Spherical Conformal Parameterization of Genus-0 Point Clouds for Meshing[*]

Gary Pui-Tung Choi[†], Kin Tat Ho[‡], and Lok Ming Lui[‡]

Abstract. The point cloud is the most fundamental representation of three-dimensional geometric objects. Analyzing and processing point cloud surfaces is important in computer graphics and computer vision. However, most of the existing algorithms for surface analysis require connectivity information. Therefore, it is desirable to develop a mesh structure on point clouds. This task can be simplified with the aid of a parameterization. In particular, conformal parameterizations are advantageous in preserving the geometric information of the point cloud data. In this paper, we extend a state-of-the-art spherical conformal parameterization algorithm for genus-0 closed meshes to the case of point clouds, using an improved approximation of the Laplace–Beltrami operator on data points. Then, we propose an iterative scheme called the north-south reiteration for achieving a spherical conformal parameterization. A balancing scheme is introduced to enhance the distribution of the spherical parameterization. High-quality triangulations and quadrangulations can then be built on the point clouds with the aid of the parameterizations. Also, the meshes generated are guaranteed to be genus-0 closed meshes. Moreover, using our proposed spherical conformal parameterization, multilevel representations of point clouds can be easily constructed. Experimental results demonstrate the effectiveness of our proposed framework.

Key words. mesh generation, triangulation, quadrangulation, spherical conformal parameterization, surface reconstruction, point cloud, multilevel representation

AMS subject classifications. 65D18, 65D99

DOI. 10.1137/15M1037561

1. Introduction. Contemporary scanning technologies enable efficient acquisitions of three-dimensional (3D) objects. Using modern 3D scanners, data points are sampled from the surfaces of 3D objects for further analyses and usages. Point clouds are widely applied in computer graphics, vision, and many other engineering fields. However, the data points acquired by laser scanners are often complex and unorganized. Moreover, the absence of the connectivity information in point cloud data poses difficulties in understanding the underlying geometry of the 3D objects. This largely hinders the applications of the data. For instance, many applications in 3D printing [51, 38] and texture mapping [52, 33] are built upon mesh structures. With the rapid development of the computer industry, finding a high-quality meshing framework for point cloud data is increasingly important.

[†]John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 (pcho@g.harvard.edu).
[‡]Department of Mathematics, Chinese University of Hong Kong, Shatin, Hong Kong (ktho@math.cuhk.edu.hk, lmlui@math.cuhk.edu.hk).

One possible approach for mesh generation on point clouds is to parameterize a point cloud to a simpler domain with the corresponding genus, such as the unit sphere for genus-0 point clouds. Then, a triangulation or a quadrangulation can be created on the parameter domain instead of the original complicated point cloud. Finally, a mesh structure on the point cloud can be defined with respect to the structure on the parameter domain. The major difficulty of computing parameterizations of point-set surfaces is the extremely limited information they can provide. Most of the existing surface parameterization methods are developed on meshes only. In other words, besides the locations of the point data, a given connectivity is also required as an input. The connectivity information plays an important role in representing the surface structure as well as in approximating continuous operators to minimize certain distortions. As a result, most conventional mesh parameterization approaches fail to work on point clouds. Without the connectivity information, the underlying geometry of the point cloud data becomes more obscure. Hence, it is more challenging to develop parameterization schemes with good quality for point cloud data.

A good parameterization scheme for a point cloud must satisfy certain criteria. In particular, it should retain the geometric information of a point cloud as completely as possible. In our case, one of the ultimate goals is to create a triangulation for a point cloud by finding a Delaunay triangulation on a simpler parameter domain. It is noteworthy that in general, a mesh structure with good quality on the parameter domain does not necessarily imply that the associated mesh structure of the original data points is satisfactory. In other words, meshing the parameter domain may provide meaningless results if the parameterization scheme is arbitrarily chosen. Note that the regularity of the mesh structures is related to the angle structure of the triangles and quadrilaterals. To ensure the regularity of the associated mesh structure on the point cloud, the parameterization should preserve the angle structure of the triangles and quadrilaterals on the parameter domain. This motivates for us the use of conformal mappings.

For smooth surfaces, it is well known that the conformal parameterizations preserve angles and hence the local geometry of the surfaces. It is natural to consider the discrete analogue of conformal parameterization for point cloud data. Since data points are sampled from real 3D surfaces, we can assume that every point cloud has an underlying geometry. Based on this important assumption, we consider finding conformal parameterizations of genus-0 point clouds. In [9], Choi, Lam, and Lui proposed a fast spherical conformal parameterization algorithm for genus-0 closed surfaces in two steps. In the first step, a Laplace equation is solved on a planar triangular domain and the inverse stereographic projection is applied to obtain an initial spherical parameterization. In the second step, quasi-conformal theories are applied to enhance the conformality of the spherical parameterization. The computation only involves solving a few sparse linear systems and the conformality distortion of the parameterization is minimal. However, the algorithm is developed on triangular meshes only. In this work, we extend and improve the algorithm for point clouds with spherical topology.

The aforementioned algorithm in [9] developed on meshes involves solving a Laplace equation. To extend the algorithm for point clouds, we propose a new weight function for enhancing the accuracy of the approximation of the Laplace–Beltrami (LB) operators on point clouds. Using our improved approximation, the LB operator in the mentioned algorithm can be accurately computed on point clouds. Also, we replace a key step of solving for a quasi-conformal map in the mentioned algorithm by an iterative scheme, called the *north-south*

(N-S) reiteration, for improving the conformality of the parameterizations. Furthermore, we introduce a balancing scheme for enhancing the distribution of the parameterization results. Experimental results demonstrate the effectiveness of our proposed parameterization algorithm for genus-0 point clouds. Our algorithm achieves global spherical parameterizations with minimal conformality distortions. Furthermore, with the aid of our parameterization schemes, we can easily generate high-quality triangulations and quadrangulations on point clouds. The meshes generated are guaranteed to be genus-0 closed meshes. Moreover, multilevel representations of the point clouds can also be easily computed with the aid of our spherical parameterization scheme.

The rest of the paper is organized as follows. The contribution of our work is highlighted in section 2. In section 3, we review the related previous works on point cloud parameterizations and approximations of differential operators on point clouds. In section 4, we introduce the mathematical background of our work. In section 5, we review a spherical conformal parameterization scheme for triangular meshes, which is closely related to our proposed framework for point clouds. In section 6, we explain our proposed framework for spherical conformal parameterization and mesh generation of point clouds. In section 7, we demonstrate the effectiveness of our proposed framework by numerous experiments. The paper is concluded in section 8.

2. Contribution. In this work, we propose a framework for meshing using spherical conformal parameterizations of genus-0 point clouds. Our proposed method is advantageous in the following aspects:

(i) We extend and improve the spherical conformal parameterization algorithm on meshes in [9] for point clouds. An accurate approximation of the LB operator is achieved using the moving least square (MLS) method [30, 35, 36] together with a new Gaussian-type weight function. A key step of the parameterization algorithm in [9] for computing a quasi-conformal map is replaced by solving a Laplace equation on the complex plane, followed by an iterative scheme, the N-S reiteration. Also, the point distribution of the parameterization is enhanced by a balancing scheme for point clouds.

(ii) Our spherical parameterization method is efficient and robust to complex geometric structures. The algorithm completes within a few minutes and can handle highly convoluted point clouds.

(iii) Unlike most of the existing approaches, our algorithm specifically minimizes the conformality distortion of the parameterizations. Since the local geometry is preserved under the global spherical conformal parameterizations, we can create an almost-Delaunay triangulation on a point cloud by computing a Delaunay triangulation of its spherical conformal parameterization. The resulting triangulation on the point cloud preserves the regularity of that on the parameterization.

(iv) High-quality quad meshes can also be generated using our spherical conformal parameterization scheme.

(v) Unlike the conventional approaches for meshing, our method is topology preserving. The meshes produced using our proposed framework are guaranteed to be genus-0 closed meshes. No postprocessing is required.

(vi) Our method is stable under geometrical and topological noises on the input point clouds.

(vii) With the aid of our spherical conformal parameterization scheme, multilevel representations of genus-0 point clouds can be easily constructed.

3. Previous works. In this section, we describe some previous works closely related to our work.

Surface parameterization has been extensively studied by different research groups. For surveys on surface parameterization methods, see [15, 16, 23, 55]. In particular, conformal parameterizations have been well established on meshes. For the recent works, Lai et al. [28] proposed a folding-free spherical conformal mapping scheme by harmonic energy minimization. In [1], Aflalo, Kimmel, and Zibulevsky proved theoretical bounds of the conformal factor and proposed a method that minimizes the area distortion and avoids numerical errors of the conformal mapping. In [9], Choi, Lam, and Lui developed a fast algorithm for spherical conformal parameterizations of genus-0 closed meshes.

In the last few decades, numerous studies have been devoted to the parameterization of point cloud data. In [13, 14], Floater and Reimers proposed the meshless parameterization method for unorganized point sets. The point sets are parameterized onto a planar domain by solving a sparse linear system. In [63], Zwicker and Gotsman presented a parameterization approach for a genus-0 point cloud using a k-nearest neighborhood (K-NN) graph of the point cloud, followed by a spherical embedding method for planar graphs. In [3, 4, 5], Azariadis and Sapidis introduced the notion of dynamic base surfaces and suggested a parameterization scheme by orthogonally projecting a point cloud onto the dynamic base surface. Guo et al. [19] computed a global conformal parameterization of point-set surfaces, based on Riemann surface theory and Hodge theory. In [57], Tewari, Gotsman, and Gortler proposed a doubly periodic global parameterization of point cloud sampled from a closed surface of genus 1 to the plane, with the aid of discrete harmonic one-forms. Wang, Yuan, and Miao [58] suggested a parameterization method for genus-0 cloud data. A point cloud is first mapped onto its circumscribed sphere, then the sphere is mapped onto an octahedron and finally unfolded to a 2D image. In [61], Zhang et al. presented an as-rigid-as-possible parameterization approach for point cloud data. A point cloud with disk topology is mapped onto the plane by a local flattening step and a rigid alignment. In [35], Liang et al. constructed spherical conformal mappings of genus-0 point clouds by adapting the harmonic energy minimization algorithm in [28]. Meng et al. [46] proposed a neural network based method for point cloud parameterization. An adaptive sequential learning algorithm is applied to dynamically adjust the parameterization.

The use of parameterization of the point cloud is widespread in computer science and engineering. One of the major applications of point cloud parameterization is mesh generation. Instead of a convoluted point cloud, mesh reconstruction is usually completed on a simpler parameter domain. In [13, 14], Floater and Reimers applied their proposed parameterization scheme for meshing point clouds with disk topology. In [22], Hormann and Reimers extended the parameterization method in [14] for surface reconstruction of point clouds with spherical topology. In [63], Zwicker and Gotsmann used their proposed parameterization method for mesh reconstruction of genus-0 point clouds. Tewari, Gotsman, and Gortler [57] performed surface reconstruction using their proposed doubly periodic global parameterization. Li et al. [34] proposed a meshless quadrangulation scheme by global parameterization. The input point cloud is cut to be with disk topology and parameterized onto the plane for meshing.

Table 1
Several previous works on meshing point clouds using parameterization.

Methods	Topology	Parameter domain	Local/global parameterization?	Distortion to be minimized
Meshless parameterization [13, 14]	Disk topology	Plane	Global	/
Meshless parameterization for spherical topology [22]	Genus-0	Planes	Local	/
Spherical embedding [63]	Genus-0	Sphere	Global	Stretch
Discrete one-forms [57]	Genus-1	Planes	Local	/
As-rigid-as-possible meshless parameterization [61]	Disk topology	Plane	Global	ARAP
Meshless quadrangulation by global parameterization [34]	Unrestricted	Plane	Global	Gradient and principal fields

Zhang et al. [61] suggested a mesh reconstruction method of point cloud data by meshless denoising and their proposed parameterization scheme. Table 1 compares several previous works on meshing point clouds using parameterizations. The above previous works reflect the importance of parameterization in surface reconstruction of point cloud data.

Finding a conformal parameterization involves solving differential equations. In particular, for conformal parameterizations of point clouds, it is necessary to build a discrete analogue of the differential operators on point clouds. Numerous works on approximating differential operators on a point cloud have been reported. In [47], Nayroles, Touzot, and Villon described a diffuse approximation method for estimating the derivatives at a given set of points. In [6], Belkin and Niyogi established a theoretical foundation for the LB operator on point clouds. Belkin, Sun, and Wang [7] proposed the PCD Laplace operator for approximating the LB operator using an integral approximation. The MLS method [56, 31] is widely used for the approximation. A number of algorithms for the approximation of derivatives have been developed based on the MLS method [39, 32, 48, 8]. In [30], Lange and Polthier proposed a point-set analogue of the LB and shape operator using the MLS method. In [35, 36], Liang and others approximated the LB operator on point clouds by the MLS method with a special weighting function. In [27], Lai, Liang, and Zhao presented a local mesh approach for solving PDEs on point clouds. A local mesh structure is constructed at each point using local principal component analysis (PCA). Macdonald, Merriman, and Ruuth [45] computed reaction-diffusion processes on point clouds. In [42], Lozes, Elmoataz, and Lézoray proposed a method to solve PDEs on point clouds for image processing using partial difference operators on weighted graphs.

4. Mathematical background. In this section, we introduce some basic mathematical concepts closely related to our work. For more details, see [53, 54, 24].

