Suppose $\mathbf{f} = (\mathbf{u}, \mathbf{v})^T$ is a point cloud map defined on P with well-defined discrete diffuse PCBC. Let $\boldsymbol{\sigma}$ be a complex-valued point cloud function. Then, $\boldsymbol{\sigma}$ is the discrete diffuse PCBC of $\boldsymbol{f}$ if and only if $M_1(\boldsymbol{\sigma})[U^T, V^T]^T = 0$.*

Proof. First, note that $\mathbf{f}$ satisfies $M_1(\boldsymbol{\sigma})[U^T, V^T]^T = 0$ if and only if the Equation (8.39) holds for all $p \in P$, which is equivalent to that

$$(q_1(p) + iq_2(p))^T A_p(U + iV) = \boldsymbol{\sigma}(p)(q_1(p) - iq_2(p))^T A_p(U + iV) \tag{8.41}$$

holds for all $p \in P$. Since the discrete diffuse PCBC is well-defined for all $p \in P$, it is straightforward to show that the above equation holds if and only if $\boldsymbol{\sigma}(p)$ is the diffuse PCBC of $\mathbf{f}$ on p. The result follows. □

The above lemma shows that given an admissible Beltrami coefficient $\boldsymbol{\mu}$, the solution to the linear system $M_1(\boldsymbol{\mu})[U^T, V^T]^T = 0$ is associated with a diffuse PCBC close to $\boldsymbol{\mu}$. Therefore, the accuracy of the computation of quasi-conformal maps on point clouds is ensured.

Improving the approximation of Equation (8.35)

Next, we introduce our proposed approximation scheme for the discretization in Equation (8.35) on disk-type point clouds.

The existing MLS approximation schemes work well for interior points on disk-type point clouds. However, unlike the approximations of the LB operator on point clouds and the generalized LB operator on meshes, the Beltrami coefficients for point cloud maps are not locally constant. Therefore, the matrix A in Equation (3.12) is not constant. Therefore, it is necessary to take its derivatives into considerations. We have

$$\begin{aligned}
&\nabla \cdot \left(A \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right) \\
=&\partial_x(\alpha_1 u_x + \alpha_2 u_y) + \partial_y(\alpha_2 u_x + \alpha_3 u_y) \\
=&(\partial_x \alpha_1 + \partial_y \alpha_2) u_x + (\partial_x \alpha_2 + \partial_y \alpha_3) u_y + \alpha_1 u_{xx} + 2\alpha_2 u_{xy} + \alpha_3 u_{yy}.
\end{aligned} \tag{8.42}$$

By applying the MLS method on the derivatives of α_i and u, we can obtain the approximation linear system. It follows that if the point $p = p_i$ is an interior point, then the i-th row of the matrix $M_3(\boldsymbol{\mu})$ is

$$\begin{aligned}
(M_3(\boldsymbol{\mu}))_i = &[\alpha_1(P)]^T B_{11}(p) + [\alpha_2(P)]^T (B_{12}(p) + B_{21}(p)) + [\alpha_3(P)]^T B_{22}(p) \\
&+ 2\alpha_1(p)(A_p)_4 + 2\alpha_2(p)(A_p)_5 + 2\alpha_3(p)(A_p)_6
\end{aligned} \tag{8.43}$$

where

$$B_{jl}(p) = A_p^T q_j(p) q_l^T(p) A_p. \tag{8.44}$$

Here, $[\alpha_j(P)]$ denotes a vector whose l-th element is $\alpha_j(p_l)$, and E_j denotes the j-th row of any matrix E.

However, for the boundary points $p \in \partial\Omega$, the above approach does not work well as the second order MLS method requires more points for getting an accurate result. To overcome this problem, we apply the local mesh method [46] on the boundary. For each boundary point $p_i \in \partial\Omega$, a 1-ring structure $\mathcal{T}_i$

is constructed by making use of the Delaunay triangulation. If a point p_j is not included in $\mathcal{T}_i$, then $(M_3(\boldsymbol{\mu}))_{ij}$ is zero. Otherwise, the entry is given by

$$(M_3(\boldsymbol{\mu}))_{ij} = \begin{cases} \displaystyle\sum_{T=\{i,j,k\}\in\mathcal{T}_i} \frac{1}{|T|}(p_k - p_i)^T A(T)(p_k - p_j) & \text{if } j \neq i \\ \displaystyle\sum_{T=\{i,l,k\}\in\mathcal{T}_i} \frac{1}{|T|}(p_k - p_l)^T A(T)(p_k - p_l) & \text{if } j = i \end{cases} \tag{8.45}$$

where

$$A(\{i,j,k\}) = \frac{1}{3}(A(p_i) + A(p_j) + A(p_k)). \tag{8.46}$$

Our proposed combined scheme effectively improves the accuracy of the approximation of the generalized Laplace-Beltrami operator in Equation (8.35) on disk-type point clouds. Numerous experiments are reported in Section 8.6.

8.5.3 Teichmüller parameterization of disk-type point clouds with landmark constraints

Let $\mathcal{M}$ be a simply-connected open surface, and P be a point cloud sampled from $\mathcal{M}$. Our *TEMPO* algorithm aims to compute a landmark aligned parameterization $\varphi : P \to R \subset \mathbb{R}^2$ with suitable boundary conditions, such that φ approximates a Teichmüller extremal mapping (T-map) from $\mathcal{M}$ to R. Here, R is a rectangle. In this section, we explain the algorithmic details of our *TEMPO* algorithm.

Our algorithm mainly consists of two major stages:

1. Conformally parameterizing a disk-type point cloud P onto a rectangular planar domain $\tilde{R}$, with the following sub-steps:

   (a) Mapping P onto the unit disk D by a harmonic map,

   (b) Mapping the unit disk D onto the unit square R by solving the generalized Laplace equation, and

(c) Optimizing the height of R and obtain $\tilde{R}$.

2. Computing a landmark constrained Teichmüller parameterization on $\tilde{R}$.

Mapping the point cloud onto the unit disk by a harmonic map

In the continuous case, we consider the parameterization of $\mathcal{M}$. As an initialization, the first sub-step of our algorithm is to compute the harmonic map $\phi_0 : \mathcal{M} \to D$ that satisfies

$$\Delta\phi_0 = 0 \tag{8.47}$$

with arc-length parameterization boundary constraints.

In the discrete case, we first recall that given an arbitrary diffeomorphism $\varphi : \mathcal{M} \to \varphi(\mathcal{M}) \subset \mathbb{C}$, solving for a harmonic map $\phi_0 : \mathcal{M} \to D$ is equivalent to solving the generalized Laplace equation from $\varphi(\mathcal{M})$ to D with input Beltrami coefficient $\mu(\varphi^{-1})$. Therefore, we do not need to approximate the above harmonic equation. We further note that a diffeomorphism from $\mathcal{M}$ to a planar domain is not easy to solve. Thus, we locally solve the generalized Laplace equation at each point p, and set φ near p to be the projection from the neighborhood of p to its fitting plane obtained by Principal Component Analysis (PCA). This results in one linear equation at each point. By combining all points together, we get a linear system with solution $\boldsymbol{\phi}_0 : P \to D$ being an approximation of the desired harmonic map for point clouds.

Mapping the unit disk onto the unit square by solving the generalized Laplace equation

Recall that our goal in the first major stage is to conformally parameterize the input point cloud onto a rectangular domain $\tilde{R}$, with four specified boundary points mapped to the four corners of $\tilde{R}$. Without loss of generality, we assume

that $\tilde{R}$ is with height h and width 1. It is important to find a suitable h such that such conformal parameterization exists. To find such h, our strategy is to find a quasi-conformal map from D to the unit square R, and then optimally adjust the height of R to get the optimal h. The following proposition justifies our strategy.

Proposition 8.5.2. *Let $f = u + iv : D \to \tilde{R}$ be a quasi-conformal map associated with the Beltrami coefficient μ, and $\tilde{R}$ has height h and width 1. Suppose that $f_1 = u_1 + iv_1$ is a quasi-conformal map from D to R. If u_1 and v_1 satisfy the generalized Laplace equation (3.12) with the coefficient matrix A computed using μ, and with the corresponding boundary condition, then $f = u_1 + ihv_1$.*

Proof. Let g be the metric tensor on D induced by μ. Then $f : (D, g) \to R$ is a conformal map. Therefore, u, v are solutions to the Laplace equation defined on (D, g). That is,

$$\frac{1}{\det(g)}\partial_i(\sqrt{\det(g)}g^{ij}\partial_j u) = 0, \tag{8.48}$$

$$\frac{1}{\det(g)}\partial_i(\sqrt{\det(g)}g^{ij}\partial_j v) = 0. \tag{8.49}$$

Also, as the LB operator is linear, u and v/h are solutions to the Laplace equation with respect to the metric tensor. This implies that u and v/h are solutions to the following PDE

$$\nabla \cdot (\sqrt{\det(g)}(g^{ij})\nabla x) = 0. \tag{8.50}$$

By a straightforward calculation, the coefficient matrix A in the generalized Laplace equation (3.12) with μ satisfies

$$A = \sqrt{\det(g)}(g^{ij}). \tag{8.51}$$

It follows that

$$A = \frac{1}{1-|\mu|^2}\begin{pmatrix} 1+|\mu|^2-2Re(\mu) & -2Im(\mu) \\ -2Im(\mu) & 1+|\mu|^2+2Re(\mu) \end{pmatrix}. \tag{8.52}$$

Thus, u and v/h are solutions of the generalized Laplace equation with Beltrami coefficient μ from D to R, with specific boundary conditions. Since the generalized Laplace equation is an elliptic PDE whose solution is unique with the given boundary condition, it follows that $u = u_1$ and $v/h = v_1$. □

By the above proposition, the final result $\phi = (u_1, hv_1)^T \circ \phi_0$ can be obtained by finding a quasi-conformal map $\phi_1 = (u_1, v_1)^T : D \to R$, and scaling its height by a scalar h.

