Parallelizable global conformal parameterization of simply-connected surfaces via partial welding*

Gary P. T. Choi†, Yusan Leung-Liu‡, Xianfeng Gu§, and Lok Ming Lui¶

arXiv:1903.12359v2 [cs.CG] 2 Mar 2020

Abstract.
Conformal surface parameterization is useful in graphics, imaging and visualization, with applications to texture mapping, atlas construction, registration, remeshing and so on. With the increasing capability in scanning and storing data, dense 3D surface meshes are common nowadays. While meshes with higher resolution better resemble smooth surfaces, they pose computational difficulties for the existing parameterization algorithms. In this work, we propose a novel parallelizable algorithm for computing the global conformal parameterization of simply-connected surfaces via partial welding maps. A given simply-connected surface is first partitioned into smaller subdomains. The local conformal parameterizations of all subdomains are then computed in parallel. The boundaries of the parameterized subdomains are subsequently integrated consistently using a novel technique called partial welding, which is developed based on conformal welding theory. Finally, by solving the Laplace equation for each subdomain using the updated boundary conditions, we obtain a global conformal parameterization of the given surface, with bijectivity guaranteed by quasi-conformal theory. By including additional shape constraints, our method can be easily extended to achieve disk conformal parameterization for simply-connected open surfaces and spherical conformal parameterization for genus-0 closed surfaces. Experimental results are presented to demonstrate the effectiveness of our proposed algorithm. When compared to the state-of-the-art conformal parameterization methods, our method achieves a significant improvement in both computational time and accuracy.

Key words. Conformal parameterization, conformal welding, parallelization, simply-connected surface

AMS subject classifications. 65D18, 68U05, 52C26, 68W10

1. Introduction.

From mobile games to high-resolution movies, from small 3D printed desk toys to large aircraft engines, 3D geometric models are everywhere nowadays. With the advancement of computing technologies and scanning devices, 3D geometric data can be created or acquired easily. However, at the same time, the scale of the data grows rapidly. In many situations, it is necessary to handle dense geometric data with hundreds of thousands, or even millions of vertices.

In geometry processing, a common representation of 3D objects is triangulated 3D surfaces. To simplify various tasks that are to be performed on the 3D surfaces, one possible way is to transform the 3D surfaces into a simpler 3D shape or a 2D shape. This process is known as surface parameterization. With the aid of surface parameterization, we can perform the tasks on the simpler domain and transform the results back to the original 3D surfaces instead of working on them directly. For instance, under surface parameterization, PDEs on complicated

*Submitted to the editors DATE.

Funding: This work was supported in part by the Croucher Foundation (to Gary P. T. Choi) and HKRGC GRF with Project ID: 2130447 (to Lok Ming Lui).

†John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA (pchoi@g.harvard.edu).

‡Department of Mathematics, The Chinese University of Hong Kong, Hong Kong (ylleung@math.cuhk.edu.hk).

§Department of Computer Science, Stony Brook University, Stony Brook, NY 11794, USA (gu@cs.stonybrook.edu).

¶Department of Mathematics, The Chinese University of Hong Kong, Hong Kong (lmlui@math.cuhk.edu.hk).

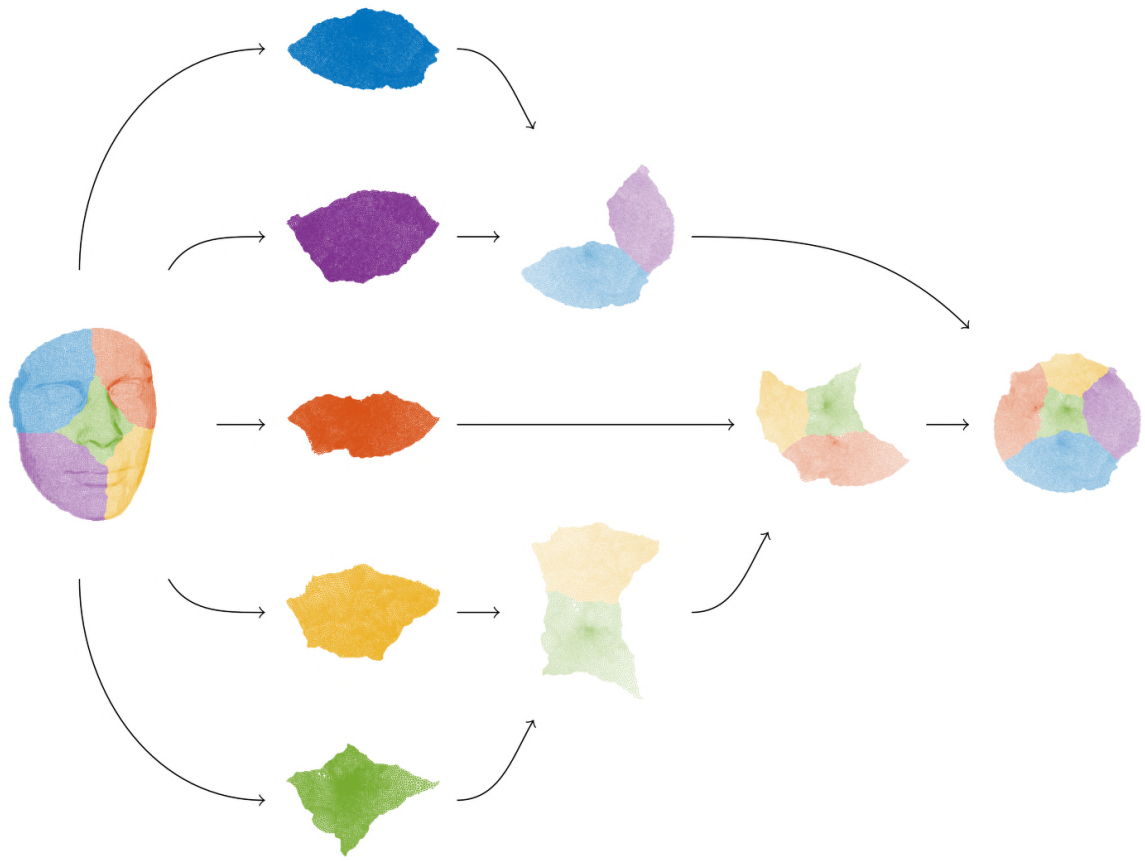

Figure 1. *An overview of our proposed parallelizable global conformal parameterization (PGCP) algorithm. A simply-connected surface is first partitioned into small subdomains. The subdomains are then conformally flattened onto the plane in parallel. The flattened subdomains are subsequently stitched seamlessly by a novel partial welding technique along the common boundary arcs, thereby producing a global conformal parameterization of the surface. Note that the partial welding step only involves the boundary points of the subdomains but not their interior. The mesh structures of the subdomains are shown only for visualization purpose.*

surfaces can be reduced to PDEs on the parameter domain, which are much easier to solve. Also, texture mapping on a 3D surface can be done by parameterizing it onto the 2D plane, in which textures can be easily designed. Among all surface parameterizations, one special type of parameterization is called conformal parameterization, which preserves angle and hence the local geometry of the surfaces. This is particularly important for applications such as texture mapping and remeshing, in which the angle structure plays an important role in the computation. To avoid creating computational burdens or introducing distortions, a fast and accurate method for computing conformal parameterization of surfaces is desired.

In this work, we propose a novel *parallelizable global conformal parameterization* method (abbreviated as *PGCP*) for simply-connected surfaces. Unlike the existing methods, our method uses a divide and conquer approach and exploits the nature of conformal parameterization, making the computation highly parallelizable. Figure 1 gives an overview of our proposed

method. We begin with partitioning a given surface into smaller subdomains. Then, the local conformal parameterizations of the subdomains are computed in parallel. Note that the local parameterization results are not necessarily consistent along their boundaries. Motivated by the theory of conformal welding in complex analysis, we develop a method called *partial welding* to update the boundaries of the flattened subdomains for enforcing the consistency between them. Finally, we solve the Laplace equation with the updated boundary constraints to find conformal parameterizations of the subdomains such that all of them can be glued seamlessly, ultimately forming a global conformal parameterization of the given dense surface. The bijectivity of the parameterization is guaranteed by quasi-conformal theory.

The rest of the paper is organized as follows. In Section 2, we review the related works in surface parameterization. In Section 3, we introduce the mathematical concepts involved in our work. In Section 4, we describe our proposed method for computing a global conformal parameterization of simply-connected surfaces via partial welding. Experimental results and applications are presented in Section 5 for demonstrating the effectiveness of our proposed method. In Section 6, we discuss the conformality improvement achieved by our method, an extension of our work for reducing the area distortion, and alternative approaches for accelerating the computation. We conclude our work and discuss possible future works in Section 7.

2. Related works. Surface parameterization has been widely studied in geometry processing. For an overview of the subject, readers are referred to the surveys [1–3]. It is well-known that only developable surfaces can be isometrically flattened without any distortions in area and angle. For general surfaces, it is unavoidable to introduce distortions in area or angle (or both) under parameterization. This limitation leads to two major classes of surface parameterization algorithms, namely the area-preserving parameterizations and angle-preserving (conformal) parameterizations.

Existing methods for area-preserving parameterizations include the locally authalic map [4], Lie advection [5], optimal mass transport (OMT) [6, 7], density-equalizing map (DEM) [8, 9] and stretch energy minimization (SEM) [10]. While the area elements can be preserved under area-preserving parameterizations, the angular distortion is uncontrolled. Since the angular distortion is related to the local geometry of the surfaces, it is important to minimize the angular distortion in many applications such as remeshing, texture mapping and cartography. In those cases, it is preferable to use conformal parameterization.

Existing conformal parameterization methods for simply-connected open surfaces include the discrete natural conformal parameterization (DNCP) [4]/least-square conformal mapping (LSCM) [11], Yamabe flow [12], angle-based flattening (ABF) [13–15], circle patterns [16], spectral conformal mapping (SCP) [17], conformal equivalence of triangle meshes (CETM) [18], discrete Ricci flow [19–21], quasi-conformal compositions [22–25] and conformal energy minimization (CEM) [26]. There are also some notable works on the spherical conformal parameterization of genus-0 closed surfaces, including linearization of Laplace equation [27, 28], Dirichlet energy minimization [29], folding-free global conformal mapping [30], FLASH [31] and north-south iterative scheme [32]. More recently, Sawhney and Crane [56] proposed the boundary first flattening (BFF) method, which is capable of computing free-boundary, fixed-boundary and spherical conformal parameterizations for simply-connected surfaces.

Note that all the above-mentioned methods compute a global conformal parameterization of a given surface by handling the entire surface directly. In case the given surface mesh is dense, the computation may be expensive. Also, in case the geometry of the input mesh is complicated, performing a global computation may lead to inaccuracy. Our work aims to overcome these problems by decomposing the input surface mesh into smaller domains and parameterizing them in parallel. The consistency between the domains is ensured by a novel technique called partial welding, thereby forming a global conformal parameterization efficiently.

3. Mathematical background.

3.1. Harmonic map and conformal map.

Following [33, 34], we introduce the following definitions of harmonic map and conformal map. Let D and Ω be simply-connected regions in $\mathbb{R}^2$.

Definition 3.1 (Harmonic map). *A map $\varphi : D \to \Omega$ is said to be* harmonic *if it minimizes the Dirichlet energy*

$$E_D(\varphi) = \frac{1}{2}\int_D |\nabla\varphi|^2. \tag{3.1}$$

Definition 3.2 (Conformal map). *A map $\varphi : D \to \Omega$ is said to be* conformal *if it satisfies*

$$J\frac{\partial\varphi}{\partial x} = \frac{\partial\varphi}{\partial y}, \tag{3.2}$$

where J is a rotation by $\frac{\pi}{2}$ in the tangent plane. If we write $\varphi = (\varphi_x, \varphi_y)$, the above equation can be reformulated as the following equations, known as the Cauchy-Riemann equations*:*

$$\begin{cases} \frac{\partial\varphi_x}{\partial x} - \frac{\partial\varphi_y}{\partial y} = 0, \\ \frac{\partial\varphi_x}{\partial y} + \frac{\partial\varphi_y}{\partial x} = 0. \end{cases} \tag{3.3}$$

To achieve conformality, we could minimize the *conformal energy*

$$E_C(\varphi) = \frac{1}{2}\int_D \left[\left(\frac{\partial\varphi_x}{\partial x} - \frac{\partial\varphi_y}{\partial y}\right)^2 + \left(\frac{\partial\varphi_x}{\partial y} + \frac{\partial\varphi_y}{\partial x}\right)^2\right]. \tag{3.4}$$

As shown by Hutchinson [33], if we define the *area* $A(\varphi)$ by

$$A(\varphi) = \int_D \left\|\frac{\partial\varphi}{\partial x} \times \frac{\partial\varphi}{\partial y}\right\|, \tag{3.5}$$

then the conformal energy can be expressed in terms of the Dirichlet energy and area:

$$E_C(\varphi) = E_D(\varphi) - A(\varphi). \tag{3.6}$$

Since the conformal energy is nonnegative, it follows that the Dirichlet energy is always bounded below by the area. In particular, the equality holds if and only if φ is conformal.

Moreover, given the area term $A(\varphi)$, minimizing the conformal energy is equivalent to minimizing the Dirichlet energy. Note that the area depends on how φ maps the boundary. In other words, given a "good" boundary condition, a conformal map can be obtained by simply finding the harmonic map under the given boundary condition.

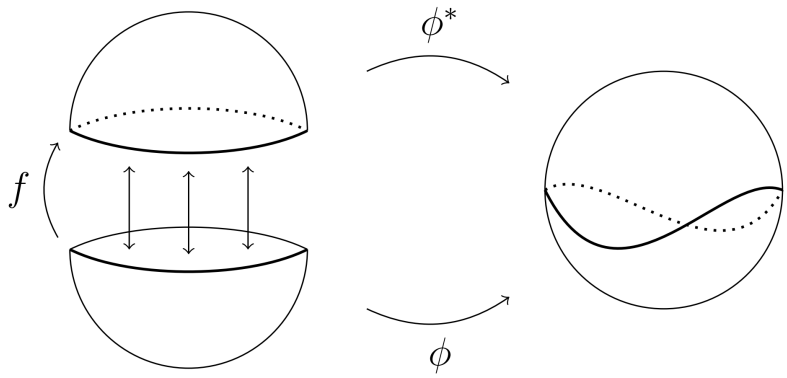

Figure 2. *An illustration of the closed welding problem. The entire boundaries of the two parts are glued consistently.*

3.2. Möbius transformation. A special type of conformal maps on the extended complex plane $\overline{\mathbb{C}}$ are the *Möbius transformations*, also known as the *linear fractional transformations*:

Definition 3.3 (Möbius transformation). *A function $f : \overline{\mathbb{C}} \to \overline{\mathbb{C}}$ is said to be* Möbius transformation *if it is of the form*

$$f(z) = \frac{az + b}{cz + d}, \tag{3.7}$$

where a, b, c, d are complex numbers with $ab - bc \neq 0$.

