

# SPHERICAL CONFORMAL PARAMETERIZATION OF GENUS-0 POINT CLOUDS FOR MESHING

PUI TUNG CHOI, KIN TAT HO AND LOK MING LUI

**Abstract.** Point cloud is the most fundamental representation of 3D geometric objects as it provides the most exact information of them. 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

**1. Introduction.** Contemporary scanning technologies enable efficient acquisitions of 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, mesh structures are necessary for 3D printing [38, 27] and texture mapping [39, 24]. With the rapid development of the computer industry, finding a high quality meshing framework for point cloud data is increasingly important.

One possible approach for mesh generation on point clouds is to parameterize a point cloud to a simpler domain, such as the unit sphere. 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 become more obscure. Hence, it is more challenging in developing parameterization schemes with good quality for point cloud data.

A good parameterization scheme of point cloud must satisfy certain criteria. In particular, it should retain the geometric information of a point cloud as complete 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 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 analog 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 [6], Choi *et al.* 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 is linear 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 [6] 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 operators on point clouds. Using our improved approximation, the Laplace-Beltrami operator in the mentioned algorithm can be accurately computed on point clouds. Also, we replace a key step in the mentioned algorithm by an iterative scheme called the North-South reiteration. 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, multi-level 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. In Section 2, we review the related previous works on point cloud parameterizations and approximations of differential operators on point clouds. The contribution of our work is highlighted in Section 3. 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.

Methods	Topology	Parameter domain	Local/global parameterization?	Distortion to be minimized
Meshless parameterization [10, 11]	Disk topology	Plane	Global	/
Meshless parameterization for Spherical Topology [18]	Genus-0	Planes	Local	/
Spherical embedding [50]	Genus-0	Sphere	Global	Stretch
Discrete one-forms [44]	Genus-1	Planes	Local	/
As-rigid-as-possible meshless parameterization [48]	Disk topology	Plane	Global	ARAP

TABLE 1.1

*Several previous works on meshing point clouds using parameterization.*

**2. 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, please refer to [12, 13, 19, 42]. In the last few decades, numerous studies have been devoted to the parameterization of point cloud data. In [10, 11], 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 [50], Zwicker and Gotsmann presented a parameterization approach for a genus-0 point cloud using a  $k$ -nearest neighborhood graph of the point cloud, followed by a spherical embedding method for planar graphs. In [1, 2, 3], 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.* [16] computed a global conformal parameterization of point-set surfaces, based on Riemann surface theory and Hodge theory. In [44], Tewari *et al.* 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 *et al.* [45] 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 [48], 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 [25], Liang *et al.* constructed spherical conformal mappings of genus-0 point clouds by adapting the harmonic energy minimization algorithm in [21]. Meng *et al.* [34] 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 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 [10, 11], Floater and Reimers applied their proposed parameterization scheme for meshing point clouds with disk topology. In [18], Hormann and Reimers extended the parameterization method in [11] for surface reconstruction of point clouds with spherical topology. In [50], Zwicker and Gotsmann used their proposed parameterization method for mesh reconstruction of genus-0 point clouds. Tewari *et al.* [44] performed surface reconstruction using their proposed doubly-periodic global parameterization. Zhang *et al.* [48] suggested a mesh reconstruction method of point cloud data by meshless denoising and their proposed parameterization scheme. Table 1.1 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 analog of the differential operators on point clouds. Numerous works on approximating differential operators on point cloud have been reported. In [35], Naylor *et al.* described a diffuse approximation method for estimating the derivatives at a given set of points. Belkin *et al.* [4] proposed the PCD Laplace operator for approximating the Laplace-Beltrami operator using an integral approximation. The moving least square (MLS) method [43, 22] is widely used for the approximation. A number of algorithms for the approximation of derivatives are developed based on the MLS method [28, 23, 36, 5]. In [25, 26], Liang *et al.* approximated the Laplace-Beltrami operator on point clouds by the MLS method with a special weighting function. In [20], Lai *et al.* 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 *et al.* [33] computed reaction-diffusion processes on point clouds. In [31], Lozes *et al.* proposed a method to solve PDEs on point clouds for image processing using partial difference operators on weighted graphs.

**3. 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 [6] for point clouds. An accurate approximation of the Laplace-Beltrami operator is achieved using the moving least square method [25, 26] together with our proposed weight function. A key step of the parameterization algorithm in [6] is replaced by solving a Laplace equation on the complex plane, followed by an iterative scheme called the North-South (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 post-processing 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.

**4. Mathematical background.** In this section, we introduce some basic mathematical concepts closely related to our work. For more details, readers are referred to [40, 41].

**4.1. Conformal maps.** 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  $r_1, r_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^3$  are two vector-valued functions. The first fundamental form of  $\mathcal{M}$  is defined by

$$ds_{\mathcal{M}}^2 = \sum_{i,j} g_{ij} dx^i dx^j, \quad (4.1)$$

where

$$g_{ij} = \left\langle \frac{\partial r_1}{\partial x^i}, \frac{\partial r_1}{\partial x^j} \right\rangle. \quad (4.2)$$

Similarly, the first fundamental form of  $\mathcal{N}$  is defined by

$$ds_{\mathcal{N}}^2 = \sum_{i,j} \tilde{g}_{ij} dx^i dx^j, \quad (4.3)$$

where

$$\tilde{g}_{ij} = \left\langle \frac{\partial r_2}{\partial x^i}, \frac{\partial r_2}{\partial x^j} \right\rangle. \quad (4.4)$$

Let  $f : \mathcal{M} \rightarrow \mathcal{N}$  be a map between the two Riemann surfaces. In local coordinate systems,  $f$  can be regarded as  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , with

$$f(x^1, x^2) = (f^1(x^1, x^2), f^2(x^1, x^2)). \quad (4.5)$$

The pull-back metric  $f^* ds_{\mathcal{N}}^2$  defined on  $\mathcal{M}$ , induced by  $f$  and  $ds_{\mathcal{N}}^2$ , is the metric

$$f^* ds_{\mathcal{N}}^2 = \sum_{m,n} \left( \sum_{i,j} \tilde{g}_{ij}(f(x^1, x^2)) \frac{\partial f^m}{\partial x^i} \frac{\partial f^n}{\partial x^j} \right) dx^m dx^n. \quad (4.6)$$

The map  $f : \mathcal{M} \rightarrow \mathcal{N}$  is said to be *conformal* if there exists a positive scalar function  $\lambda(x^1, x^2)$ , called the *conformal factor*, such that

$$f^* ds_{\mathcal{N}}^2 = \lambda ds_{\mathcal{M}}^2. \quad (4.7)$$

In other words, the surface metric is preserved up to a multiplicative factor.

An immediate consequence of the above is that every conformal map preserves angles. With this angle-preserving property, a conformal map effectively preserves 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 *harmonic energy functional* for a map  $f : \mathcal{M} \rightarrow \mathbb{S}^2$  is defined as

$$E(f) = \int_{\mathcal{M}} |\nabla f|^2 dv_{\mathcal{M}}. \quad (4.8)$$

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. Hence, the problem of finding a conformal map  $f : \mathcal{M} \rightarrow \mathbb{S}^2$  is equivalent to an energy minimization problem.

**4.3. Point cloud and local system.** A *point cloud*  $P = \{z_1, z_2, \dots, z_n\} \subset \mathbb{R}^3$  is a set of sample points representing a Riemann surface  $\mathcal{M}$ . For triangulated surfaces, we can find the neighborhood of each vertex by the associated triangulation and hence calculate the local geometric information. However, in the case of point cloud, we do not have such information. Given a point cloud of an orientable surface in  $\mathbb{R}^3$ , we construct a local coordinate system on each point  $z_s$  and approximate the derivatives on the point cloud.

To construct a local coordinate system, we need to define an atlas  $(U_s, \phi_s)$  for each point  $z_s$ , where  $U_s$  is an open cover and  $\phi_s$  is the associated local coordinate function. For point clouds, one common approach for forming the open cover  $U_s$  of the point  $z_s$  is to use the collection of all neighboring points of  $z_s$ , denoted by  $\mathcal{N}(z_s)$ . We apply the *k-Nearest-Neighbors (k-NN)* algorithm to define the neighborhood of each point  $z_s$ .

Let  $P = \{z_1, z_2, \dots, z_n\} \subset \mathbb{R}^N$  be a given point cloud with  $n$  points. The *k-nearest neighborhood*  $\mathcal{N}^k(z_s)$  of  $z_s$  is a set with the  $k$  distinct elements in  $P$  (including  $z_s$ ) closest from  $z_s$ . Here the distance is defined by the Euclidean 2-norm. For convenience, we denote  $\mathcal{N}^k(z_s) = \{z_s^1, z_s^2, \dots, z_s^k\}$  with  $z_s^1 = z_s$ .

After introducing an open cover of  $P$ , we now construct a local coordinate system for  $P$  and a coordinate function  $\phi_s$  for each neighborhood  $\mathcal{N}^k(z_s)$ . 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  $k$ -plane clustering method and the principal component analysis (PCA) method [17]. Using the PCA method, for every point  $z_s \in P$ , we can obtain three vectors  $\{e_s^1, e_s^2, e_s^3\}$  which form an orthonormal basis of  $\mathbb{R}^3$ .

Then, we project each neighborhood  $\mathcal{N}^k(z_s)$  to the plane spanned by  $\{e_s^1, e_s^2\}$  by

$$\hat{z}_s^i = z_s^i - \langle z_s^i - z_s, e_s^3 \rangle e_s^3, \quad (4.9)$$

for  $i = 1, 2, \dots, k$ .

Now we have the projection

$$\hat{\mathcal{N}}^k(z_s) = \{\hat{z}_s^1, \hat{z}_s^2, \dots, \hat{z}_s^k\} \quad (4.10)$$

and also the local coordinates

$$\{(x_s^1, y_s^1), (x_s^2, y_s^2), \dots, (x_s^k, y_s^k)\}, \quad (4.11)$$

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  $i = 1, 2, \dots, k$ . Therefore, we can define the local coordinate function  $\phi_s : \mathcal{N}_s \rightarrow \mathbb{R}^2$  by

$$\phi_s(z_s^i) = (x_s^i, y_s^i). \quad (4.12)$$

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 [6] for computing a spherical conformal parameterization of a genus-0 closed triangular mesh. This approach motivates our proposed parameterization scheme for genus-0 point clouds.