4.1. Conformal maps. An altas of a manifold is said to be *conformal* if all of its transition maps are biholomorphic. A conformal structure is the maximal conformal altas, and a surface with a conformal structure is called a *Riemann surface.* Suppose $\mathcal{M}$, $\mathcal{N}$ are 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

$$(4.1) \qquad ds_{\mathcal{M}}^2 = \sum_{i,j} g_{ij} dx^i dx^j \quad \text{and} \quad ds_{\mathcal{N}}^2 = \sum_{i,j} \tilde{g}_{ij} dx^i dx^j,$$

where $g_{ij} = \langle \frac{\partial r_1}{\partial x^i}, \frac{\partial r_1}{\partial x^j} \rangle$, $\tilde{g}_{ij} = \langle \frac{\partial r_2}{\partial x^i}, \frac{\partial r_2}{\partial x^j} \rangle$. 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 $f^* ds_{\mathcal{N}}^2$ defined on $\mathcal{M}$, induced by f and $ds_{\mathcal{N}}^2$, is the metric

$$(4.2) \qquad f^* ds_{\mathcal{N}}^2 = \sum_{m,n} \left(\sum_{i,j} \tilde{g}_{ij} \left(f \left(x^1, x^2 \right) \right) \frac{\partial f^m}{\partial x^i} \frac{\partial f^n}{\partial x^j} \right) dx^m dx^n.$$

f is said to be *conformal* if there exists a positive scalar function $\lambda(x^1, x^2)$, called the *conformal factor*, such that $f^* ds_{\mathcal{N}}^2 = \lambda ds_{\mathcal{M}}^2$. An immediate consequence of the above is that every conformal map preserves angles and hence the local geometry of the surface.

4.2. Harmonic maps. By the uniformization theorem, every genus-0 closed surface is conformally equivalent to $\mathbb{S}^2$. Hence, it is natural to consider mappings between a genus-0 closed surface and the unit sphere. The *Dirichlet energy* for a map $f : \mathcal{M} \to \mathbb{S}^2$ is defined as

$$(4.3) \qquad E(f) = \int_{\mathcal{M}} |\nabla f|^2 dv_{\mathcal{M}}.$$

In the space of mappings, the critical points of $E(f)$ are called *harmonic mappings*. For genus-0 closed surfaces, conformal maps are equivalent to harmonic maps [24]. Hence, the problem of finding a conformal map $f : \mathcal{M} \to \mathbb{S}^2$ is equivalent to an energy minimization problem.

4.3. Quasi-conformal maps. Quasi-conformal maps are a generalization of conformal maps. Mathematically, $f : \mathbb{C} \to \mathbb{C}$ is a quasi-conformal map if it satisfies the Beltrami equation

$$(4.4) \qquad \frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z}$$

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

$$(4.5) \qquad \frac{\partial f}{\partial z} := \frac{1}{2} \left(\frac{\partial f}{\partial x} - i \frac{\partial f}{\partial y} \right) \quad \text{and} \quad \frac{\partial f}{\partial \bar{z}} := \frac{1}{2} \left(\frac{\partial f}{\partial x} + i \frac{\partial f}{\partial y} \right).$$

μ is called the *Beltrami coefficient* of the quasi-conformal map f. Note that the quasi-conformal map f is conformal around a small neighborhood of p if and only if $\mu(p) = 0$, as (4.4) becomes the Cauchy–Riemann equation in this situation.

Suppose $f : \Omega_1 \to \Omega_2$ and $g : \Omega_2 \to \Omega_3$ are quasi-conformal maps with the Beltrami coefficients μ_f and μ_g, respectively. Then, the Beltrami coefficient of the composition map $g \circ f : \Omega_1 \to \Omega_3$ is explicitly given by

$$(4.6) \qquad \mu_{g \circ f} = \frac{\mu_f + \frac{\overline{f_z}}{f_z}(\mu_g \circ f)}{1 + \frac{\overline{f_z}}{f_z} \overline{\mu_f}(\mu_g \circ f)}.$$

Quasi-conformal maps are also defined between two Riemann surfaces M and N. A *Beltrami differential* $\mu(z)\frac{\overline{dz}}{dz}$ on 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 $f : M \to N$ is called *quasi-conformal* associated with $\mu(z)\frac{\overline{dz}}{dz}$ if for any chart (U_α, ϕ_α) on M and any chart (U_β, ψ_β) on N, the mapping $f_{\alpha\beta} := \psi_\beta \circ f \circ f_\alpha^{-1}$ is quasi-conformal associated with $\mu_\alpha \frac{\overline{dz_\alpha}}{dz_\alpha}$. See [17] for more details of quasi-conformal maps.

4.4. Stereographic projection. In our work, we frequently make use of the stereographic projection. Mathematically, the *stereographic projection* is a conformal map $P_N : \mathbb{S}^2 \to \overline{\mathbb{C}}$ with

$$(4.7) \qquad P_N(x, y, z) = \frac{x}{1-z} + i\frac{y}{1-z}.$$

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

$$(4.8) \qquad P_N^{-1}(x + iy) = \left(\frac{2x}{1+x^2+y^2}, \frac{2y}{1+x^2+y^2}, \frac{-1+x^2+y^2}{1+x^2+y^2} \right).$$

Similarly, we define the *south-pole stereographic projection* $P_S : \mathbb{S}^2 \to \overline{\mathbb{C}}$ by

$$(4.9) \qquad P_S(x, y, z) = \frac{x}{1+z} + i\frac{y}{1+z}.$$

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

$$(4.10) \qquad P_S^{-1}(x + iy) = \left(\frac{2x}{1+x^2+y^2}, \frac{2y}{1+x^2+y^2}, \frac{1-x^2-y^2}{1+x^2+y^2} \right).$$

4.5. 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 construct a local coordinate system for P on each point z_s and approximate the derivatives. To achieve this, we 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 all neighboring points of z_s, denoted by $\mathcal{N}(z_s)$. Specifically, we apply the k-NN algorithm to define the neighborhood. The k-NN $\mathcal{N}^k(z_s)$ of z_s is a set with k distinct elements in P (including z_s) closest from z_s under the Euclidean 2-norm. In this work, we apply the KD-tree implementation by Lin [37] for the computation. We denote $\mathcal{N}^k(z_s) = \{z_s^1, z_s^2, \ldots, z_s^k\}$ with $z_s^1 = z_s$. Then, one common approach for constructing a local coordinate system is to define the normal vector as the z-axis, which is more convenient for further computation. There are various methods to obtain the tangent planes and the normal vectors for point clouds, such as the PCA method [21]. Using the PCA method 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 $\hat{z}_s^i = z_s^i - \langle z_s^i - z_s, e_s^3 \rangle e_s^3$, $i = 1, 2, \ldots, k$. Now we have 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 $i = 1, 2, \ldots, k$. Therefore, 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, $z_s^i = z_s + x_s^i e_s^1 + y_s^i e_s^2 + f_s(x_s^i, y_s^i)e_s^3$.

5. An overview of the fast spherical conformal parameterization algorithm for triangular meshes. In this section, we briefly describe the approach in [9] for computing a spherical conformal parameterization of a genus-0 closed triangular mesh M. This approach motivates our proposed parameterization scheme for genus-0 point clouds.

To compute a conformal mapping $f : M \to \mathbb{S}^2$, it suffices to solve (4.3). This can be achieved by solving the Laplace equation $\Delta^T f = 0$ subject to $\|f\| = 1$, where $\Delta^T f$ is the tangential component of Δf on the tangent plane of $\mathbb{S}^2$. This tangential approach was applied by Oberknapp and Polthier in [49]. Note that this problem is nonlinear because of the constraint $\|f\| = 1$. In [2, 20], Angenent and others linearize this problem by solving the equation on the complex plane,

$$(5.1) \qquad \qquad \Delta f = 0,$$

given three boundary constraints $f(a_i) = b_i$, where $a_i, b_i \in \mathbb{C}, i = 1, 2, 3$, such that the triangle $[a_1, a_2, a_3]$ and the triangle $[b_1, b_2, b_3]$ are with the same angle structures. Note that $\Delta^T f = \Delta f = 0$ since the target domain is now $\mathbb{C}$. As the nonlinear constraint $\|f\| = 1$ is removed, the above problem becomes linear and can be solved using the cotangent formula [50].

After solving (5.1), the inverse stereographic projection P_N^{-1} is applied for obtaining a spherical parameterization. However, unlike in the continuous case, the spherical parameterization in the discrete case is with large conformality distortion at the north pole of the sphere due to the discretization and the approximation errors. Hence, Choi, Lam, and Lui [9] proposed to apply the south-pole stereographic projection P_S to map the sphere to a planar domain $R \subset \mathbb{C}$. Note that the region with large distortion is the innermost region of R, while the outermost region of R has negligible distortion. Denote the above steps by a map $g : M \to R$. To correct the distortion of g, Choi, Lam, and Lui made use of the quasi-conformal theory.

Let $\mu_{g^{-1}}$ be the Beltrami coefficient of the map g^{-1}. Fixing the outermost region on R, Choi, Lam, and Lui [9] composed the map g with a quasi-conformal map $h : R \to \mathbb{S}^2$ with the associated Beltrami coefficient $\mu_h = \mu_{g^{-1}}$. Let $h = u + iv$ and $\mu_h = \rho + i\tau$. Specifically, by considering the Beltrami equation (4.4), each pair of the partial derivatives v_x, v_y and u_x, u_y can be expressed as linear combinations of the other [26],

$$(5.2) \qquad \begin{array}{ll} -v_y = \alpha_1 u_x + \alpha_2 u_y; \\ v_x = \alpha_2 u_x + \alpha_3 u_y, \end{array} \quad \text{and} \quad \begin{array}{ll} -u_y = \alpha_1 v_x + \alpha_2 v_y; \\ u_x = \alpha_2 v_x + \alpha_3 v_y, \end{array}$$

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 $\nabla \cdot \left(\begin{array}{c} -v_y \\ v_x \end{array} \right) = 0$ and $\nabla \cdot \left(\begin{array}{c} -u_y \\ u_x \end{array} \right) = 0$, the map h can be constructed by solving the following equations:

$$(5.3) \qquad \nabla \cdot \left(A \left(\begin{array}{c} u_x \\ u_y \end{array} \right) \right) = 0 \ \text{ and } \ \nabla \cdot \left(A \left(\begin{array}{c} v_x \\ v_y \end{array} \right) \right) = 0,$$

where $A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_3 \end{pmatrix}$. In the discrete case, the above elliptic PDEs (5.3) can be discretized into sparse symmetric positive definite linear systems as described in [43, 9]. In [25], Jones and Mahadevan derived the system (5.2) from the conjugate Beltrami equation $\frac{\partial h}{\partial \bar{z}} = \nu(z)\overline{\left(\frac{\partial h}{\partial z}\right)}$ and proposed an alternative approach for solving the system. Specifically, the authors considered minimizing the functional

$$(5.4) \qquad \frac{1}{2} \int_R \left(\alpha_1 |\nabla u|^2 + 2\alpha_2 \nabla u \cdot \nabla v + \alpha_3 |\nabla v|^2 \right) dS$$

using an Euler–Lagrange variational approach. Despite the different implementations, both of the two above-mentioned methods effectively solve for a quasi-conformal map on triangulated meshes. Then, by (4.6), the composition map $h \circ g : M \to \mathbb{S}^2$ is with the Beltrami coefficient $\mu_{h \circ g} = 0$ and hence $h \circ g$ is conformal. See [9] for more details.

Note that a key step above is the computation of the quasi-conformal map h for improving the conformality, which is guaranteed by the composition formula (4.6). However, the Beltrami coefficients in the above algorithm are approximated on the triangular faces of a mesh. Hence, the above algorithm cannot be directly applied for point clouds. Moreover, even if we can define the discrete Beltrami coefficients on point clouds, (4.6) may not hold anymore. Therefore, we need to replace this key step by a new method suitable for point clouds.

6. Meshing genus-0 point clouds using spherical conformal parameterization. In this section, we discuss our proposed framework for meshing genus-0 point clouds. The main steps involved include solving a series of Laplace equations on the complex plane for the spherical conformal parameterization of a genus-0 point cloud and creating a mesh structure with the aid of the global parameterization.

6.1. Approximation of the LB operator. In this subsection, we explain our approximation scheme for the LB operator in the Laplace equation (5.1) on a point cloud P by the MLS method. The MLS method is widely used for approximation [39, 32, 48, 8, 30, 35, 36]. In particular, Liang and others [35, 36] approximated the LB operator on point clouds using the MLS method with a special weight function. Our approximation scheme is built upon the method in [30, 35, 36]. In this work, we propose a new weight function to achieve a more accurate approximation of the LB operator.

First, we discuss our approximation method for the derivatives on the point cloud $P = \{z_1, z_2, \ldots, z_n\}$. To simplify the discussion, we only consider 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, $z_s^i = z_s + x_s^i e_s^1 + y_s^i e_s^2 + f_s(x_s^i, y_s^i) e_s^3$. Denote the derivatives of f_s along the e_s^1-direction and the e_s^2-direction by f_{sx} and f_{sy}, respectively. We 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 some coefficients to be determined. In our work, we use $\{1, x, y, x^2, xy, y^2\}$ as the basis of the space of all polynomials with second order or below, which means $m = 6$. We add a remark here that $m = 6$ is an appropriate choice for our approximation. Since second derivatives are considered in approximating the LB operator, polynomials with at least second order are needed. On the other hand, if we fit a polynomial with third order ($m = 10$) or higher, it will be too sensitive to noises and the approximation gets worse. Therefore, $m = 6$ is a suitable dimension for our approximation.

Table 2
Some common weighting functions for the MLS approximation.

Weight	Formula of $w(d)$
Constant weight	$w(d) = 1$
Exponential weight	$w(d) = \exp\left(-\dfrac{d^2}{h^2}\right)$
Inverse of squared distance weight	$w(d) = \dfrac{1}{d^2 + \epsilon^2}$
Wendland weight [59, 60]	$w(d) = \left(1 - \dfrac{d}{D}\right)^4 \left(\dfrac{4d}{D} + 1\right)$
Special weight [35]	$w(d) = \begin{cases} 1 & \text{if } d = 0 \\ \frac{1}{k} & \text{if } d \neq 0 \end{cases}$

In the approximation, we aim to minimize

$$(6.1) \qquad \sum_{i=1}^{n} w_i \left(\sum_{j=1}^{m} c_j f_s^j(x_i, y_i) - f_s(x_i, y_i) \right)^2,$$

where $w_i = w(\|z_i - z_s\|)$ for some weighting function $w : \mathbb{R} \to \mathbb{R}$. The weight function w significantly affects the accuracy and robustness of the approximation. Hence, one must carefully choose a suitable weight function. Table 2 lists some common weighting functions.