Given two sets of distinct points $\{z_1, z_2, z_3\}$ and $\{w_1, w_2, w_3\}$, there exists a unique Möbius transformation satisfying $f(z_i) = w_i$, $i = 1, 2, 3$. Therefore, Möbius transformations provides us with a simple way of fixing three points conformally.

3.3. Conformal welding. *Conformal welding*, also known as *sewing* or simply *welding*, is a problem in complex analysis which concerns with gluing two surfaces in a conformal way so that they fit together consistently according to certain correspondence.

Given a diffeomorphism f from a curve (e.g. the unit circle) to itself, we want to find two Jordan domains $D, \Omega \subset \overline{\mathbb{C}}$ and two conformal maps $\phi : D \to \Omega$ and $\phi^* : D^* \to \Omega^*$ such that $\phi = \phi^* \circ f$ on the curve [35]. Here, D^* and Ω^* are the exterior of D and Ω respectively. Since $\overline{\mathbb{C}} \cong \mathbb{S}^2$, the two domains D, Ω can be regarded as two disk-like surfaces on $\mathbb{S}^2$. Intuitively, given a correspondence between the boundaries of the two surfaces, the problem of conformal welding is to find two conformal deformations such that the surfaces are stitched together seamlessly (see Figure 2). We refer this classical welding problem as a *closed* welding problem.

For a general homeomorphism f, the closed welding problem may not have any solution. However, if f satisfies certain conditions, the problem is solvable. We introduce the concept of *quasisymmetric function* below:

Definition 3.4 (Quasisymmetric function [36]). *Let f be a continuous, strictly increasing function defined on an interval I of the x-axis. We call f k-quasisymmetric (or simply*

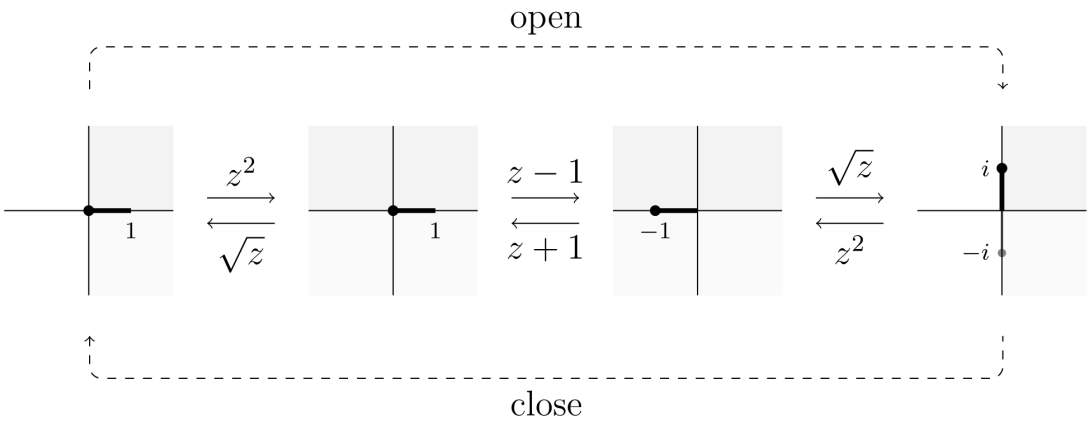

Figure 3. *An illustration of the opening map and the closing map. The top part shows the processes involved in the opening map $z \mapsto \sqrt{z^2-1}$, which ultimately map $\{0,1\}$ to $\{i,0\}$ or $\{-i,0\}$, depending on the choice of branching. The closing map $z \mapsto \sqrt{z^2+1}$ reverses the processes such that $\pm i$ will be mapped back to 0.*

quasisymmetric) on I if there exists a positive constant k such that

$$\frac{1}{k} \leq \frac{f(x+t)-f(x)}{f(x)-f(x-t)} \leq k \tag{3.8}$$

for all $x, x-t, x+t \in I$ with $t > 0$.

One can show that the closed welding problem is solvable if f is a quasisymmetric function from the real axis to itself:

Theorem 3.5 (Sewing theorem [36]). *Let f be a quasisymmetric function on the real axis. Then the upper and lower half-planes can be mapped conformally onto disjoint Jordan domains D, Ω by two maps ϕ, ϕ^*, with $\phi(x) = \phi^*(f(x))$ for all $x \in \mathbb{R}$.*

The theorem was first proven by Pfluger based on the existence of solutions to the Beltrami equation [37]. Another way to prove the result is to use some approximation techniques on the quasisymmetric function [36].

3.4. Geodesic algorithm. A conformal mapping method called the *zipper algorithm* was proposed independently by Kühnau [38] and Marshall and Morrow [39] in the 1980s. In particular, Marshall and Rohde [40] proved the convergence of a variant of it called the *geodesic algorithm*. The geodesic algorithm computes a conformal map from a region in the complex plane to the upper half-plane $\mathbb{H}$. Below, we briefly describe the geodesic algorithm.

The key ingredients of the geodesic algorithm are two maps: the *opening map* $z \mapsto \sqrt{z^2-1}$ and the *closing map* $z \mapsto \sqrt{z^2+1}$. Intuitively, they are operations analogous to opening and closing a slit, behaving like a zipper (see Figure 3). Suppose we have a simple closed region Ω, and a sequence of boundary points $\{z_0, z_1, \ldots, z_k\}$ on $\partial\Omega$. To initiate the process, define a map $g_1 : \Omega \to \mathbb{C}$ by

$$g_1(z) = \sqrt{\frac{z-z_1}{z-z_0}}, \tag{3.9}$$

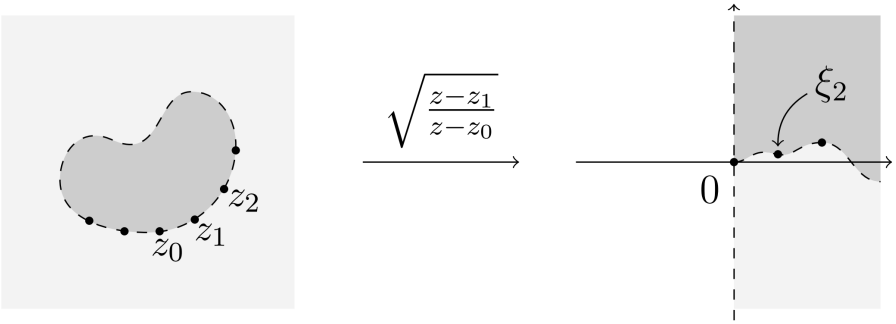

Figure 4. *In the geodesic algorithm, the first map g_1 maps z_0 to ∞ and z_1 to 0. It initiates the process so that we can apply the opening maps for the remaining data points.*

assuming the branching $(-1)^{1/2} = i$. This maps Ω to the right half-plane. In particular, the line segment between z_0 and z_1 is mapped onto the imaginary axis, with z_0 mapped to ∞ and z_1 mapped to 0 (see Figure 4).

Analogously, one can define a map g_2 such that the line segment between $g_1(z_1) = 0$ and $\xi_2 := g_1(z_2)$ is mapped to the imaginary axis, while the remaining points are still in the right half-plane. By further repeating the above process, all the boundary points can be pushed onto the imaginary axis one by one. More explicitly, suppose the point z_j has already been transformed to the position ξ_j after applying the opening maps $g_1, g_2, \ldots, g_{j-1}$, i.e.

$$\xi_j = g_{j-1} \circ g_{j-2} \circ \cdots \circ g_1(z_j). \tag{3.10}$$

Consider a Möbius transformation

$$L_{\xi_j}(z) := \frac{\frac{\mathrm{Re}(\xi_j)}{|\xi_j|^2} z}{1 + \frac{\mathrm{Im}(\xi_j)}{|\xi_j|^2} zi}. \tag{3.11}$$

It can be easily checked that L_{ξ_j} maps $\{0, \xi_j, \rho\}$ to $\{0, 1, \infty\}$, where ρ is a point on the imaginary axis at which the orthogonal circular arc from 0 to ξ_k extends to. Now, as $L_{\xi_j}(\xi_j) = 1$, we can map the segment between $L_{\xi_j}(g_{j-1} \circ g_{j-2} \circ \cdots \circ g_1(z_{j-1})) = 0$ and $L_{\xi_j}(\xi_j) = 1$ onto the imaginary axis as illustrated in Figure 3. Define g_j as the composition of L_{ξ_j} with the opening map $f(z) = \sqrt{z^2 - 1}$:

$$g_j(z) := \sqrt{L_{\xi_j}(z)^2 - 1}. \tag{3.12}$$

Note that $g_j((g_{j-1} \circ \cdots \circ g_1(z_{j-1})) = \sqrt{0-1} = i$ (assuming the branching $(-1)^{1/2} = i$), $g_j(\xi_j) = \sqrt{L_{\xi_j}(\xi_j)^2 - 1} = \sqrt{1-1} = 0$, and the entire region will remain in the right half-plane. Therefore, the requirements for g_j are satisfied.

After obtaining the maps $g_1, g_2, \ldots, g_k$ such that $g_k \circ g_{k-1} \circ \cdots \circ g_1$ maps all boundary points $\{z_0, z_1, \ldots, z_k\}$ onto the imaginary axis, define a final map

$$g_{k+1}(z) = \left(\frac{z}{1 - \frac{z}{g_k \circ g_{k-1} \circ \cdots \circ g_1(z_0)}} \right)^2. \tag{3.13}$$

g_{k+1} maps the transformed region $g_k \circ g_{k-1} \circ \cdots \circ g_1(\Omega)$ onto the upper half-plane $\mathbb{H}$. Since all the above maps are analytic and, in particular, a square and a square root map are applied in each step, the composition map $(g_{k+1} \circ g_k \circ \cdots \circ g_1) : \Omega \to \mathbb{H}$ is conformal.

3.5. Quasi-conformal map. Quasi-conformal map is an extension of conformal map in the sense that it allows for bounded conformal distortion. Intuitively, conformal maps map infinitesimal circles to infinitesimal circles , while quasi-conformal maps map infinitesimal circles to infinitesimal ellipses with bounded eccentricity. The formal definition of quasi-conformal map is given below.

Definition 3.6 (Quasi-conformal map [41]). *A map $\varphi : D \to \Omega$ is said to be* quasi-conformal *if it satisfies the* Beltrami equation

$$\frac{\partial \varphi}{\partial \bar{z}} = \mu_\varphi(z) \frac{\partial \varphi}{\partial z} \tag{3.14}$$

for some complex-valued function μ_φ with $\|\mu_\varphi\|_\infty < 1$. μ_φ is said to be the Beltrami coefficient *of φ.*

The Beltrami coefficient μ_φ captures the conformal distortion of φ. In particular, if $\mu_\varphi = 0$, then the Beltrami equation becomes the Cauchy-Riemann equations and hence φ is conformal. Also, the Jacobian J_φ of φ is given by

$$J_\varphi = \left| \frac{\partial \varphi}{\partial z} \right|^2 (1 - |\mu_\varphi|^2). \tag{3.15}$$

Therefore, a map is folding-free if and only if its Beltrami coefficient is with sup norm less than 1.

Moreover, one can correct the conformal distortion and non-bijectivity of a map by composing it with another map. If $\varphi_1 : \mathbb{C} \to \mathbb{C}$ and $\varphi_2 : \mathbb{C} \to \mathbb{C}$ are two maps with Beltrami coefficients μ_{φ_1} and μ_{φ_2}, then $\varphi_2 \circ \varphi_1$ is a quasi-conformal map with Beltrami coefficient

$$\mu_{\varphi_2 \circ \varphi_1}(z) = \frac{\mu_{\varphi_1}(z) + \frac{\overline{\varphi_{1_z}}}{\varphi_{1_z}} \mu_{\varphi_2}(\varphi_1(z))}{1 + \frac{\overline{\varphi_{1_z}}}{\varphi_{1_z}} \overline{\mu_{\varphi_1}(z)} \mu_{\varphi_2}(\varphi_1(z))}. \tag{3.16}$$

In particular, if $\mu_{\varphi_2} = \mu_{\varphi_1^{-1}}$, then $\mu_{\varphi_2 \circ \varphi_1} = 0$ and hence the composition map $\varphi_2 \circ \varphi_1$ is conformal and folding-free. This idea of quasi-conformal composition has been used in [24, 25, 31], and the details of the theory and computation of it can be found therein.

4. Proposed method. Let $\mathcal{S}$ be a simply-connected surface in $\mathbb{R}^3$, with a triangle mesh representation $(\mathcal{V}, \mathcal{F})$ where $\mathcal{V}$ is the vertex set and $\mathcal{F}$ is the face set. Our goal is to compute a global conformal parameterization of $\mathcal{S} = (\mathcal{V}, \mathcal{F})$ in an efficient and accurate way.