Recall that for genus-0 closed surfaces, conformal maps are equivalent to harmonic maps. To compute a conformal mapping  $f : M \rightarrow \mathbb{S}^2$  from a genus-0 closed triangular mesh  $M$  to the unit sphere  $\mathbb{S}^2$ , it suffices to solve Equation (4.8). This can be achieved by solving the following Laplace equation

$$\Delta^T f = 0 \quad (5.1)$$

subject to  $\|f\| = 1$ , where  $\Delta^T f$  is the tangential component of  $\Delta f$  on the tangent plane of  $\mathbb{S}^2$ . Note that this problem is nonlinear because of the constraint  $\|f\| = 1$ . In [6], the authors linearize this problem by solving the equation on the complex plane:

$$\Delta f = 0 \quad (5.2)$$

given the constraints of three boundary points

$$f(a_i) = b_i, \quad (5.3)$$

where  $a_i$  and  $b_i$  are complex numbers for  $i = 1, 2, 3$  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  $\mathbb{C}$ . Since the nonlinear constraint  $\|f\| = 1$  in the problem (5.1) is removed, the above problem (5.2) becomes linear.

After solving Equation (5.2), the inverse stereographic projection 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 *et al.* [6] proposed to apply the south-pole stereographic projection to map the sphere to a planar domain  $R$  on the complex plane. Note that the region with large conformality distortion is the innermost region of the planar domain  $R$  while the outermost region of  $R$  is with negligible distortion. Denote the abovementioned steps by a map  $g : M \rightarrow R$ . To correct the conformality distortion of  $g$ , Choi *et al.* made use of the quasi-conformal theory. Quasi-conformal maps are a generalization of conformal maps. Mathematically,  $f : \mathbb{C} \rightarrow \mathbb{C}$  is a quasi-conformal map if it satisfies the Beltrami equation:

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z} \quad (5.4)$$

for some complex-valued function  $\mu$  satisfying  $\|\mu\|_\infty < 1$  and  $\frac{\partial f}{\partial \bar{z}}$  is non-vanishing almost everywhere.  $\mu$  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  $\mu(p) = 0$ . Readers are referred to [14] for more details of quasi-conformal maps.

Let  $\mu_{g^{-1}}$  be the Beltrami coefficient of the map  $g^{-1}$ . Fixing the outermost region on  $R$ , the authors in [6] composed the map  $g$  with a quasi-conformal map  $h : R \rightarrow \mathbb{S}^2$

with the associated Beltrami coefficient  $\mu_h = \mu_g^{-1}$ . The map  $h$  is constructed by solving the following equation

$$\nabla \cdot \left( A \begin{pmatrix} u_x \\ u_y \end{pmatrix} \right) = 0 \quad \text{and} \quad \nabla \cdot \left( A \begin{pmatrix} v_x \\ v_y \end{pmatrix} \right) = 0 \quad (5.5)$$

where  $A = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_2 & \alpha_3 \end{pmatrix}$  depends on the prescribed Beltrami coefficient  $\mu_h = \mu_g^{-1}$ . In the discrete case, the above elliptic PDEs (5.5) can be discretized into sparse symmetric positive definite linear systems.

Then, by the composition formula of quasi-conformal maps, the composition map  $h \circ g : M \rightarrow \mathbb{S}^2$  is with the Beltrami coefficient

$$\begin{aligned} \mu_{h \circ g} &= \frac{\mu_g + (\overline{g_z}/g_z)(\mu_h \circ g)}{1 + (\overline{g_z}/g_z)\overline{\mu_g}(\mu_h \circ g)} \\ &= \frac{\mu_g + (\overline{g_z}/g_z)(-(g_z/\overline{g_z})\mu_g)}{1 + (\overline{g_z}/g_z)\overline{\mu_g}(-(g_z/\overline{g_z})\mu_g)} \\ &= 0, \end{aligned} \quad (5.6)$$

which indicates that the map  $h \circ g$  is conformal. Readers are referred to [6] for more details.

The above spherical conformal parameterization algorithm works perfectly for genus-0 closed triangular meshes. It is noteworthy that there are two key steps in the above algorithm. The first key step is to solve the Laplace equation (5.2) on the complex plane. In this step, the Laplace-Beltrami operator is computed using the well known cotangent formula in [37] for triangular meshes. The second key step is to compose a map with a quasi-conformal map. In this step, the conformality of the composition mapping is guaranteed by the composition formula (5.6). However, in our case, the coordinates of the point clouds are the only given information. In other words, unlike the case of triangular meshes, no connectivity information about the data points is given. Note that 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 all data points of a point cloud, the composition formula (5.6) may not hold for the discrete Beltrami coefficients on point clouds. Therefore, we need to modify the second key step for computing the spherical conformal parameterization of genus-0 point clouds. Nevertheless, for the first key step, alternative approaches which do not require triangulations are available for computing the Laplace-Beltrami operator in Equation (5.2). In Section 6, we introduce a method to replace the second key step by solving another Laplace equation.

**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 Laplace-Beltrami operator.** In this subsection, we explain our approximation scheme for the Laplace-Beltrami operator in the Laplace Equation (5.2) on a point cloud  $P$  by the moving least-square method. The moving least-square method is widely used for approximation [28, 23, 36, 5, 25, 26]. In particular, Liang *et al.* [25, 26] approximated the Laplace-Beltrami operator on point

Weight	Formula of $w(d)$
Constant weight	$w(d) = 1$
Exponential weight	$w(d) = \exp\left(-\frac{d^2}{h^2}\right)$
Inverse of squared distance weight	$w(d) = \frac{1}{d^2 + \epsilon^2}$
Wendland weight [46, 47]	$w(d) = \left(1 - \frac{d}{D}\right)^4 \left(\frac{4d}{D} + 1\right)$
Special weight [25]	$w(d) = \begin{cases} 1 & \text{if } d = 0 \\ \frac{1}{K} & \text{if } d \neq 0 \end{cases}$

TABLE 6.1

*Some common weighting functions.*

clouds using the MLS method with a special weight function. Our approximation scheme is built upon the method in [25, 26]. In this work, we propose a new weight function to achieve a more accurate approximation of the Laplace-Beltrami operator.

First, we discuss our approximation method for the derivatives on the point cloud  $P = \{z_1, z_2, \dots, z_n\}$ . 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  $\{f_s^1, f_s^2, \dots, f_s^m\}$  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, which means  $m = 6$ .

In the approximation, we aim to minimize

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

where  $w_i = w(\|z_i - z_s\|)$  for some weighting function  $w : \mathbb{R} \rightarrow \mathbb{R}$ . The weight function  $w$  significantly affects the accuracy and robustness of the approximation. Hence, one must carefully choose a suitable weight function. Table 6.1 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 of a weight function of the Gaussian type. We propose the following weight function:

$$\begin{cases} w_s = w(0) = 1 \\ w_i = w(\|z_i - z_s\|) = \frac{\exp(-\frac{\sqrt{n}\|z_i - z_s\|^2}{h^2})}{n} \quad \text{for all } i \neq s. \end{cases} \quad (6.2)$$

Numerical experiments are demonstrated in Section 7 to support our proposed weight function. It can be observed that our proposed weight results in more accurate approximations of the Laplace-Beltrami 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  $\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}$ ,  $\vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}$ ,  $\vec{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}$ , 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:

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

We can solve it by quadratic programming or using the least-square method, namely solving

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

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

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

Similarly, the coefficients  $\hat{c}_i$  can be approximated. Let  $\vec{A} = \begin{pmatrix} f_{s,1}^1 & f_{s,1}^2 & \cdots & f_{s,1}^m \\ f_{s,2}^1 & f_{s,2}^2 & \cdots & f_{s,2}^m \\ \vdots & \vdots & \ddots & \vdots \\ f_{s,n}^1 & f_{s,n}^2 & \cdots & f_{s,n}^m \end{pmatrix}$ ,

$$\vec{D} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix}, \vec{\hat{c}} = \begin{pmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_m \end{pmatrix}, \text{ and } \vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}. \text{ We can find the coefficients}$$

$\hat{c}_i$  by solving the following least-square problem

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

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

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

and

$$\frac{\partial u}{\partial y} = \sum_{i=1}^m \vec{c}_i \frac{\partial f_s^i}{\partial y}. \quad (6.8)$$

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

$$\Delta u(z) = \frac{1}{W} \sum_{i,j=1}^2 \partial_i (g^{ij} W \partial_j (u(z))), \quad (6.9)$$

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

$$(g_{ij}) = \begin{pmatrix} 1 + (f_s)_x^2 & (f_s)_x(f_s)_y \\ (f_s)_x(f_s)_y & 1 + (f_s)_y^2 \end{pmatrix} \quad (6.10)$$

and

$$(g^{ij}) = \frac{1}{W^2} \begin{pmatrix} 1 + (f_s)_y^2 & -(f_s)_x(f_s)_y \\ -(f_s)_x(f_s)_y & 1 + (f_s)_x^2 \end{pmatrix}, \quad (6.11)$$

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

We use Equation (6.7) and Equation (6.8) 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  $\Delta u$  and the Laplace-Beltrami operator is a second order differential operator, by differentiating Equation (6.9), we get

$$\Delta u(z_s) = \alpha_1 \frac{\partial u}{\partial x}(z_s) + \alpha_2 \frac{\partial u}{\partial y}(z_s) + \alpha_3 \frac{\partial^2 u}{\partial x^2}(z_s) + \alpha_4 \frac{\partial^2 u}{\partial x \partial y}(z_s) + \alpha_5 \frac{\partial^2 u}{\partial y^2}(z_s) \quad (6.12)$$

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 Laplace-Beltrami 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  $\tilde{f} : P \rightarrow \mathbb{S}^2$  which effectively resembles the conformal map  $f : \mathcal{M} \rightarrow \mathbb{S}^2$ . By the previous section, we can approximate the Laplace-Beltrami operator  $\Delta$  on  $P$ . Denote the approximated Laplace-Beltrami operator on the point cloud by  $\Delta_{PC}$ . The approximation allows us to solve the Laplace equation (5.2) on point clouds for a map  $\phi : P \rightarrow \mathbb{C}$ . More specifically, we solve the following equation

$$\Delta_{PC}\phi = 0 \quad (6.13)$$

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  $\phi$ . 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 [6]. Here, the regularity of a triangle  $[a_1, a_2, a_3]$  is defined by

$$\text{Regularity}[a_1, a_2, a_3] = \left| \alpha - \frac{\pi}{3} \right| + \left| \beta - \frac{\pi}{3} \right| + \left| \gamma - \frac{\pi}{3} \right|, \quad (6.14)$$

where  $\alpha, \beta$  and  $\gamma$  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 Laplace-Beltrami operator, it is necessary to find the  $k$  nearest neighboring data points  $z_s^1, z_s^2, \dots, z_s^k$  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 Equation (6.13) by considering

$$\min_{s,i,j} \text{Regularity}[z_s, z_s^i, z_s^j] \quad (6.15)$$

among all combinations of  $s, i$  and  $j$ .