In the second sub-step of our algorithm, we compute the abovementioned quasi-conformal map ϕ_1. In the continuous case, by Theorem 3.3.3, we can obtain $\phi_1 : D \to R$ that improves the conformality of φ_0 by solving the generalized Laplace equation (3.12) with Beltrami coefficient $\mu(\phi_0^{-1})$.

In the discrete case, to approximate $\mu(\phi_0^{-1})$, the diffuse PCBC in Definition 8.3.6 is adopted. With a given PCBC, the desired quasi-conformal map $\boldsymbol{\phi}_1 = (\mathbf{u}_1, \mathbf{v}_1)^T$ from P to R can be solved by the generalized Laplace equation (8.35).

Optimizing the height of the unit square to achieve conformality

In the third sub-step, we determine the scalar h such that $\phi_2 = (u_1, hv_1)^T$ is a quasi-conformal map with Beltrami coefficient equals $\mu(\phi_0^{-1})$. To solve this problem, it is natural to consider the following energy minimization problem

$$h = argmin \int_{\mathbb{D}} |\mu(\phi_2) - \mu(\phi_0^{-1})|^2. \tag{8.53}$$

Note that the energy is zero when h is optimal. After obtaining the optimal h, we have the desired conformal parameterization $\phi : \mathcal{M} \to \tilde{R}$, where

$$\phi = \phi_2 \circ \phi_0. \tag{8.54}$$

In the discrete case, as the point clouds are assumed to be quasi-uniform, we can simplify the computation of the optimal height h by replacing the integral in Equation (8.53) by a summation. It suffices to solve the minimization problem

$$h = argmin \ \ \|\tilde{\boldsymbol{\sigma}} - \tilde{\boldsymbol{\mu}}\|^2, \tag{8.55}$$

where $\tilde{\boldsymbol{\sigma}}$ and $\tilde{\boldsymbol{\mu}}$ are respectively the discrete diffuse PCBCs of $\boldsymbol{\phi}_2 = (\mathbf{u}_1, h\mathbf{v}_1)^T$ and $\boldsymbol{\phi}_0^{-1}$. The computation for h then becomes straightforward.

Our first major stage is summarized in Algorithm 10.

Algorithm 10: Conformal parameterization of disk-type point clouds

Input: A disk-type point cloud P, four boundary points $p_i, i = 1, 2, 3, 4$
Output: The conformal parameterization $\boldsymbol{\phi}$ of P to a rectangle $\tilde{R}$

1 Find a harmonic map $\boldsymbol{\phi}_0 : P \to D$ with arc-length parameterization boundary constraints;
2 Compute the discrete diffuse PCBC $\tilde{\boldsymbol{\mu}}$ of $\boldsymbol{\phi}_0^{-1}$;
3 Solve the generalized Laplace equation (8.35) with the associated PCBC $\tilde{\boldsymbol{\mu}}$, and obtain a map $\boldsymbol{\phi}_1 = (\mathbf{u}_1, \mathbf{v}_1)^T$ from D to the unit square R, with four points p_1, p_2, p_3, p_4 mapped to the corresponding four corners on R;
4 Adjust the height of R to be h by minimizing $\|\tilde{\boldsymbol{\sigma}} - \tilde{\boldsymbol{\mu}}\|^2$, where $\tilde{\boldsymbol{\sigma}}$ is the discrete diffuse PCBC of $(\mathbf{u}_1, h\mathbf{v}_1)^T$;
5 Obtain the conformal parameterization $\boldsymbol{\phi} = (\mathbf{u}_1, h\mathbf{v}_1)^T \circ \boldsymbol{\phi}_0$ from P to a rectangle $\tilde{R}$ with height h and width 1.

Teichmüller parameterization of disk-like point clouds with landmark constraints

In the second major stage, after computing the conformal parameterization of P onto a rectangle $\tilde{R}$, we now develop an algorithm for computing landmark aligned Teichmüller parameterization on $\tilde{R}$. In the following, we invent an iterative algorithm for PCT-maps on disk-type point clouds and study the convergence properties.

Recall that T-maps are associated with Beltrami coefficients which have constant norms and satisfy a specific form for the argument part. Our

proposed algorithm aims to achieve PCT-maps by iteratively manipulating the PCBC. First, we initialize the map to be the identity map. Then, in each iteration, we perform four sub-steps as explained below on the associated discrete diffuse PCBC of the map, and update the map using the modified PCBC.

In the first sub-step of each iteration, we compute the PCBC $\tilde{\boldsymbol{\mu}}_n$ of a given point cloud map $\mathbf{f}_n$.

In the second step, we manipulate the norm of the PCBC so that the new PCBC gets closer to the Beltrami coefficient of a T-map. Specifically, a suitable constant k_n is chosen to be the norm of the new PCBC:

$$k_n = \begin{cases} \frac{\sum_{p\in S_n} |\tilde{\boldsymbol{\mu}}_n(p)|}{|S_n|} & \text{if } |S_n| > 0 \\ 0 & \text{if } |S_n| = 0, \end{cases} \tag{8.56}$$

where

$$S_n = \{p \in P : |\tilde{\boldsymbol{\mu}}_n(p)| < 1\}. \tag{8.57}$$

This sub-step guarantees that the new norm is feasible.

After updating the norm, we update the argument part of the PCBC in the third sub-step. Define $\boldsymbol{\nu}_n$ by

$$\boldsymbol{\nu}_n(p_i) = \begin{cases} \dfrac{\tilde{\boldsymbol{\mu}}_n(p_i)}{|\tilde{\boldsymbol{\mu}}_n(p_i)|} & \text{if } \tilde{\boldsymbol{\mu}}_n(p_i) \neq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{8.58}$$

Then, we apply the weighted Laplacian smooth operator and the normalization operator on $\boldsymbol{\nu}_n$. More explicitly, we have

$$\boldsymbol{\tau}_n(p_i) = \begin{cases} \dfrac{L_i\boldsymbol{\nu}_n}{|L_i\boldsymbol{\nu}_n|} & \text{if } L_i\boldsymbol{\nu}_n \neq 0, \\ \boldsymbol{\nu}_n(p_i) & \text{otherwise,} \end{cases} \tag{8.59}$$

where L is the approximation of the LB operator on point clouds as described in Section 8.5.2, and $L_i\boldsymbol{\nu}_n$ denotes $(L\boldsymbol{\nu}_n)(p_i)$. Using the new norm k_n and

the new argument part $\boldsymbol{\tau}_n$, we construct a new complex-valued point cloud function

$$\boldsymbol{\sigma}_n = k_n \boldsymbol{\tau}_n. \tag{8.60}$$

In the last sub-step of each iteration, a new landmark-constrained map $\mathbf{f}_{n+1}$ is computed by solving the hybrid equation (8.37) with the PCBC $\boldsymbol{\sigma}_n$. It is noteworthy that as the landmark constraints are added and $\boldsymbol{\sigma}_n$ may not be a feasible PCBC, $\mathbf{f}_{n+1}$ may not be associated with $\boldsymbol{\sigma}_n$ perfectly. In this case, this sub-step can be regarded as a projection of $\boldsymbol{\sigma}_n$ onto the space of all feasible PCBCs.

By repeating the abovementioned sub-steps, the map converges to the landmark aligned Teichmüller parameterization as both the landmark constraints and the constant norm are achieved. Our second major stage is summarized in Algorithm 11.

To justify our proposed algorithm, we prove that the limit function obtained by the algorithm is indeed a PCT-map.

Proposition 8.5.3. *Let P be a planar point cloud. Assume that the sequence $\mathbf{f}_n$ in Algorithm 11 converges to a point cloud map $\mathbf{f}$, and its discrete diffuse PCBC $\tilde{\boldsymbol{\mu}}$ exists. Further assume that $\lim_n \gamma_n = 0$, where γ_n denotes the parameter in Equation (8.37) in the n-th iteration. Then, $\mathbf{f}$ is a PCT-map. Furthermore, both $\tilde{\boldsymbol{\mu}}_n$ and $\boldsymbol{\sigma}_n$ converge to $\tilde{\boldsymbol{\mu}}$.*

Proof. We first show that $\tilde{\boldsymbol{\mu}}_n \to \tilde{\boldsymbol{\mu}}$. Let D_1, D_2 be as defined in the proof of Proposition 8.4.3. By definition, for all $x \in P$,

$$\begin{aligned} &|\tilde{\boldsymbol{\mu}}_n(x) - \tilde{\boldsymbol{\mu}}(x)| \\ &= \left| \frac{D_1\left[\begin{pmatrix} U_n \\ V_n \end{pmatrix} - \begin{pmatrix} U \\ V \end{pmatrix}\right] + D_1\begin{pmatrix} U \\ V \end{pmatrix}}{D_2\left[\begin{pmatrix} U_n \\ V_n \end{pmatrix} - \begin{pmatrix} U \\ V \end{pmatrix}\right] + D_2\begin{pmatrix} U \\ V \end{pmatrix}} - \frac{D_1\begin{pmatrix} U \\ V \end{pmatrix}}{D_2\begin{pmatrix} U \\ V \end{pmatrix}} \right|. \end{aligned} \tag{8.61}$$

Algorithm 11: Landmark-matching Teichmüller parameterization of planar point clouds

Input: A planar point cloud P, target rectangle R and the landmark constraints

Output: The landmark-matching Teichmüller parameterization $\mathbf{f} : P \to R$

Initialize $\mathbf{f}_0$ to be identity map;
while $\|F_n - F_{n-1}\|_2 \geq \epsilon$ **do**
  Calculate the discrete diffuse PCBC $\tilde{\boldsymbol{\mu}}_n$ of $\mathbf{f}_n$;
  Compute $k_n = \begin{cases} \frac{\sum_{p\in S_n} |\tilde{\boldsymbol{\mu}}_n(p)|}{|S_n|} & \text{if } |S_n| > 0 \\ 0 & \text{if } |S_n| = 0, \end{cases}$, where $S_n = \{p \in P : |\tilde{\boldsymbol{\mu}}_n(p)| < 1\}$;
  Calculate the argument part of $\tilde{\boldsymbol{\mu}}_n$ and denote it as $\boldsymbol{\nu}_n$;
  Construct a complex valued point cloud function $\boldsymbol{\tau}_n$ by

$$\boldsymbol{\tau}_n(p_i) = \begin{cases} \dfrac{L_i\boldsymbol{\nu}_n}{|L_i\boldsymbol{\nu}_n|} & \text{if } L_i\boldsymbol{\nu}_n \neq 0, \\ \boldsymbol{\nu}_n(p_i) & \text{otherwise} \end{cases}$$

  where $L = M_2(0)$ is the approximation of the LB operator in Section 8.5.2;
  Construct a new complex valued point cloud function $\boldsymbol{\sigma}_n = k_n\boldsymbol{\tau}_n$;
  Compute the point cloud map $\mathbf{f}_{n+1}$ by solving the hybrid equation (8.37) with a given parameter $\gamma_n \in [0, \infty)$, the input BC $\boldsymbol{\sigma}_n$ and the prescribed landmark constraints;
  Denote the matrix form of $\mathbf{f}_{n+1}$ by F_{n+1};
  Update n by $n + 1$;
Obtain the final map $\mathbf{f} = \mathbf{f}_n$.