4.1. Surface partition. The first step is to partition $\mathcal{S}$ into submeshes based on a prescribed set of edges $\tilde{\mathcal{E}}$. Here, $\tilde{\mathcal{E}}$ can either be defined manually by the user or computed automatically using some existing partitioning methods. This allows the user to have full control of the number of subdomains to be used and how the surface is to be partitioned. More specifically, denote the edge set of $\mathcal{S}$ by $\mathcal{E}$, and the set of boundary edges of $\mathcal{S}$ by $\mathcal{E}_{\text{bdy}}$. Consider the set

$E = \mathcal{E} \setminus \left(\tilde{\mathcal{E}} \cup \mathcal{E}_{\text{bdy}}\right)$. We construct a graph G using E and find all connected components in G. Suppose there are K connected components in G, where each of them consists of a sub-face set $\mathcal{F}_i$, $i = 1, \ldots, K$. By tracking all vertices that are contained in $\mathcal{F}_i$, we obtain a sub-vertex set $\mathcal{V}_i$. In other words, we have obtained K simply-connected open submeshes $\mathcal{S}_1 = (\mathcal{V}_1, \mathcal{F}_1), \mathcal{S}_2 = (\mathcal{V}_2, \mathcal{F}_2), \ldots, \mathcal{S}_K = (\mathcal{V}_k, \mathcal{F}_K)$ that satisfy the following properties:

(i) The union of the vertex sets of all submeshes is exactly $\mathcal{V}$:

$$\bigcup_{i=1}^{K} \mathcal{V}_i = \mathcal{V}. \tag{4.1}$$

(ii) The union of the face sets of all submeshes is exactly $\mathcal{F}$:

$$\bigcup_{i=1}^{K} \mathcal{F}_i = \mathcal{F}. \tag{4.2}$$

(iii) The intersection of any two different sub-vertex sets is the intersection of the boundary sets, which is either an empty set or a boundary segment:

$$\mathcal{V}_i \cap \mathcal{V}_j = \partial\mathcal{S}_i \cap \partial\mathcal{S}_j \text{ for all } i, j. \tag{4.3}$$

(iv) The intersection of any two different sub-face sets is empty:

$$\mathcal{F}_i \cap \mathcal{F}_j = \emptyset \text{ for all } i \neq j. \tag{4.4}$$

4.2. Local conformal parameterization of submeshes. The next step is to compute a conformal parameterization of every $\mathcal{S}_i$. To find a conformal parameterization $\varphi_i : \mathcal{S}_i \to \mathbb{R}^2$, the DNCP method [4] is used. In short, DNCP minimizes the Dirichlet energy $E_D(\varphi)$ and maximizes the area $A(\varphi)$, based on the fact that the Dirichlet energy is bounded below by the area and conformality is attained when equality holds. We briefly describe the method below.

Let $\mathcal{V}_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_{n_i}}\}$ be the vertices in $\mathcal{S}_i$, and $\varphi_i : \mathcal{S}_i \to \mathbb{R}^2$ be a flattening map. Denote $\mathbf{u} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_i}]^t = \left[\varphi_i(v_{i_1}), \varphi_i(v_{i_2}), \ldots, \varphi_i(v_{i_{n_i}})\right]^t$. The Dirichlet energy is discretized using the cotangent formula [34]:

$$E_D(\mathbf{u}) = \frac{1}{2} \sum_{(v_{i_p}, v_{i_q}) \text{ adjacent}} (\cot\alpha_{pq} + \cot\beta_{pq})|\mathbf{u}_p - \mathbf{u}_q|^2 = \mathbf{u}^t L^{\text{cotan}} \mathbf{u}, \tag{4.5}$$

where α_{pq}, β_{pq} are the two angles opposite to the edge $[v_{i_p}, v_{i_q}]$ in $\mathcal{S}_i$, and L^{cotan} is a $|\mathcal{V}_i| \times |\mathcal{V}_i|$ sparse symmetric positive definite matrix also known as the cotangent Laplacian:

$$L^{\text{cotan}}_{p,q} = \begin{cases} \frac{1}{2}(\cot\alpha_{pq} + \cot\beta_{pq}) & \text{if } (v_{i_p}, v_{i_q}) \text{ are adjacent,} \\ -\sum_{r \neq p} L^{\text{cotan}}_{p,r} & \text{if } p = q, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

The area is discretized using the boundary vertices of $\mathcal{S}_i$:

$$A(\mathbf{u}) = \frac{1}{2} \sum_{[v_{i_p}, v_{i_q}] \subset \partial\mathcal{S}_i} (x_p y_q - y_p x_q) = \begin{pmatrix} \mathbf{x}^t & \mathbf{y}^t \end{pmatrix} M^{\text{area}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}, \tag{4.7}$$

where $\mathbf{u}_j = (x_j, y_j)$ for all j, $\mathbf{x} = [x_1, x_2, \ldots, x_{n_i}]^t$ is the collection of all x-coordinates of $\mathbf{u}$, $\mathbf{x} = [y_1, y_2, \ldots, y_{n_i}]^t$ is the collection of all y-coordinates of $\mathbf{u}$, and M^{area} is a $2|\mathcal{V}_i| \times 2|\mathcal{V}_i|$ sparse symmetric matrix. More explicitly, if $[v_{i_p}, v_{i_q}] \subset \partial\mathcal{S}_i$, we have

$$M^{\text{area}}_{p,q+|\mathcal{V}|_i} = M^{\text{area}}_{q+|\mathcal{V}|_i,p} = 1 \quad \text{and} \quad M^{\text{area}}_{q,p+|\mathcal{V}|_i} = M^{\text{area}}_{p+|\mathcal{V}|_i,q} = -1. \tag{4.8}$$

DNCP minimizes the discrete conformal energy

$$E_C(\mathbf{u}) = E_D(\mathbf{u}) - A(\mathbf{u}) \tag{4.9}$$

subject to the prescribed positions of two boundary vertices that remove the freedom of rigid motion and scaling. It suffices to solve a $2|\mathcal{V}_i| \times 2|\mathcal{V}_i|$ sparse linear system

$$\left(\begin{pmatrix} L^{\text{cotan}} & 0 \\ 0 & L^{\text{cotan}} \end{pmatrix} - M^{\text{area}} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = 0 \tag{4.10}$$

subject to four boundary constraints (two in $\mathbf{x}$ and two in $\mathbf{y}$ for the two pinned boundary vertices). The resultant map φ_i satisfying $\varphi_i(\mathcal{V}_i) = \mathbf{u} = [\mathbf{x}, \mathbf{y}]$ is the desired conformal parameterization of $\mathcal{S}_i$.

DNCP is suitable for our framework since it is a free-boundary linear method. As discussed above, obtaining each φ_i only requires solving a $2|\mathcal{V}_i| \times 2|\mathcal{V}_i|$ sparse matrix equation, which is highly efficient. Also, the free-boundary condition ensures that no additional conformal distortion will be introduced at the boundaries. This is particularly important in our subsequent welding step.

It is noteworthy that the parameterization of each submesh is independent, and hence this step of computing local conformal parameterizations is highly parallelizable.

4.3. Partial welding. Note that the local parameterizations we obtained via DNCP are not necessarily consistent along the boundaries. Therefore, we need a step for gluing the boundaries of them consistently. To preserve the conformality of the parameterization, the gluing step should be conformal. This problem of gluing subdomains is different from the closed welding problem introduced in Section 3. More explicitly, the closed welding problem considers gluing the entire boundaries of two domains, while in general only a portion of the boundaries of two neighboring subdomains in our case should be glued. In other words, the problem that we need to tackle is a *partial* welding problem that involves gluing two subdomains along only a pair of boundary arcs.

Below, we first rigorously derive a theoretical construction for solving the partial welding problem. Then, we devise an efficient algorithm for solving it.

4.3.1. Theoretical construction. We formulate the problem mathematically. Given two Jordan regions $A, B \subset \overline{\mathbb{C}}$, let $\gamma_A \subset \partial A$ and $\gamma_B \subset \partial B$ be some arcs of the boundaries of A and B respectively. Suppose we have a correspondence function $f : \gamma_A \to \gamma_B$ that relates points on γ_A and points on γ_B. The partial welding problem is to find two conformal maps $\Phi_A : A \to A'$ and $\Phi_B : B \to B'$, with A' and B' being disjoint, such that

$$\Phi_A(\gamma_A) = (\Phi_B \circ f)(\gamma_A). \tag{4.11}$$

Recall that the closed welding problem is solvable for quasisymmetric function on the real axis. For the partial welding problem, we make use of the following lemma.

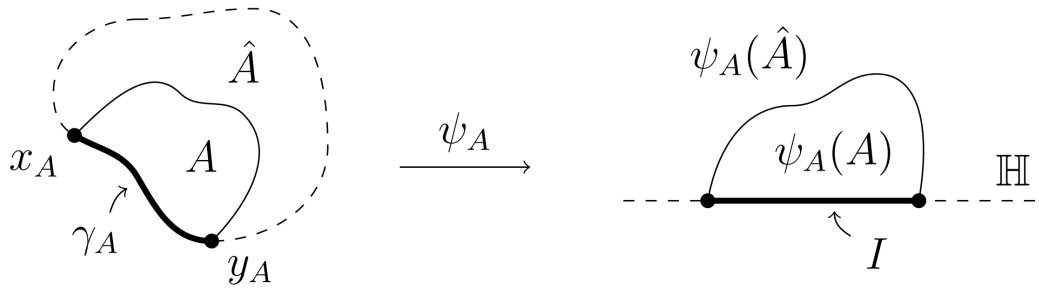

Figure 5. *Construction of $\hat{A}$ and the mapping from it to the upper half plane.*

Lemma 4.1 (Lehto and Virtanen [36]). *Every function f which is k-quasisymmetric on an interval $I = [a, b]$ can be extended to a $\tilde{k}$-quasisymmetric function on the entire x-axis, where the constant $\tilde{k}$ is less than a number depending only on k.*

To make use of the above lemma, we further suppose that $\hat{A}$ is a larger domain that contains A while sharing the boundary segment γ_A, i.e. $A \subset \hat{A}$ and $\gamma_A \subset \partial\hat{A}$ (see Figure 5 left). Denote the endpoints of γ_A by x_A and y_A. Similarly, let $\hat{B}$ be a domain such that $B \subset \hat{B}$ and $\gamma_B \subset \partial\hat{B}$, and denote the endpoints of γ_B by x_B and y_B.

By the Riemann mapping theorem, $\hat{A}$ and $\hat{B}$ can be mapped to the upper and lower half plane respectively by some conformal maps ψ_A and ψ_B. Now, we fix x_A and y_A at the endpoints of some interval I on the x-axis. For simplicity, we take $I = [-1, 1]$ and fix x_A and y_A at -1 and 1 respectively (see Figure 5 right). Similarly, we fix x_B at -1 and y_B at 1. The homeomorphic extensions to the closures define a map $g : I \to I$ by $g = \psi_B \circ f \circ \psi_A^{-1}$. In other words, we have $f = \psi_B^{-1} \circ g \circ \psi_A$ by construction. Assuming that g is a quasisymmetric function from I to itself, we get a quasisymmetric extension $\hat{g} : \mathbb{R} \to \mathbb{R}$ of g using Lemma 4.1.

Then, we apply Theorem 3.5 with this $\hat{g}$, which gives us two conformal maps $\phi_A : \hat{A} \to \hat{A}'$ and $\phi_B : \hat{B} \to \hat{B}'$ with $\hat{A}'$ and $\hat{B}'$ being disjoint, such that the boundary values satisfy $\phi_A(x) = \phi_B(\hat{g}(x))$ for all $x \in \mathbb{R}$. In particular, $\phi_A(x) = \phi_B(g(x))$ for all $x \in I$. Figure 6 shows an illustration of the construction.

Since the composition of conformal maps is conformal, we have constructed two conformal maps $\Phi_A = \phi_A \circ \psi_A$ and $\Phi_B = \phi_B \circ \psi_B$, which respectively map A to some $A' \subset \hat{A}'$ and B to some $B' \subset \hat{B}'$. Note that $f = \psi_B^{-1} \circ g \circ \psi_A$ when we restrict f on γ_A. Also,

$$(\phi_B \circ \psi_B) \circ f = \phi_B \circ \psi_B \circ \psi_B^{-1} \circ g \circ \psi_A = \phi_B \circ g \circ \psi_A = \phi_A \circ \psi_A, \tag{4.12}$$

where the last equality follows from Theorem 3.5. This solves the partial welding problem.

4.3.2. Algorithmic construction. The theoretical construction above provides us with a continuous approach for solving the partial welding problem. We proceed to develop an algorithm to solve the problem over discrete boundary data points. Suppose we have two sequences of boundary points $\partial A = \{a_0, \ldots, a_k, \ldots, a_m\}$ and $\partial B = \{b_0, \ldots, b_k, \ldots, b_n\}$, where a_j corresponds to b_j (i.e. a_j should be glued with b_j) for $j = 0, \ldots, k$. This gives a correspondence function $f : \gamma_A \subset \partial A \to \gamma_B \subset \partial B$, where $\gamma_A = \{a_0, \ldots, a_k\}$ and $\gamma_B =$

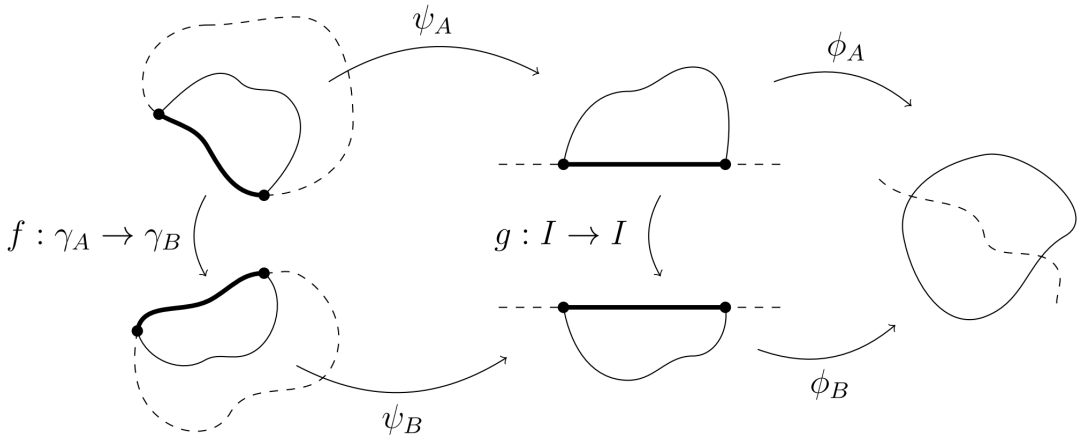

Figure 6. *The construction of conformal maps for solving the partial welding problem.*

$\{b_0, \ldots, b_k\}$, with $f(a_j) = b_j$ for $j = 0, \ldots, k$. Our goal is to construct the two maps Φ_A, Φ_B for gluing the two boundary curves conformally along the corresponding points. As discussed in the theoretical construction, a good way for the construction of Φ_A, Φ_B is to make use of two maps ψ_A, ψ_B that map A, B onto the upper and lower half-planes respectively. We propose an algorithm that makes use of a variant of the geodesic algorithm [40].