Note that the information provided by the data points near the center point z_s should be more reliable than that of the data points distant from z_s. The closer the data points are to z_i, the more reliable they are. Hence, it is natural to consider a smooth weight function which concentrates at z_s. This motivates us to use a weight function of the Gaussian type. As a remark, in [6, 7], Belkin and others used a Gaussian weight function in the form of $\exp\left(-\|x - y\|^2/4t\right)$ for integral approximation. In our MLS approximation, we propose another Gaussian-type weight function:

$$(6.2) \qquad \begin{cases} w_s = w(0) = 1, \\ w_i = w(\|z_i - z_s\|) = \dfrac{1}{k} \exp\left(-\dfrac{\sqrt{k}}{h^2} \|z_i - z_s\|^2\right) & \text{for all } i \neq s, \end{cases}$$

where h is the maximum distance from z_s in $\mathcal{N}^k(z_s)$. Numerical experiments are demonstrated in section 7 to support our proposed weight function with the specific factor $\sqrt{k}/h^2$ inside the exponent. It can be observed that our proposed weight results in more accurate approximations of the LB operator on point clouds.

With the proposed weight function, we now solve the minimization problem (6.1). Denote $f_{s,i}^j = f_s^j(x_i, y_i)$ and $f_{s,i} = f_s(x_i, y_i)$.

Let

$$\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}, \vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}, \vec{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}, \text{and} \vec{b} = \begin{pmatrix} f_{s,1} \\ f_{s,2} \\ \vdots \\ f_{s,n} \end{pmatrix}.$$

The minimization problem in (6.1) can be written as follows:

$$(6.3) \qquad \min_{c \in \mathbb{R}^n} \left\langle \vec{D}(\vec{A}c - \vec{b}), \vec{A}c - \vec{b} \right\rangle.$$

We can solve it using the least square method, namely, solving

$$(6.4) \qquad \vec{A}^T \vec{D} \vec{A} \vec{c} = \vec{A}^T \vec{D} \vec{b}.$$

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

$$(6.5) \qquad u = f_s(x, y) \approx \sum_{i=1}^{m} \hat{c}_i f_s^i(x, y).$$

Similarly, the coefficients $\hat{c}_i$ can be approximated. Let

$$\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}, \quad \vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}, \quad \vec{c} = \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_m \end{pmatrix}, \text{ and } \vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}.$$

We can find the coefficients $\hat{c}_i$ by solving the following least square problem:

$$(6.6) \qquad \vec{A}^T \vec{D} \vec{A} \vec{c} = \vec{A}^T \vec{D} \vec{u}.$$

Since we know the explicit formula of the derivatives of each f_s^i, we can compute the approximated derivatives of u, such as

$$(6.7) \qquad \frac{\partial u}{\partial x} = \sum_{i=1}^{m} \vec{c}_i \frac{\partial f_s^i}{\partial x} \quad \text{and} \quad \frac{\partial u}{\partial y} = \sum_{i=1}^{m} \vec{c}_i \frac{\partial f_s^i}{\partial y}.$$

Now, we are ready to introduce the construction of the LB operator of a smooth function u on $\mathcal{N}(z_s)$. For any smooth real-valued function u on the $\mathcal{N}(z)$, the LB operator of u is given by

$$(6.8) \qquad \Delta u(z) = \frac{1}{W} \sum_{i,j=1}^{2} \partial_i(g^{ij} W \partial_j(u(z))),$$

where z is a point in $\mathcal{N}(z)$, (g_{ij}) is the metric of the surface 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

$$(6.9) \quad (g_{ij}) = \begin{pmatrix} 1 + (f_s)_x^2 & (f_s)_x(f_s)_y \\ (f_s)_x(f_s)_y & 1 + (f_s)_y^2 \end{pmatrix} \quad \text{and} \quad (g^{ij}) = \frac{1}{W^2} \begin{pmatrix} 1 + (f_s)_y^2 & -(f_s)_x(f_s)_y \\ -(f_s)_x(f_s)_y & 1 + (f_s)_x^2 \end{pmatrix},$$

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

We use (6.7) to calculate the first order partial derivatives of f_s. Then, we proceed to compute $\Delta u(z_s)$. Since we have a closed form of Δu and the LB operator is a second order differential operator, by differentiating (6.8), we get

$$(6.10) \qquad \Delta u(z_s) = \alpha_1 \frac{\partial u}{\partial x}(z_s) + \alpha_2 \frac{\partial u}{\partial y}(z_s) + \alpha_3 \frac{\partial^2 u}{\partial x^2}(z_s) + \alpha_4 \frac{\partial^2 u}{\partial x \partial y}(z_s) + \alpha_5 \frac{\partial^2 u}{\partial y^2}(z_s),$$

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 scheme for the LB operator on point clouds. With this approximation, we are now ready to describe our proposed spherical conformal parameterization algorithm for genus-0 point clouds.

6.2. Spherical conformal parameterization of genus-0 point clouds. In this subsection, we introduce our proposed method for the spherical conformal parameterizations of genus-0 point clouds.

Given a point cloud P sampled from a genus-0 closed surface $\mathcal{M}$, our goal is to find a conformal map $\tilde{f} : P \to \mathbb{S}^2$ which effectively resembles the conformal map $f : \mathcal{M} \to \mathbb{S}^2$. By the previous section, we can approximate the LB operator Δ on P. Denote the approximated LB operator on the point cloud by Δ_{PC}. The approximation allows us to solve the Laplace equation (5.1) on point clouds for a map $\phi : P \to \mathbb{C}$. More specifically, we solve the equation

$$(6.11) \qquad\qquad\qquad\qquad \Delta_{PC}\phi = 0$$

subject to the constraints $\phi(a_i) = b_i$ for $i = 1, 2, 3$, where $a_i, b_i \in \mathbb{C}$. The choice of the three boundary points a_1, a_2, a_3 affects the conformality of the map ϕ. In the case of triangular meshes, a_1, a_2, a_3 are chosen to be the three vertices of the most regular triangle among all triangles on the input mesh [9]. Here, the regularity of a triangle $[a_1, a_2, a_3]$ is defined by

$$(6.12) \qquad \text{Regularity}[a_1, a_2, a_3] = \left| \alpha - \frac{\pi}{3} \right| + \left| \beta - \frac{\pi}{3} \right| + \left| \gamma - \frac{\pi}{3} \right|,$$

where α, β, and γ are the three angles in the triangle $[a_1, a_2, a_3]$. However, in the case of point clouds, we do not have the required connectivity information. Hence, we choose the three points a_1, a_2, a_3 in a different way.

Recall that in approximating the LB operator, it is necessary to find the k-nearest neighboring data points $z_s^1, z_s^2, \ldots, z_s^k$ for each point z_s on the point cloud P. We consider forming a triple using z_s and two other neighboring points z_s^i and z_s^j, where $i \neq j$. Different combinations of i and j result in different triples $[z_s, z_s^i, z_s^j]$. Then, we propose to choose the three boundary points a_1, a_2, a_3 in the constraint of (6.11) by considering

$$(6.13) \qquad\qquad\qquad \min_{s,i,j} \text{Regularity}[z_s, z_s^i, z_s^j]$$

among all combinations of s, i and j.

After solving (6.11) with our proposed boundary constraints, we apply the inverse stereographic projection P_N^{-1} on $\phi(P)$ to obtain a spherical point cloud. Recall that the conformality distortion around the north pole is large due to the approximation error in the stereographic projection. Note that the key step in the method in [9] for correcting the distortion via a

composition of quasi-conformal maps does not work for the case of point clouds. Now, we propose a new method to correct the conformality distortion by solely using the LB operator.

We begin with the south-pole stereographic projection P_S to project the spherical point cloud back onto the complex plane. Under the projection, the north pole of the sphere, which corresponds to the outermost region of $\phi(P) \subset \mathbb{C}$, is mapped to the innermost region on the complex plane. It follows that the outermost region is now with very low distortion while the innermost region is with large distortion. We use the outermost low-distortion data points as the boundary constraints and solve the Laplace equation $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \to \mathbb{C}$ again,

$$(6.14) \qquad\qquad\qquad\qquad\qquad \Delta_{PC}\psi = 0$$

subject to the boundary constraints $\psi(x) = x$ for all data points x in the outermost low-distortion region. The low-distortion boundary constraints provide us with a more accurate result in the inner part of the planar region. Finally, we apply the inverse south-pole stereographic projection P_S^{-1} and obtain a composition map

$$(6.15) \qquad\qquad\qquad\qquad \widetilde{f} = P_S^{-1} \circ \psi \circ P_S \circ P_N^{-1} \circ \phi.$$

This step effectively replaces the step in the mesh parameterization algorithm in [9] which involves computing a quasi-conformal map.

Altogether, by solving (6.11) and (6.14) and using a number of projections, we can obtain a conformal map $\widetilde{f} : P \to \mathbb{S}^2$. Note that the method in [9] is based on certain manipulations of Beltrami coefficients and quasi-conformal maps. In contrast, our new method only involves solving Laplace equations. The equivalence between the two approaches can be explained as follows.

In the first step, the conformality distortion of the spherical parameterization is due to the error in the stereographic projection. Then in the approach in [9], the entire initial parameterization result is used in (5.3) for computing a quasi-conformal map in order to cancel the distortion. The method is theoretically guaranteed by the composition formula (4.6) of quasi-conformal maps. In contrast, in our new approach, we only make use of the most accurate part in the initial parameterization result. More explicitly, we use the southernmost regions as the boundary constraints and compute the remaining part of the spherical parameterization again, with the aid of the LB operator. The replacement of the south-pole step in [9] by our new south-pole step can be justified by the following theorem.

Theorem 6.1. *Let $(S_1, \sigma|dz|^2)$ and $(S_2, \rho|dw|^2)$ be two Riemann surfaces, and μ is a prescribed Beltrami differential on S_1. Then, the map solved by (5.3) is a harmonic map between $(S_1, |dz + \mu d\bar{z}|^2)$ and $(S_2, \rho|dw|^2)$. Consequently, solving the Laplace equation (6.14) is equivalent to solving (5.3).*

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

$$(6.16) \qquad\qquad\qquad E_{harm}(h) = \int_{S_1} \rho(h(\zeta))(|h_\zeta|^2 + |h_{\bar{\zeta}}|^2)dxdy.$$

On the other hand, by definition of the Beltrami equation (4.4), the solution to (5.3) is the critical point of the following energy functional:

$$(6.17) \qquad E_{QC}(f) = \int_{S_1} \rho(f(z))(|f_{\bar{z}} - \mu f_z|^2) dx dy.$$

It is shown in [44] that the above two energy functionals have the same set of critical points. Hence, solving the generalized Laplace equation (5.3) for a quasi-conformal map is equivalent to solving the Laplace equation (6.14) for a harmonic map under the distorted metric. Then, the conformality of our approach is again guaranteed by the composition formula (4.6) of quasi-conformal maps. ∎

Therefore, in the continuous case, under suitable boundary conditions in (5.3) and (6.14), both of the methods are theoretically guaranteed for producing a conformal map.

However, in the discrete case, the two methods perform differently. For the case of triangular meshes, the Beltrami coefficients can be accurately computed and the composition formula (4.6) of quasi-conformal maps is accurate under the discretization. In this situation, the method in [9] can be effectively applied. Yet, for the case of point clouds, we only have an approximation of the LB operator but not the Beltrami coefficients, and there is no guarantee about the composition formula (4.6) of quasi-conformal maps. Consequently, it is more suitable to use our proposed method as it only involves solving the Laplace equation (6.14). However, since the accuracy of our proposed method depends on the boundary constraints in solving the Laplace equation (6.14), the boundary constraints obtained from the initial parameterization result may contain small errors and hence slightly affect the result in solving (6.14). Therefore, it is desirable to perform more iterations for obtaining a more accurate result.

It is noteworthy that in the parameterization algorithm in [9] for triangular meshes, no further steps are required after the second step. However, because of the above-mentioned issue about the boundary constraints in the Laplace equation (6.14), further iterations are necessary for enhancing the parameterization result. These are the N-S reiterations. In each N-S reiteration, two Laplace equations are solved again after the north-pole stereographic projection and the south-pole stereographic projection, respectively. For solving each Laplace equation, we fix the outermost $r\%$ points on $\mathbb{C}$ to ensure the existence of the solution.

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

$$(6.18) \qquad \Delta_{PC}\widetilde{\phi} = 0$$

with the boundary constraints $\widetilde{\phi}(x) = x$ for the outermost $r\%$ of the data points on $\mathbb{C}$. After obtaining $\widetilde{\phi}$, the inverse north-pole stereographic projection is again applied, followed by the south-pole stereographic projection. Then, we 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

$$(6.19) \qquad \Delta_{PC}\widetilde{\psi} = 0$$

Algorithm 1. Our proposed spherical conformal parameterization algorithm.

Input: A genus-0 point cloud P.

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

1 Approximate the LB operator on P and denote it by Δ_{PC};

2 Find the most regular triple of points by solving problem (6.13);

3 Obtain a map $\phi : P \to \overline{\mathbb{C}}$ by solving the Laplace equation (6.11);

4 Apply the inverse stereographic projection $P_N^{-1} : \overline{\mathbb{C}} \to \mathbb{S}^2$ on $\phi(P)$;

5 Apply the south-pole stereographic projection $P_S : \mathbb{S}^2 \to \overline{\mathbb{C}}$ on $(P_N^{-1} \circ \phi)(P)$;

6 Solve the Laplace equation (6.14) for $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \to \mathbb{C}$;

7 Apply the inverse south-pole stereographic projection P_S^{-1} and denote the overall composition of the maps by $f = P_S^{-1} \circ \psi \circ P_S \circ P_N^{-1} \circ \phi$;

8 **repeat**

9   Update $\widetilde{f}$ by f;

10   Solve the Laplace equation (6.18) for $\widetilde{\phi} : (P_N \circ \widetilde{f})(P) \to \mathbb{C}$;

11   Solve the Laplace equation (6.19) for $\widetilde{\psi} : (P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f})(P) \to \mathbb{C}$;

12   Update f by $P_S^{-1} \circ \widetilde{\psi} \circ P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f}$;

13 **until** $mean(\|f(p_i) - \widetilde{f}(p_i)\|^2) < \epsilon$;

---

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

$$(6.20) \qquad P_S^{-1} \circ \widetilde{\psi} \circ P_S \circ P_N^{-1} \circ \widetilde{\phi} \circ P_N \circ \widetilde{f}.$$

We check whether the above updated parameterization result is close to the previous parameterization result $\widetilde{f}$. If yes, then the parameterization is stable and we complete the algorithm. If no, we apply another N-S reiteration on the updated parameterization point by repeating the procedures and so on. In practice, we choose $r = 10$. Our proposed spherical conformal parameterization scheme is outlined in Algorithm 1.