By the MLS error estimate in Equation (8.12), for $j = 1, 2$,

$$|q_j^T(x)A_x(U_n - U)| \leq \|U_n - U\|_\infty \|q_j^T(x)A_x\|_1 \leq \|U_n - U\|_\infty O(h^{-1}), \quad (8.62)$$

where h is the fill distance of P and hence constant. Since $\mathbf{f}_n \to \mathbf{f}$ and both of them are defined on a finite set P, we have $\|U_n - U\|_\infty \to 0$. It follows that $|q_j^T(x)A_x(U_n - U)| \to 0$. A similar result holds for V. Hence, for $j = 1, 2$,

$$\lim_n \left| D_j \left[\begin{pmatrix} U_n \\ V_n \end{pmatrix} - \begin{pmatrix} U \\ V \end{pmatrix} \right] \right| = 0. \quad (8.63)$$

Therefore, $\tilde{\boldsymbol{\mu}}_n \to \tilde{\boldsymbol{\mu}}$.

Next, we show that $\boldsymbol{\sigma}_n \to \tilde{\boldsymbol{\mu}}$. Let

$$E_n = -\gamma_n M_2(\boldsymbol{\sigma}_n) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix}. \tag{8.64}$$

Then

$$M_1(\boldsymbol{\sigma}_n) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} = E_n. \tag{8.65}$$

A straightforward calculation yields

$$\begin{aligned}
&D_1(x) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} \\
=&q_1^T(x)A_xU_{n+1} + iq_2^T(x)A_xU_{n+1} + iq_1^T(x)A_xV_{n+1} - q_2^T(x)A_xV_{n+1} \\
=&q_1^T(x)A_xU_{n+1} + iq_2^T(x)A_xU_{n+1} - i(\alpha_2(\boldsymbol{\sigma}_n|_x)q_1^T(x) + \alpha_3(\boldsymbol{\sigma}_n|_x)q_2^T(x))A_xU_{n+1} \\
&- (\alpha_1(\boldsymbol{\sigma}_n|_x)q_1^T(x) + \alpha_2(\boldsymbol{\sigma}_n|_x)q_2^T(x))A_xU_{n+1} + i(E_n)_{j+N} + (E_n)_j \\
=&\left(2\frac{\boldsymbol{\sigma}_n(x) - |\boldsymbol{\sigma}_n(x)|^2}{1-|\boldsymbol{\sigma}_n(x)|^2}q_1(x) - 2i\frac{\boldsymbol{\sigma}_n(x) + |\boldsymbol{\sigma}_n(x)|^2}{1-|\boldsymbol{\sigma}_n(x)|^2}q_2(x)\right)^T A_xU_{n+1} \\
&+ i(E_n)_{j+N} + (E_n)_j,
\end{aligned} \tag{8.66}$$

for any $x = p_j \in P$. Similarly, we have

$$\begin{aligned}
&D_2(x) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} \\
=&\left(2\frac{1-\overline{\boldsymbol{\sigma}_n(x)}}{1-|\boldsymbol{\sigma}_n(x)|^2}q_1(x) - 2i\frac{1+\overline{\boldsymbol{\sigma}_n(x)}}{1-|\boldsymbol{\sigma}_n(x)|^2}q_2(x)\right)^T A_xU_{n+1} + i(E_n)_{j+N} - (E_n)_j.
\end{aligned} \tag{8.67}$$

Then,

$$\boldsymbol{\sigma}_n(x) = \frac{D_1(x) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} - i(E_n)_{j+N} - (E_n)_j}{D_2(x) \begin{pmatrix} U_{n+1} \\ V_{n+1} \end{pmatrix} - i(E_n)_{j+N} + (E_n)_j}. \tag{8.68}$$

Since P is fixed and $|\boldsymbol{\sigma}_n(x)| \le k_n < 1$ is bounded, the matrix $M_2(\boldsymbol{\sigma}_n)$ is also bounded. Besides, since $\mathbf{f}_n$ is a point cloud map within a fixed rectangle

$\tilde{R}$, the vectors U_n and V_n are bounded. Therefore, $E_n \to 0$ since $\gamma_n \to 0$. Furthermore, from Equation (8.63) and the assumption that $\tilde{\boldsymbol{\mu}}$ is well-defined, we conclude that $|D_2(x)[U_{n+1}^T, V_{n+1}^T]^T|$ is bounded below by a positive number when n is large. Hence, the right hand side of Equation (8.68) is well-defined, and it converges to $\lim_n \tilde{\boldsymbol{\mu}}_{n+1}(x) = \tilde{\boldsymbol{\mu}}(x)$. It follows that $\boldsymbol{\sigma}_n$ converges to $\tilde{\boldsymbol{\mu}}$.

As $|\boldsymbol{\sigma}_n(x)| = k_n < 1$ is a constant for any $x \in P$, $k := \lim_n k_n = |\tilde{\boldsymbol{\mu}}(x)|$ is also a constant.

Let p_i be any interior point. It remains to prove that $(L_i\tilde{\boldsymbol{\mu}})\overline{\tilde{\boldsymbol{\mu}}(p_i)} \in \mathbb{R}$. The statement is trivial when $L_i\tilde{\boldsymbol{\mu}} = 0$. Therefore, we only consider the case when $L_i\tilde{\boldsymbol{\mu}} \neq 0$ and $k \neq 0$. Let $\tilde{\boldsymbol{\mu}}_n = \tilde{\boldsymbol{\mu}}_{n,1} + i\tilde{\boldsymbol{\mu}}_{n,2}$, $\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{\rho}}_1 + i\tilde{\boldsymbol{\rho}}_2$, and $\boldsymbol{\sigma}_n = \boldsymbol{\sigma}_{n,1} + i\boldsymbol{\sigma}_{n,2}$. Without loss of generality, let $\tilde{\boldsymbol{\mu}}_2(p_i) \neq 0$. Then, when n is sufficiently large, we have

$$\frac{\tilde{\boldsymbol{\rho}}_1(p_i)}{\tilde{\boldsymbol{\rho}}_2(p_i)} = \lim_n \frac{\boldsymbol{\sigma}_{n,1}(p_i)}{\boldsymbol{\sigma}_{n,2}(p_i)} = \lim_n \frac{L_i(\tilde{\boldsymbol{\mu}}_{n,1}/|\tilde{\boldsymbol{\mu}}_n|)}{L_i(\tilde{\boldsymbol{\mu}}_{n,2}/|\tilde{\boldsymbol{\mu}}_n|)} = \frac{L_i(\tilde{\boldsymbol{\rho}}_1/k)}{L_i(\tilde{\boldsymbol{\rho}}_2/k)} = \frac{L_i\tilde{\boldsymbol{\rho}}_1}{L_i\tilde{\boldsymbol{\rho}}_2} \tag{8.69}$$

Hence, $(L_i\tilde{\boldsymbol{\rho}}_1)\tilde{\boldsymbol{\rho}}_2(p_i) = (L_i\tilde{\boldsymbol{\rho}}_2)\tilde{\boldsymbol{\rho}}_1(p_i)$. This concludes our proof. □

The above proposition guarantees that if Algorithm 11 converges, then $\mathbf{f}_n$ converges to the PCT-map we desired. Also, $\tilde{\boldsymbol{\mu}}_n$, $\boldsymbol{\sigma}_n$ converge to the discrete diffuse PCBC associated with the desired PCT-map. Moreover, the limit PCBC satisfies the requirements in Definition 8.4.1. As a remark, since Proposition 8.4.3 holds when the fill distance of a sequence of point clouds converges to zero, it follows that the accuracy of our algorithm increases with the density of the point clouds.

8.6 Experimental results

In this section, we demonstrate the effectiveness of our proposed *TEMPO* algorithm. Our algorithms are implemented in MATLAB. The point clouds

are adopted from the AIM@SHAPE shape repository [89] and the TF3DM repository [95]. The sparse linear systems are solved using the built-in backslash operator (\) in MATLAB. All experiments are performed on a PC with an Intel(R) Core(TM) i7-4770 CPU @3.40 GHz processor and 16.00 GB RAM.

8.6.1 The performance of our proposed approximation schemes

We first evaluate the performance of our proposed approximation schemes for computing conformal/quasi-conformal maps on disk-type point clouds.