We begin with designing an algorithm that maps a sequence of boundary points to a standard shape. The algorithm is based on a key observation that the geodesic algorithm can be paused halfway. Suppose we have a sequence of boundary points $\{z_0, \ldots, z_k, \ldots, z_n\}$. Consider applying the first k maps $g_1, g_2, \ldots, g_k$ in the geodesic algorithm on $\{z_0, \ldots, z_k, \ldots, z_n\}$ with branching $(-1)^{1/2} = i$. The composition $g_k \circ \cdots \circ g_1$ maps the first $k+1$ points $\{z_0, \ldots, z_k\}$ onto the imaginary axis, with z_k mapped to 0, while the remaining boundary data points $\{z_{k+1}, \ldots, z_n\}$ are all mapped onto the right half-plane. Note that each of $g_2, \ldots, g_k$ is a composition of a Möbius transformation, a square map and a square root map. Therefore, they are all conformal.

Now, instead of the final map (3.13) in the geodesic algorithm, we apply the following Möbius transformation:

$$g_{k+1}(z) = \frac{z}{1 - \frac{z}{g_k \circ g_{k-1} \circ \cdots \circ g_1(z_0)}}. \tag{4.13}$$

It is easy to check that $g_{k+1}(g_k \circ \cdots \circ g_1(z_0)) = \infty$ and $g(0) = 0$. In other words, the new composition $g_{k+1} \circ g_k \circ \cdots g_1$ maps the first data point z_0 to ∞ and the $(k+1)$-th data point z_k to 0. Note that the first $k+1$ data points are on the upper half of the imaginary axis, and the remaining boundary data points $\{z_{k+1}, \ldots, z_n\}$ are on the right half-plane. We call such a half-opened (i.e. half-unzipped) shape an *intermediate form*. Note that by using another branching $(-1)^{1/2} = -i$ throughout the maps above, we have an alternative way to transform a sequence of boundary data points onto the right half-plane, with the first $k+1$ data points mapped onto the lower half of the imaginary axis. Algorithm 1 summarizes the proposed intermediate form transformation procedure.

Algorithm 1: Intermediate form transformation

Input: A sequence of boundary points $\{z_0, \ldots, z_k, \ldots, z_n\}$ and a choice of branching.
Output: A sequence of transformed boundary points $\{Z_0, \ldots, Z_k, \ldots, Z_n\}$, where $Z_0, \ldots, Z_k$ are on the imaginary axis.

Set $g_1(z) = \sqrt{\frac{z-z_1}{z-z_0}}$ with the choice of branching;
for $j = 2, \ldots, k$ **do**
  Compute $\xi_j = (g_{j-1} \circ \cdots \circ g_1)(z_j)$;
  Set $g_j(z) = \sqrt{L_{\xi_j}(z)^2 - 1}$ with the choice of branching, where $L_{\xi_j}(z) := \frac{\frac{\mathrm{Re}(\xi_j)}{|\xi_j|^2} z}{1 + \frac{\mathrm{Im}(\xi_j)}{|\xi_j|^2} zi}$;
Set $g_{k+1}(z) = \frac{z}{1 - \frac{z}{g_k \circ g_{k-1} \circ \cdots \circ g_1(z_0)}}$;
Compute $P_l = (g_{k+1} \circ \cdots \circ g_1)(z_l)$ for $l = 0, \ldots, k, \ldots, n$;

Coming back to the problem of aligning the two sequences of boundary points $\partial A = \{a_0, \ldots, a_k, \ldots, a_m\}$ and $\partial B = \{b_0, \ldots, b_k, \ldots, b_n\}$, we define auxiliary data points $a_{m+1} = b_{n+1} = 0, a_{m+2} = b_{n+2} = \infty$ to keep track of the transformation. Now, using Algorithm 1 with two different choices of branching $(-1)^{1/2} = i$ and $(-1)^{1/2} = -i$, we map $\{a_0, \ldots, a_k, \ldots, a_m, a_{m+1}, a_{m+2}\}$ and $\{b_0, \ldots, b_k, \ldots, b_n, b_{n+1}, b_{n+2}\}$ onto the right half-plane. Denote the transformed data points by $\{A_0, \ldots, A_k, \ldots, A_m, A_{m+1}, A_{m+2}\}$ and $\{B_0, \ldots, B_k, \ldots, B_n, B_{n+1}, B_{n+2}\}$. Note that $\{A_0, \ldots, A_k\}$ are all on the upper half of the imaginary axis with $A_0 = \infty$ and $A_k = 0$, while $\{B_0, \ldots, B_k\}$ are all on the lower half of the imaginary axis with $B_0 = \infty$ and $B_k = 0$. The next step is to align A_j with B_j for all $j = 0, \ldots, k$ conformally, such that the two boundary curves ∂A and ∂B are welded based on the partial correspondence between γ_A and γ_B.

Suppose $\alpha = ai$ and $\beta = bi$ are two corresponding points originally on γ_A and γ_B under the intermediate form transformations, where $a > 0 > b$. A Möbius transformation that takes $\{\alpha, 0, \beta\}$ to $\{i, 0, -i\}$ is explicitly given by

$$T_\beta^\alpha(z) = \frac{z}{\frac{-2ab}{a-b} - \frac{a+b}{a-b} zi}. \tag{4.14}$$

This transformation provides us with a simple way to align each pair of corresponding points. Note that $A_k = 0 = B_k$ is automatically aligned, and so we start with aligning A_{k-1} and B_{k-1}. Applying the Möbius transformation $T_{B_{k-1}}^{A_{k-1}}$ onto the two sets of boundary data points, we map A_{k-1} to i and B_{k-1} to $-i$. Then, we compose the map with the closing map $z \mapsto \sqrt{z^2+1}$ so that i and $-i$ are both mapped to 0. More explicitly, we define

$$h_{k-1}(z) := \sqrt{T_{B_{k-1}}^{A_{k-1}}(z)^2 + 1} \tag{4.15}$$

and apply it to all data points. The branching for the computation of each point is determined using the previous choice in the intermediate form transformation. Then, we repeat the above

process for $j = k-2, \ldots, 1$ by defining

$$h_j(z) := \sqrt{T_{\beta_j}^{\alpha_j}(z)^2 + 1}, \tag{4.16}$$

where

$$\alpha_j = (h_{j+1} \circ \cdots \circ h_{k-1})(A_j) \tag{4.17}$$

and

$$\beta_j = (h_{j+1} \circ \cdots \circ h_{k-1})(B_j). \tag{4.18}$$

Now, all pairs of corresponding points $(A_1, B_1), \ldots, (A_k, B_k)$ have been consistently aligned under the composition map $h_1 \circ h_2 \circ \cdots \circ h_{k-1}$. The first pair of corresponding points $A_0 = \infty = B_0$ are also automatically aligned. Note that each of $h_{k-1}, \cdots, h_1$ is a composition of a Möbius transformation, a square map and a square root map. Hence, they are all conformal.

Then, we define a closing map h_0 similar to (3.13) in the geodesic algorithm:

$$h_0(z) := \left(\frac{z}{1 - \frac{z}{(h_1 \circ \cdots \circ h_k)(\infty)}} \right)^2. \tag{4.19}$$

Note that h_0 maps all points onto the upper half plane $\mathbb{H}$, with 0 mapped to 0 and $(h_1 \circ \cdots \circ h_{k-1})(\infty)$ (i.e. $(h_1 \circ \cdots \circ h_{k-1})(A_0)$) mapped to ∞. We obtain the transformed data points

$$\tilde{a}_l = (h_0 \circ \cdots \circ h_{k-1})(A_l) \tag{4.20}$$

for $l = 0, \ldots, m+2$ with branching $(-1)^{1/2} = i$, and

$$\tilde{b}_l = (h_0 \circ \cdots \circ h_{k-1})(B_l) \tag{4.21}$$

for $l = 0, \ldots, n+2$ with branching $(-1)^{1/2} = -i$.

Considering the entire composition $h_0 \circ h_1 \circ \cdots h_{k-1} \circ g_{k+1} \circ \cdots g_1$ starting from the beginning to here, it can be observed that $g_2, \cdots, g_{k+1}, h_{k-1}, \cdots, h_2$ are all conformal, while g_2 is a square root map and h_0 is a square map. Therefore, the entire composition is conformal. In other words, we have conformally transformed the two sequences of boundary data points $\{a_0, \ldots, a_k, \ldots, a_m\}$ and $\{b_0, \ldots, b_k, \ldots, b_n\}$ into $\{\tilde{a}_0, \ldots, \tilde{a}_k, \ldots, \tilde{a}_m\}$ and $\{\tilde{b}_0, \ldots, \tilde{b}_k, \ldots, \tilde{b}_n\}$ such that the partial correspondence between them is satisfied, i.e. $\tilde{a}_j = \tilde{b}_j$ for $j = 0, \ldots, k$.

Finally, we perform a normalization by tracking the transformation of the auxiliary data points $a_{m+1} = b_{n+1} = 0, a_{m+2} = b_{n+2} = \infty$. More explicitly, we apply a Möbius transformation T that takes $\{\tilde{a}_{m+1}, \tilde{b}_{n+1}, \frac{1}{2}(\tilde{a}_{m+2} + \tilde{a}_{n+2})\}$ to $\{-1, 1, \infty\}$ on all the transformed points. This regularizes the transformation and prevents the boundary data points from being mapped far away. Note that Möbius transformations are conformal and hence the conformality of the composition map is preserved. This completes the process of gluing two boundary curves based on a partial correspondence between them. Algorithm 2 summarizes the proposed partial welding algorithm.

Algorithm 2: Partial welding

Input: Two sequences of boundary data points $\{a_0, \ldots, a_k, \ldots, a_m\}$ and $\{b_0, \ldots, b_k, \ldots, b_n\}$, where a_j should be glued with b_j for $j = 0, \ldots, k$.

Output: Conformally transformed data points $\{\tilde{a}_0, \ldots, \tilde{a}_k, \ldots, \tilde{a}_m\}$ and $\{\tilde{b}_0, \ldots, \tilde{b}_k, \ldots, \tilde{b}_n\}$ such that $\tilde{a}_j = \tilde{b}_j$ for $j = 0, \ldots, k$.

Define auxiliary points $a_{m+1} = b_{n+1} = 0, a_{m+2} = b_{n+2} = \infty$;

Apply Algorithm 1 on $\{a_0, \ldots, a_k, \ldots, a_m, a_{m+1}, a_{m+2}\}$ with branching $(-1)^{1/2} = i$ and obtain the transformed boundary data points $\{A_0, \ldots, A_k, \ldots, A_m, A_{m+1}, A_{m+2}\}$;

Apply Algorithm 1 on $\{b_0, \ldots, b_k, \ldots, b_n, b_{n+1}, b_{n+2}\}$ with branching $(-1)^{1/2} = -i$ and obtain the transformed boundary data points $\{B_0, \ldots, B_k, \ldots, B_n, B_{n+1}, B_{n+2}\}$;

Set $h_{k-1}(z) := \sqrt{T_{B_{k-1}}^{A_{k-1}}(z)^2 + 1}$;

for $j = k-2, \ldots, 1$ **do**

  Compute $\alpha_j = (h_{j+1} \circ \cdots \circ h_{k-1})(A_j)$ with branching $(-1)^{1/2} = i$;

  Compute $\beta_j = (h_{j+1} \circ \cdots \circ h_{k-1})(B_j)$ with branching $(-1)^{1/2} = -i$;

  Set $h_j(z) := \sqrt{T_{\beta_j}^{\alpha_j}(z)^2 + 1}$;

Set $h_0(z) := \left(\frac{z}{1 - \frac{z}{(h_1 \circ \cdots \circ h_{k-1})(\infty)}} \right)^2$;

Compute $\tilde{a}_l = (h_0 \circ \cdots \circ h_{k-1})(A_l)$ for $l = 0, \ldots, m+2$, with branching $(-1)^{1/2} = i$;

Compute $\tilde{b}_l = (h_0 \circ \cdots \circ h_{k-1})(B_l)$ for $l = 0, \ldots, n+2$, with branching $(-1)^{1/2} = -i$;

Apply a Möbius transformation T that takes $\{\tilde{a}_{m+1}, \tilde{b}_{n+1}, \frac{1}{2}(\tilde{a}_{m+2} + \tilde{a}_{n+2})\}$ to $\{-1, 1, \infty\}$ on all the transformed points;

An illustration of the partial welding algorithm is given in Figure 7. As a remark, to weld two subdomains obtained by the local parameterization step partially, we only need to extract their boundary points on $\mathbb{C}$ and apply Algorithm 2. The interior points of the two flattened subdomains are not needed. With the updated coordinates of the boundary points of the subdomains, we can then easily obtain the desired global conformal parameterization by solving a number of sparse linear systems. The details will be described in Section 4.5.

4.4. Enforcing additional constraints. Before moving on to the step of obtaining the final global parameterization, it is possible for us to include an optional step here and enforce additional constraints for achieving disk conformal parameterization and spherical conformal parameterization.

4.4.1. Constraints for disk conformal parameterization. If the input simply-connected surface $\mathcal{S}$ is open, one can further restrict the target parameter domain to be the unit disk in the proposed method, thereby achieving a disk conformal parameterization. This is done by adding an extra step of applying the geodesic algorithm introduced in Section 3.4 to the global boundary $\partial\mathcal{S}$. Note that the points on $\partial\mathcal{S}$ are distributed into various subdomains. Therefore, we first extract the coordinates of those boundary points from the partial welding result. Once the mapping that takes those points to the unit circle is determined, we apply

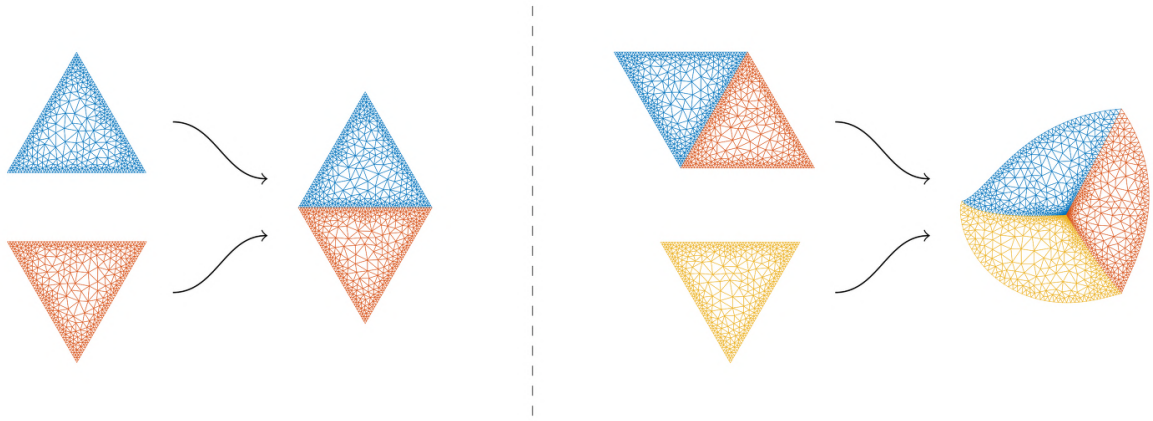

Figure 7. *An illustration of partial welding. Suppose we are given a pyramid-like surface with three triangular faces (colored in blue, red, yellow). Each of them have been flattened onto the plane. In order to weld the boundaries of the three triangles, we apply Algorithm 2 twice. First, we glue the common boundaries of the blue and red triangles via partial welding. Then, we glue the common boundaries of the yellow triangle and the other two triangles via partial welding. Note that the interior points and the mesh structure of each triangle are plotted for a better visualization only. In the actual computation of partial welding, only the boundary points are involved.*

the map for transforming the boundary coordinates of every flattened subdomain onto the unit disk. This results in boundary coordinates for the subdomains that yield a disk conformal parameterization upon solving the Laplace equation (details to be described in Section 4.5).