To explain the motivation of our proposed N-S reiteration scheme, we define the N-S Dirichlet energy by

$$(6.21) \qquad \tilde{E}(f) = \frac{1}{2} \left(E(P_N(f)) + E(P_S(f)) \right),$$

where $E(f)$ is the Dirichlet energy. It follows that the minimum of $\tilde{E}$ is attained 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 f, we can consider minimizing the N-S Dirichlet energy $\tilde{E}(f)$. More specifically, we aim to minimize both $E(P_N(f))$ and $E(P_S(f))$. Note that these two energies are respectively minimized by solving the Laplace equations (6.18) and (6.19) in our proposed N-S reiteration. Introducing the N-S reiteration for minimizing the energies is advantageous for two reasons. First, it linearizes the computation as we only need to solve linear systems on $\mathbb{C}$. Second, it avoids the error induced by the

stereographic projection as we consider both the north-pole step and the south-pole step in each reiteration.

Note that theoretically we only need to take away the two points of infinity, and at least three points on $\mathbb{C}$ are required to be fixed to guarantee the existence of the solution of the two Laplace equations. However, in terms of the numerical computations, the large matrix equations may be ill-posed if only three points are fixed as boundary constraints. Therefore, we consider fixing the outermost $r\%$ of data points in solving (6.18) and (6.19). Nevertheless, fixing these extra points may not affect the accuracy of the solution too much. This can be explained with the aid of the Beltrami holomorphic flow [17] as follows.

Theorem 6.2 (Beltrami holomorphic flow on $\overline{\mathbb{C}}$ [17]). *There is a one-to-one correspondence between the set of quasi-conformal diffeomorphisms of $\overline{\mathbb{C}}$ that fix the points $0, 1$, and ∞ and the set of smooth complex-valued functions μ on $\overline{\mathbb{C}}$ with $\|\mu\|_\infty = k < 1$. Furthermore, the solution f^μ to the Beltrami equation (4.4) depends holomorphically on μ. Let $\{\mu(t)\}$ be a family of Beltrami coefficients depending on a real or complex parameter t. Suppose also that $\mu(t)$ can be written in the form*

$$(6.22) \qquad \mu(t)(z) = \mu(z) + t\nu(z) + t\epsilon(t)(z)$$

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 $w \in \mathbb{C}$,

$$(6.23) \qquad f^{\mu(t)}(w) = f^\mu(w) + tV(f^\mu, \nu)(w) + o(|t|)$$

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

$$(6.24) \qquad V(f^\mu, \nu)(w) = -\frac{f^\mu(w)(f^\mu(w) - 1)}{\pi} \int_{\mathbb{C}} \frac{\nu(z)((f^\mu)_z(z))^2}{f^\mu(z)(f^\mu(z) - 1)(f^\mu(z) - f^\mu(w))} dxdy.$$

In our case, as the conformality distortion of the outermost region is negligible, ν is compactly supported around origin. Hence, it can be deduced from (6.24) in the above theorem that the data points located farther away from the origin will be associated with a smaller flow V, as the denominator in the integral becomes larger. Therefore, in each iteration, the outermost points will remain almost unchanged, while the innermost points (which have the largest conformality distortion) will be adjusted and improved. In other words, fixing more outermost points for the numerical stability does not affect the solution much. Numerical experiments are presented in section 7 to verify the convergence of our N-S reiteration scheme. Intuitively, the boundary constraints in (6.18) and (6.19) are adjusted to the positions associated with a conformal map by the iterations. They are observed to eventually stabilize and hence we obtain the desired conformal map by solving (6.18) and (6.19) with these boundary constraints.

Finally, we make an important remark about our proposed spherical conformal parameterization algorithm for genus-0 point clouds. In addition to genus-0 point clouds, our proposed algorithm also efficiently works on genus-0 triangular meshes. Note that for triangular meshes, the LB operator can be easily constructed by computing the cotangent weights on the given mesh structures. Also, solving Laplace equations on the complex plane only

requires solving sparse linear systems. Hence, our proposed algorithm can serve as an alternative linear formulation for computing spherical conformal parameterizations of genus-0 closed triangular meshes.

6.3. Improving the distribution of the spherical parameterization. It is obvious that spherical conformal parameterizations are unique only up to Möbius transformations. Although the conformality does not change under the Möbius transformations, the distribution of the points on the sphere does. The distribution is crucial for meshing. Hence, it is desirable to obtain an even distribution of the points on the sphere.

In the spherical conformal parameterization algorithm for triangular meshes [9], Choi, Lam, and Lui proved the following theorem.

Theorem 6.3 (see [9, p. 75]). *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.*

With this theorem, Choi, Lam, and Lui achieved an even distribution of a spherical parameterization mesh by applying the stereographic projection on the sphere and then considering the outermost triangle T and the innermost triangle t on the complex plane. They scaled the planar domain by a factor so that T and t are with the same perimeters on the sphere, under the inverse stereographic projection.

In our case, the above idea does not work as we do not have any information about the connectivity of the point clouds. However, we can extend Theorem 6.3 for point clouds by considering two sets of points. The extension is as follows.

Theorem 6.4. *Let $\{u_i\}_{i=0}^m$ and $\{v_j\}_{j=0}^n$ be two sets of points on $\mathbb{C}$. Then*

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

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

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

Proof. We prove the theorem using the approach in [9]. Note that for any $z = x + iy$, we have

$$P_S(P_N^{-1}(z)) = P_S(P_N^{-1}(x + iy))$$

(6.25)
$$= \frac{-\frac{2x}{1+x^2+y^2}}{1 + \frac{-1+x^2+y^2}{1+x^2+y^2}} + i\frac{\frac{2y}{1+x^2+y^2}}{1 + \frac{-1+x^2+y^2}{1+x^2+y^2}}$$

$$= \frac{-x}{x^2+y^2} + i\frac{y}{x^2+y^2} = \frac{-\mathrm{Re}(z)}{|z|^2} + i\frac{\mathrm{Im}(z)}{|z|^2}.$$

Hence, for any scaling factor $\lambda \neq 0$, we have

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

Therefore, the product is an invariance. ■

To apply this theorem for obtaining an even distribution of our spherical parameterization result, we propose to use the average distance between the poles on the unit sphere and their k-NN neighborhoods. More specifically, suppose v_N and v_S are the northernmost point and the southernmost point on the spherical parameterization result $f(P)$ obtained by Algorithm 1, respectively. By the north-pole stereographic projection P_N, v_N is mapped to the point x_N on the complex plane. On the other hand, by the south-pole stereographic projection P_S, v_S is mapped to the point x_S on the complex plane. Denote the average distances of x_N and x_S to their k-NN neighborhood on their corresponding planar domain by d_N and d_S, respectively; d_N and d_S are explicitly given by

$$
(6.27) \qquad d_p = \operatorname{mean}(\{|P_N(f(z)) - x_N| : z \in \mathcal{N}^k(f^{-1}(v_N))\})
$$

and

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

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

$$
(6.29) \qquad \lambda = \frac{\sqrt{d_p \times d_s}}{d_p}.
$$

Now, denote the two updated average distances by $\tilde{d}_p$ and $\tilde{d}_s$. It follows that

$$
(6.30) \qquad \tilde{d}_p = \lambda d_p = \frac{\sqrt{d_p \times d_s}}{d_p} \times d_p = \sqrt{d_p \times d_s}.
$$

Also, by Theorem 6.4, we have

$$
(6.31) \qquad \tilde{d}_p \times \tilde{d}_s = d_p \times d_s.
$$

Algorithm 2. 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 north-pole stereographic projection P_N on $f(P)$;

2 Denote the northernmost and the southernmost points of $f(P)$ by v_N and v_S.

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 north-pole stereographic projection P_N^{-1} on $\lambda(P_N(f(P)))$;

Therefore,

$$\tilde{d}_s = d_p \times d_s \times \frac{1}{\tilde{d}_p} = \sqrt{d_p \times d_s}. \tag{6.32}$$

In other words, the two updated average distances $\tilde{d}_p$ and $\tilde{d}_s$ defined on the new spherical parameterization result $P_N^{-1}(\lambda(P_N(f(P))))$ are equal. This indicates that the distribution of the points at the two poles is balanced. Hence, Algorithm 1 together with the described balancing scheme provide us with a spherical conformal parameterization with an even distribution. Our balancing scheme is summarized in Algorithm 2.

6.4. Meshing using spherical conformal parameterization. In this subsection, we present our meshing framework for genus-0 point clouds. Directly triangulating a point cloud is difficult because of its complicated geometry. However, with the aid of the spherical conformal parameterization of point clouds, the difficulty is significantly alleviated. Instead of triangulating a point cloud, we triangulate the unit sphere obtained by our spherical conformal parameterization algorithm. Algorithms for triangulating a spherical point cloud are well-established. In particular, the spherical Delaunay triangulation method, which computes a Delaunay triangulation on the unit sphere, is the most suitable method for our purpose.

Delaunay triangulations are widely used in computer graphics because of their good triangle quality. More specifically, Delaunay triangulations are advantageous as they maximize the minimum angle in every triangle in the triangulations and hence avoid skinny triangles. With this important property, the triangulations generated by this method are more regular than the common triangulation methods.

By applying the spherical Delaunay triangulation method on the spherical conformal parameterization of a genus-0 point cloud, we obtain a nice triangulation on the spherical point cloud. Since the points on the original point cloud and those obtained by the spherical conformal parameterization have a 1-1 correspondence, the triangulation on the spherical point cloud naturally induces a triangulation on the original point cloud. It is noteworthy that since the parameterization is conformal, the angles of the new triangulation on the original point cloud are well preserved. In other words, the regularity of the triangulation defined on the original point cloud closely resembles that of the spherical Delaunay triangulation. Moreover, the meshing result is guaranteed to be a genus-0 closed triangular mesh because of

---

Algorithm 3. Our proposed meshing framework for genus-0 point clouds.

Input: A genus-0 point cloud P.

Output: A triangular mesh $M = (P, T)$ where T is a triangulation of P.

1 Apply Algorithm 1 and Algorithm 2 to obtain a spherical conformal parameterization $f : P \to \mathbb{S}^2$;

2 Compute a triangulation T on $f(P)$ using the spherical Delaunay algorithm;

3 Use T to form a triangular mesh $M = (P, T)$;

---

the spherical Delaunay method. This completes our goal of meshing a genus-0 point cloud. Our meshing framework is described in Algorithm 3.

Before ending this section, we make an important remark about a possible extension of our proposed framework. In fact, our proposed parameterization and meshing scheme can be possibly extended for point clouds with disk topology. In this case, we can first extend the double covering technique [18, 10] to turn a point cloud with disk topology into a genus-0 point cloud. More specifically, given a point cloud P of a simply connected open surface $\mathcal{M}$, we turn P into a point cloud $\widetilde{P}$ with spherical topology and approximate the derivatives on it by the following steps:

Step 1: Approximate the derivatives on P using the k-NN algorithm and the MLS method.
Step 2: Duplicate P and denote the copy of it by P'.
Step 3: Define the derivatives on P' using the results in Step 1, with reversed orientations.
Step 4: Identify the boundary points of P, P' and obtain a genus-0 point cloud $\widetilde{P}$.
Step 5: Create the LB operator for $\widetilde{P}$ using the derivatives on P and P'.

Then, we can apply the above-mentioned spherical conformal parameterization algorithm on $\widetilde{P}$ to obtain a spherical point cloud. After that, by applying the stereographic projection on the southern hemisphere, we obtain a planar conformal parameterization of P. Finally, we can easily compute a Delaunay triangulation on the planar parameter domain. Since both the parameterization algorithm and the stereographic projection produce conformal results, this triangulation on the planar domain accurately induces a regular mesh structure on P. This completes the task of meshing a point cloud with disk topology.

7. Experimental results. In this section, we demonstrate the effectiveness of our proposed framework for meshing genus-0 point clouds using spherical conformal parameterization. In the following, we assess the performance of our proposed framework in different aspects. The datasets used in the experiments are freely adapted from the AIM@SHAPE Shape Repository [64], the Stanford 3D Scanning Repository [65], and the RGB-D Scenes Dataset v.2 [29]. The mentioned algorithms are implemented in MATLAB. The sparse linear systems for the Laplace equations are solved using the built-in backslash operator (\) in MATLAB. All experiments are performed on a PC with an Intel Core i5-3470 CPU @3.20 GHz processor and 8.00 GB RAM.

7.1. The performance of our approximation of the LB operator. In this work, we apply the MLS method with a new weight function for approximating the LB operator. It is natural to ask whether our proposed weight function produces better results. It is also necessary to

compare other approximation approaches such as the local mesh method to justify our choice. In this subsection, we compare the numerical accuracy of the local mesh method and the MLS method with several weighting functions for approximating the LB operator on point clouds. More specifically, we compare the performance of the following methods:

1. The local mesh method [27],
2. The MLS method with the Wendland weight in [59, 60],
3. The MLS method with the Gaussian weight in [6, 7],
4. The MLS method with the special weight in [35], and
5. The MLS method with our proposed weight function.