We first compare our proposed scheme with the local mesh method [46] and the MLS method with special weight [52, 53] for computing conformal maps. In each experiment, we generate a random point cloud on a rectangle. Then, we map the planar point cloud to 3D using a conformal map with an explicit formula. On the transformed point cloud, we approximate the LB operator using the above approximation schemes. Using the approximated LB operator, we solve the Laplace equation to map the point cloud back onto the rectangular domain and calculate the resulting position errors. An example of such a conformal map with an explicit formula is shown in Figure 8.2. Table 8.1 records several experiments with this conformal map. Our proposed scheme results in better approximations.

Then, we compare the mentioned approximation schemes for computing quasi-conformal maps with prescribed PCBCs. This time, in each experiment, we map a randomly generated planar point cloud to 3D using a quasi-conformal map with an explicit formula. Figure 8.3 shows an example of such quasi-conformal map. On the 3D point cloud, we approximate the generalized LB operator using different schemes and solve the generalized

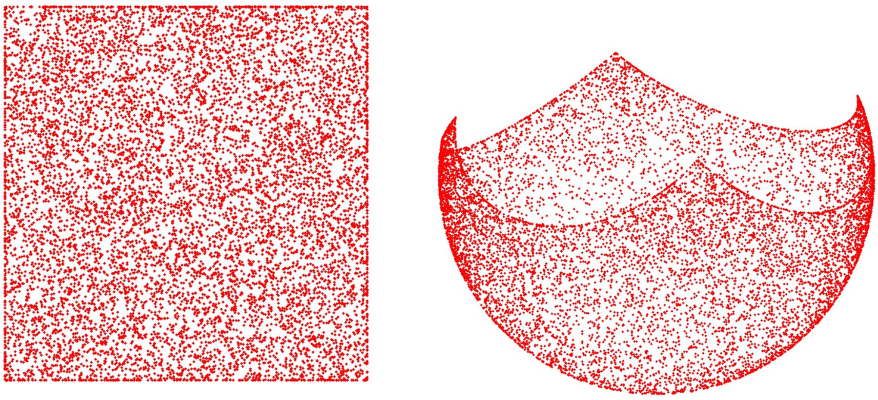

Figure 8.2: An example (the stereographic projection) used in evaluating different numerical schemes for computing conformal maps on point clouds. The explicit formula of it is $f(x,y) = \left(\frac{2x}{x^2+y^2+1}, \frac{2y}{x^2+y^2+1}, \frac{x^2+y^2-1}{x^2+y^2+1}\right)$.

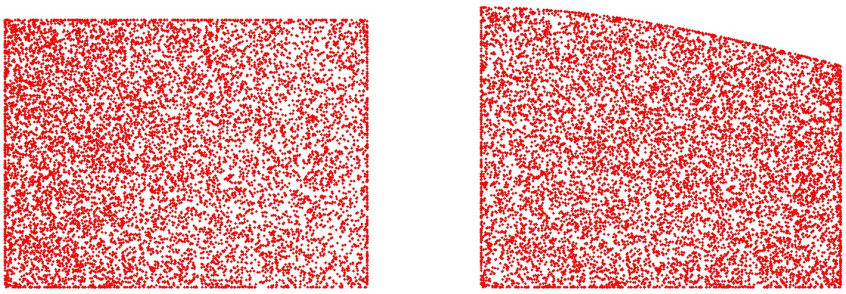

Figure 8.3: A quasi-conformal map used in evaluating different schemes for computing quasi-conformal maps on point clouds. The explicit formula of it is $f(x,y) = \left(\log(x+1), \arcsin\left(\frac{y}{2+(\log(x+1))^2}\right)\right)$.

Laplace equation to map the point cloud back onto the rectangle. Table 8.2 records several experiments. Again, our proposed combined scheme produces better approximations.

Example 1

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.048	0.017	0.00031
MLS with special weight [52, 53]	1.821	0.275	0.13026
Our proposed method	0.048	0.017	0.00031

Example 2

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.022	0.0043	0.000020
MLS with special weight [52, 53]	0.853	0.2529	0.071753
Our proposed method	0.011	0.0035	0.000014

Example 3

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.011	0.0041	0.000017
MLS with special weight [52, 53]	0.182	0.0659	0.004212
Our proposed method	0.010	0.0035	0.000013

Example 4 (Noisy)

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.050	0.0150	0.000202
MLS with special weight [52, 53]	0.379	0.0881	0.009463
Our proposed method	0.026	0.0083	0.000063

Example 5 (Noisy)

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.090	0.0128	0.000238
MLS with special weight [52, 53]	0.501	0.0574	0.004941
Our proposed method	0.049	0.0097	0.000105

Table 8.1: The performances of different numerical schemes for approximating conformal maps on point clouds.

8.6.2 Landmark constrained Teichmüller parameterizations

After demonstrating the advantage of our proposed approximation scheme, we illustrate the effectiveness of our *TEMPO* algorithm for the landmark constrained Teichmüller parameterization of disk-type point clouds.

We first compare the Teichmüller parameterization results with different

Example 1

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.0756	0.01150	0.00014248
MLS with special weight [52, 53]	0.0940	0.02524	0.00089952
Our proposed method	0.0252	0.00836	0.00006811

Example 2

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.0467	0.01011	0.00010035
MLS with special weight [52, 53]	0.0212	0.01219	0.00012290
Our proposed method	0.0183	0.00750	0.00005012

Example 3

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.0172	0.00813	0.00005617
MLS with special weight [52, 53]	0.6373	0.11550	0.02536965
Our proposed method	0.0172	0.00796	0.00005446

Example 4 (Noisy)

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.0181	0.00685	0.00004237
MLS with special weight [52, 53]	0.0565	0.01594	0.00034875
Our proposed method	0.0177	0.00652	0.00003953

Example 5 (Noisy)

Method	maximum position error	average 1-norm error	average 2-norm error
Local Mesh [46]	0.0170	0.00769	0.00005009
MLS with special weight [52, 53]	0.0429	0.01948	0.00034534
Our proposed method	0.0170	0.00762	0.00004964

Table 8.2: The performances of different schemes for computing quasi-conformal maps on disk-type point clouds.

choices of γ in the hybrid equation (8.37) . Two categories of the parameter γ are considered:

(i) Decreasing values: γ_n is gradually decreased throughout the iterations.

(ii) Constant values: γ remains constant throughout the iterations.

In the experiment, we compute the Teichmüller parameterization of a real 3D point cloud and calculate the variance of $|\boldsymbol{\mu}|$. A negligible variance of

Result with decreasing γ

γ	0	1	50	100	500	1000	∞
Var($\lvert\boldsymbol{\mu}\rvert$)	NaN	0.0018	0.0024	0.0020	0.0105	0.0085	0.0050

Result with constant γ

γ	0	1	50	100	500	1000	∞
Var($\lvert\boldsymbol{\mu}\rvert$)	NaN	0.0018	0.0022	0.0017	0.0031	0.0022	0.0050

Table 8.3: The results obtained by different γ in Algorithm 11. The row of γ refers to the initial values.

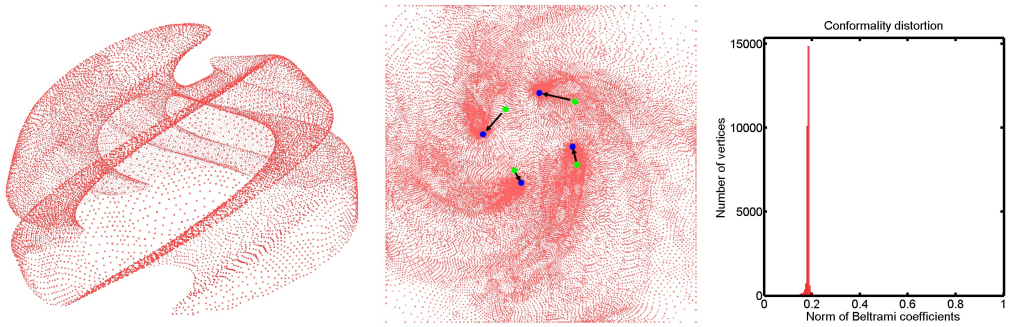

Figure 8.4: Teichmüller parameterization of a spiral point cloud. Left: The input. Middle: The parameterization. The green and the blue points respectively represent the original and the target locations of the landmark constraints. Right: The histogram of the norm of the Beltrami coefficients.

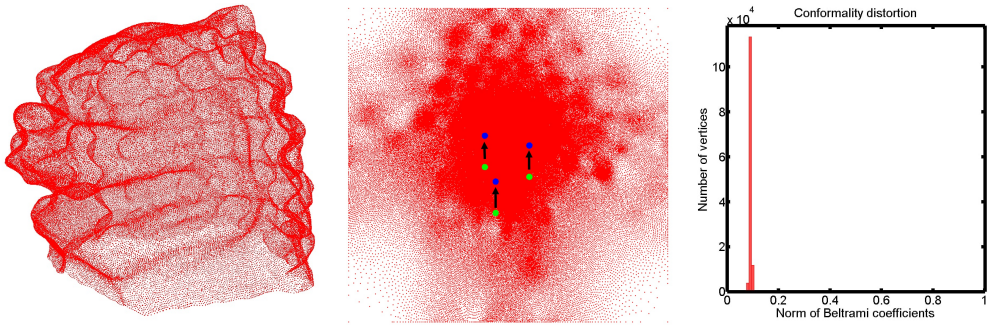

Figure 8.5: Teichmüller parameterization of a lion point cloud. Left: The input. Middle: The parameterization. The green and the blue points respectively represent the original and the target locations of the landmark constraints. Right: The histogram of the norm of the Beltrami coefficients.