4.4.2. Constraints for spherical conformal parameterization. For genus-0 closed surfaces, one common choice of the parameter domain is the unit sphere $\mathbb{S}^2$. In case the input surface $\mathcal{S}$ is a genus-0 closed surface, we can modify our framework so that the partial welding procedure is repeated until two large components are left. Then, for the last welding, we use a closed welding instead of a partial welding to glue the entire boundaries of the two large components. As all boundaries are glued, the resulting boundary coordinates of the subdomains on the extended complex plane yield a spherical conformal parameterization upon solving the Laplace equation (details to be described in Section 4.5).

4.5. Obtaining the global conformal parameterization. After obtaining the new boundary constraints that satisfy the consistency condition, we can compute the global conformal parameterization of the input surface $\mathcal{S}$ by finding a harmonic map $\tilde{\varphi}_i : \mathcal{S}_i \to \mathbb{R}^2$ for each submesh with the new boundary constraints. More explicitly, it suffices to solve the Laplace equation

$$\Delta \tilde{\varphi}_i = 0 \tag{4.22}$$

subject to the new boundary constraints. Again, note that the computations for the K submeshes are independent and so this step is parallelizable. Because of the consistency between the boundaries of all subdomains, the new local parameterization results can be glued

seamlessly, thereby forming a global conformal parameterization. One can further ensure the bijectivity of each subdomain using the idea of quasi-conformal composition (see Section 3.5). More specifically, we compute the Beltrami coefficient of the inverse mapping $\tilde{\varphi}_i^{-1}$ (denoted by $\mu_{\tilde{\varphi}_i^{-1}}$). We can then determine whether $\tilde{\varphi}_i$ is folding-free by checking if $\|\mu_{\tilde{\varphi}_i^{-1}}\|_\infty > 1$ (or close to 1 in the discrete case). If so, we compose $\tilde{\varphi}_i$ with another mapping that is associated with the Beltrami coefficient $\mu_{\tilde{\varphi}_i^{-1}}$ to fix the fold-overs as guaranteed by quasi-conformal theory. With this additional step, we can ensure the bijectivity of the resulting global conformal parameterization.

Note that the resulting global parameterization lies in the extended complex plane. In case $\mathcal{S}$ is a genus-0 closed surface, we add a stereographic projection step to convert it to a spherical parameterization. Algorithm 3 summarizes the proposed method.

Algorithm 3: Parallelizable global conformal parameterization of simply-connected surfaces (PGCP)

Input: A simply-connected surface mesh $\mathcal{S} = (\mathcal{V}, \mathcal{F})$, a set of edges $\tilde{\mathcal{E}}$ for the partition.
Output: A global conformal parameterization $\varphi : \mathcal{S} \to \mathbb{R}^2$ or $\mathbb{S}^2$.

Partition the mesh into K submeshes based on $\tilde{\mathcal{E}}$;
for $i = 1, \ldots, K$ **do**
  Compute a conformal parameterization of $\mathcal{S}_i = (\mathcal{V}_i, \mathcal{F}_i)$ using DNCP. Only the boundary coordinates of the parameterization are kept;
Perform partial welding as described in Algorithm 2 to update the boundary coordinates;
(Optional) To achieve disk conformal parameterization, further apply the geodesic algorithm [40]. To achieve spherical conformal parameterization, perform conformal welding on the last two components obtained by partial welding;
for $i = 1, \ldots, K$ **do**
  Solve the Laplace equation $\Delta\tilde{\varphi}_i = 0$ with the new boundary constraints for each $\mathcal{S}_i$;
  Compute the Beltrami coefficient $\mu_{\tilde{\varphi}_i^{-1}}$ to check whether $\tilde{\varphi}_i$ is folding-free. If not, fix the fold-overs in $\tilde{\varphi}_i$ using quasi-conformal composition;
The solutions $\tilde{\varphi}_i$ for all $\mathcal{S}_i$ together form a global conformal parameterization φ. For spherical conformal parameterization, further apply the stenographic projection to map the result onto $\mathbb{S}^2$;

As a remark, the novel combination of local parameterization and partial welding in our proposed method significantly improves the computational efficiency of global conformal parameterization. Efficient sparse linear system solvers for the Laplace equation for the entire mesh ($2|\mathcal{V}| \times 2|\mathcal{V}|$) typically require a complexity of $O(|\mathcal{V}|^{1.5})$ (SOR), $O(|\mathcal{V}| \log |\mathcal{V}|)$ (FFT), $O(|\mathcal{V}|)$ (multigrid) etc. [55]. By contrast, one can see that the interior parts of the submeshes are not used in the partial welding step in our method. The partial welding step only involves $\mathcal{B}$, the collection of boundary points of the subdomains, with $|\mathcal{B}| \ll |\mathcal{V}|$. Also, the computation of the local parameterizations at the beginning and the harmonic maps at the end of our proposed method can both be parallelized, so that each computation involves $\mathcal{V}_i$ only.

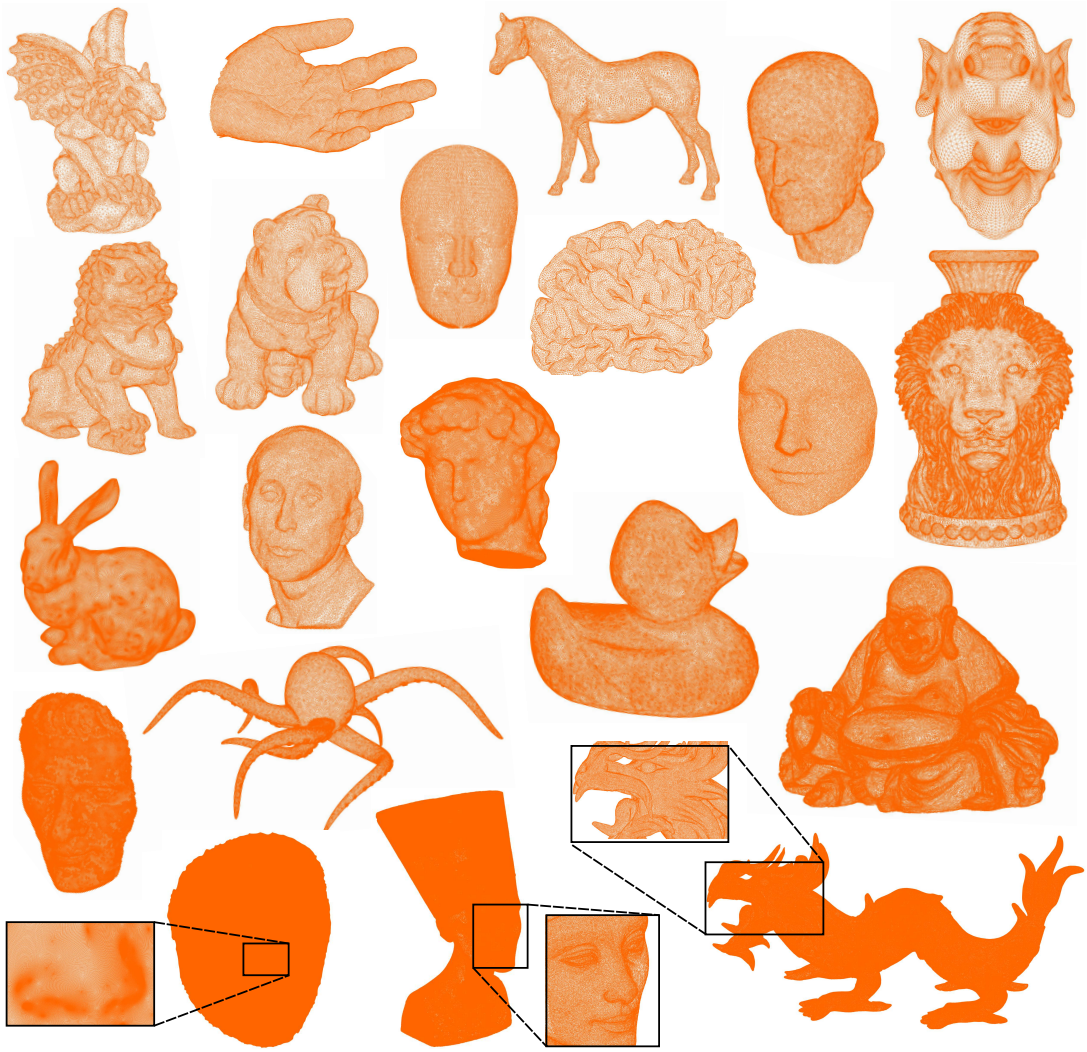

Figure 8. *A gallery of simply-connected surface meshes used in our experiments. Our PGCP method is capable of handling a wide range of simply-connected surfaces with different geometry, mesh quality and resolution.*

5. Experiments. Our proposed PGCP method is implemented in MATLAB, with the Parallel Computing Toolbox utilized for achieving parallelization. The sparse linear systems are solved using the backslash operator in MATLAB. All experiments in this section (except the experiment in Section 5.6) are performed on a Windows PC with Intel i7-6700K quad-core CPU and 16 GB RAM. To evaluate the performance of our proposed method, we adapt various simply-connected surface meshes from multiple free 3D model repositories [46–49] (see Figure 8). As for the distortion measure, we define the angular distortion of an angle $[v_i, v_j, v_k]$ (in degree) under the conformal parameterization φ by

$$d([v_i, v_j, v_k]) = \angle[\varphi(v_i), \varphi(v_j), \varphi(v_k)] - \angle[v_i, v_j, v_k]. \tag{5.1}$$

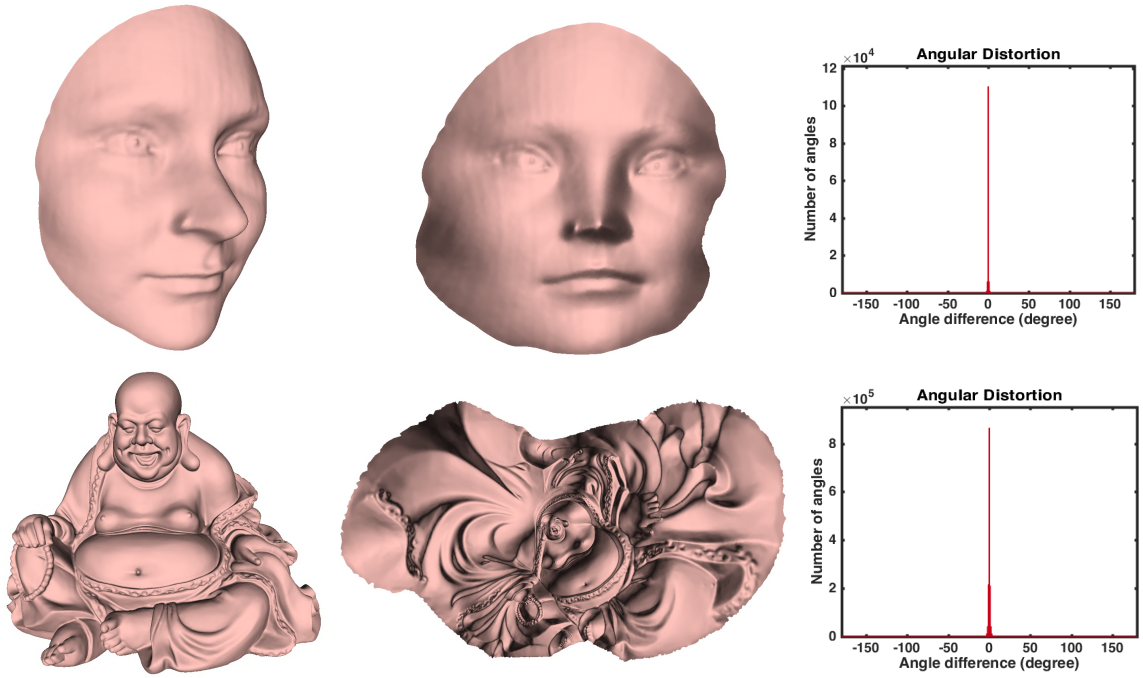

Figure 9. *Free-boundary conformal parameterizations of simply-connected open surfaces obtained by our proposed PGCP method, rendered with normal map shader.*

Surface	# vertices	SCP [17]		CETM [18]		PGCP	
		Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)
Sophie	21K	1.1	0.2	1.2	0.2	0.6	0.2
Niccolò da Uzzano	25K	1.3	0.6	Failed		0.7	0.6
Mask	32K	1.6	0.2	7.3	0.2	0.9	0.2
Max Planck	50K	2.6	0.5	5.5	0.5	1.5	0.5
Bunny	85K	4.4	0.5	18.8	0.5	2.0	0.5
Julius	220K	14.2	0.1	19.5	0.1	6.6	0.1
Buddha	240K	13.7	0.6	49.0	0.6	9.2	0.6
Face	1M	85.2	< 0.1	98.1	< 0.1	47.6	< 0.1

Table 1

The performance of spectral conformal parameterization (SCP) [17], conformal equivalence of triangle meshes (CETM) [18] and PGCP for free-boundary conformal parameterization of simply-connected open surfaces.