Experiments are carried out for assessing the numerical accuracies of the above-mentioned approaches. Figure 1 shows the setups in two of the experiments. In each experiment, we first generate a point cloud on the unit disk. This serves as the ground truth in our analysis. Then, we transform the point cloud using a conformal map with an explicit formula. We apply the mentioned approximation schemes for approximating the LB operator on the transformed point cloud. Then, we solve the Laplace equation with the circular boundary constraints on the original unit disk. Theoretically, the result obtained by the disk harmonic map should be exactly the same as the original point cloud, as the transformation is given by a conformal map
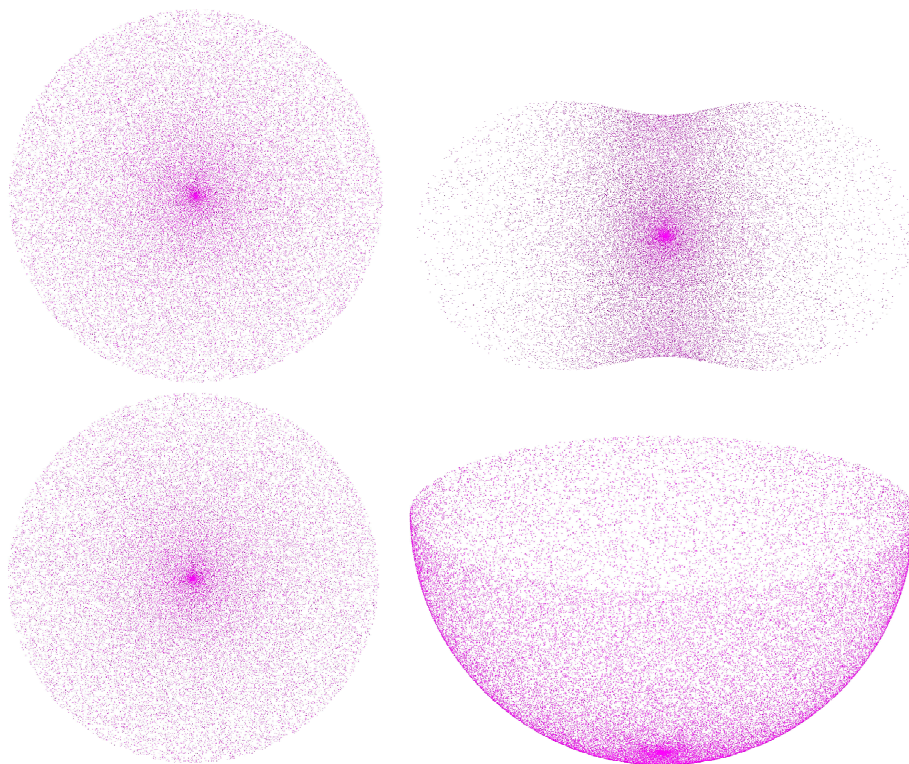

Figure 1. *Two experiments for assessing the approximation accuracies of the LB operator. In each experiment, we generate a point cloud on the unit disk and transform it using a conformal map with an explicit formula. We then approximate the LB operator on the transformed point cloud and solve the Laplace equation back onto the unit disk. Top: the first experiment. Bottom: the second experiment.*

Table 3

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

Method	Maximum position error	Average position error
Local mesh method [27]	1.3427	0.0179
MLS with the Wendland weight in [59, 60]	3.3074	0.1696
MLS with the Gaussian weight in [6, 7]	0.5697	0.0114
MLS with the special weight in [35]	0.0427	0.0006
MLS with our proposed weight	0.0245	0.0004

Method	Maximum position error	Average position error
Local mesh method [27]	1.5148	0.0271
MLS with the Wendland weight in [59, 60]	2.0082	0.0803
MLS with the Gaussian weight in [6, 7]	1.5460	0.0925
MLS with the special weight in [35]	0.0110	0.0001
MLS with our proposed weight	0.0103	0.0002

with an explicit formula. In other words, the ideal position error between the disk harmonic map and the original point cloud should be 0. By measuring the maximum and average position error between the pairs of points, we can evaluate the accuracy of the aforementioned approximation schemes for approximating the LB operator.

Table 3 illustrates the approximation error of different approaches in the two experiments. It is noteworthy that in both experiments the MLS method with our proposed weight function produces approximations which are much more accurate than those produced by the local mesh method and the MLS method with the Wendland weight [59, 60] and the Gaussian weight in [6, 7]. With similar and negligible average position errors, our proposed scheme reduces the maximum position errors by about 25% on average when compared with the MLS method with the special weight [35]. The comparisons reflect the advantage of our proposed method for approximating the LB operator.

7.2. Performance of our proposed spherical conformal parameterization. After demonstrating the advantage of our approximation scheme for the LB operator, we investigate the performance of our proposed spherical conformal parameterization algorithm for genus-0 point clouds. Figures 2 and 3 show the results of parameterizing a lion point cloud and a bulldog point cloud using our proposed parameterization method. Two more convoluted examples are shown in Figure 4. The experiments demonstrate the effectiveness of our proposed algorithm for convoluted point cloud data.

Moreover, with the aid of the spherical conformal parameterization, we can create a Delaunay triangulation on the spherical parameterization result by the spherical Delaunay algorithm and define an induced triangulation on the input point cloud. Using the mesh structures, we can measure the angle differences of the two meshed point clouds and hence effectively evaluate the conformality of our parameterization scheme. In particular, we define the conformality distortion of the parameterization by the angular distortion between the two meshes. The angle difference provides an accurate measurement of the conformality distortion of the parameterizations. It can be easily observed in Figures 2 and 3 that the histograms of the angle
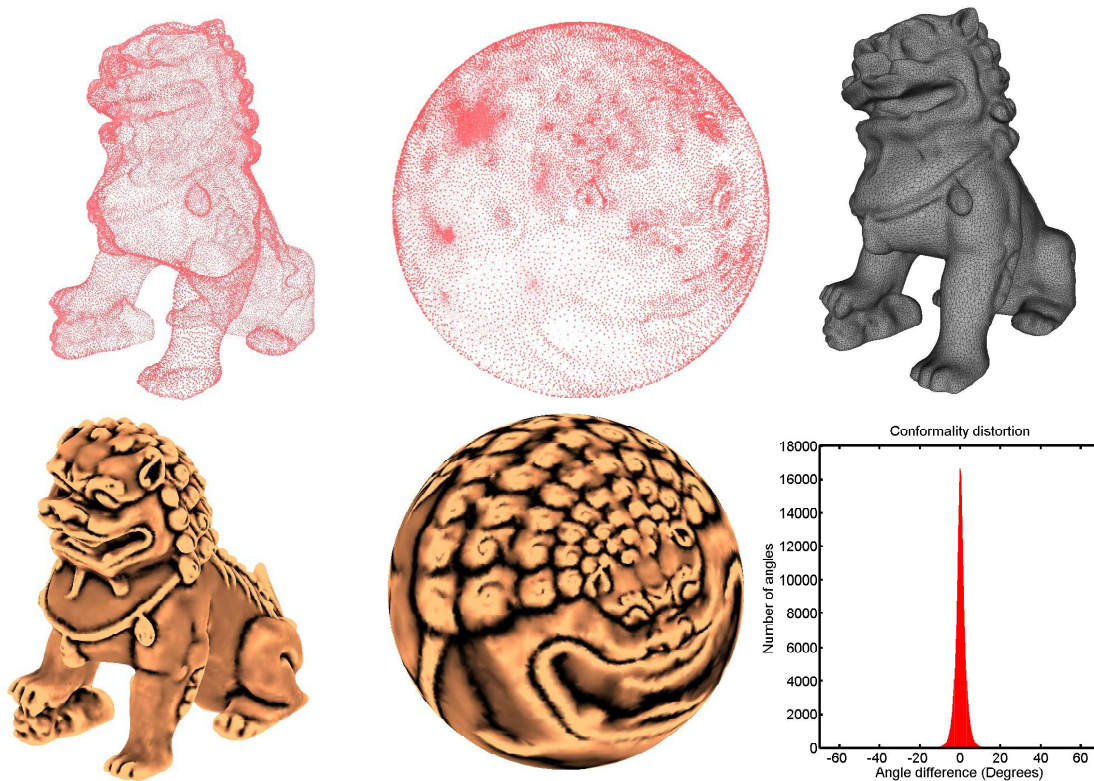
Figure 2. *Parameterizing a lion point cloud. Top left: A lion point cloud. Top middle: The spherical conformal parameterization of the lion point cloud. Top right: A triangulation created by our method. Bottom left and middle: The triangulated point cloud and the spherical parameterization result colored with the approximated mean curvature at each vertex. Bottom right: The conformality distortion of the parameterization based on the triangulation.*

differences highly concentrate at 0. In addition, for better visualizations of the spherical conformal parameterization results, we color the surfaces by the approximated mean curvature on the source surfaces. It can be observed from the colored figures that the local geometries of the point clouds are well preserved under our proposed spherical conformal parameterization algorithm. Figure 5 shows the difference plots of several experiments. The plots demonstrate the convergence of our N-S reiteration scheme.

We compare our proposed spherical conformal parameterization method for genus-0 point clouds with the spherical embedding method proposed by Zwicker and Gotsman [63] and the global conformal map [35]. In our experiment, $k = 25$ nearest neighbors of every point are used for approximating the LB operator in Algorithm 1. The stopping threshold for the N-S reiteration is set to be $\epsilon = 0.0001$. Table 4 summarizes the computational time and the conformality distortion of the three schemes. It is noteworthy that our proposed method produces spherical conformal parameterizations with better conformality. The better conformality obtained by our method is attributed to the south-pole step in our algorithm, which is conceptually equivalent to a composition of quasi-conformal maps. With this specific step, our method can hence further reduce the conformality distortion of the parameterization.
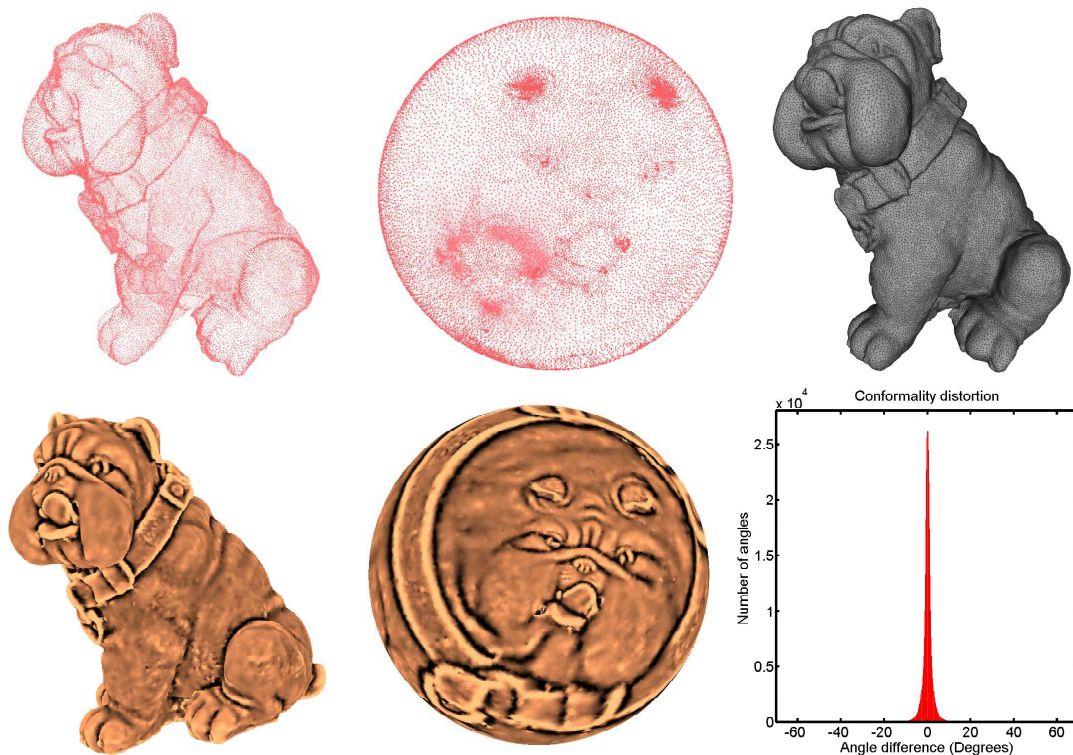
Figure 3. *Parameterizing a bulldog point cloud. Top left: A bulldog point cloud. Top middle: The spherical conformal parameterization of the bulldog point cloud. Top right: A triangulation created by our method. Bottom left and middle: The triangulated point cloud and the spherical parameterization result colored with the approximated mean curvature at each vertex. Bottom right: The conformality distortion of the parameterization based on the triangulation.*

Also, note that the global conformal map [35] fails for a few examples, which may be due to the uneven distribution of points in those examples. Moreover, our method is more efficient than the algorithms in [63] and [35]. The above results indicate that our parameterization algorithm preserves the local geometry of the point clouds very well. Last, we make a remark that the most time-consuming part in our algorithm is the MLS approximation of the LB operator. Therefore, as the size of the input point cloud increases, the computational time for the MLS approximation of the LB operator will outweigh the time for the parameterization steps in our algorithm. This possibly diminishes the advantage in the overall computational time of our algorithm over the algorithms in [63] and [35].