After solving Equation (6.13) 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 second key step in the method in [6] 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 Laplace-Beltrami 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  $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \rightarrow \mathbb{C}$  again:

$$\Delta_{PC}\psi = 0 \quad (6.16)$$

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

$$\tilde{f} = P_S^{-1} \circ \psi \circ P_S \circ P_N^{-1} \circ \phi. \quad (6.17)$$

This step effectively replaces the second key step in the mesh parameterization algorithm in [6] which involves computing the Beltrami coefficients.

Altogether, by solving Equation (6.13) and Equation (6.16) and using a number of projections, we can obtain a conformal map  $\tilde{f} : P \rightarrow \mathbb{S}^2$ . Note that the method in [6] 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 [6], the entire initial parameterization result is used for computing a quasi-conformal map in order to cancel the distortion. The method is theoretically guaranteed by the composition formula (5.6) 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 southern-most regions as the boundary constraints and compute the remaining part of the spherical parameterization again, with the aid of the Laplace-Beltrami operator. The conformality of the final result is based on the accuracy of the boundary constraints, as well as the fact that harmonic maps are conformal maps under suitable boundary constraints. Therefore, in the continuous case, both methods are theoretically guaranteed for producing a conformal map.



However, in the discrete case, the two methods are advantageous under different conditions. For the case of triangular meshes, the Beltrami coefficients can be accurately computed and the composition formula (5.6) of quasi-conformal maps is accurate under the discretization. In this situation, the method in [6] is likely to produce a more accurate result since it is less dependent on the boundary constraints. While for the case of point clouds, we only have an approximation of the Laplace-Beltrami operator and there is no guarantee about the composition formula (5.6) of quasi-conformal maps. Hence, it is more suitable to use our proposed method as it only involves solving the Laplace Equation (6.16). However, since our proposed method is more dependent on the boundary constraints, the boundary constraints obtained from the initial parameterization result may contain small error and hence slightly affect the result in solving Equation (6.16). Therefore, it is desirable to perform some more iterations for obtaining a more accurate result.

It is noteworthy that in the parameterization algorithm in [6] for triangular meshes, no further steps are required after the second step. However, because of the abovementioned issue about the boundary constraints in the Laplace Equation (6.16), a few further iterations are necessary for enhancing the parameterization result. We call them the *North-South (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 the complex plane to guarantee 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  $\tilde{\phi} : (P_N \circ f)(P) \rightarrow \mathbb{C}$  by solving the Laplace equation

$$\Delta_{PC}\tilde{\phi} = 0 \quad (6.18)$$

with the boundary constraints  $\tilde{\phi}(x) = x$  for the outermost  $r\%$  of the data points on  $\mathbb{C}$ . After obtaining  $\tilde{\phi}$ , the inverse north-pole stereographic projection is again applied, followed by the south-pole stereographic projection. Then, we compute another harmonic map  $\tilde{\psi} : (P_S \circ P_N^{-1} \circ \tilde{\phi} \circ P_N \circ f)(P) \rightarrow \mathbb{C}$  by solving the Laplace equation

$$\Delta_{PC}\tilde{\psi} = 0 \quad (6.19)$$

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

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

We check whether the above updated parameterization result is close to the previous parameterization result  $\tilde{f}$ . 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.

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 Laplace-Beltrami operator can be easily constructed by computing the cotangent weights on the given mesh structures. Also,

solving Laplace equations on the complex plane requires only linear time. Hence, our proposed algorithm can serve as an alternative approach for computing spherical conformal parameterizations of genus-0 closed triangular meshes in linear time.

---

**Algorithm 1:** Our proposed spherical conformal parameterization algorithm.

---

**Input:** A genus-0 point cloud  $P$ .

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

- 1 Approximate the Laplace-Beltrami operator on  $P$  and denote the approximation by  $\Delta_{PC}$ ;
  - 2 Find the most regular triple of points  $a_1, a_2, a_3$  by solving the minimization problem (6.15);
  - 3 Obtain a map  $\phi : P \rightarrow \overline{\mathbb{C}}$  by solving the Laplace equation (6.13);
  - 4 Apply the inverse stereographic projection  $P_N^{-1} : \overline{\mathbb{C}} \rightarrow \mathbb{S}^2$  on  $\phi(P)$ ;
  - 5 Apply the south-pole stereographic projection  $P_S : \mathbb{S}^2 \rightarrow \overline{\mathbb{C}}$  on  $(P_N^{-1} \circ \phi)(P)$ ;
  - 6 Solve the Laplace equation (6.16) for  $\psi : (P_S \circ P_N^{-1} \circ \phi)(P) \rightarrow \mathbb{C}$ ;
  - 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  $\tilde{f}$  by  $f$ ;
  - 10    Solve the Laplace equation (6.18) for  $\tilde{\phi} : (P_N \circ \tilde{f})(P) \rightarrow \mathbb{C}$ ;
  - 11    Solve the Laplace equation (6.19) for  $\tilde{\psi} : (P_S \circ P_N^{-1} \circ \tilde{\phi} \circ P_N \circ \tilde{f})(P) \rightarrow \mathbb{C}$ ;
  - 12    Update  $f$  by  $P_S^{-1} \circ \tilde{\psi} \circ P_S \circ P_N^{-1} \circ \tilde{\phi} \circ P_N \circ \tilde{f}$ ;
  - 13 **until**  $\text{mean}(\|f(p_i) - \tilde{f}(p_i)\|^2) < \epsilon$ ;
- 

**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 [6], Choi *et al.* proved the following theorem:

**THEOREM 6.1** (See [6], 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 *et al.* 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 6.1 for point clouds by considering two sets of points. The extension is as follows:

THEOREM 6.2. Let  $\{u_i\}_{i=0}^m$  and  $\{v_j\}_{j=0}^n$  be two sets of points on  $\mathbb{C}$ . Then

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

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 [6]. Note that for any  $z = x + iy$ , we have

$$\begin{aligned} P_S(P_N^{-1}(z)) &= P_S(P_N^{-1}(x + iy)) \\ &= \frac{-\frac{2x}{1+x^2+y^2}}{1 + \frac{-1+x^2+y^2}{1+x^2+y^2}} + i \frac{\frac{2y}{1+x^2+y^2}}{1 + \frac{-1+x^2+y^2}{1+x^2+y^2}} \\ &= \frac{-x}{x^2+y^2} + i \frac{y}{x^2+y^2} \\ &= \frac{-Re(z)}{|z|^2} + i \frac{Im(z)}{|z|^2}. \end{aligned} \tag{6.21}$$

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

$$\begin{aligned} & \left( \sum_{i=1}^m \|\lambda u_i - \lambda u_0\| \right) \left( \sum_{j=1}^n \|P_S(P_N^{-1}(\lambda v_j)) - P_S(P_N^{-1}(\lambda v_0))\| \right) \\ &= \left( \sum_{i=1}^m \|\lambda u_i - \lambda u_0\| \right) \left( \sum_{j=1}^n \left\| \frac{-Re(\lambda v_j)}{|\lambda v_j|^2} + i \frac{Im(\lambda v_j)}{|\lambda v_j|^2} - \frac{-Re(\lambda v_0)}{|\lambda v_0|^2} + i \frac{Im(\lambda v_0)}{|\lambda v_0|^2} \right\| \right) \\ &= \left( \lambda \sum_{i=1}^m \|u_i - u_0\| \right) \left( \frac{\lambda}{\lambda^2} \sum_{j=1}^n \left\| \frac{-Re(v_j)}{|v_j|^2} + i \frac{Im(v_j)}{|v_j|^2} - \frac{-Re(v_0)}{|v_0|^2} + i \frac{Im(v_0)}{|v_0|^2} \right\| \right) \\ &= \left( \sum_{i=1}^m \|u_i - u_0\| \right) \left( \sum_{j=1}^n \|P_S(P_N^{-1}(v_j)) - P_S(P_N^{-1}(v_0))\| \right). \end{aligned} \tag{6.22}$$

Therefore, the product is an invariance.  $\square$

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

$$d_p = \text{mean}(\{|P_N(f(z)) - x_N| : z \in \mathcal{N}^k(f^{-1}(v_N))\}) \quad (6.23)$$

and

$$d_s = \text{mean}(\{|P_S(f(z)) - x_S| : z \in \mathcal{N}^k(f^{-1}(v_S))\}). \quad (6.24)$$

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

$$\lambda = \frac{\sqrt{d_p \times d_s}}{d_p}. \quad (6.25)$$

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

$$\tilde{d}_p = \lambda d_p = \frac{\sqrt{d_p \times d_s}}{d_p} \times d_p = \sqrt{d_p \times d_s}. \quad (6.26)$$

Also, by Theorem 6.2, we have

$$\tilde{d}_p \times \tilde{d}_s = d_p \times d_s. \quad (6.27)$$

Therefore,

$$\tilde{d}_s = d_p \times d_s \times \frac{1}{\tilde{d}_p} = \sqrt{d_p \times d_s}. \quad (6.28)$$

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.