5.1. Free-boundary conformal parameterization of simply-connected open surfaces. We first consider computing free-boundary global conformal parameterization of simply-connected open surfaces using our proposed PGCP method (see Figure 9 for examples). To assess the performance of our method, we compare it with the spectral conformal parameterization (SCP) [17] and conformal equivalence of triangle meshes (CETM) [18] in terms of the computation time and the angular distortion (see Table 1). The MATLAB version of SCP is

implemented by the authors, and the MATLAB version of CETM can be found at [50]. The experimental results show that our proposed method is significantly faster than both SCP and CETM by over 40% and 70% respectively on average, while maintaining comparable accuracy in terms of the average angular distortion. This demonstrates the effectiveness of our method for free-boundary global conformal parameterization.

5.2. Disk conformal parameterization of simply-connected open surfaces. Besides free-boundary global conformal parameterization, our proposed PGCP method can also achieve disk conformal parameterization of simply-connected open surfaces (see Figure 10 for examples). To evaluate the performance of our method, we compare it with the state-of-the-art linear disk conformal map (LDM) method [25] and the conformal energy minimization (CEM) method [26] (see Table 2). The MATLAB version of LDM can be found at [51], and the MATLAB version of CEM can be found at [52]. It can be observed that our method is significantly faster than LDM and CEM by over 50% and 30% on average respectively. Also, our method achieves comparable or smaller angular distortion when compared to the two other methods. This shows that our method is advantageous for disk conformal parameterization.

5.3. Spherical conformal parameterization of simply-connected closed surfaces. We then consider computing spherical conformal parameterization of genus-0 closed surfaces using our proposed PGCP method (see Figure 11 for examples). To evaluate the performance, we compare our proposed method with the state-of-the-art folding-free global conformal mapping (FFGCM) algorithm [30] and the FLASH algorithm [31] (see Table 3). The MATLAB version of FFGCM is kindly provided by its authors, and the MATLAB version of FLASH can be found at [53]. Because of the "divide-and-conquer" nature of our method, our method is capable of producing spherical conformal parameterizations with a smaller angular distortion when compared to the two state-of-the-art algorithms. In particular, the FLASH algorithm involves puncturing a triangle from the input surface and flattening the punctured surface onto a big triangular domain. This step unavoidably creates squeezed regions and produces certain angular distortions. While the distortions are alleviated in the subsequent step using quasi-conformal composition, the step again involves a domain where most vertices are squeezed at the interior, which leads to some distortions. By contrast, our proposed PGCP method flattens each submesh naturally, with the shape of the submesh boundary taken into consideration. This effectively reduces the angular distortions, thereby producing a spherical conformal parameterization with a better accuracy. Moreover, because of the ability of exploiting parallelism, our method achieves a significant reduction in computational time by over 90% on average when compared to FFGCM. When compared to FLASH, our method achieves comparable efficiency for moderate meshes and a notable reduction in computational time by around 25% for dense meshes. This shows the advantages of our method for spherical conformal parameterization.

5.4. Comparison with boundary first flattening (BFF). While most of the existing methods can only handle a single type of global conformal parameterization, the recently proposed boundary first flattening (BFF) method [56], with code and executable files available at [57], is capable of computing multiple types of global conformal parameterizations, including free-boundary conformal parameterization for simply-connected open surfaces, disk conformal

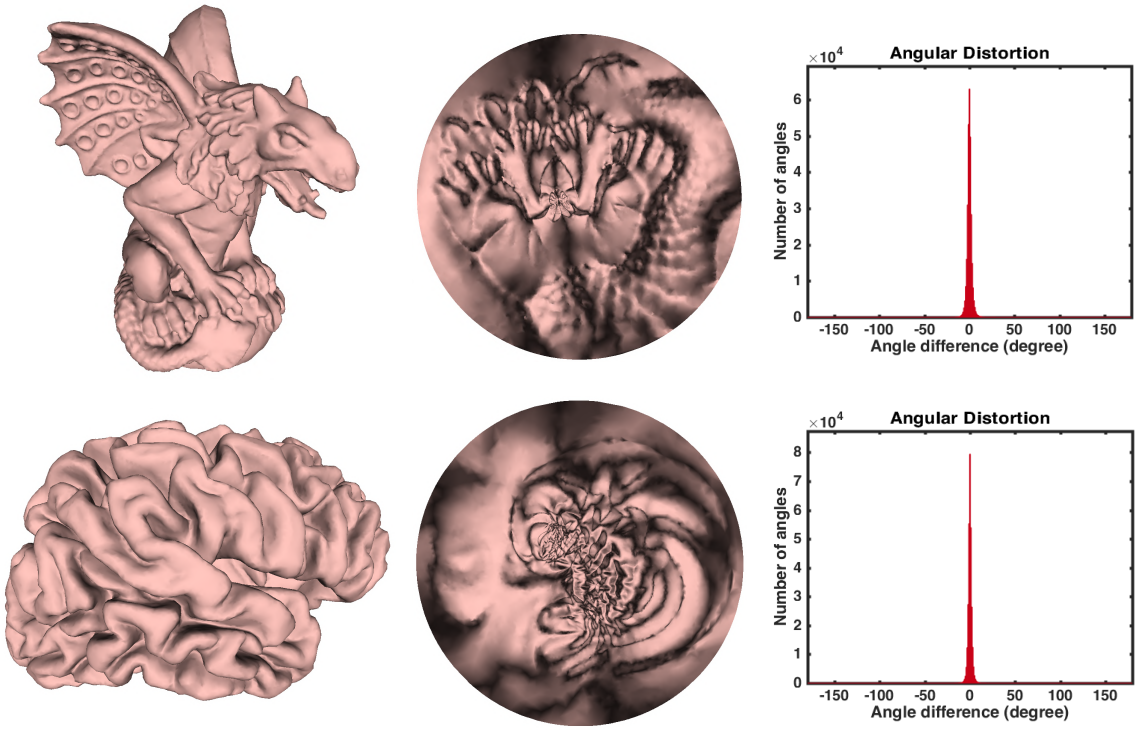

Figure 10. *Disk conformal parameterizations of simply-connected open surfaces obtained by our proposed PGCP method, rendered with normal map shader.*

Surface	# vertices	LDM [25]		CEM [26]		PGCP	
		Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)
Ogre	20K	1.1	1.5	0.3	2.6	0.5	1.5
Niccolò da Uzzano	25K	1.6	0.8	1.4	1.3	0.8	0.8
Brain	48K	2.9	1.6	2.9	1.5	1.3	1.5
Gargoyle	50K	3.1	1.9	2.8	2.1	1.4	1.9
Hand	53K	3.4	1.2	3.4	1.2	1.4	1.2
Octopus	150K	15.4	7.2	10.4	24.0	8.9	5.6
Buddha	240K	22.4	0.7	25.1	0.7	11.4	0.7
Nefertiti	1M	87.9	2.9	83.2	4.2	52.7	2.9

Table 2

The performance of linear disk conformal map (LDM) [25], conformal energy minimization (CEM) [26] and PGCP for disk conformal parameterization of simply-connected open surfaces.

parameterization for simply-connected open surfaces, and spherical conformal parameterization for genus-0 closed surfaces. It is therefore natural to compare our proposed method and the BFF method.

Table 4 shows the comparison between the two methods. For free-boundary and disk

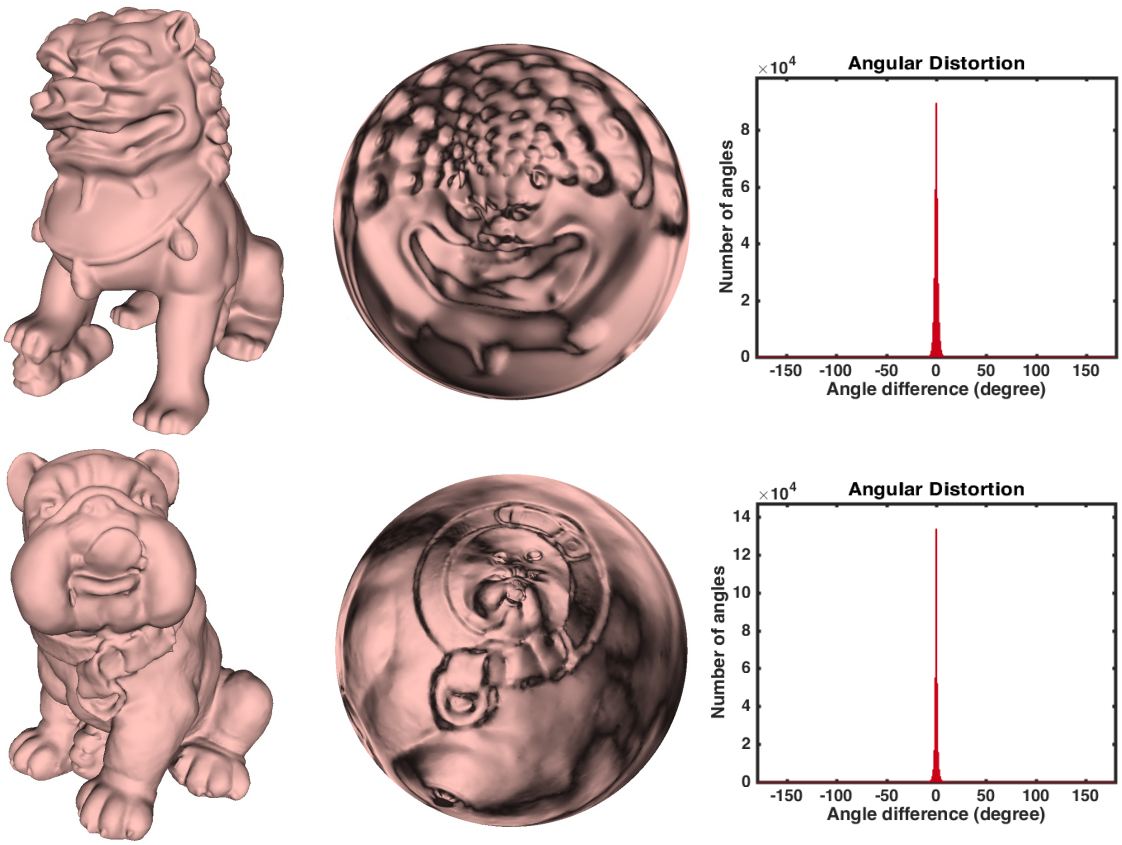

Figure 11. *Spherical conformal parameterizations of genus-0 closed surfaces obtained by our proposed PGCP method, rendered with normal map shader.*

Surface	# vertices	FFGCM [30]		FLASH [31]		PGCP	
		Time (s)	mean(d)	Time (s)	mean(d)	Time (s)	mean(d)
Horse	20K	12.1	11.0	0.4	3.0	0.4	2.7
Bulldog	50K	22.0	1.0	0.9	1.1	1.0	1.0
Chinese Lion	50K	29.3	1.3	1.1	1.3	1.1	1.3
Duck	100K	100.4	1.1	2.2	0.4	2.4	0.3
David	130K	46.6	0.2	3.5	0.2	3.4	0.2
Octopus	150K	112.3	37.2	10.1	6.9	7.1	2.6
Lion Vase	210K	222.7	14.4	4.5	0.8	4.7	0.7
Asian Dragon	1M	Failed		64.4	1.3	48.5	0.9

Table 3

The performance of folding-free global conformal mapping (FFGCM) [30], FLASH [31] and PGCP for spherical conformal parameterization of genus-0 closed surfaces.

conformal parameterization, it can be observed that our method achieves at least comparable and sometimes better conformality, with a shorter computational time. For spherical conformal

Parameterization	Surface	# vertices	BFF [56]		PGCP	
			Time (s)	mean($\lvert d \rvert$)	Time (s)	mean($\lvert d \rvert$)
Free-boundary	Sophie	21K	0.9	0.2	0.6	0.2
	Niccolò da Uzzano	25K	1.0	0.5	0.7	0.6
	Mask	32K	1.3	0.2	0.9	0.2
	Max Planck	50K	3.0	0.5	1.5	0.5
	Bunny	85K	4.7	1.2	2.0	0.5
	Julius	220K	16.8	0.1	6.6	0.1
	Buddha	240K	14.4	0.6	9.2	0.6
	Face	1M	Failed		47.6	< 0.1
Disk-boundary	Ogre	20K	0.7	1.5	0.5	1.5
	Niccolò da Uzzano	25K	1.0	1.1	0.8	0.8
	Brain	48K	2.4	1.6	1.3	1.5
	Gargoyle	50K	2.1	1.9	1.4	1.9
	Hand	53K	2.6	1.6	1.4	1.2
	Octopus	150K	5.5	24.7	8.9	5.6
	Buddha	240K	14.5	0.9	11.4	0.7
	Nefertiti	1M	Failed		52.7	2.9
Spherical	Horse	20K	1.0	74.52	0.4	2.7
	Bulldog	50K	3.1	11.5	1.0	1.0
	Chinese Lion	50K	3.0	4.4	1.1	1.3
	Duck	100K	7.6	5.7	2.4	0.3
	David	130K	15.9	2.6	3.4	0.2
	Octopus	150K	10.2	74.3	7.1	2.6
	Lion Vase	210K	14.2	4.9	4.7	0.7
	Asian Dragon	1M	Failed		48.5	0.9

Table 4

Comparison between BFF [56] and PGCP for free-boundary conformal parameterization for simply-connected open surfaces, disk conformal parameterization for simply-connected open surfaces, and spherical conformal parameterization for genus-0 closed surfaces.

parameterization, our method is advantageous in both the conformality and efficiency. A possible reason is that BFF handles genus-0 closed surfaces by removing an arbitrary vertex star, flattening the punctured surface onto a disk conformally followed by a suitable rescaling, and finally mapping the disk onto the sphere using stereographic projection and filling the punctured vertex star at the pole. The choice of the vertex star greatly affects the overall shape of the disk parameterization of the punctured surface and the final angular distortion of the spherical parameterization. By contrast, our "divide and conquer" approach enables us to tackle the conformal flattening problem of subdomains which are obtained from a more natural partition of the surface, thereby achieving better conformality.