7.3. Performance of our meshing scheme. As mentioned in the last subsection, we generate mesh structures on genus-0 point clouds by building Delaunay triangulations on their spherical conformal parameterizations. Our meshing scheme has two important advantages. First, the regularity of the triangulations generated is guaranteed by the preservation of the 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
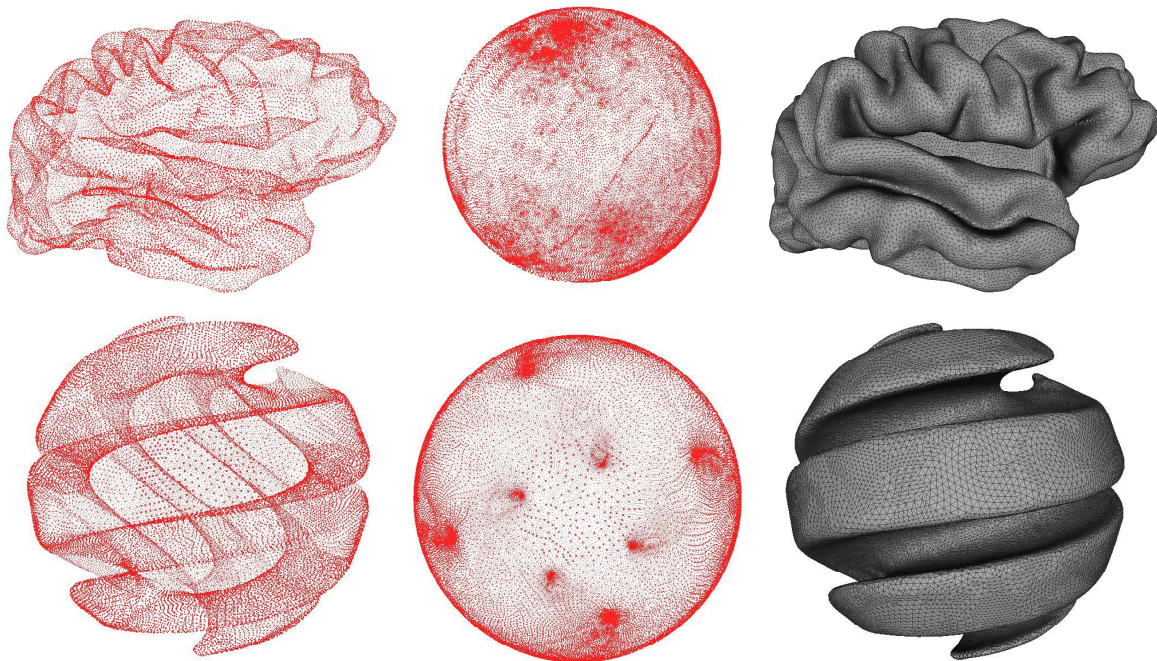
Figure 4. *Parameterizing two convoluted point clouds of a human brain and a spiral. Left: The input point clouds. Middle: The spherical conformal parameterizations obtained by our proposed algorithm. Right: The triangulations created by our method.*
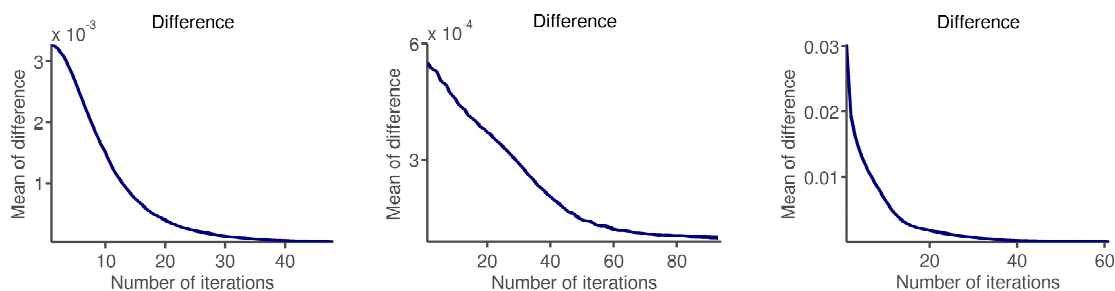

Figure 5. *Several plots of the difference mean $(\|f(p_i) - \widetilde{f}(p_i)\|^2)$ with the number of iterations in our N-S reiteration scheme. Left: Cereal box. Middle: Hippocampus. Right: Bulldog. The plots demonstrate the convergence of our proposed algorithm.*

and almost-Delaunay triangulation on the original point clouds. Besides Figures 2, 3, and 4, some more examples of triangulations created by our approach are shown in Figure 6. It can be observed that our meshing method can handle point clouds with different geometry. High quality triangulations can be created even with the presence of sharp, nonconvex, and convoluted regions on the input point clouds. Second, unlike most of the existing meshing methods, the meshes produced by our proposed scheme are guaranteed to be genus-0 closed meshes. No holes or unwanted boundaries will be present in our meshing result. Hence, postprocessing steps are not required in our meshing scheme.

Table 4

Performances of three spherical conformal parameterization methods for genus-0 point clouds. To quantitatively evaluate the conformality of the parameterization, we build a mesh structure on the spherical parameterization using the spherical Delaunay method and then create an induced mesh structure on the original point cloud. The conformality distortion (denoted by δ) of the parameterization is assessed using the angle difference (in degrees) between an angle on a meshed point cloud and the mapped angle on the meshed spherical parameterization result.

| Point clouds	Number of points	Performance	Our proposed method	Spherical embedding [63]	Global conformal map [35]
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	Fail
		Mean($|\delta|$)	0.6523	12.3573	
		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	Fail
		Mean($|\delta|$)	1.4167	23.2354	
		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
Fossil	305002	Time (s)	1690.3412	1793.5970	2158.9806
		Mean($|\delta|$)	1.0008	18.9758	1.0305
		SD($|\delta|$)	1.4658	21.3011	1.4782
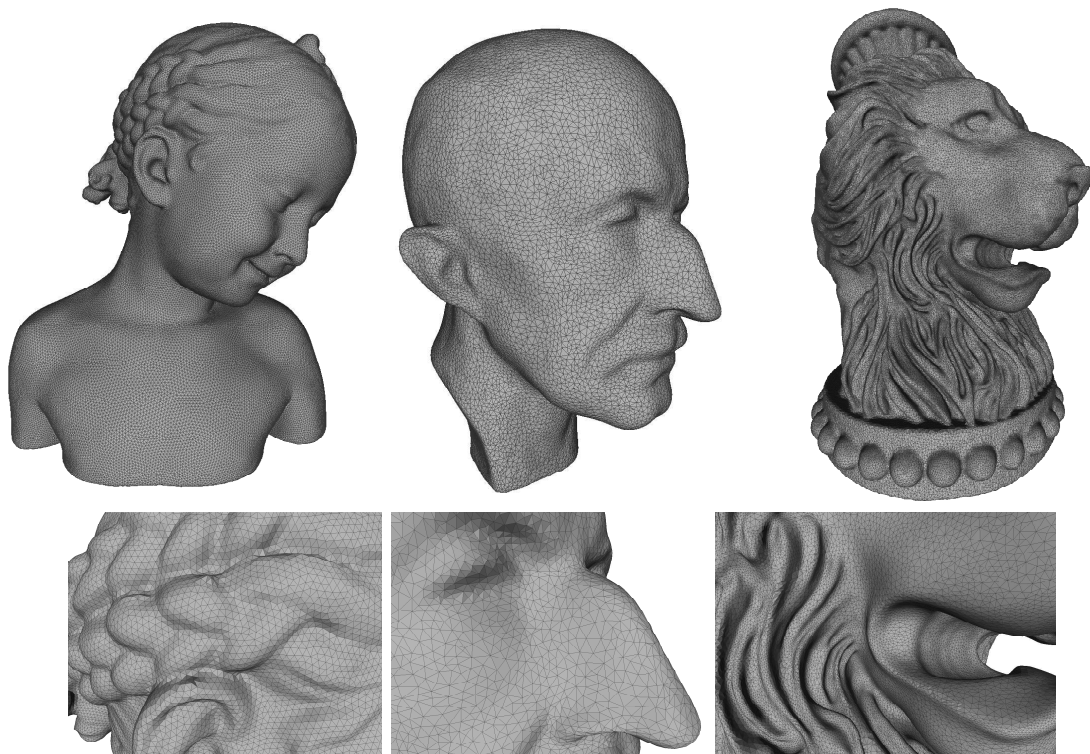
Figure 6. *Meshes generated by our proposed method and a zoom-in of them. The regularity of the triangulations is attributed to our spherical conformal parameterization and the spherical Delaunay method.*

We compare our meshing method with three existing meshing approaches. As an example of parameterization-based approaches, Zwicker and Gotsman [63] generate triangulations for a genus-0 point cloud with the aid of the spherical embedding algorithm and the spherical Delaunay triangulation method. On the other hand, two typical methods for meshing without using parameterizations are the marching cubes algorithm [41] and the Tight Cocone algorithm [11]. Figure 7 provides a comparison between our method and the three mentioned approaches. It can be observed that our meshing scheme and the Tight Cocone algorithm [11] produce high-quality triangulations, while the triangulations produced by the approaches in [63] and [41] consist of certain sharp and irregular triangles. Also, the result by the marching cubes algorithm contains holes, while our method is topology preserving. Therefore, unlike the marching cubes algorithm, no further postprocessing is needed in our meshing scheme.

To quantitatively assess the "almost-Delaunay" property of our meshing results induced by the spherical Delaunay triangulations, we recall that a Delaunay triangulation satisfies the opposite angle sum property: For every edge in a Delaunay triangulation, the two angles α and β opposite to the edge satisfy

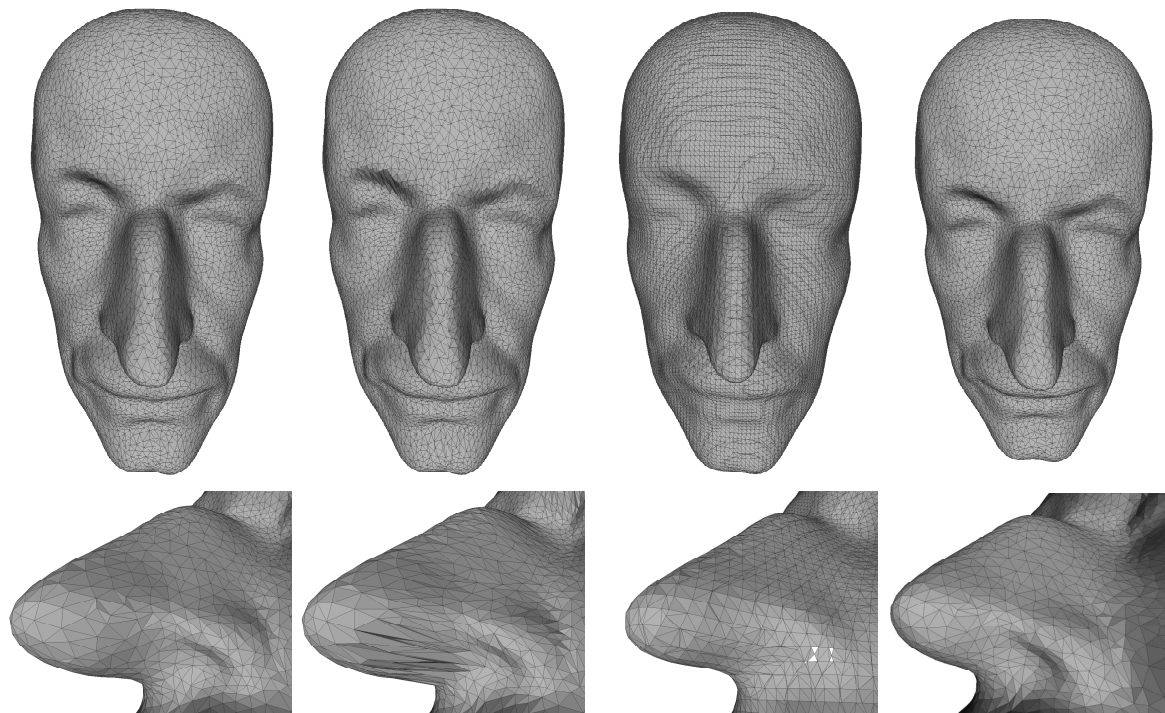$$(7.1) \qquad\qquad \alpha + \beta \leq \pi.$$

Figure 7. *A comparison between our meshing scheme and other approaches. A front view of the triangulated point cloud and a zoom-in of the nose are shown for each method. Left to right: Our meshing result, the method in* [63], *the marching cubes algorithm* [41], *and the Tight Cocone algorithm* [11].

For the above-mentioned meshing approaches, we define the *Delaunay ratio* of the resulting triangulation by

$$(7.2) \qquad \frac{\text{\# of edges with opposite angles } \alpha, \beta \text{ s.t. } \alpha + \beta \leq \pi}{\text{Total \# of edges in the triangulation}}.$$

A higher Delaunay ratio indicates that the triangulation is closer to a perfect Delaunay triangulation. The Delaunay ratios by different meshing algorithms are presented in Table 5. Because of the high accuracy of our spherical conformal parameterizations, our proposed scheme achieves significantly better triangulation results when compared with the spherical embedding algorithm [63] and the marching cubes algorithm [41]. Also, our results are comparable to or even slightly better than those of the Tight Cocone algorithm [11]. The comparisons demonstrate the advantages of our proposed meshing scheme. A further comparison between our method and the Tight Cocone algorithm [11] is given in the following subsection.

In addition, we can generate quadrangulations of point clouds with the aid of the spherical conformal parameterization. Two examples of the quad meshes generated by our method are given in Figure 8. To create quad meshes of point clouds, we make use of a standard spherical quad mesh and our spherical conformal parameterization results. With the aid of the spherical conformal parameterizations, we can interpolate the standard quad mesh onto the input point

Table 5
The Delaunay ratios of different meshing approaches. 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 that exactly equals 1 indicates that the triangulation is Delaunay.

Point clouds	Our proposed method	Spherical embedding [63]	Marching cubes [41]	Tight Cocone [11]
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
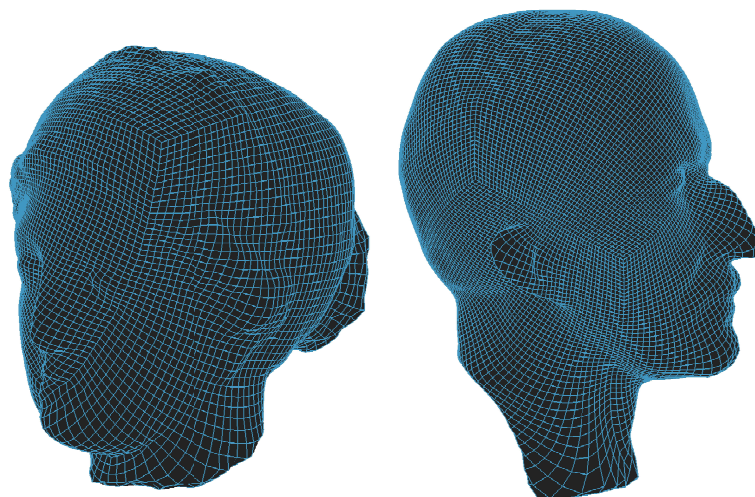
Figure 8. *Quad mesh generation on point clouds using our proposed method.*

clouds and thus generate quad mesh representations. Because of the conformality of our parameterization scheme, the resulting quad meshes are of high quality. Also, the meshes are guaranteed to be topology preserving.