---

**Algorithm 2:** Our proposed balancing scheme for better distribution.

---

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

**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

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

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  $\lambda(P_N(f(P)))$ ;
-

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

---

**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  $f : P \rightarrow \mathbb{S}^2$ ;
  - 2 Compute a triangulation  $T$  on  $f(P)$  using the spherical Delaunay algorithm;
  - 3 Use  $T$  to form a triangular mesh  $M = (P, T)$ ;
- 

Before ending this section, we make an important remark about an extension of our proposed framework. In fact, our proposed parameterization and meshing scheme can be easily extended for point clouds with disk topology. In this case, we first extend the double covering technique [15, 7] 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  $\tilde{P}$  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 moving least square 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  $P$  and  $P'$  and obtain a genus-0 point cloud

Method	maximum position error	average position error
Local mesh method [20]	1.3427	0.0179
MLS with the Wendland weight [46, 47]	3.3074	0.1696
MLS with the special weight [25]	0.0427	0.0006
MLS with our proposed weight	0.0245	0.0004

Method	maximum position error	average position error
Local mesh method [20]	1.5148	0.0271
MLS with the Wendland weight [46, 47]	2.0082	0.0803
MLS with the special weight [25]	0.0110	0.0001
MLS with our proposed weight	0.0103	0.0002

TABLE 7.1

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

$\tilde{P}$ .

Step 5: Create the Laplace-Beltrami operator for  $\tilde{P}$  using the derivatives on  $P$  and  $P'$ .

Then, we can apply the abovementioned spherical conformal parameterization algorithm on  $\tilde{P}$  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 [51] and the Stanford 3D Scanning Repository [52]. The mentioned algorithms are implemented in MATLAB. The sparse linear systems for the Laplace equations are solved using the built-in backslash operator ( $\backslash$ ) in MATLAB. All experiments are performed on a PC with an Intel(R) Core(TM) i5-3470 CPU @3.20 GHz processor and 8.00 GB RAM.

**7.1. The performance of our approximation of the Laplace-Beltrami operator.** In this work, we apply the moving least square method with a new weight function for approximating the Laplace-Beltrami 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 moving least square method with several weighting functions for approximating the Laplace-Beltrami operator on point clouds. More specifically, we compare the performance of the following methods:

1. The local mesh method [20],
2. The moving least square method with the Wendland weight function [46, 47],
3. The moving least square method with the special weight function [25], and

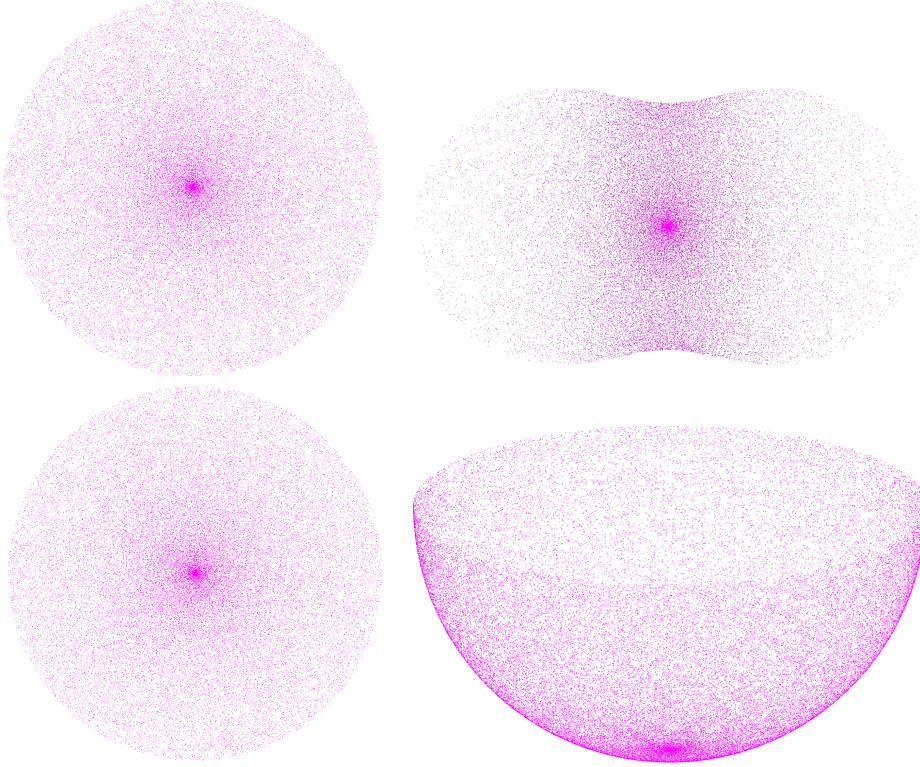


FIG. 7.1. *Two experiments for assessing the approximation accuracies of the Laplace-Beltrami operator. In each experiment, we generate a point cloud on the unit disk and transform it using a conformal map with an explicit formula. We then approximate the Laplace-Beltrami operator on the transformed point cloud and solve the Laplace equation back onto the unit disk. Top: the first experiment. Bottom: the second experiment.*

#### 4. The moving least square method with our proposed weight function.

Experiments are carried out for assessing the numerical accuracies of the above-mentioned approaches. Figure 7.1 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 Laplace-Beltrami 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 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 Laplace-Beltrami operator.

Table 7.1 illustrates the approximation error of different approaches in the two experiments. It is noteworthy that in both experiments, the moving least square method with our proposed weight function produces approximations which are much

Point clouds	No. of points	Our proposed method	Spherical embedding [50]
		Time (s) / Mean( $ distortion $ ) / SD( $ distortion $ )	
Hippocampus	10242	13.0919 / 1.2855 / 1.4701	37.4124 / 14.3072 / 19.6461
Max Planck	21530	30.7785 / 0.7326 / 1.0803	87.0887 / 8.6058 / 14.0857
Bulldog	49797	88.9297 / 1.5432 / 2.9183	206.9920 / 16.2010 / 21.1544
Chinese Lion	50002	95.8935 / 1.8474 / 1.9286	212.5685 / 19.1579 / 22.7259
Bimba	74764	198.6064 / 0.6227 / 0.8129	360.7178 / 18.0340 / 20.6272
Igea	134345	427.7658 / 0.7076 / 1.4273	731.8661 / 5.0853 / 8.2623
Armadillo	172974	676.4106 / 1.4167 / 1.6855	995.7537 / 23.2354 / 23.9892
Lion Vase	256094	1305.9013 / 2.0920 / 4.1052	1484.7682 / 17.8501 / 21.9588

TABLE 7.2

Performance of two spherical 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 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.

more accurate those produced by the local mesh method and the moving least square method with the Wendland weight [46, 47]. With similar and negligible average position errors, our proposed scheme reduces the maximum position errors by about 25% on average when compared with the moving least square method with the special weight [25]. The comparisons reflect the advantage of our proposed method for approximating the Laplace-Beltrami operator.

**7.2. Performance of our proposed spherical conformal parameterization.** After demonstrating the advantage of our approximation scheme for the Laplace-Beltrami operator, we investigate the performance of our proposed spherical conformal parameterization algorithm for genus-0 point clouds. Figure 7.2 and Figure 7.3 show the results of parameterizing a lion point cloud and a bulldog point cloud using our proposed parameterization method.

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 Figure 7.2 and Figure 7.3 that the histograms of the angle differences highly concentrate at 0. Besides, 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.

We compare our proposed spherical conformal parameterization method with the spherical embedding method proposed by Zwicker and Gotsman [50], which also computes a spherical parameterization for a genus-0 point cloud. In our experiment,  $k = 25$  nearest neighbors of every point are used for approximating the Laplace-