5.5. Robustness of the proposed parameterization method to the choice of cut edges. In our proposed PGCP method, there is a flexibility for the user to prescribe the cut paths and supply the set of chosen edges as an input of the parameterization algorithm. This

Surface	Cut paths	mean($\|d\|$)	sd($\|d\|$)	median($\|d\|$)	iqr($\|d\|$)
Face	Figure 12 (top, leftmost)	0.25	0.44	0.13	0.23
	Figure 12 (top, second left)	0.26	0.44	0.16	0.24
	Figure 12 (top, second right)	0.24	0.44	0.12	0.24
	Figure 12 (top, rightmost)	0.26	0.44	0.15	0.25
Bunny	Figure 12 (middle, leftmost)	0.49	0.70	0.33	0.48
	Figure 12 (middle, second left)	0.49	0.70	0.33	0.48
	Figure 12 (middle, second right)	0.49	0.70	0.33	0.48
	Figure 12 (middle, rightmost)	0.49	0.70	0.33	0.49
Duck	Figure 12 (bottom, leftmost)	0.27	0.46	0.16	0.27
	Figure 12 (bottom, second left)	0.27	0.46	0.16	0.27
	Figure 12 (bottom, second right)	0.28	0.48	0.17	0.30
	Figure 12 (bottom, rightmost)	0.27	0.47	0.16	0.27

Table 5

The performance of our proposed PGCP method with different choices of cut paths. The mean, standard deviation, median, and interquartile range of the absolute angular distortion $|d|$ are evaluated.

allows the user to freely choose the number of subdomains to be used and how the surface is partitioned. It is natural to ask whether the performance of our method is robust to the choice of the cut edges. As shown in Figure 12, we consider different choices of cut paths on several surfaces and supplying them as an input of our proposed method. The accuracy of the resulting parameterizations is recorded in Table 5. It can be observed that the angular distortions produced by different choices of cut paths are highly consistent, which indicates that our method is robust to the choice of the cut paths.

5.6. Performance of our proposed PGCP method with different number of subdomains.

By partitioning a given surface into n subdomains evenly, one can reduce the original problem with size $|\mathcal{V}|$ to subproblems each with size $\frac{|\mathcal{V}|}{n}$. To check the parallel efficiency of our proposed method, we consider subdividing the bunny surface into different number of subdomains (see Figure 13) and apply our PGCP method for conformal parameterization. Since the maximum number of workers MATLAB creates depends on the number of physical cores, we use another machine with six physical cores for this experiment and focus on the ratio between the computational time with different number of subdomains.

Table 6 shows the experimental results with $2, 3, 4, 5, 6$ subdomains. Note that 2 is used as the baseline as the welding algorithm requires at least two subdomains. From the experimental result, we observe a speedup achieved by exploiting parallelization. To evaluate the parallel efficiency, we consider the ratio $E_n = \frac{S_n}{n/2}$, where S_n is the speedup achieved by n subdomains compared to the baseline and $n/2$ is the subdomain ratio. It can be observed that E_n is close to 1 for small n but shows a decreasing trend as n increases. A possible reason is that in the numerical implementation, there are some technical steps which are not parallelizable and hence will cost additional time as the number of subdomains increases. For instance, both the step of partitioning the surface into subdomains and the step of combining the parameterization results of all subdomains as a large $|\mathcal{V}| \times 2$ or $|\mathcal{V}| \times 3$ matrix involve non-sliced variables (i.e. matrices that cannot be broken up into segments by MATLAB), such as `face(subdomain_id == i,:)` where `face` is the original $|\mathcal{F}| \times 3$ triangulation, `subdomain_id` is the ID of the

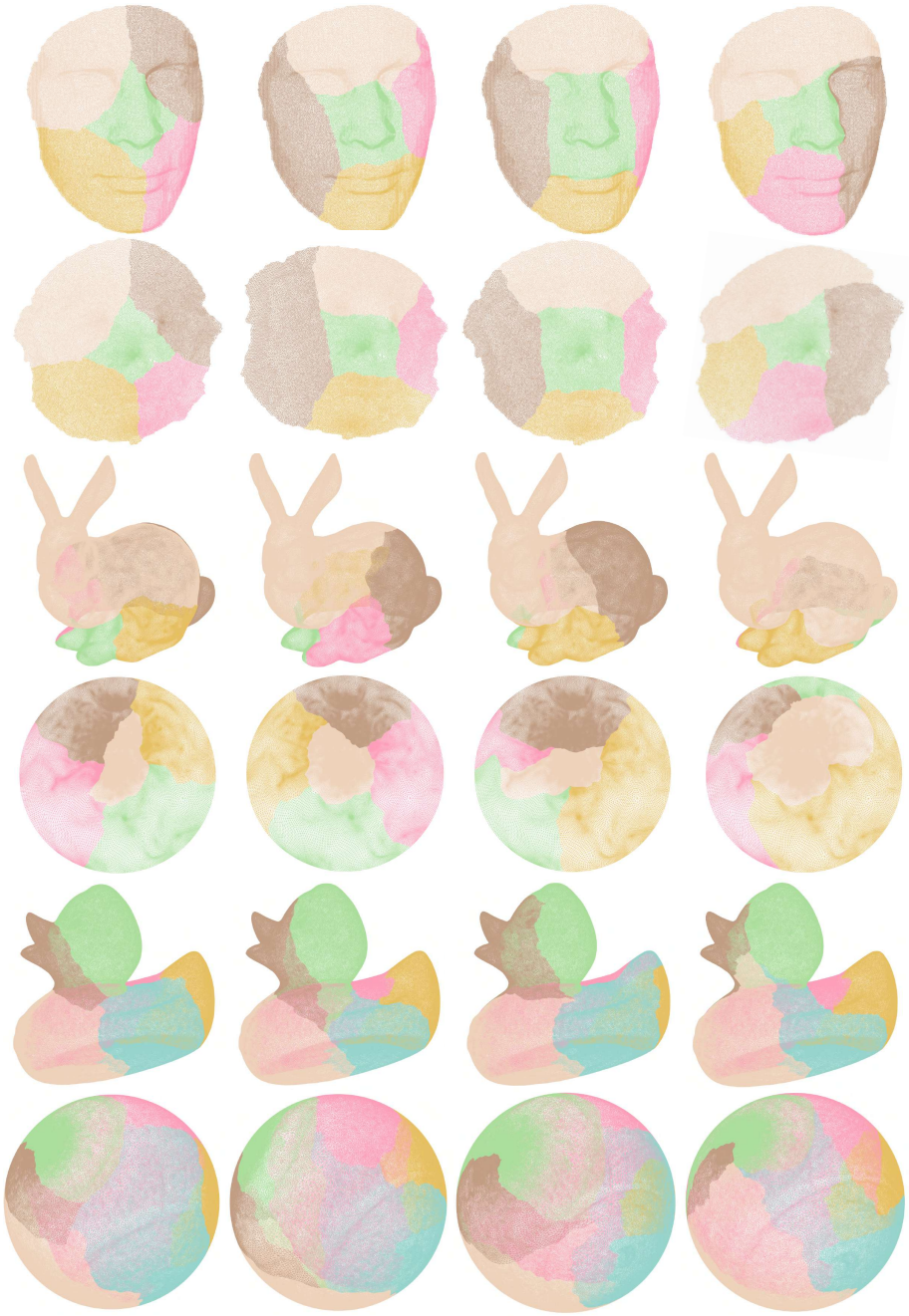

Figure 12. *The global conformal parameterizations produced by our PGCP method with different choices of cut paths. Each subdomain is with a distinct color. Top example: a simply-connected open human face surface with different cut paths, and the corresponding free-boundary conformal parameterization results. Middle example: a simply-connected open bunny surface with different cut paths, and the corresponding disk conformal parameterization results. Bottom example: a genus-0 closed duck surface with different cut paths, and the corresponding spherical conformal parameterization results.*

Figure 13. *The bunny model divided into different number of subdomains for the speedup experiment. Left to right:* $2, 3, 4, 5, 6$.

# of subdomains n	2	3	4	5	6
Average mesh size $\lvert\mathcal{V}\rvert/n$	42K	28K	21K	17K	14K
Parallel speedup S_n	1	1.31	1.66	1.91	2.05
$n/2$	1	1.5	2	2.5	3
Parallel efficiency E_n	1	0.87	0.83	0.76	0.68

Table 6

The performance of our proposed PGCP method for parameterizing the bunny model, with different number of subdomains used. Note that the welding algorithm requires at least two subdomains and hence we use $n = 2$ *as the baseline. For this reason, we define the parallel speedup to be* $S_n = \frac{T_2}{T_n}$ *where* T_i *is the time taken with* i *subdomains used, and the parallel efficiency to be* $E_n = \frac{S_n}{n/2}$.

connected component each triangle belongs to (after the mesh is cut along the set of cut edges $\mathcal{E}$). With the non-sliced variables, those computations are not parallelizable under the current MATLAB parallel computing (`parfor`) framework. For this reason, the ideal parallel efficiency cannot be achieved as n increases. We anticipate that this issue can be alleviated by performing code optimization or considering alternative implementation languages.

5.7. Applications. The above experiments demonstrate the improvement of our proposed PGCP method over the state-of-the-art conformal parameterization algorithms. In this section, we discuss the applications of it.

5.7.1. Texture mapping. One application of our proposed PGCP method is texture mapping. After conformally flattening a surface onto the plane using our method, we can design a texture on the parameter domain. Since there is a 1-1 correspondence between the input surface and the parameter domain, we can then use the inverse mapping to map the texture back onto the surface, thereby obtaining a surface with the desired texture on it. Several examples are shown in Figure 14. It is noteworthy that our method is conformal and hence the local geometry of the texture pattern is well preserved. For instance, the checkerboard texture shown in Figure 14 can maintain its orthogonality on the Ogre surface.

5.7.2. Surface remeshing. Our proposed PGCP method can also be applied to surface remeshing, which aims at improving the mesh quality of a given surface. By conformally parameterizing the surface and constructing a regular mesh structure on the parameter domain, we can use the inverse mapping to map the mesh structure back onto the surface,

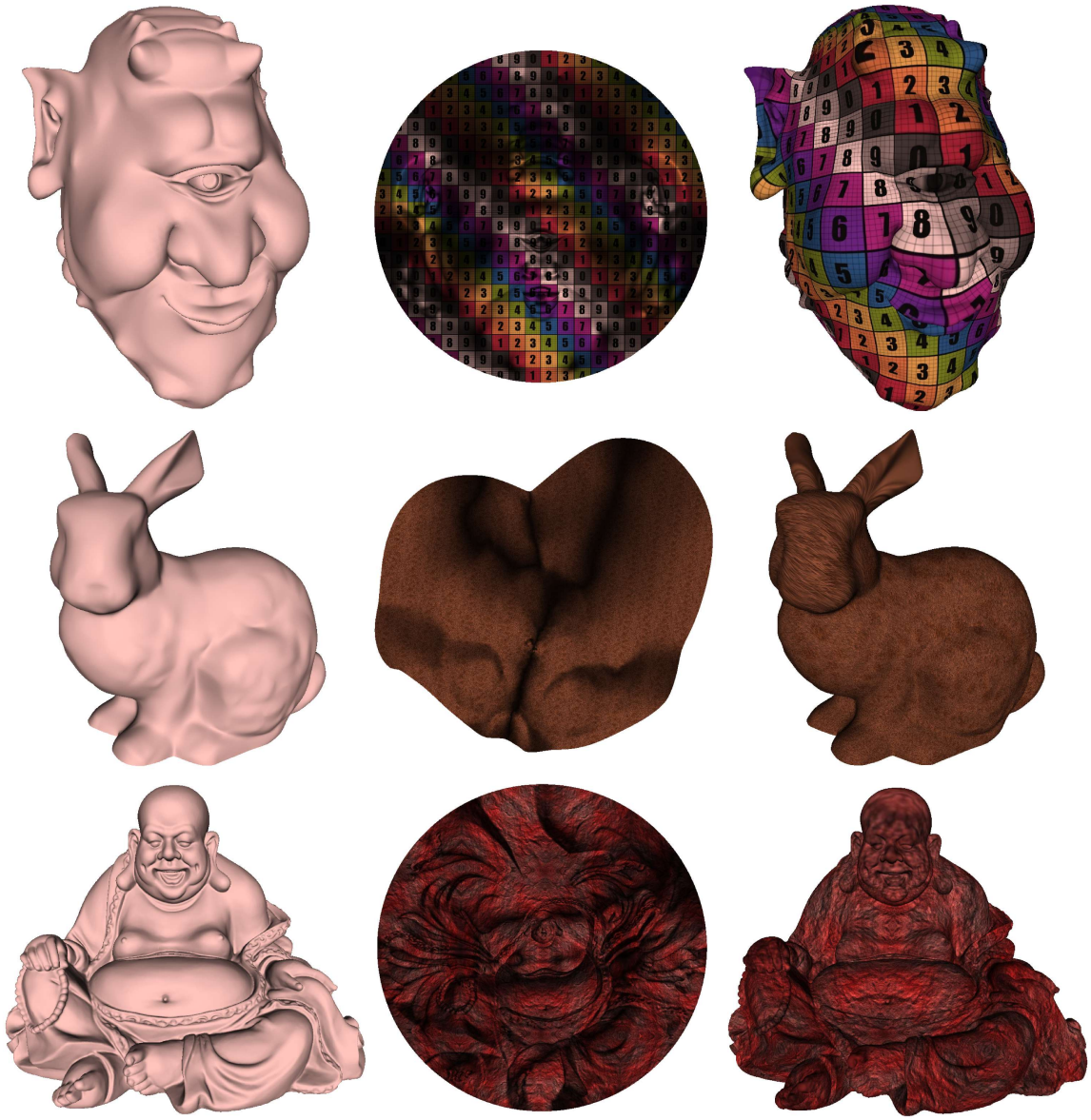

Figure 14. *Texture mapping via our proposed PGCP method. Left: the input mesh. Middle: the parameterization achieved by PGCP rendered with normal map shader, overlaid with a texture (colored checkerboard/hair/stone). Right: the texture mapping result.*

thereby remeshing the surface (see Figure 15 for example). It is noteworthy that since the parameterization is conformal, the regularity of the mesh structure defined on the parameter domain is well-preserved on the surface.