Before ending this subsection, we demonstrate the significance of our proposed balancing scheme in the spherical conformal parameterization. The redistribution is vital for the meshing quality. Figure 9 shows the meshing results with and without the redistribution scheme. It can be easily observed that if the spherical parameterization of a genus-0 point cloud is unbalanced, then on the mesh generated by interpolation with the aid of the spherical conformal parameterization, most of the vertices will be concentrated at one small region of
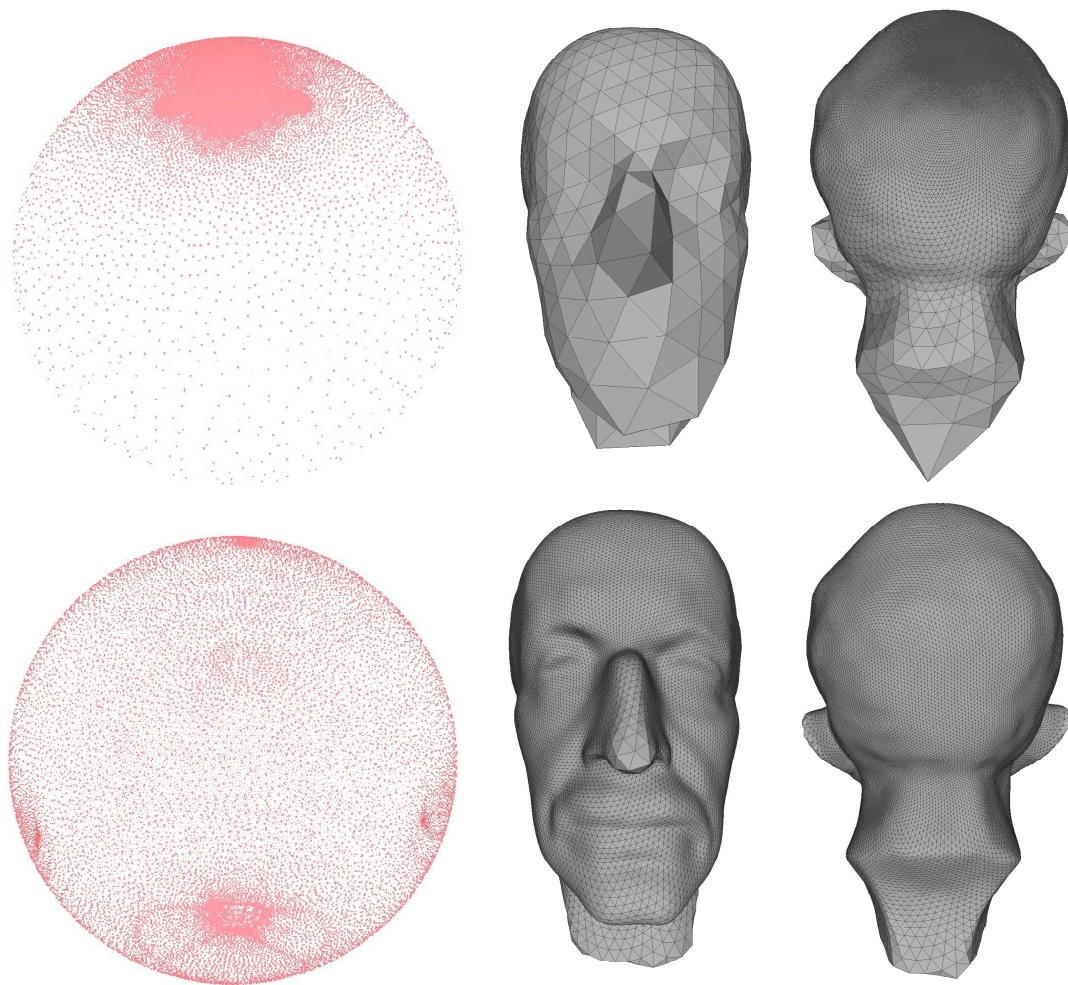
Figure 9. *The effect of our balancing scheme on meshing a genus-0 point cloud of Max Planck. Top left: A spherical conformal parameterization without the balancing scheme. Bottom left: A spherical conformal parameterization with the balancing scheme. Middle: The front view of the meshing results by interpolation with the aid of the parameterizations. Right: The back view.*

the mesh. As a result, most features of the underlying surface are lost. In contrast, with our proposed balancing scheme, a high-quality mesh can be effectively generated. The above results reflect the importance of our balancing scheme in the point cloud parameterizations for meshing.

7.4. Stability under geometrical and topological noises. Our meshing framework is stable under geometrical and topological noises of the input genus-0 point clouds. In some situations, the point clouds obtained by 3D cameras are geometrically noisy. To compute triangulations which represent the underlying surfaces, we can first apply a Poisson filtering on the noisy point clouds. Then, with the aid of our spherical conformal parameterization, we can obtain high-quality triangulations on a uniform spherical point cloud and interpolate
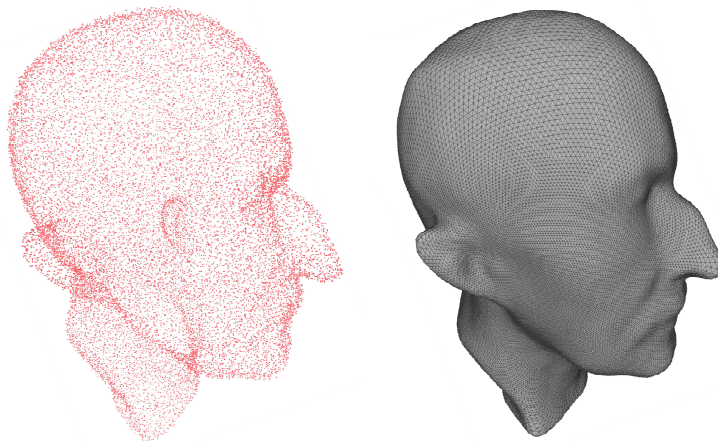
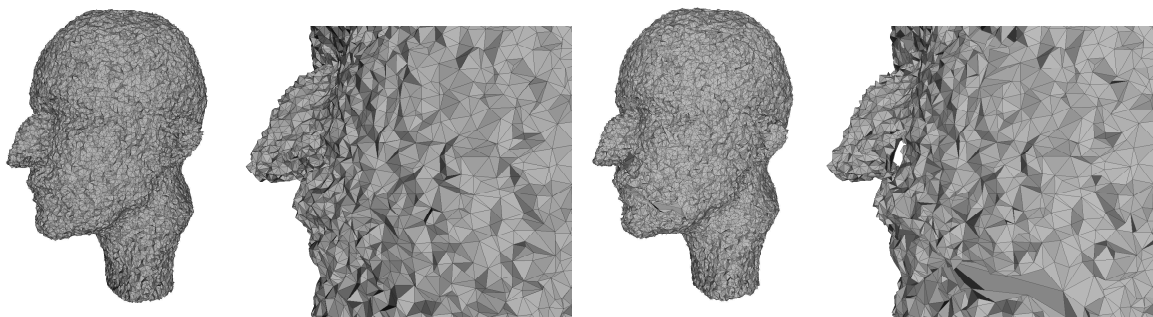**Figure 10.** *Meshing a geometrically noisy point cloud.*

Figure 11. *Comparison of our meshing scheme and the Tight Cocone algorithm* [11] *on a geometrically noisy point cloud. All points are considered in the computations. Left: Our meshing result with a zoom-in of the nose. Right: The result of the Tight Cocone algorithm with a zoom-in of the nose.*
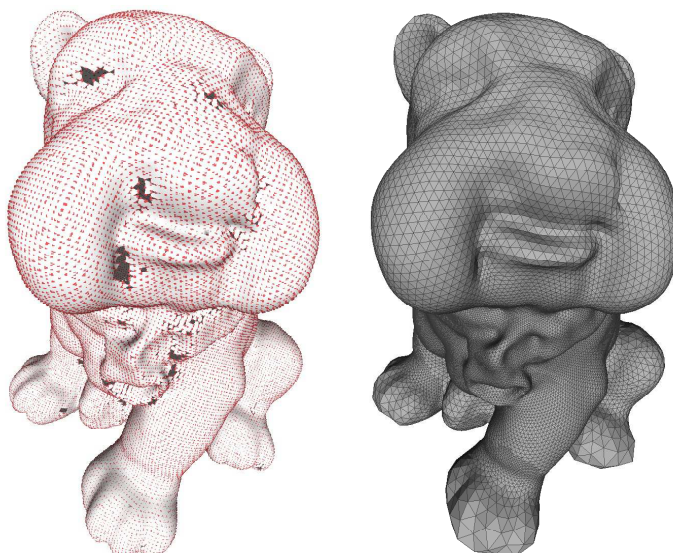

Figure 12. *Meshing a topologically noisy point cloud with unwanted holes.*

Figure 13. *Meshing real* 3D *scanned noisy point cloud data of a cereal box and a soda can in the RGB-D Scenes Dataset v.*2 [29]. *Left: The raw point cloud data. Middle: The meshing results. Right: The meshing results without showing the triangulations.*

them back onto the filtered point clouds to produce meshed surfaces. We first demonstrate the effectiveness of our algorithm with two synthetic examples.

Figure 10 shows a synthetic point cloud with 3% uniformly distributed random noise and our meshing result. We can also construct a faithful triangulated mesh on the geometrically noisy point cloud without any filtering or sampling procedure. Figure 11 shows the triangulation result of our meshing scheme and the Tight Cocone algorithm [11] on the noisy point cloud in Figure 10. All points of the point cloud are considered and fixed in the construction of the triangulation. It can be observed that there are irregular triangulations and topological
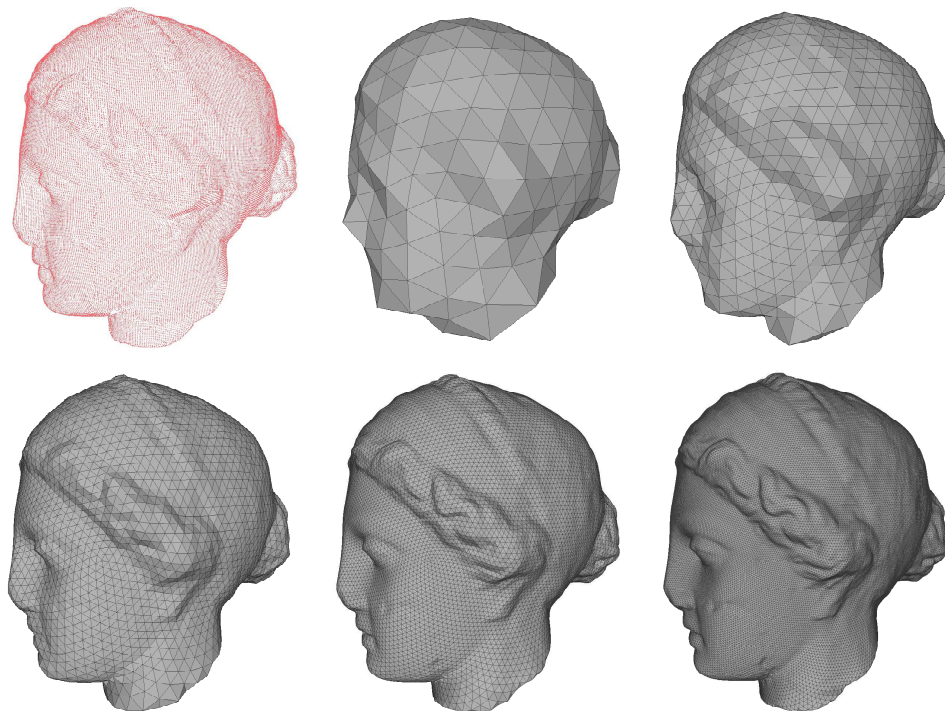
Figure 14. *Multilevel representations of a genus-0 point cloud of Igea. For better visualizations, we create mesh structures on the representations. Top left: A point cloud with 134345 points of Igea. Top middle to bottom right: The multilevel representations with 0, 1, 2, 3, and 4 subdivisions. The representations are with 642, 2562, 10242, 40962, and 163842 points, respectively.*

holes on the result by [11], while our meshing scheme guarantees a regular and topology preserving triangulation even for noisy input point clouds.

In addition, it is common that the sampling processes result in nonuniformly sampled point clouds. In particular, there may be large holes on certain parts of the point clouds sampled from genus-0 objects, which create topological ambiguities and hinder mesh generations. Our parameterization and meshing scheme produce satisfactory results with these topological noises. Moreover, the meshes generated are guaranteed to be genus-0 closed meshes. Figure 12 shows a synthetic point cloud with 1021 randomly created topological holes. It can be observed that our algorithm produces a satisfactory meshing result.

Then, we apply our algorithm for real 3D scanned noisy point cloud data. Much raw point cloud data are adapted from the RGB-D Scenes Dataset v.2 [29]. Figure 13 shows two point clouds of a soda can and a cereal box and shows our meshing results. The above experiments demonstrate the stability and robustness of our proposed method for noisy point clouds.

7.5. Multilevel representations of genus-0 point clouds. With our proposed spherical conformal parameterization scheme, multilevel representations of a genus-0 point cloud can be easily achieved. We start with a coarse spherical point cloud. The vertices on the sphere can be interpolated onto the genus-0 point cloud with the aid of its spherical parameterization.