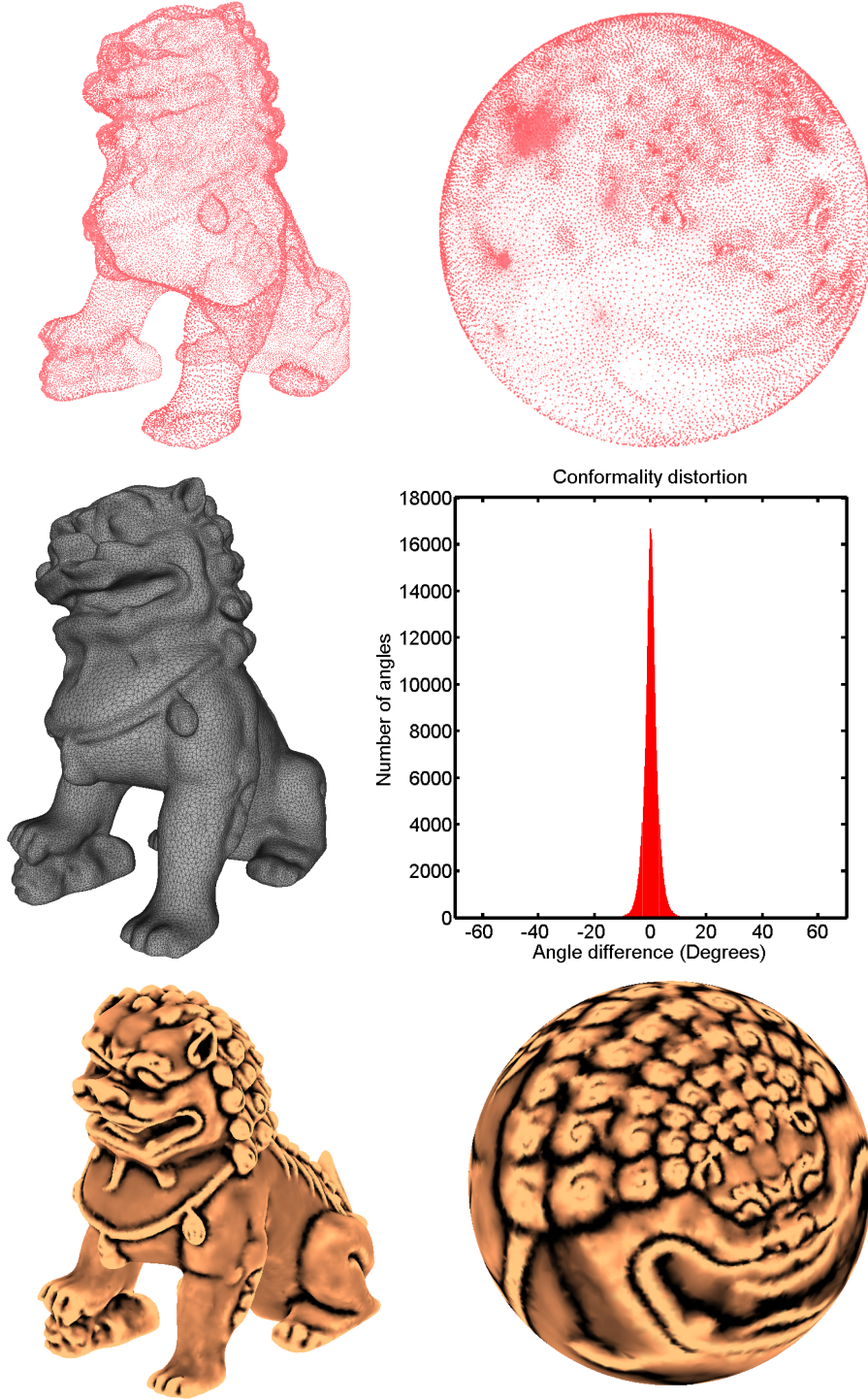


FIG. 7.2. Parameterizing a lion point cloud. Top left: A lion point cloud. Top right: The spherical conformal parameterization of the lion point cloud. Middle left: A triangulation created by our method. Middle right: The conformality distortion of the parameterization based on the triangulation. Bottom: The triangulated point cloud and the spherical parameterization result colored with the approximated mean curvature at each vertex.

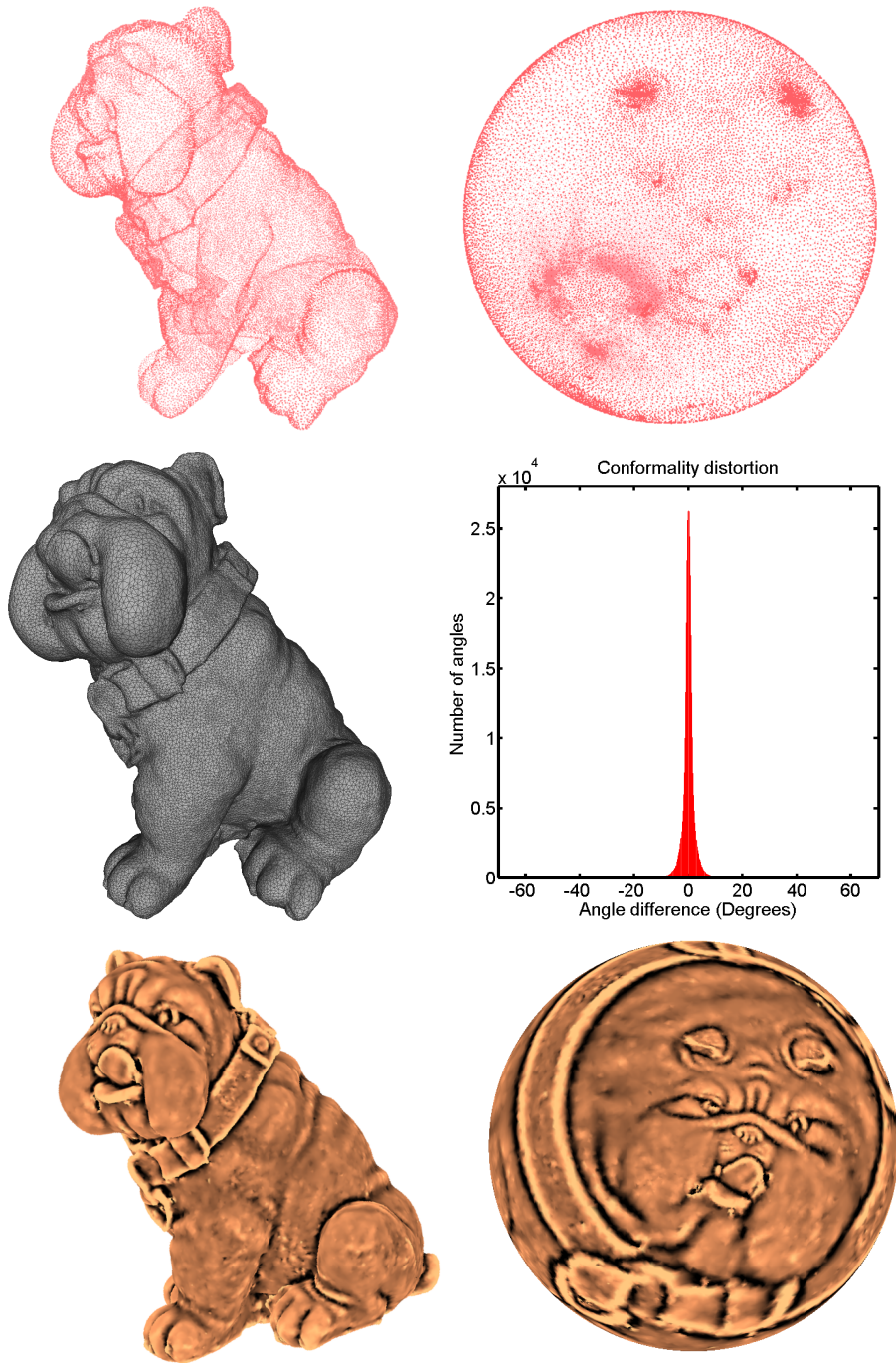


FIG. 7.3. *Parameterizing a bulldog point cloud. Top left: A bulldog point cloud. Top right: The spherical conformal parameterization of the bulldog point cloud. Middle left: A triangulation created by our method. Middle right: The conformality distortion of the parameterization based on the triangulation. Bottom: The triangulated point cloud and the spherical parameterization result colored with the approximated mean curvature at each vertex.*

Beltrami operator in Algorithm 1. The stopping threshold for the N-S reiteration is set to be  $\epsilon = 0.0001$ . Table 7.2 summarizes the computational time and the conformality distortion of our proposed scheme and the spherical embedding method for computing the spherical parameterizations of genus-0 point clouds. In all of the experiments, our proposed method produces parameterizations with better conformality. Moreover, our method is more efficient than the algorithm in [50]. The above results indicate that our parameterization algorithm preserves the local geometry of the point clouds very well.

**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 triangulation on the original point clouds. Besides Figure 7.2 and Figure 7.3, some more examples of triangulations created by our approach are shown in Figure 7.4. 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, non-convex 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, post-processing steps are not required in our meshing scheme.

We compare our meshing method with three existing meshing approaches. As an example of parameterization-based approaches, Zwicker and Gotsman [50] 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 [30] and the Tight Cocone algorithm [8]. Figure 7.5 provides a comparison between our method and the three mentioned approaches. It can be observed that our meshing scheme and the Tight Cocone algorithm [8] produce high quality triangulations, while the triangulations produced by the approaches in [50] and [30] 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 post-processing is needed in our meshing scheme. The comparison demonstrates the advantages of our proposed meshing scheme. A further comparison between our method and the Tight Cocone algorithm [8] 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 7.6. 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 clouds and thus generate quad mesh representations. Because of the conformality of our parameterization scheme, the resulting quad meshes are with high quality. Also, the meshes are guaranteed to be topology preserving.