$|\boldsymbol{\mu}|$ is desired because of the Teichmüller property. For fairness, we perform the same number of iterations in Algorithm 11. It is noteworthy that the

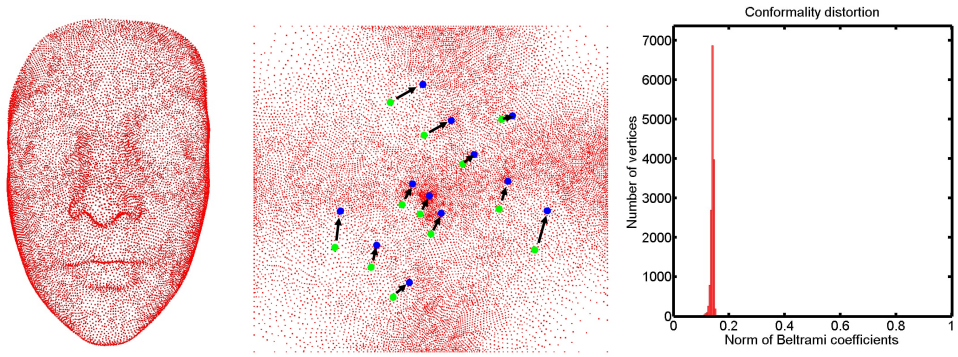

Figure 8.6: Teichmüller parameterization of Lucy point cloud. Left: The input. Middle: The parameterization. The green and the blue points respectively represent the original and the target locations of the landmark constraints. Right: The histogram of the norm of the Beltrami coefficients.

Point clouds	# of points	Time (m)	Mean($\|\boldsymbol{\mu}\|$)	Var($\|\boldsymbol{\mu}\|$)
Cesar	9755	0.3081	0.2402	6.1474e-04
Muscle guy	14240	0.3241	0.2934	9.8858e-04
Obese man	14415	0.3568	0.2508	7.7045e-04
Lucy	15167	0.6486	0.1523	1.4981e-04
Spiral	28031	1.1648	0.1986	8.0233e-04
Human face (neutral)	31350	0.6517	0.0713	3.3564e-05
Human face (angry)	31468	0.7428	0.0796	3.8812e-05
Human face (sad)	31543	0.6117	0.0693	5.6557e-05
Human face (happy)	31878	0.8536	0.0909	7.7383e-05
Bumps	114803	2.5353	0.1070	3.4902e-05
Lion	129957	2.5842	0.0893	2.8239e-05
Noisy face 1	20184	0.9172	0.1111	0.0140
Noisy face 2	19743	0.5191	0.1077	0.0130
Noisy face 3	20194	0.4905	0.0787	0.0157

Table 8.4: The performance of our *TEMPO* algorithm.

cases $\gamma = 0$ or ∞ respectively correspond to solely solving Equation (3.11) or Equation (3.12).

In Table 8.3, it can be observed that the algorithm fails if $\gamma = 0$. Also, the result associated with $\gamma = \infty$ is not as accurate as those associated with $0 < \gamma < +\infty$. It follows that our hybrid scheme outperforms the two existing

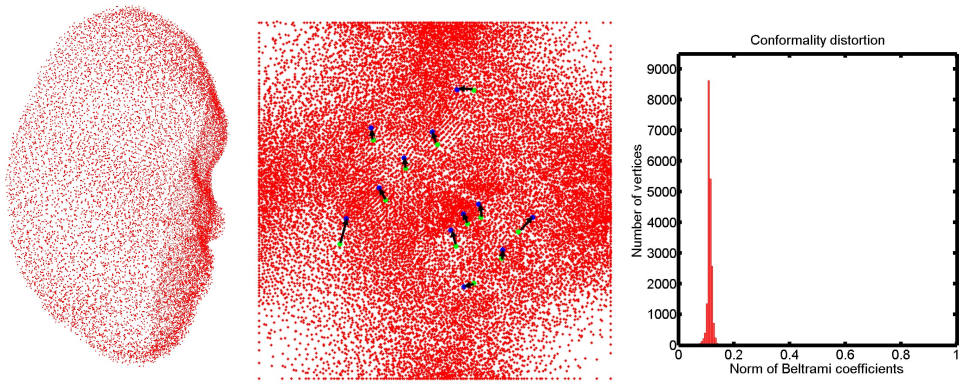

Figure 8.7: Teichmüller parameterization of a noisy facial point cloud. Left: The input. Middle: The parameterization. The green and the blue points respectively represent the original and the target locations of the landmark constraints. Right: The histogram of the norm of the Beltrami coefficients.

methods.

Besides, it can be observed that the results associated with different $0 < \gamma < +\infty$ are highly similar, regardless of whether γ is decreasing or constant. The observation suggests that the assumption $\lim_n \gamma_n = 0$ in Proposition 8.5.3 can be relaxed in practice.

For simplicity, in the following experiments, we set $\gamma \equiv 0.5$ in the hybrid equation (8.37). The stopping criterion is set to be $\epsilon = 10^{-6}$.

Figure 8.4 shows the landmark aligned Teichmüller parameterization of a spiral point cloud. Even under the large landmark displacements, our parameterization result ensures a uniform conformality distortion. Figure 8.5 and Figure 8.6 show two more examples. Again, the norms of the resulting Beltrami coefficients accumulate at a specific value, which implies that the parameterizations obtained by our *TEMPO* algorithm are Teichmüller.

We also test our *TEMPO* algorithm on noisy point clouds. Figure 8.7 shows the Teichmüller parameterization of a point cloud with 5% uniformly distributed random noise. Again, the histogram of the resulting Beltrami

coefficient indicates that the resulting parameterization is with uniform conformality distortion. This demonstrates the effectiveness of our *TEMPO* algorithm on noisy point clouds.

The statistics of numerous experiments are listed in Table 8.4. It is noteworthy that our *TEMPO* algorithm is highly efficient. The computations complete within 1 minute on average for point clouds with moderate size. Also, the variances of the resulting Beltrami coefficients are close to zero in all experiments, which indicates that our parameterizations achieve uniform conformality distortions. It can also be observed that for denser point clouds, the variances of the results are in general smaller. All of the above experimental results coincide with the theories we have established.

8.7 Application in point cloud classification

In this section, we explain the use of our proposed landmark-matching Teichmüller parameterization for point cloud classification.

8.7.1 Shape analysis via Teichmüller metric

We first introduce the use in the continuous setting. Suppose Ω_1 and Ω_2 are two simply-connected open surfaces, and the landmark correspondences are given by

$$p_i \leftrightarrow q_i \tag{8.70}$$

for $i = 1, 2, \cdots, n$. To compute the landmark-matching Teichmüller registration map $f : \Omega_1 \to \Omega_2$ such that $f(p_i) = q_i$ for all $i = 1, 2, \cdots, n$, we start by computing the conformal parameterizations $g_1 : \Omega_1 \to \tilde{R}_1$, $g_2 : \Omega_2 \to \tilde{R}_2$. Then, we compute the landmark-matching Teichmüller parameterization

$h : \tilde{R}_1 \to \tilde{R}_2$ between the two planar domains with

$$h(g_1(p_i)) = g_2(q_i) \tag{8.71}$$

for $i = 1, 2, \cdots, n$. Finally, $f : \Omega_1 \to \Omega_2$ can be obtained by

$$f = g_2^{-1} \circ h \circ g_1. \tag{8.72}$$

Obviously, f satisfies

$$f(p_i) = q_i. \tag{8.73}$$

Moreover, note that since h is a Teichmüller map, the Beltrami coefficient μ_h is with constant norm. Also, since g_1, g_2 are conformal maps, the Beltrami coefficient μ_{g_1}, μ_{g_2} is zero. Therefore, by Theorem 3.3.3,

$$\mu_f = \mu_{g_2^{-1} \circ h \circ g_1} = (\mu_h \circ g_1) \frac{\overline{(\partial_z g_1)}}{\partial_z g_1} = k \frac{\overline{(\partial_z g_1)^2 (\varphi \circ g_1)}}{|(\partial_z g_1)^2 (\varphi \circ g_1)|} \tag{8.74}$$

where $k = |\mu_h|$ is a constant, and φ is holomorphic. Also, since g_1 is conformal, $(\partial_z g_1)^2 (\varphi \circ g_1)$ is holomorphic. Hence, f is a landmark-matching Teichmüller map.

In the discrete case, the landmark-matching Teichmüller registration between two feature-endowed disk-type point clouds is summarized in Algorithm 12.

With the abovementioned registration scheme, a metric shape space can be built for point cloud classification. More explicitly, a dissimilarity metric called the *Teichmüller metric* is naturally induced by Teichmüller maps.

Definition 8.7.1 (Teichmüller metric)**.** *For all i, let M_i be a Riemann surface with feature landmarks $\{p_{ik}\}_{k=1}^n$. Let Ω be a template surface with landmarks $\{q_k\}_{k=1}^n$. Suppose every M_i is parameterized onto Ω by a landmark aligned quasi-conformal map $f_i : M_i \to \Omega$, with p_{ik} matching q_k for all k. The*

Algorithm 12: Landmark-matching Teichmüller registration between disk-type point clouds

Input: Two disk-type point clouds P_1 and P_2, with landmark correspondences $p_i \leftrightarrow q_i$, $p_i \in P_1$, $q_i \in P_2$, $i = 1, 2, ..., n$

Output: The Teichmüller map $\mathbf{f} : P_1 \to P_2$ with $\mathbf{f}(p_i) = q_i$ for all i

1 Apply Algorithm 10 to compute the conformal parameterizations $\mathbf{g}_t : P_t \to \tilde{R}_t$, where $t = 1, 2$;

2 Apply Algorithm 11 to compute the Teichmüller parameterization $\mathbf{h} : \tilde{R}_1 \to \tilde{R}_2$, with the landmark constraints $\mathbf{h}(\mathbf{g}_1(p_i)) = \mathbf{g}_2(q_i)$ for all i;

3 Obtain the registration map $\mathbf{f} = \mathbf{g}_2^{-1} \circ \mathbf{h} \circ \mathbf{g}_1$.

Teichmüller metric *between* (f_i, M_i) *and* (f_j, M_j) *is defined by*

$$d_T((f_i, M_i), (f_j, M_j)) = \inf_{\varphi} \frac{1}{2} \log K(\varphi), \tag{8.75}$$

where $\varphi : M_i \to M_j$ *varies over all quasi-conformal maps with* $\{p_{ik}\}_{k=1}^n$ *corresponds to* $\{p_{jk}\}_{k=1}^n$*, which is homotopic to* $f_j^{-1} \circ f_i$*, and* K *is the maximal quasi-conformal dilation.*

Note that the maximal quasi-conformal dilation K is not affected by compositions of conformal maps. Therefore, if $\varphi = g_j^{-1} \circ h \circ g_i$, where g_i, g_j are conformal parameterizations, then

$$K(\varphi) = K(h). \tag{8.76}$$

Thus, the Teichmüller metric is uniquely determined by the maximal quasi-conformal dilation of quasi-conformal maps on the rectangular domains.