Figure 15. *Multilevel representations of a genus-0 lion vase point cloud. For better visualizations, we create mesh structures on the representations. Top left: A 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 with* 696, 2778, 11106, 44418, *and* 177666 *points, respectively.*

Then, we can progressively subdivide the sphere using existing subdivision methods, such as the butterfly subdivision method [12] and the loop subdivision method [40]. For each subdivided sphere, we can repeat the mentioned interpolation procedure and obtain a coarse representation of the point cloud. This method results in multilevel representations of the point cloud. As the subdivision level increases, more details of the point cloud are represented. Examples of multilevel representations of genus-0 point clouds are given in Figures 14 and 15. In our examples, the subdivisions are generated using the loop subdivision method [40]. The subdivision connectivity of the results can be easily observed. The results indicate that our method can effectively generate the multilevel representations of genus-0 point clouds.

8. Conclusion and future work. In this paper, we presented a novel framework for meshing genus-0 point clouds via global spherical conformal parameterizations. We extended and improved the parameterization algorithm for triangular meshes in [9]. First, we enhanced

the accuracy for approximating the LB operator on point clouds by using a new Gaussian-type weight function. Second, we proposed an iterative scheme called the N-S reiteration to replace the step of solving for a quasi-conformal map in [9] for achieving better conformality. Third, we introduced a balancing scheme for guaranteeing an even distribution of the spherical point cloud parameterization. Experimental results show that our proposed algorithm is highly efficient and accurate. With the aid of the spherical conformal parameterizations, almost-Delaunay triangulations and high-quality quadrangulations of genus-0 point clouds can be effectively created. The meshes generated are guaranteed to be of genus-0 and no post-processing is needed. In addition, our meshing method is stable under geometrical and topological noises on point clouds. Moreover, multilevel representations of genus-0 point clouds can be easily computed. As a remark, our proposed spherical conformal parameterization algorithm also works efficiently on triangular meshes. In the future, we plan to establish a rigorous theoretical proof of the convergence of our parameterization scheme and extend our method to handle disk-type point clouds and point clouds with arbitrary topology.

REFERENCES

[1] Y. AFLALO, R. KIMMEL, AND M. ZIBULEVSKY, *Conformal mapping with as uniform as possible conformal factor*, SIAM J. Imaging Sci., 6 (2013), pp. 78–101.

[2] S. ANGENENT, S. HAKER, A. TANNENBAUM, AND R. KIKINIS, *Conformal geometry and brain flattening*, in Medical Image Computing and Computer-Assisted Intervention, Lecture Notes in Comput. Sci. 1679, Springer, New York, 1999, pp. 271–278.

[3] P. AZARIADIS, *Parameterization of clouds of unorganized points using dynamic base surfaces*, Comput. Aided Design, 36 (2004), pp. 607–623.

[4] P. AZARIADIS AND N. SAPIDIS, *Efficient parameterization of 3D point-sets using recursive dynamic base surfaces*, in Advances in Informatics, Lecture Notes in Comput. Sci. 3746, Springer, New York, 2005, pp. 296–306.

[5] P. AZARIADIS AND N. SAPIDIS, *Product design using point-cloud surfaces: A recursive subdivision technique for point parameterization*, Computers in Industry, 58 (2007), pp. 832–843.

[6] M. BELKIN AND P. NIYOGI, *Towards a theoretical foundation for Laplacian-based manifold methods*, J. Comput. System Sci., 74 (2008), pp. 1289–1308.

[7] M. BELKIN, J. SUN, AND Y. WANG, *Constructing Laplace operator from point clouds in $\mathbb{R}^d$*, in Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 1031–1040.

[8] P. BREITKOPF, H. NACEUR, A. RASSINEUX, AND P. VILLON, *Moving least squares response surface approximation: Formulation and metal forming applications*, Comput. Structures, 83 (2005), pp. 1411–1428.

[9] P. T. CHOI, K. C. LAM, AND L. M. LUI, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces*, SIAM J. Imaging Sci., 8 (2015), pp. 67–94.

[10] P. T. CHOI AND L. M. LUI, *A Linear Algorithm for Disk Conformal Parameterization of Simply-Connected Open Surfaces*, preprint, arXiv:1508.00396v1, 2015.

[11] T. K. DEY AND S. GOSWAMI, *Tight Cocone: A water tight surface reconstructor*, in Proceedings of the 8th ACM Symposium on Solid Modeling and Applications, 2003, pp. 127–134.

[12] N. DYN, D. LEVINE, AND J. A. GREGORY, *A butterfly subdivision scheme for surface interpolation with tension control*, ACM Trans. Graph., 9 (1990), pp. 160–169.

[13] M. S. FLOATER, *Meshless parameterization and B-spline surface approximation*, in Proceedings of the 9th IMA Conference on the Mathematics of Surfaces, 2000, pp. 1–18.

[14] M. S. FLOATER AND M. REIMERS, *Meshless parameterization and surface reconstruction*, Comput. Aided Geom. Design, 18 (2001), pp. 77–92.

[15] M. FLOATER AND K. HORMANN, *Parameterization of triangulations and unorganized points*, in Tutorials on Multiresolution in Geometric Modelling, Math. Vis., Springer, Berlin, 2002, pp. 287–316.

[16] M. FLOATER AND K. HORMANN, *Surface parameterization: A tutorial and survey*, in Advances in Multiresolution for Geometric Modelling, Math. Vis., Springer, Berlin, 2005, pp.157–186.

[17] F. GARDINER AND N. LAKIC, *Quasiconformal Teichmuller Theory*, AMS, Providence, RI, 2000.

[18] X. GU AND S. T. YAU, *Global conformal surface parameterization*, in Proceedings of the Eurographics Symposium on Geometry Processing, 2003, pp. 127–137.

[19] X. GUO, X. LI, Y. BAO, X. GU, AND H. QIN, *Meshless thin-shell simulation based on global conformal parameterization*, IEEE Trans. Vis. Comput. Graphics., 12 (2006), pp. 375–385.

[20] S. HAKER, S. ANGENENT, A. TANNENBAUM, R. KIKINIS, G. SAPIRO, AND M. HALLE, *Conformal surface parameterization for texture mapping*, IEEE Trans. Vis. Comput. Graphics., 6 (2000), pp. 181–189.

[21] H. HOPPE, T. DEROSE, T. DUCHAMP, J. MCDONALD, AND W. STUETZLE, *Surface reconstruction from unorganized points*, in Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, 1992, pp. 71–78.

[22] K. HORMANN AND M. REIMERS, *Triangulating point clouds with spherical topology*, Curve and Surface Design: Saint-Malo 2002, Modern Methods in Applied Mathematics, Nashboro Press, 2003, pp. 215–224.

[23] K. HORMANN, B. LÉVY, AND A. SHEFFER, *Mesh parameterization: Theory and practice*, SIGGRAPH 2007 Course Notes, Vol. 2, 2007, pp. 1–122.

[24] J. JOST, *Riemannian Geometry and Geometry Analysis*, 6th ed., Universitext, Springer, Berlin, 2011.

[25] G. W. JONES AND L. MAHADEVAN, *Planar morphometry, shear and optimal quasi-conformal mappings*, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci., 469 (2013).

[26] K. C. LAM, *Surface Registration Using Quasi-Conformal Teichmüller Theory and Its Application to Texture Mapping*, M. Phil. thesis, Chinese University of Hong Kong, 2013.

[27] R. LAI, J. LIANG, H. ZHAO, *A local mesh method for solving PDEs on point clouds*, Inverse Probl. Imaging, 7 (2013), pp. 737–755.

[28] 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*, J. Sci. Comput., 58 (2014), pp. 705–725.

[29] K. LAI, L. BO, AND D. FOX, *Unsupervised feature learning for 3D scene labeling*, in Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2014.

[30] C. LANGE AND K. POLTHIER, *Anisotropic smoothing of point sets*, Comput. Aided Geom. Des., 22 (2005), pp. 680–692.

[31] P. LANCASTER AND K. SALKAUSKAS, *Surfaces generated by moving least squares methods*, Math. Comp., 37 (1981), pp. 141–158.

[32] D. LEVIN, *The approximation power of moving least-squares*, Math. Comp., 67 (1998), pp. 1517–1531.

[33] B. LÉVY, *Constrained texture mapping for polygonal meshes*, in Proceedings of ACM SIGGRAPH, 2001, pp. 417–424.

[34] E. LI, B. LÉVY, X. ZHANG, W. CHE, W. DONG, AND J.-C. PAUL, *Meshless quadrangulation by global parametrization*, Computers & Graphics, 35 (2011), pp. 992–1000.

[35] J. LIANG, R. LAI, A. WONG, AND H. ZHAO, *Geometric understanding of point clouds using Laplace-Beltrami operator*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[36] J. LIANG AND H. ZHAO, *Solving partial differential equations on point clouds*, SIAM J. Sci. Comput., 35 (2013), pp. 1461–1486.

[37] D. LIN, *Statistical Learning Toolbox*, MathWorks Community File Exchange, 2006, http://www.mathworks.com/matlabcentral/fileexchange/12333-statistical-learning-toolbox.

[38] H. LIPSON AND M. KURMAN, *Fabricated: The New World of 3D Printing*, Wiley, New York, 2013.

[39] W. K. LIU, S. LI, AND T. BELYTSCHKO, *Moving least-square reproducing kernel methods* (I). *Methodology and convergence*, Comput. Methods Appl. Mech. Eng., 143 (1997), pp. 113–154.

[40] C. T. LOOP, *Smooth Subdivision Surfaces Based on Triangles*, M.S. thesis, University of Utah, 1987.

[41] W. E. LORENSEN AND H. E. CLINE, *Marching cubes: A high resolution 3D surface construction algorithm*, in Proceedings of ACM SIGGRAPH, 1987, pp. 163–169.

[42] F. LOZES, A. ELMOATAZ, AND O. LÉZORAY, *Partial difference operators on weighted graphs for image processing on surfaces and point clouds*, IEEE Trans. Image Process., 23 (2014), pp. 3896–3909.

[43] L. M. LUI, K. C. LAM, T. W. WONG, AND X. F. GU, *Texture map and video compression using Beltrami representation*, SIAM J. Imaging Sci., 6 (2013), pp. 1880–1902.

[44] L. M. LUI, X. GU, AND S.-T. YAU, *Convergence of an iterative algorithm for Teichmüller maps via harmonic energy minimization*, Math. Comp., 84 (2015), pp. 2823–2842.

[45] C. B. MACDONALD, B. MERRIMAN, AND S. J. RUUTH, *Simple computation of reaction-diffusion processes on point clouds*, Proc. Natl. Acad. Sci. USA, 110 (2014), pp. 9209–9214.

[46] Q. MENG, B. LI, H. HOLSTEIN, AND Y. LIU, *Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBF networks*, Pattern Recognit., 46 (2013), pp. 2361–2375.

[47] B. NAYROLES, G. TOUZOT, AND P. VILLON, *Generalizing the finite element method: Diffuse approximation and diffuse elements*, Comput. Mech., 10 (1992), pp. 307–318.

[48] 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.

[49] B. OBERKNAPP AND K. POLTHIER, *An algorithm for discrete constant mean curvature surfaces*, in Visualization and Mathematics, Springer, Berlin, 1997, pp. 141–161.

[50] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Experiment. Math., 2 (1993), pp. 15–36.

[51] F. RENGIER, A. MEHNDIRATTA, H. VON TENGG-KOBLIGK, C. M. ZECHMANN, R. UNTERHINNINGHOFEN, H. U. KAUCZOR, AND F. L. GIESEL, 3*D printing based on imaging data: Review of medical applications*, Int. J. Comput. Assist. Radiol. Surg., 5 (2010), pp. 335–341.

[52] P. V. SANDER, J. SNYDER, S. J. GORTLER, AND H. HOPPE, *Texture mapping progressive meshes*, in Proceedings of ACM SIGGRAPH, 2001, pp. 409–416.

[53] R. SCHOEN AND S. T. YAU, *Lectures on Differential Geometry*, International Press, Cambridge, MA, 1994.

[54] R. SCHOEN AND S. YAU, *Lectures on Harmonic Maps*, International Press, Cambridge, MA, 1997.

[55] A. SHEFFER, E. PRAUN, AND K. ROSE, *Mesh parameterization methods and their applications*, Found. Trends Comput. Graphic Vis., 2 (2006), pp. 105–171.

[56] D. SHEPARD, *A two-dimensional interpolation function for irregularly spaced points*, in Proceedings of the 23rd ACM National Conference, 1968, pp. 517–524.

[57] G. TEWARI, C. GOTSMAN, AND S. J. GORTLER, *Meshing genus-*1 *point clouds using discrete one-forms*, Computers & Graphics, 30 (2006), pp. 917–926.

[58] L. WANG, B. YUAN, AND Z. MIAO, 3*D Point clouds parameterization algorithm*, in Proceedings of the 9th International Conference on Signal Processing, 2008, pp. 1410–1413.

[59] H. WENDLAND, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*, Adv. Comput. Math., 4 (1995), pp. 389–396.

[60] H. WENDLAND, *Local polynomial reproduction and moving least squares approximation*, IMA J. Numer. Anal., 21 (2001), pp. 285–300.

[61] L. ZHANG, L. LIU, C. GOTSMAN, AND H. HUANG, *Mesh reconstruction by meshless denoising and parameterization*, Computers & Graphics, 34 (2010), pp. 198–208.

[62] W. ZENG, L. M. LUI, F. LUO, T. F. CHAN, S. T. YAU, AND X. GU, *Computing quasiconformal maps using an auxiliary metric and discrete curvature flow*, Numer. Math., 121 (2012), pp. 671–703.

[63] M. ZWICKER AND C. GOTSMAN, *Meshing point clouds using spherical parameterization*, in Proceedings of the Eurographics Symposium on Point-Based Graphics, 2004.

[64] *AIM@SHAPE Shape Repository*, http://visionair.ge.imati.cnr.it/.

[65] *Stanford 3D Scanning Repository*, http://graphics.stanford.edu/data/3Dscanrep/.