Before ending this subsection, we demonstrate the significance of our proposed

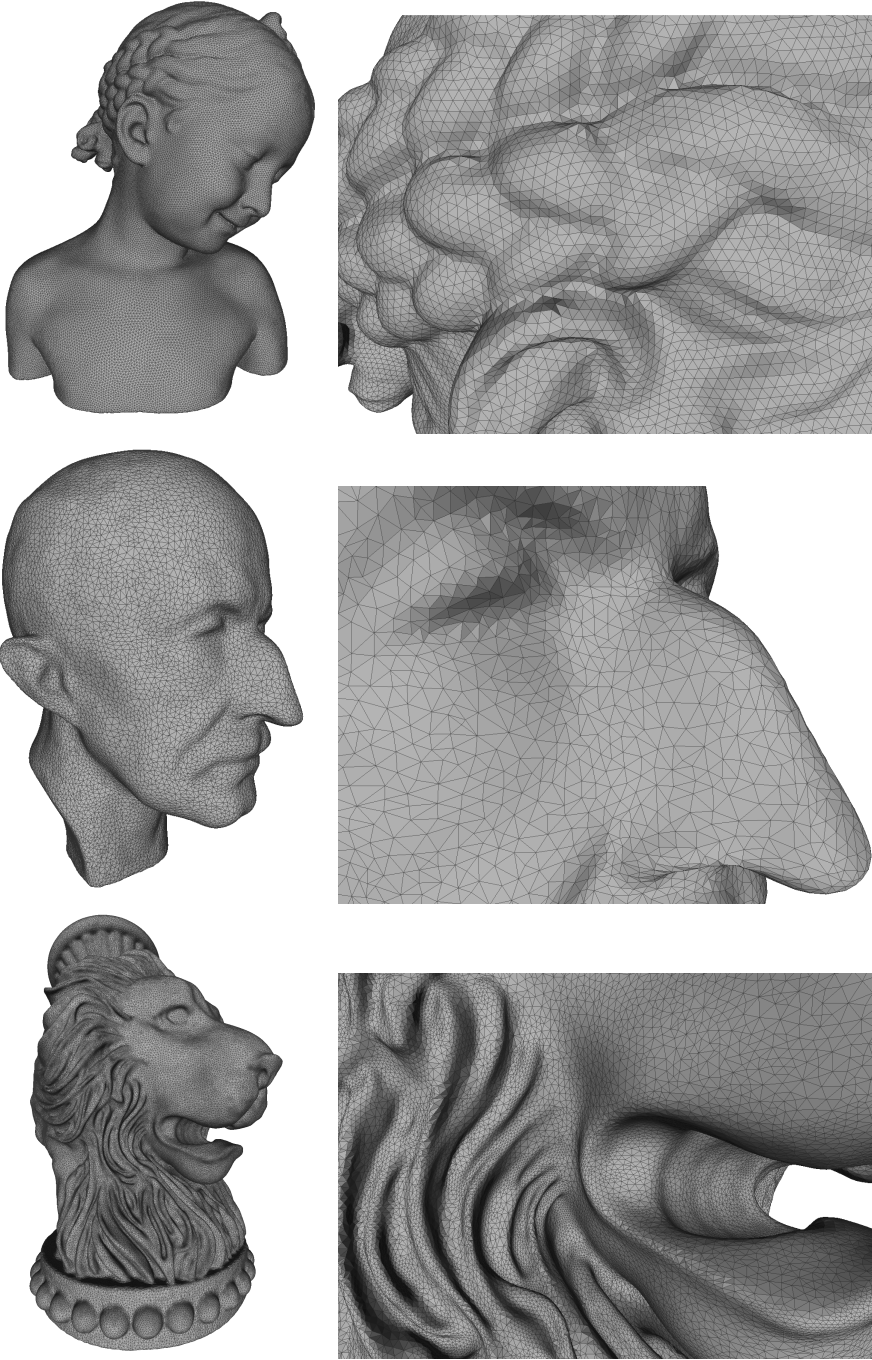


FIG. 7.4. Meshes generated by our proposed method and a zoom-in of them. The regularity of the triangulations is attributed to our spherical conformal parameterization and the spherical Delaunay method.



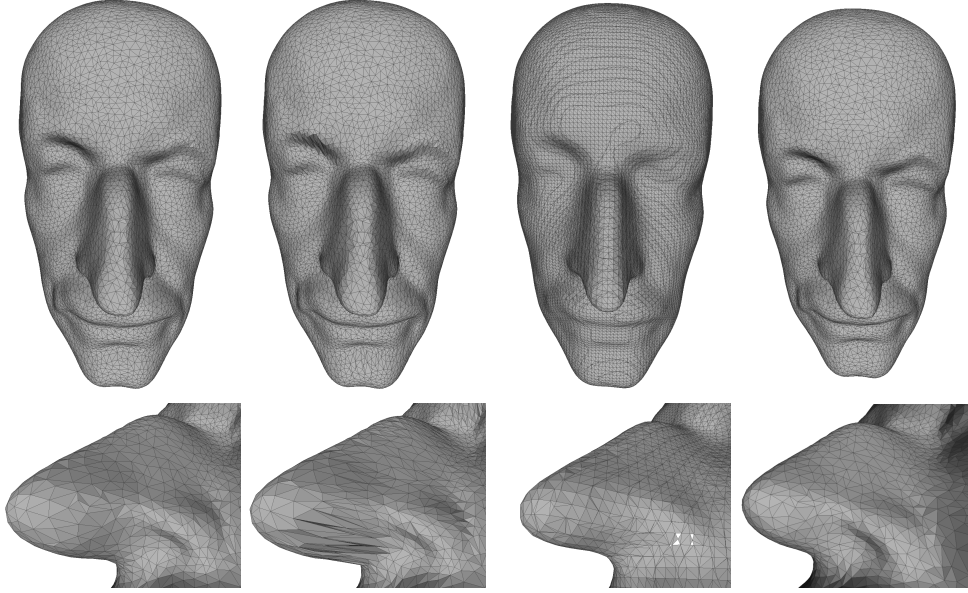


FIG. 7.5. A comparison between our meshing scheme and other approaches. A front view of the triangulated point cloud and a zoom-in of the nose are shown for each method. Left to right: Our meshing result, the method in [50], the marching cubes algorithm [30] and the Tight Cocone algorithm [8].

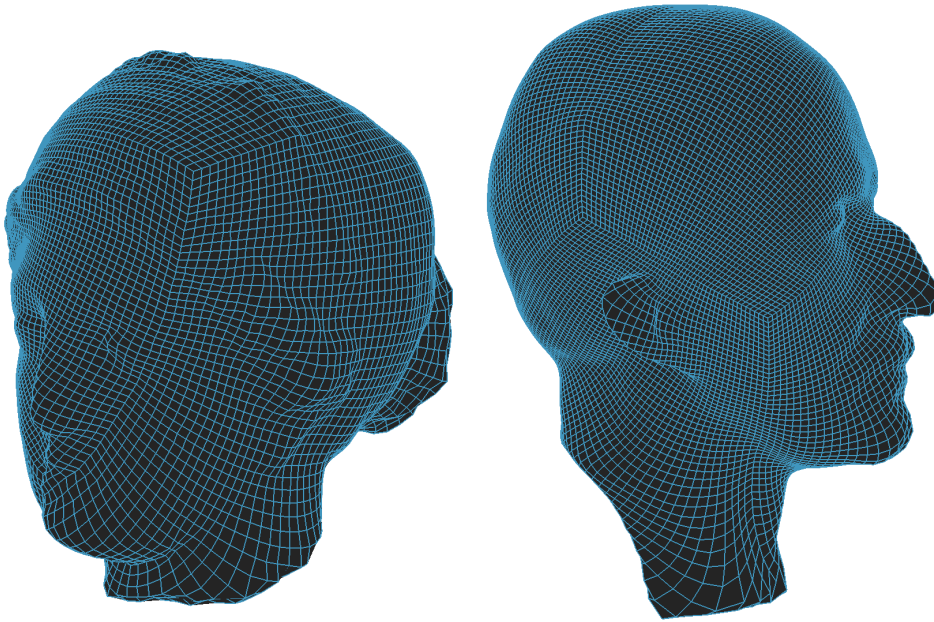


FIG. 7.6. Quad mesh generation on point clouds using our proposed method.

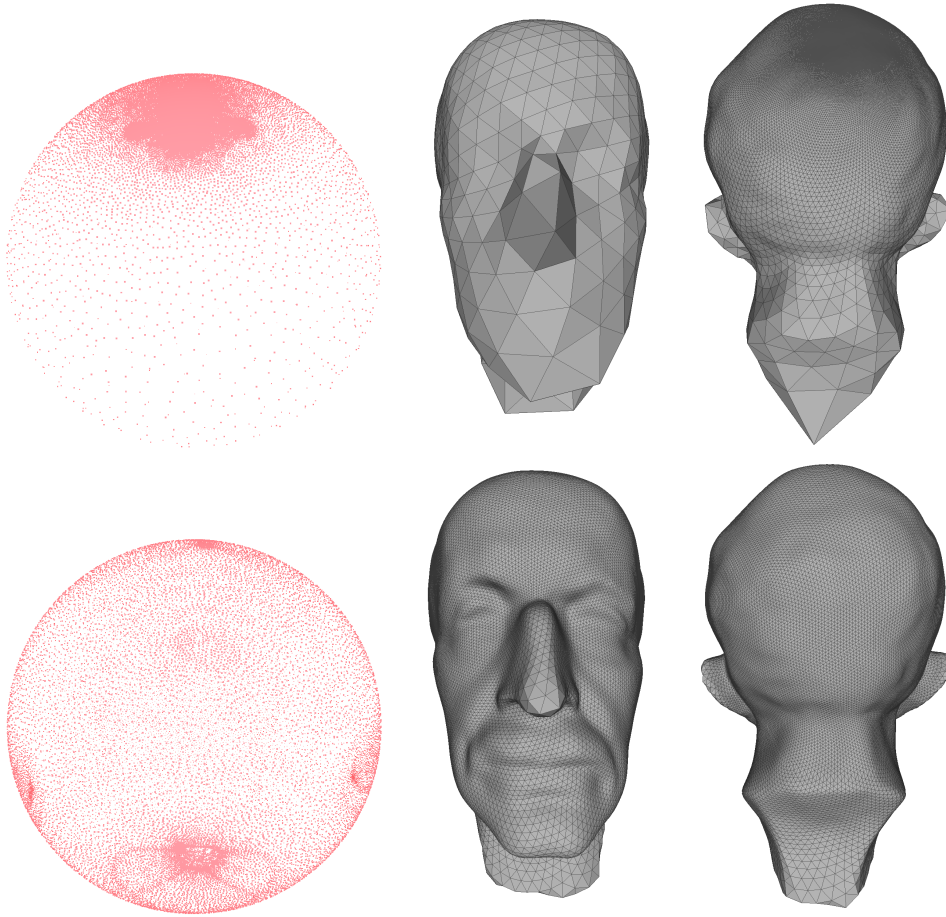


FIG. 7.7. *The effect of our balancing scheme on meshing a genus-0 point cloud of Max Planck. Top left: A spherical conformal parameterization without the balancing scheme. Bottom left: A spherical conformal parameterization with the balancing scheme. Middle: The front view of the the meshing results by interpolation with the aid of the parameterizations. Right: The back view.*

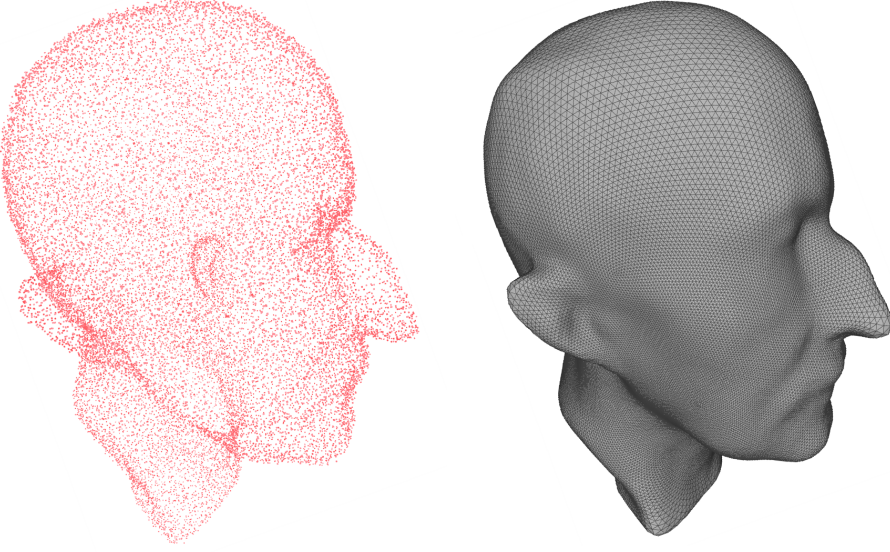


FIG. 7.8. *Meshing a geometrically noisy point cloud.*

balancing scheme in the spherical conformal parameterization. The redistribution is vital for the meshing quality. Figure 7.7 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 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 them back onto the filtered point clouds to produce meshed surfaces. Figure 7.8 shows an example of geometrically noisy point cloud and our meshing result.