By Theorem 3.4.3, with a suitable boundary condition, the T-map between the unit disks is extremal and unique. A similar result also holds on rectangular domains. Therefore, the infimum in Equation (8.75) is achieved by the unique Teichmüller map between $\tilde{R}_i$ and $\tilde{R}_j$. In other words,

$$d_T((f_i, M_i), (f_j, M_j)) = \frac{1}{2} \log K(h) \tag{8.77}$$

where $h : \tilde{R}_i \to \tilde{R}_j$ is a landmark-constrained Teichmüller map with $g_j^{-1} \circ h \circ g_i$ homotopic to $f_j^{-1} \circ f_i$. Hence, the difference of two feature-endowed point clouds can be evaluated in terms of the quasi-conformal dilation of the landmark aligned Teichmüller parameterization between $\tilde{R}_i$ and $\tilde{R}_j$.

In this discrete case, let $\{P_t\}$ be a collection of feature-endowed point clouds. To accurately classify them, for all i, j we compute the landmark-matching Teichmüller map $\mathbf{f}_{ij} : P_i \to P_j$. Then, we evaluate the associated discrete diffuse PCBCs $\boldsymbol{\mu}_{ij}$ and denote the Teichmüller distance between P_i and P_j by

$$d_{ij} = \frac{1}{2} \log \frac{1 + \|\boldsymbol{\mu}_{ij}\|_\infty}{1 - \|\boldsymbol{\mu}_{ij}\|_\infty}. \tag{8.78}$$

Note that if P_i and P_j are with similar shapes, then d_{ij} is small. By analyzing the distance matrix (d_{ij}), we can effectively classify different feature-endowed point clouds. For instance, the multidimensional scaling (MDS) method can be applied for the classification here.

In practice, to avoid the effect of very minor outlying Beltrami coefficients in the approximations, we replace $\|\mu\|_\infty$ by the average of $|\mu|$ in calculating the Teichmüller distances. This modification is valid as the variance of $|\mu|$ is negligible. It is also noteworthy that the Teichmüller distance d_{ij} (induced by the PCT-map $f_{ij} : P_i \to P_j$) may be slightly different from d_{ji} (induced by the PCT-map $f_{ji} : P_j \to P_i$) because of the numerical errors. For symmetry, we replace the distance matrix $D = (d_{ij})$ by $(D + D^T)/2$. Our proposed algorithm for the dissimilarity metric is summarized in Algorithm 13.

8.7.2 Evaluating the classification result

Using our *TEMPO* algorithm, landmark-matching Teichmüller registration between disk-type point clouds can be efficiently computed as described in Algorithm 12. Figure 8.8 shows the registration of two human face point

Algorithm 13: Building a distance matrix using the Teichmüller metric

Input: A set of point clouds $\{P_t\}_{t=1}^{M}$ with landmark correspondences
Output: The distance matrix $D = (d_{ij})$, $i, j = 1, 2, \cdots, M$

1 Apply Algorithm 12 to compute the landmark-matching PCT-map $\mathbf{f}_{ij} : P_i \to P_j$;
2 Compute the Teichmüller distance $d_{ij} = \frac{1}{2}\log(K(\mathbf{f}_{ij}))$ and form the matrix $D = (d_{ij})$;
3 Update D by $(D + D^T)/2$;

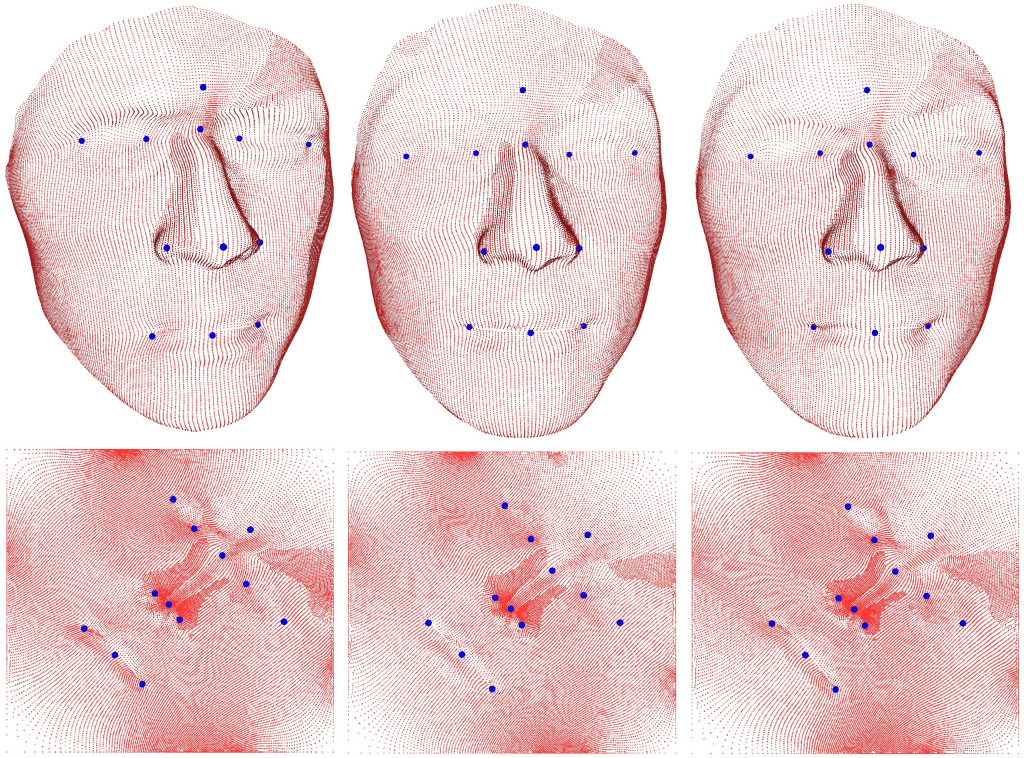

Figure 8.8: Teichmüller registration between two human face point clouds with prescribed landmark constraints (highlighted in blue). Top left: The source point cloud. Top middle: The target point cloud. Top right: The registration resul. Bottom: The conformal parameterizations of the source and target point clouds, and the landmark constrained Teichmüller parameterization.

clouds by our proposed method. In the registration result, the landmark constraints are fulfilled and the geometric features are optimally preserved.

To demonstrate the effectiveness of the Teichmüller metric induced, we

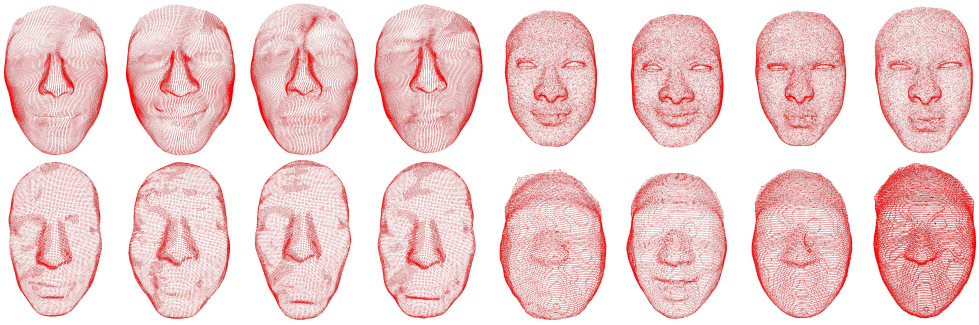

Figure 8.9: The dataset of human face point clouds with different facial expressions used in our first experiment.

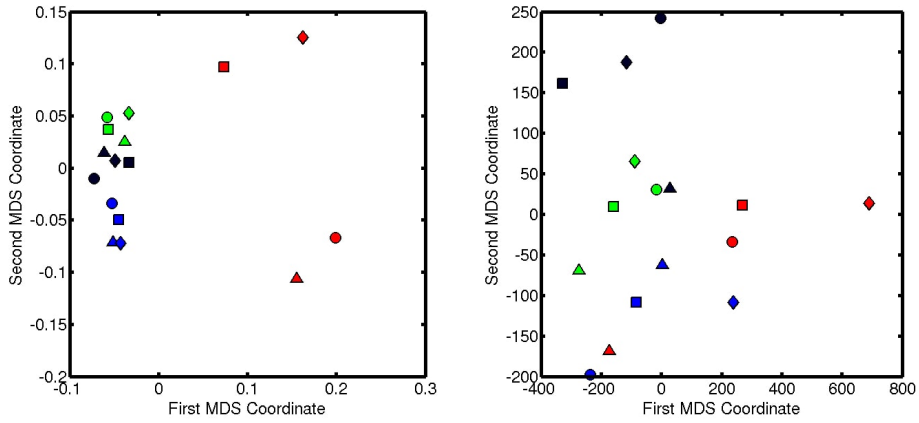

Figure 8.10: The MDS results for our proposed distance matrix (left) and the distance matrix in [41] (right). Each color represents one human, and each shape represents a type of facial expressions (circle: neutral, square: happy, triangle: sad, diamond: angry).

consider a facial point cloud classification problem. Given a set of facial point clouds with multiple facial expressions, we aim to correctly classify the point clouds into several groups, where each group represents one human.