We can also construct a faithful triangulated mesh on a geometrically noisy point cloud without any filtering or sampling procedure. Figure 7.9 shows the triangulation result of our meshing scheme and the Tight Cocone algorithm [8] on the noisy point cloud in Figure 7.8. 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 holes on the result by [8], while our meshing scheme guarantees a regular and topology preserving triangulation even for noisy input point clouds.

Besides, it is common that the sampling processes result in non-uniformly sampled point clouds. In particular, there may be large holes on certain parts of the point

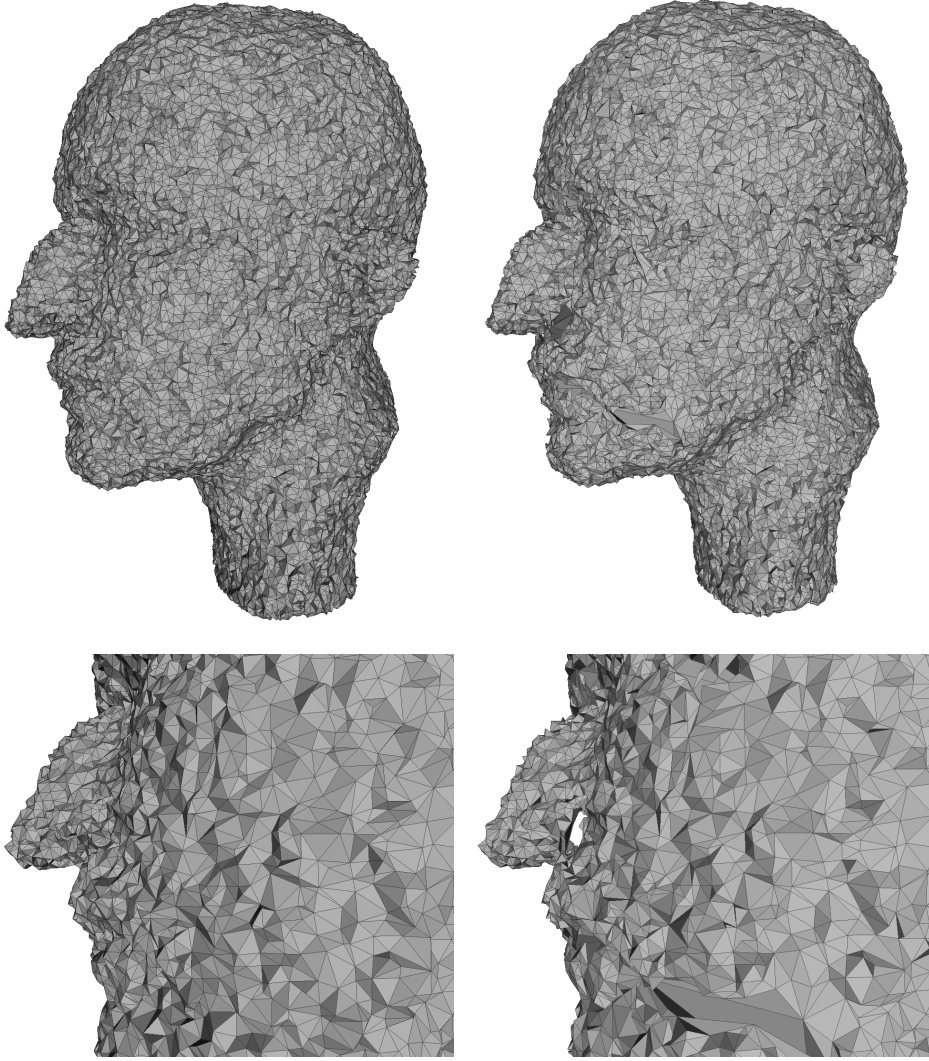


FIG. 7.9. Comparison of our meshing scheme and the Tight Cocone algorithm [8] on a geometrically noisy point cloud. All points are considered in the computations. Left: Our meshing result with a zoom-in of the nose. Right: The result of the Tight Cocone algorithm with a zoom-in of the nose.



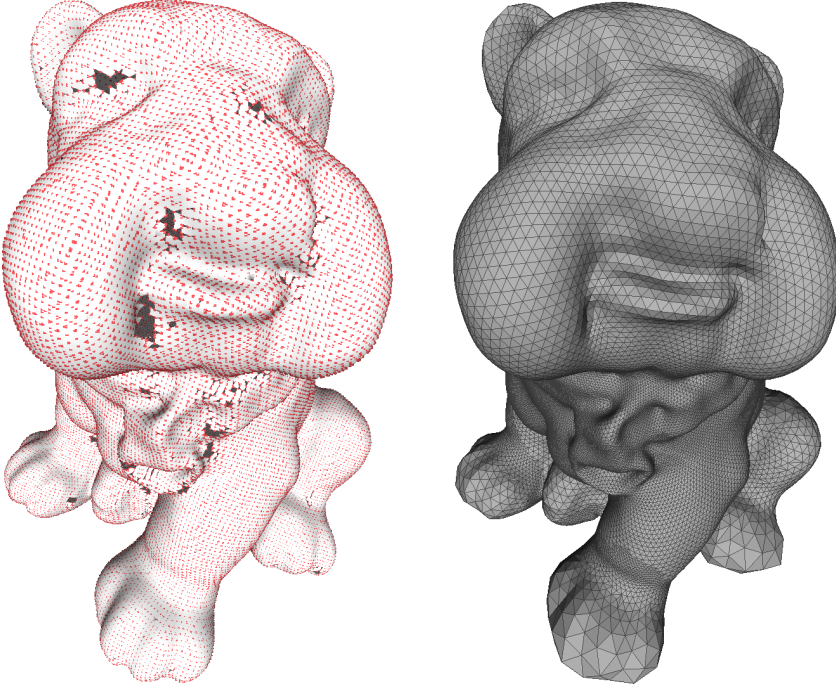


FIG. 7.10. *Meshing a topologically noisy point cloud with unwanted holes.*

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 7.10 shows the performance of our algorithm on topologically noisy point clouds. 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. Then, we can progressively subdivide the sphere using existing subdivision methods, such as the butterfly subdivision method [9] and the loop subdivision method [29]. 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 Figure 7.11 and Figure 7.12. In our examples, the subdivisions are generated using the loop subdivision method [29]. 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.

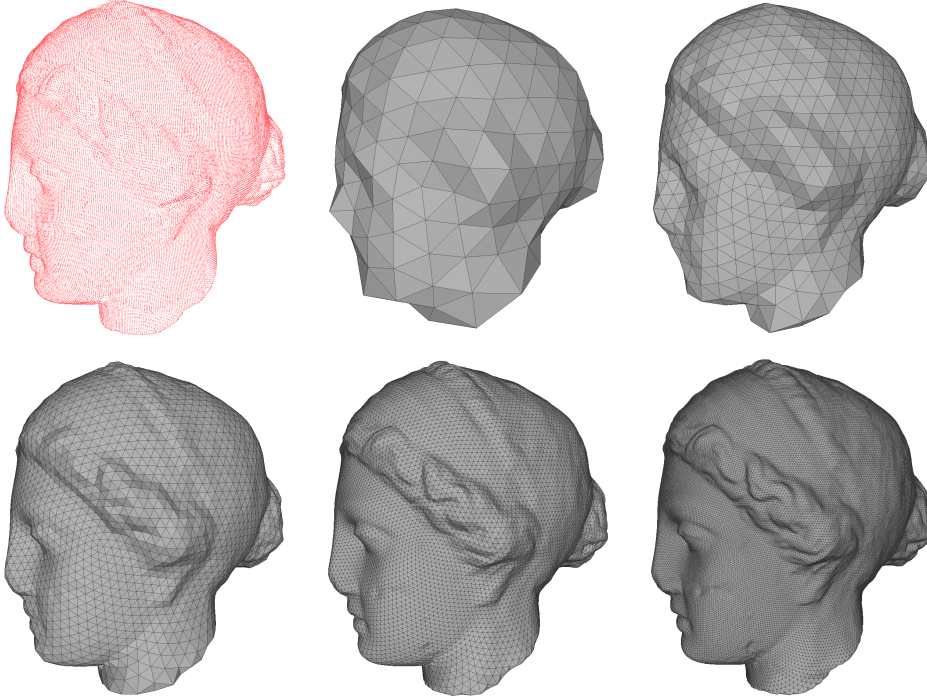


FIG. 7.11. *Multilevel representations of a genus-0 point cloud of Igea. For better visualizations, we create mesh structures on the representations. Top left: a point cloud with 134345 points of Igea. Top middle to Bottom right: the multilevel representations with 0, 1, 2, 3 and 4 subdivisions. The representations are with 642, 2562, 10242, 40962 and 163842 points respectively.*

**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 [6]. Firstly, we enhanced the accuracy for approximating the Laplace-Beltrami operator on point clouds by introducing a new weight function. Secondly, we proposed an iterative scheme called the N-S reiteration to replace a key step in [6] for achieving better conformality. Thirdly, 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. Besides, 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. Furthermore, our meshing framework for genus-0 point clouds can be easily extended for meshing point clouds with disk topology. In the future, we plan to extend our method to handle point clouds with arbitrary topology.