Landmark constraints at the most prominent parts of the faces, such as the eyes, nose and mouth of every point cloud are manually labeled to ensure the accuracy of the classification. We compare our proposed Teichmüller metric with another dissimilarity metric computed by a feature-endowed point cloud mapping algorithm in [41]. For a fair comparison, we replace the

Figure 8.11: A partial set of the facial point clouds used in our second experiment, adopted from the 3D human face database [15].

automatically detected landmarks in [41] by our manually labeled landmarks. In the experiments, we compute our proposed distance matrix using Algorithm 13 and another distance matrix using [41]. The multidimensional scaling (MDS) method is then applied on the two distance matrices. Two experiments are presented below.

In the first experiment, we are interested in classifying facial point clouds with specific facial expressions. 16 facial point clouds with 4 specific expressions (neutral, happy, sad, angry) are adopted from [91, 17, 86] or sampled by Kinect (see Figure 8.9). 12 landmark constraints are manually labeled

on each point cloud. Figure 8.10 shows the MDS results. It can be observed that distinct subjects are effectively clustered using our distance matrix, while the result based on the distance matrix in [41] cannot identify distinct subjects. For a more detailed comparison, the leave-one-out cross-validation (LOOCV) is applied to evaluate the classification accuracies. Our distance matrix results in a classification accuracy of 94%, while the distance matrix in [41] results in a classification accuracy of 50%. The comparison demonstrates the effectiveness of our Teichmüller metric.

In the second experiment, we consider a larger dataset. We adopt the 3D human face database [15], which is the database used in [41]. Each subject in the database is with 3 random facial expressions. Figure 8.11 shows a partial set of the facial point clouds in [15]. We label 10 landmark constraints on each point cloud for the computation. It is noteworthy that unlike the experiments in [41], we do not perform any triangulation or smoothing procedure on the raw point clouds. Our distance matrix results in a LOOCV accuracy of 79%, while the distance matrix in [41] results in a LOOCV accuracy of 50%. The results again reflect the effectiveness of our proposed dissimilarity metric for point cloud classification.

Chapter 9

Conclusion

Surface conformal/quasi-conformal parameterizations are important in medical morphometry, computer graphics and engineering. In the discrete case, two major discretizations of surfaces are triangulated meshes and point clouds. In this thesis, we have invented effective conformal/quasi-conformal parametrization algorithms for triangulated meshes and point clouds, including

(i) spherical conformal/optimized-conformal parametrization of genus-0 closed meshes,

(ii) disk conformal parametrization of simply-connected open meshes,

(iii) spherical quasi-conformal/Teichmüller parametrization of genus-0 closed meshes,

(iv) spherical conformal parametrization of genus-0 point clouds, and

(v) landmark aligned Teichmüller parametrization of point clouds with disk topology.

Experimental results have demonstrated the advantages of our proposed parametrization algorithms over the existing approaches. Specifically, our

proposed algorithms significantly accelerate the computation of the parameterizations with improved accuracies. Also, the bijectivity of our parameterizations is mathematically supported by quasi-conformal theory. Various applications of our proposed parametrization algorithms are exhibited.

In the future, we plan to extend our parametrization algorithms for high-genus meshes and point clouds, and investigate the use of them in physical and biological applications.

Bibliography

[1] P. T. Choi, K. C. Lam, and L. M. Lui, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces*. SIAM Journal on Imaging Sciences, Volume 8, Issue 1, pp. 67–94, 2015.

[2] P. T. Choi and L. M. Lui, *Fast disk conformal parameterization of simply-connected open surfaces*. Journal of Scientific Computing, Volume 65, Issue 3, pp. 1065–1090, 2015.

[3] P. T. Choi and L. M. Lui, *A linear algorithm for disk conformal parameterization of simply-connected open surfaces*. Preprint, arXiv:1508.00396, 2015.

[4] G. P.-T. Choi, K. T. Ho, and L. M. Lui, *Spherical conformal parameterization of genus-0 point clouds for meshing*. Preprint, arXiv:1508.07569, 2015.

[5] T. W. Meng, G. P.-T. Choi, and L. M. Lui, *TEMPO: Feature-endowed Teichmüller extremal mappings of point clouds*. Preprint, arXiv:1511.06624, 2015.

[6] G. P.-T. Choi, M. H.-Y. Man and L. M. Lui, *Linear spherical quasiconformal parameterizations of genus-0 closed surfaces for adaptive remeshing*. In Preparation.

[7] Y. Chen*, G. P.-T. Choi*, L. M. Lui, and B. Chiu, *Conformal mapping of 3D ultrasound carotid arteries images*. In Preparation. (*: Co-first authorship.)

[8] N. Aigerman and Y. Lipman, *Injective and bounded distortion mappings in 3D*. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), Volume 32, Issue 4, 106:1–106:14, 2013.

[9] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, *Conformal geometry and brain flattening*. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 271-278, 1999.

[10] P. Azariadis, *Parameterization of clouds of unorganized points using dynamic base surfaces*. Computer-Aided Design, Volume 36, pp. 607–623, 2004.

[11] P. Azariadis and N. Sapidis, *Efficient parameterization of 3D point-sets using recursive dynamic base surfaces*. Advances in Informatics, Lecture Notes in Computer Science, Volume 3746, pp 296–306, 2005.

[12] P. Azariadis and N. Sapidis, *Product design using point-cloud surfaces: A recursive subdivision technique for point parameterization*. Computers in Industry, Volume 58, pp. 832–843, 2007.

[13] M Belkin and P. Niyogi, *Towards a theoretical foundation for Laplacian-based manifold methods*. Journal of Computer and System Sciences, Volume 74, Issue 8, pp. 1289–1308, 2008.

[14] M. Belkin, J. Sun, and Y. Wang. *Constructing Laplace operator from point clouds in* $\mathbb{R}^d$. Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1031–1040, 2009.

[15] C. Beumier and M. Acheroy. *Automatic 3D face authentication*. Image and Vision Computing, Volume 18, Issue 4, pp. 315–321, 2000.

[16] P. Breitkopf, H. Naceur, A. Rassineux, and Pierre Villon, *Moving least squares response surface approximation: Formulation and metal forming applications*. Computers & Structures, Volume 83, Issues 17–18, pp. 1411–1428, 2005.

[17] A. M. Bronstein, M. M. Bronstein, R. Kimmel, *Calculus of nonrigid surfaces for geometry and texture manipulation*. IEEE Transactions on Visualization and Computer Graphics, Volume 13, Issue 5, pp. 902–913, 2007.

[18] M. Desbrun, M. Meyer, and P. Alliez, *Intrinsic parameterizations of surface meshes*, Computer Graphics Forum, Volume 21, pp. 209–218, 2002.

[19] T. K. Dey and S. Goswami. *Tight Cocone: A water tight surface reconstructor*. Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, pp. 127–134, 2003.

[20] M. do Carmo, *Differential geometry of curves and surfaces*, Prentice Hall, 1976.

[21] N. Dyn, D. Levine, and J.A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*. ACM Transactions on Graphics, Volume 9, Issue 2, pp. 160–169, 1990.

[22] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, *Multiresolution analysis of arbitrary meshes*. Proceedings of ACM SIGGRAPH, pp. 173-182, 1995.

[23] A. Fletcher and V. Markovic, *Quasiconformal maps and Teichmüller theory*. Oxford University Press, 2006.

[24] M. Floater, *Parameterization and smooth approximation of surface triangulations*, Computer Aided Geometric Design, Volume 14, Issue 3, pp. 231–250, 1997.

[25] M. S. Floater, *Meshless parameterization and B-spline surface approximation*. Proceedings of the Ninth IMA Conference on the Mathematics of Surfaces, pp. 1–18, 2000.

[26] M. S. Floater and M. Reimers, *Meshless parameterization and surface reconstruction*. Computer Aided Geometric Design, Volume 18, Issue 2, pp. 77–92, 2001.

[27] M. Floater and K. Hormann, *Parameterization of triangulations and unorganized points*, Tutorials on Multiresolution in Geometric Modelling, pp. 287–316, 2002.

[28] M. Floater, *Mean value coordinates*, Computer Aided Geometric Design, Volume 20, Issue 1, pp. 19–27, 2003.

[29] M. Floater and K. Hormann, *Surface parameterization: A tutorial and survey*, Advances in Multiresolution for Geometric Modelling, pp. 157–186, 2005.

[30] F. Gardiner and N. Lakic. *Quasiconformal Teichmüller theory*. American Mathematics Society, 2000.

[31] X. Gu and S.T. Yau, *Computing conformal structures of surfaces*. Communications and Information Systems, Volume 2, Number 2, pp. 121–146, 2002.

[32] X. Gu and S. T. Yau, *Global conformal surface parameterization*, Eurographics Symposium on Geometry Processing, pp. 127–137, 2003.

[33] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE Transactions on Medical Imaging, Volume 23, Issue 8, pp. 949–958, 2004.

[34] X. D. Gu and S.-T. Yau, *Computational conformal geometry*. Advanced Lectures in Mathematics (ALM), Volume 3, International Press, Somerville, MA, 2008.

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

[36] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle, *Conformal surface parameterization for texture mapping*. IEEE Transactions on Visualization and Computer Graphics, Volume 6, Issue 2, pp. 181–189, 2000.

[37] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface reconstruction from unorganized points*. ACM SIGGRAPH Computer Graphics, Volume 26, Issue 2, pp. 71–78, 1992.

[38] K. Hormann and G. Greiner, *MIPS: An efficient global parametrization method*, Curve and Surface Design, pp. 153–162, 2000.

[39] K. Hormann, B. Lévy, and A. Sheffer, *Mesh parameterization: Theory and practice*, ACM SIGGRAPH 2007 Course Notes, pp. 1–122, 2007.

[40] M.K. Hurdal, K. Stephenson, *Cortical cartography using the discrete conformal approach of circle packings*. NeuroImage, Volume 23, Supplement 1, pp. S119–S128, 2004.

[41] M. O. İrfanoğlu, B. Gökberk, and L. Akarun, *3D shape-based face recognition using automatically registered facial surfaces*. Proceedings of the 17th International Conference on Pattern Recognition, Volume 4, pp. 183–186, 2004.

[42] J. Jost, *Riemannian geometry and geometric analysis*. Universitext, Springer, 2011.

[43] M. Jin, Y. Wang, X. Gu, and S. T. Yau, *Optimal global conformal surface parameterization for visualization*, Communications in Information and Systems, Volume 4, Number 2, pp. 117–134, 2005.

[44] M. Jin, J. Kim, F. Luo, and X. Gu, *Discrete surface Ricci flow*, IEEE Transaction on Visualization and Computer Graphics, Volume 14, Issue 5, pp. 1030–1043, 2008.

[45] L. Kharevych, B. Springborn, and P. Schröder, *Discrete conformal mappings via circle patterns*, ACM Transactions on Graphics, Volume 25, Number 2, pp. 412–438, 2006.

[46] R. Lai, J. Liang, and H. Zhao, *A local mesh method for solving PDEs on point clouds*. Inverse Problem and Imaging, Volume 7, Issue 3, pp. 737–755, 2013.

[47] R. Lai, Z. Wen, W. Yin, X.F. Gu, and L.M. Lui, *Folding-free global conformal mapping for genus-0 surfaces by harmonic energy minimization*. Journal of Scientific Computing, Volume 58, Issue 3, pp. 705–725, 2014.

[48] K. Lai, L. Bo, and D. Fox, *Unsupervised feature learning for 3D scene labeling*. IEEE International Conference on Robotics and Automation (ICRA), May 2014.

[49] C. Lange and K. Polthier, *Anisotropic smoothing of point sets*.Computer Aided Geometric Design, Volume 22, Issue 7, pp. 680–692, 2005.

[50] D. Levin, *The approximation power of moving least-squares*. Mathematics of Computation, Volume 67, Issue 224, pp. 1517–1531, 1998.

[51] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), Volume 21, Issue 3, pp. 362–371, 2002.

[52] J. Liang, R. Lai, A. Wong and H. Zhao, *Geometric understanding of point clouds using Laplace-Beltrami operator*, IEEE Conference on Computer Vision and Pattern Recognition, pp. 214–221, 2012.

[53] J. Liang and H. Zhao, *Solving partial differential equations on point clouds*, SIAM Journal of Scientific Computing, Volume 35, Issue 3, pp. 1461–1486, 2013.

[54] Y. Lipman, *Bounded distortion mapping spaces for triangular meshes*. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), Volume 31, Issue 4, 108, 2012.

[55] W.K. Liu, S. Li and T. Belytschko, *Moving least-square reproducing kernel methods (I) methodology and convergence*, Computer Methods in Applied Mechanics and Engineering, Volume 143, pp. 113–154, 1997.

[56] C.T. Loop, *Smooth subdivision surfaces based on triangles*. M.S. Mathematics thesis, University of Utah, 1987.

[57] W.E. Lorensen and H.E. Cline, *Marching cubes: A high resolution 3D surface construction algorithm*. ACM SIGGRAPH Computer Graphics, Volume 21, Issue 4, pp. 163–169, 1987.

[58] L.M. Lui, Y. Wang, T.F. Chan, and P.M. Thompson, *Landmark constrained genus zero surface conformal mapping and its application to brain mapping research*. Applied Numerical Mathematics, Volume 57, pp. 847–858, 2007.

[59] L.M. Lui, S. Thiruvenkadam, Y Wang, P.M. Thompson, and T.F. Chan, *Optimized conformal surface registration with shape-based landmark matching*. SIAM Journal on Imaging Sciences, Volume 3, Issue 1, pp. 52–78, 2010.

[60] L.M. Lui, T.W. Wong, W. Zeng, X.F. Gu, P.M. Thompson, T.F. Chan and S.T. Yau, *Optimization of surface registrations using Beltrami holomorphic flow*. Journal of Scientific Computing, Volume 50, Issue 3, pp. 557–585. 2012.

[61] L. M. Lui, K. C. Lam, T. W. Wong, and X. Gu, *Texture map and video compression using Beltrami representation*. SIAM Journal on Imaging Sciences, Volume 6, Issue 4, pp. 1880–1902, 2013.

[62] L. M. Lui, K. C. Lam, S. T. Yau, and X. Gu, *Teichmüller mapping (T-map) and its applications to landmark matching*. SIAM Journal on Imaging Sciences, Volume 7, Number 1, pp. 391–426, 2014.

[63] L. M. Lui, X. Gu, and S.-T. Yau, *Convergence of an iterative algorithm for Teichmüller maps via harmonic energy optimization*. Mathematics of Computation, Volume 84, Number 296, pp. 2823–2842, 2015.

[64] F. Luo, *Combinatorial Yamabe flow on surfaces*, Communications in Contemporary Mathematics, Volume 06, Issue 5, pp. 765–780, 2004.

[65] D.S. Marcus, A.F. Fotenos, J.G. Csernansky, J.C. Morris, R.L. Buckner. *Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults*. Journal of Cognitive Neuroscience, Volume 22, Issue 12, pp. 2677–2684, 2010.

[66] Q. Meng, B. Li, H. Holstein, and Y. Liu, *Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBF networks*. Pattern Recognition, Volume 46, Issue 8, pp. 2361–2375, 2013.

[67] T. W. Meng and L. M. Lui, *The theory of computational quasi-conformal geometry on point clouds*. Preprint, arXiv:1510.05104, 2015.

[68] D. Mirzaei, R. Schaback, M. Dehghan, *On generalized moving least squares and diffuse derivatives*. IMA Journal of Numerical Analysis, 32, pp. 983–1000, 2012.

[69] P. Mullen, Y. Tong, P. Alliez and M. Desbrun, *Spectral conformal parameterization*. Computer Graphics Forum, Volume 27, Issue 5, pp. 1487–1494, 2008.

[70] A. Nealen, *An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered Data approximation and interpolation*, Technical Report, TU Darmstadt, 2004.

[71] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*. Experimental Mathematics, Volume 2, Number 1, pp. 15–36, 1993.

[72] R. Schoen and S.T. Yau, *Lectures on differential geometry*. International Press, Cambridge, MA, 1994.

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

[74] A. Sheffer and E. de Sturler, *Surface parameterization for meshing by triangulation flattening*, Proceedings of 9th International Meshing Roundtable, pp. 161–172, 2000.

[75] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, *ABF++: Fast and robust angle based flattening*, ACM Transactions on Graphics, Volume 24, Issue 2, pp. 311–330, 2005.

[76] A. Sheffer, E. Praun and K. Rose, *Mesh parameterization methods and their applications*, Foundations and Trends in Computer Graphics and Vision, Volume 2, Issue 2, pp. 105–171, 2006.

[77] G. Tewari, C. Gotsman, and S.J. Gortler, *Meshing genus-1 point clouds using discrete one-forms*. Computers and Graphics, Volume 30, pp 917–926, 2006.

[78] Y. Wang, L.M. Lui, T.F. Chan, and P.M. Thompson, *Optimization of brain conformal mapping with landmarks*.Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part II, pp. 675-683, 2005.

[79] L. Wang, B. Yuan and Z. Miao, *3D point clouds parameterization algorithm*. 9th International Conference on Signal Processing, pp. 1410–1413, 2008.

[80] O. Weber, A. Myles, D. Zorin, *Computing extremal quasiconformal maps*. Computer Graphics Forum, Volume 31, Issue 5, pp. 1679–1689, 2012.

[81] L.M. Lui and C. Wen, *Geometric Registration of High-Genus Surfaces*. SIAM Journal on Imaging Sciences, Volume 7, Issue 1, pp. 337–365, 2014.

[82] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*. Advances in Computational Mathematics, Volume 4, Issue 1, pp. 389–396, 1995.

[83] H. Wendland, *Local polynomial reproduction and moving least squares approximation*. IMA Journal of Numerical Analysis, Volume 21, Issue 1, pp. 285–300, 2001.

[84] H. Wendland, *Scattered data approximation*. Cambridge University Press, Volume 17, 2004.

[85] Y. L. Yang, R. Guo, F. Luo, S. M. Hu, and X. F. Gu, *Generalized discrete Ricci flow*, Computer Graphics Forum, Volume 28, Issue 7, pp. 2005–2014, 2009.

[86] L. Zhang, N. Snavely, B. Curless, and S. Seitz, *Spacetime faces: High-resolution capture for modeling and animation*. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH), Volume 23, Issue 3, pp. 548–558, 2004.

[87] L. Zhang, L. Liu, C. Gotsman, and H. Huang, *Mesh reconstruction by meshless denoising and parameterization*. Computers and Graphics, Volume 34, Issue 3, pp. 198–208, 2010.

[88] M. Zwicker and C. Gotsman, *Meshing point clouds using spherical parameterization*. Proceedings of the Eurographics Symposium on Point-Based Graphics, 2004.

[89] *AIM@SHAPE shape repository*. http://visionair.ge.imati.cnr.it/

[90] *GoTools*. http://www.sintef.no/projectweb/geometry-toolkits/gotools/

[91] *Laboratory for computational longitudinal neuroimaging (LCLN) shape database*. http://reuter.mit.edu/research/shapeanalysis/shapedb1/

[92] *A benchmark for 3D nesh segmentation*. http://segeval.cs.princeton.edu/

[93] *RiemannMapper: A mesh parameterization toolkit*.http://www3.cs.stonybrook.edu/~gu/software/RiemannMapper/

[94] *Stanford 3D scanning repository*. http://graphics.stanford.edu/data/3Dscanrep/

[95] *TF3DM*. http://tf3dm.com/

[96] D. Lin, *Statistical learning toolbox*. MathWorks Community File Exchange, 2006. http://www.mathworks.com/matlabcentral/fileexchange/12333-statistical-learning-toolbox