#### REFERENCES



FIG. 7.12. 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.

- [1] P. Azariadis, *Parameterization of clouds of unorganized points using dynamic base surfaces*. Computer-Aided Design 36, 2004, pp. 607–623.
- [2] P. Azariadis and N. Sapidis, *Efficient parameterization of 3D point-sets using recursive dynamic base surfaces*. Advances in Informatics, Lecture Notes in Computer Science, Volume 3746, 2005, pp. 296–306.
- [3] 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.
- [4] M. Belkin, J. Sun, and Y. Wang. *Constructing Laplace operator from point clouds in  $\mathbb{R}^d$* . In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, 2009, pp. 1031–1040.
- [5] P. Breitenkopf, H. Naceur, A. Rassineux, and Pierre Villon, *Moving least squares response surface approximation: Formulation and metal forming applications*. Computers & Structures, Volume 83, Issues 17–18, 2005, pp. 1411–1428.
- [6] P.T. Choi, K.C. Lam, and L.M. Lui, *FLASH: Fast landmark aligned spherical harmonic parameterization for genus-0 closed brain surfaces*, SIAM Journal on Imaging Sciences, Volume 8, Issue 1, 2015, pp. 67–94.
- [7] P.T. Choi and L.M. Lui, *A linear algorithm for disk conformal parameterization of simply-connected open surfaces*. Preprint, arXiv:1508.00396v1, 2015.
- [8] T. K. Dey and S. Goswami. *Tight Cocone: A water tight surface reconstructor*. Proceedings of the eighth ACM symposium on Solid modeling and applications, 2003, pp. 127–134.
- [9] N. Dyn, D. Levine, and J.A. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control*. ACM Transactions on Graphics, Volume 9, Issue 2, 1990, pp. 160–169.
- [10] M. S. Floater, *Meshless parameterization and B-spline surface approximation*. Proceedings of the Ninth IMA Conference on the Mathematics of Surfaces, 2000, pp. 1–18.
- [11] M. S. Floater and M. Reimers, *Meshless parameterization and surface reconstruction*. Computer Aided Geometric Design, Volume 18, Issue 2, 2001, pp. 77–92.
- [12] M. Floater and K. Hormann, *Parameterization of triangulations and unorganized points*, Tutorials on Multiresolution in Geometric Modelling, 2002, pp. 287–316.
- [13] M. Floater and K. Hormann, *Surface parameterization: A tutorial and survey*, Advances in Multiresolution for Geometric Modelling, 2005, pp.157–186.
- [14] F. Gardiner and N. Lakic, *Quasiconformal Teichmüller theory*. American Mathematics Society, 2000.
- [15] X. Gu and S.T. Yau, *Global conformal surface parameterization*, Eurographics Symposium on Geometry Processing, 2003, pp. 127–137.
- [16] X. Guo, X. Li, Y. Bao, X. Gu, H. Qin, *Meshless thin-shell simulation based on global conformal parameterization*. IEEE Transactions on Visualization and Computer Graphics, Volume 12, Issue 3, 2006, pp.375–385.
- [17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface reconstruction from unorganized points*. Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, 1992, pp. 71–78.
- [18] K. Hormann and M. Reimers. *Triangulating point clouds with spherical topology*. Curve and Surface Design: Saint-Malo 2002, Modern Methods in Applied Mathematics, 2003, pp. 215–224.
- [19] K. Hormann, B. Lévy, and A. Sheffer, *Mesh parameterization: Theory and practice*, SIGGRAPH 2007 Course Notes, 2:1–122, 2007.
- [20] R. Lai, J. Liang, H. Zhao, *A local mesh method for solving PDEs on point clouds*. Inverse Problem and Imaging, Volume 7, Issue 3, 2013, pp. 737–755.
- [21] R. Lai, Z. Wen, W. Yin, X.F. Gu, and L.M. Lui, *Folding-free global conformal mapping for genus-0 surfaces by harmonic energy minimization*. Journal of Scientific Computing, Volume 58, Issue 3, 2013, pp. 705–725.
- [22] P. Lancaster and K. Salkauskas, *Surfaces generated by moving least squares methods*, Mathematics of Computation, Volume 37, 1981, pp. 141–158.
- [23] D. Levin, *The approximation power of moving least-squares*. Mathematics of Computation, Volume 67, Issue 224, 1998, pp. 1517–1531.
- [24] B. Lévy, *Constrained texture mapping for polygonal meshes*. Proceedings of ACM SIGGRAPH 2001, 2001, pp. 417–424.
- [25] J. Liang, R. Lai, A. Wong and H. Zhao, *Geometric understanding of point clouds using Laplace-Beltrami operator*, IEEE Conference on Computer Vision and Pattern Recognition, 2012.
- [26] J. Liang and H. Zhao, *Solving partial differential equations on point clouds*, SIAM Journal of Scientific Computing, Volume 35, Issue 3, 2013, pp. 1461–1486.
- [27] H Lipson and M Kurman, *Fabricated: The new world of 3D printing*. Wiley, 2013.
- [28] W.K. Liu, S. Li and T. Belytschko, *Moving least-square reproducing kernel methods (I) method-*

- ology and convergence, Computer Methods in Applied Mechanics and Engineering, Volume 143, 1997, pp. 113–154.
- [29] C.T. Loop, *Smooth subdivision surfaces based on triangles*. M.S. Mathematics thesis, University of Utah, 1987.
  - [30] W.E. Lorensen and H.E. Cline, *Marching cubes: A high resolution 3D surface construction algorithm*. ACM SIGGRAPH Computer Graphics, Volume 21, Issue 4, 1987, pp. 163–169.
  - [31] F. Lozes, A. Elmoataz, and O. Lézoray, *Partial difference operators on weighted graphs for image processing on surfaces and point clouds*. IEEE Transactions on Image Processing, Volume 23, Issue 9, 2014, pp. 3896–3909.
  - [32] L.M. Lui, K.C. Lam, T.W. Wong, and X.F. Gu, *Texture map and video compression using Beltrami representation*. SIAM Journal on Imaging Sciences, Volume 6, Issue 4, 2013, pp. 1880–1902.
  - [33] C.B. Macdonald, B. Merriman, and S.J. Ruuth, *Simple computation of reaction-diffusion processes on point clouds*. Proceedings of the National Academy of Sciences, Volume 110, Issue 23, 2014, pp. 9209–9214.
  - [34] Q. Meng, B. Li, H. Holstein, and Y. Liu, *Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBF networks*. Pattern Recognition, Volume 46, Issue 8, 2013, pp. 2361–2375.
  - [35] B. Nayroles, G. Touzot, and P. Villon, *Generalizing the finite element method: Diffuse approximation and diffuse elements*. Computational Mechanics, Volume 10, Issue 5, 1992, pp. 307–318.
  - [36] 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.
  - [37] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Experimental Mathematics, Volume 2, Issue 1, 1993, pp. 15–36.
  - [38] F. Rengier, A. Mehndiratta, H. von Tengg-Kobligk, C.M. Zechmann, R. Unterhinninghofen, H.U. Kauczor, and F.L. Giesel, *3D printing based on imaging data: review of medical applications*. International Journal of Computer Assisted Radiology and Surgery, Volume 5, Issue 4, 2010, pp. 335–341.
  - [39] P.V. Sander, J. Snyder, S.J. Gortler, and H. Hoppe, *Texture mapping progressive meshes*. Proceedings of ACM SIGGRAPH 2001, 2001, pp. 409–416.
  - [40] R. Schoen and S.T. Yau, *Lectures on differential geometry*. International Press, 1994.
  - [41] R. Schoen and S. Yau, *Lectures on harmonic maps*. Cambridge, MA: Harvard University, International Press, 1997.
  - [42] A. Sheffer, E. Praun and K. Rose, *Mesh parameterization methods and their applications*, Foundations and Trends in Computer Graphics and Vision, Volume 2, Issue 2, 2006, pp. 105–171.
  - [43] D. Shepard, *A two-dimensional interpolation function for irregularly spaced points*, in Proceedings of the 23rd ACM National Conference, 1968, pp. 517–524.
  - [44] G. Tewari, C. Gotsman, and S.J. Gortler, *Meshing genus-1 point clouds using discrete one-forms*. Computers and Graphics, Volume 30, 2006, pp. 917–926.
  - [45] L. Wang, B. Yuan and Z. Miao, *3D point clouds parameterization algorithm*. 9th International Conference on Signal Processing, 2008, pp. 1410–1413.
  - [46] H. Wendland, *Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree*. Advances in Computational Mathematics, Volume 4, Issue 1, 1995, pp. 389–396.
  - [47] H. Wendland, *Local polynomial reproduction and moving least squares approximation*. IMA Journal of Numerical Analysis, Volume 21, Issue 1, 2001, pp. 285–300.
  - [48] L. Zhang, L. Liu, C. Gotsman, and H. Huang, *Mesh Reconstruction by Meshless Denoising and Parameterization*. Computers and Graphics, Volume 34, Issue 3, 2010, pp. 198–208.
  - [49] 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*. Numerische Mathematik, Volume 121, Issue 4, 2012, pp. 671–703.
  - [50] M. Zwicker and C. Gotsman, *Meshing point clouds using spherical parameterization*. Eurographics Symposium on Point-Based Graphics, 2004.
  - [51] AIM@SHAPE Shape Repository. <http://shapes.aim-at-shape.net/>
  - [52] Stanford 3D Scanning Repository. <http://graphics.stanford.edu/data/3Dscanrep/>