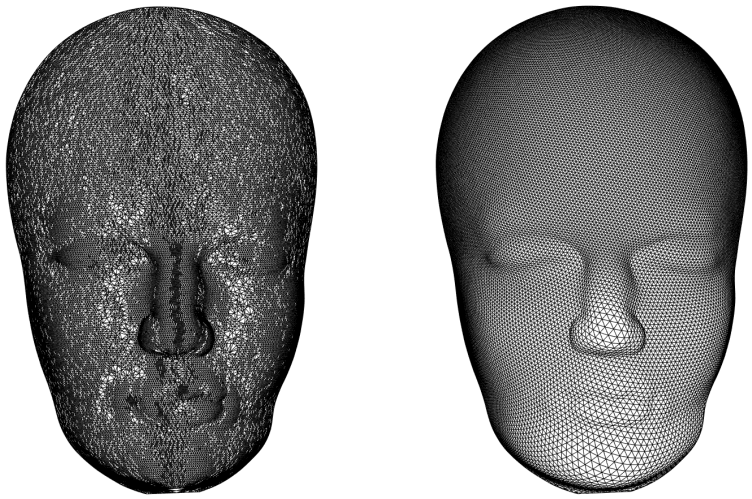

Figure 15. *Surface remeshing via our proposed PGCP method. Left: the input surface. Right: the remeshed surface with improved mesh quality.*

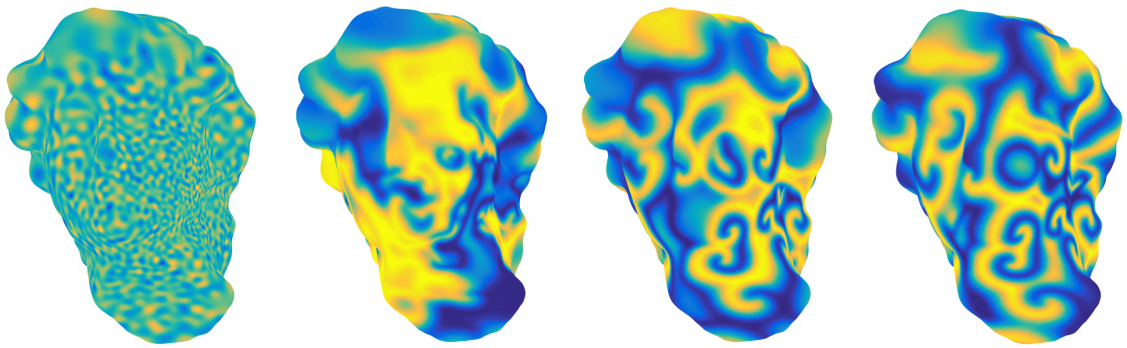

Figure 16. *Patterns formed on the genus-0 David surface by solving the Ginzburg-Landau equation on the spherical conformal parameterization. The leftmost is the initialization, and the rightmost is the final result.*

5.7.3. Solving PDEs on surfaces. Another notable application of our proposed PGCP method is solving PDEs on surfaces [42]. While solving PDEs on a general surface is difficult, solving them on a standard parameter domain such as the unit sphere or the unit disk is relatively easy. Figure 16 shows an example of patterns formed on the genus-0 David surface by solving the time-dependent Ginzburg-Landau equation on the spherical conformal parameterization obtained by our proposed PGCP method. The PDE on the sphere is solved using Chebfun [54]. The example demonstrates the use of our method for PDE-based surface decoration.

5.7.4. Other applications. Some other possible applications of conformal parameterizations include surface registration [31], medical visualization [24] and surface morphing [26].

As our proposed PGCP method is advantageous over the state-of-the-art algorithms in both the computational time and the conformal distortion, these tasks can be done with higher efficiency and accuracy using our method.

6. Discussion.

6.1. Conformality improvement. From our experimental results as shown in Tables 2, 3, 4, it can be observed that the conformality improvement achieved by our method is particularly significant for meshes with elongated parts, such as the horse model and the octopus model (see Figure 8). A possible explanation is that unlike the prior methods which compute the global conformal parameterization of a given surface by directly handling the entire surface, our method takes advantage of surface splitting. Note that in the step of solving the Laplace equation for the entire surface, the elongated parts of the surface are extremely squeezed relative to the boundary of the parameter domain (here the boundary corresponds to either the actual boundary of an open surface, or a triangle/vertex star of a closed surface), leading to numerical inaccuracy. Also, all elongated parts have to be taken into account in one single solve. By contrast, our method divides the meshes into subdomains and solves the Laplace equation for each subdomain, making the elongated parts less squeezed relative to the boundary of each flattened subdomain. Also, the number of elongated parts involved in each solve is reduced by the surface partition.

6.2. Area distortion. Note that one common issue of global conformal parameterizations is that the area is largely distorted [16]. While we have demonstrated the improvement in efficiency and conformality by our proposed method for computing global conformal parameterizations, one may also be interested in the area distortion produced by our method. To quantify the area distortion, for a surface mesh $\mathcal{S} = (\mathcal{V}, \mathcal{F})$ and a parameterization mapping $\varphi : \mathcal{S} \to \mathbb{R}^2$ or $\mathbb{S}^2$, we define the area distortion of a triangle $T \in \mathcal{F}$ by

$$d_{\text{area}}(T) = \log_e \frac{\text{Area}(f(T)) / \left(\sum_{T' \in \mathcal{F}} \text{Area}(f(T'))\right)}{\text{Area}(T) / \left(\sum_{T' \in \mathcal{F}} \text{Area}(T')\right)}. \tag{6.1}$$

In other words, d_{area} measures the logged area ratio between the triangle in the original mesh and the corresponding triangle in the parameter domain, with two normalization factors removing the global area difference between the original mesh and the parameter domain. $d_{\text{area}} \approx 0$ indicates that the area distortion is small, and a large value of $|d_{\text{area}}|$ indicates that the area distortion is large (i.e. the triangle is either shrunk or magnified).

Table 7 shows the area distortion of the BFF method [56] and the proposed PGCP method for various types of global conformal parameterizations. For spherical conformal parameterization, our method achieves a lower area distortion. For disk conformal parameterization, the two methods achieve similar area distortions. For free-boundary conformal parameterization, the BFF method possesses a lower area distortion.

The larger area distortion produced by our method in some cases is due to the lack of area control throughout the algorithm. To reduce the area distortion, one possible way is to include an extra step of composing the parameterization mapping φ with some Möbius transformations in our method. The conformality of the parameterization will be preserved as Möbius transformations are conformal, and the area distortion can be reduced by choosing a

Parameterization	Surface	BFF [56]	PGCP	PGCP with an additional Möbius transformation
Free-boundary	Sophie	0.18	0.21	0.18
	Niccolò da Uzzano	0.60	0.84	0.57
	Mask	0.24	0.58	0.32
	Max Planck	2.49	2.62	2.50
	Bunny	2.68	3.32	2.95
	Julius	0.28	1.04	0.52
	Buddha	0.78	1.20	1.16
Disk-boundary	Ogre	1.21	1.21	1.21
	Niccolò da Uzzano	0.76	0.86	0.57
	Brain	2.17	2.13	2.13
	Gargoyle	3.90	3.90	3.87
	Hand	5.29	5.25	5.25
	Octopus	6.79	8.13	8.13
	Buddha	0.78	0.79	0.77
Spherical	Horse	27.03	8.90	6.54
	Bulldog	6.74	1.09	1.08
	Chinese Lion	4.46	1.93	1.74
	Duck	7.92	1.00	0.84
	David	0.85	0.85	0.36
	Octopus	26.95	26.44	26.19
	Lion Vase	7.13	0.92	0.84

Table 7

The area distortion mean($|d_{area}|$) of the global conformal parameterizations produced by the boundary first flattening (BFF) method [56], the proposed PGCP method, and the proposed PGCP method with an additional step of composing with a Möbius transformation ((6.2) for free-boundary conformal parameterization, (6.3) for disk-boundary conformal parameterization, and (6.2) together with the stereographic projection for spherical conformal parameterization).

suitable Möbius transformation. For instance, for free-boundary conformal parameterization, one can search for an optimal Möbius transformation

$$f(z) = \frac{az+b}{cz+d}, \tag{6.2}$$

where $a, b, c, d \in \mathbb{C}$ with $ad - bc \neq 0$, such that the composition $f \circ \varphi$ minimizes the area distortion. For disk conformal parameterization, one can similarly search for an optimal automorphism

$$f(z) = \frac{z-\alpha}{1-\bar{\alpha}z}, \tag{6.3}$$

where $\alpha \in \mathbb{C}$ with $|\alpha| < 1$, such that the composition $f \circ \varphi$ minimizes the area distortion. For spherical conformal parameterization, denote the stereographic projection by $\tau : \mathbb{S}^2 \to \overline{\mathbb{C}}$.

One can search for an optimal Möbius transformation as described in (6.2) such that the composition $\tau^{-1} \circ f \circ \tau \circ \varphi$ minimizes the area distortion.

To verify this simple idea, we solve the above optimization problems using the MATLAB's optimization solver `fmincon`. For free-boundary and spherical conformal parameterizations, there are eight real parameters to be optimized (the real and imaginary parts of a, b, c, d). For disk conformal parameterizations, there are two real parameters to be optimized (the modulus and argument of α). The final results are recorded in the rightmost column of Table 7. It can be observed that the area distortion is effectively reduced with the aid of Möbius transformations in many cases, especially for free-boundary conformal parameterizations and spherical conformal parameterizations. The improvement for disk conformal parameterizations is relatively less significant, possibly due to the smaller number of free parameters. To achieve an even lower area distortion may require the composition with some other conformal transformations, which we plan to explore in the future.

6.3. Alternative numerical approaches for accelerating the computation. While we have demonstrated the advantages of our proposed method over the prior conformal parameterization methods in terms of the computational time and conformality using the idea of partial welding, another path for accelerating the computation is to consider alternative numerical approaches for the prior methods.

Note that the MATLAB code of CETM [18] solves an unconstrained optimization problem for the discrete conformal functional in each step using MATLAB's `fminunc` function, which is not parallelizable under the current MATLAB parallel computing framework (MATLAB only supports using parallel computing to estimate the numerical gradients for `fminunc` in case they are not supplied, but in the CETM code the gradients are already supplied). SCP [17] involves solving a generalized eigenvalue problem which is done in MATLAB using the `eigs` function. However, multithreading is currently not supported for this function for sparse matrices. Replacing these MATLAB functions by external routines is not straightforward and so we proceed to consider accelerating the CEM method [26] for disk conformal parameterizations. Note that CEM is an iterative method that solves two Laplace equations (one for the boundary nodes and one for the interior nodes) for each step, and the Laplacian matrices are unchanged throughout the iterations. Here we consider combining CEM with the Combinatorial Multigrid (CMG) method [58], which is a hybrid graph-theoretic algebraic multigrid solver that combines the strengths of multigrid with those of combinatorial preconditioning, with MATLAB implementation publicly available [59]. We replace the backslash solve (\) in the MATLAB implementation of CEM [52] by the CMG solver. Table 8 shows the performance of CEM, CEM combined with CMG, and our proposed method. It can be observed that while CEM combined with CMG demonstrates an improvement in efficiency for large problems when compared to the original CEM, our proposed method is still more advantageous in terms of both the efficiency and conformality.

There are many other numerical approaches and solvers which are worth exploring as alternative paths toward parallelization, such as the parallel sparse linear system solvers by Koutis and Miller [60], Peng and Spielman [61] and Kyng *et al.* [62], the *Lean Algebraic Multigrid (LAMG)* method [63] for solving the sparse linear systems involved in the prior parameterization methods. Incorporating these alternative numerical approaches into the

Surface	# vertices	CEM [26]		CEM [26] with CMG		PGCP	
		Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)	Time (s)	mean($\|d\|$)
Ogre	20K	0.3	2.6	0.3	2.6	0.5	1.5
Niccolò da Uzzano	25K	1.4	1.3	1.5	1.4	0.8	0.8
Brain	48K	2.9	1.5	2.8	1.5	1.3	1.5
Gargoyle	50K	2.8	2.1	3.2	2.1	1.4	1.9
Hand	53K	3.4	1.2	3.2	1.4	1.4	1.2
Octopus	150K	10.4	24.0	9.3	26.6	8.9	5.6
Buddha	240K	25.1	0.7	18.5	0.9	11.4	0.7
Nefertiti	1M	83.2	4.2	74.4	4.2	52.7	2.9

Table 8

The performance of the conformal energy minimization (CEM) [26] method, CEM combined with the Combinatorial Multigrid (CMG) [58], and our proposed PGCP method for disk conformal parameterization of simply-connected open surfaces.

current codes will require a careful consideration of the linkers and the cost of communication between different software and external libraries. Moreover, even if exploiting parallelization for solving the sparse linear systems can speed up the computation of the prior conformal parameterization methods, unlike our proposed partial welding approach, this approach is unable to improve their conformality or allow them to handle a wider class of surfaces.

7. Conclusion. In this work, we have proposed a novel parallelizable global conformal parameterization method called PGCP for simply-connected surfaces. Given a triangle mesh, we partition it into submeshes and conformally flatten each of them using DNCP. As the local parameterization results do not yield a consistent global parameterization, we extract their boundary points to integrate them using a novel technique called partial welding. Using the modified boundaries for all submeshes, harmonic maps can be computed to yield a global conformal parameterization, with bijectivity guaranteed by quasi-conformal theory. Additional steps can be included to produce disk conformal parameterizations for surfaces with boundary, and spherical conformal parameterizations for genus-0 closed surfaces.

Most parts of our proposed method, such as the initial local conformal parameterization step and the last harmonic mapping step, can be computed independently in a distributed manner. The only global computation involved in our algorithm takes merely boundary data points of the submeshes, which are much fewer than the vertices of the entire mesh. Experimental results have demonstrated the significant improvement in efficiency and accuracy achieved by our proposed method when compared to the state-of-the-art approaches for free-boundary conformal parameterization, disk conformal parameterization and spherical conformal parameterization.

For future work, we plan to explore the possibility of extending our method for quasi-conformal parameterizations and mappings [43–45]. More specifically, note that the partial welding step in our proposed method is conformal, and the quasi-conformal dilatation of a map is preserved under the composition with conformal maps. Therefore, it should be possible for us to compute quasi-conformal parameterizations and mappings for dense meshes by a combination of local quasi-conformal maps of submeshes and partial welding. Another possible

future work is the extension of our method for point clouds. As the partial welding approach uses only the boundary data points of the flattened submeshes but not the mesh structure of them, it should also be applicable for subdomains of a point cloud. Combining the partial welding approach with some existing conformal parameterization methods for disk-type point clouds will then yield a parallelizable global conformal parameterization method for point clouds.

REFERENCES

[1] M. S. Floater, K. Hormann, *Surface parameterization: A tutorial and survey*, Advances in Multiresolution for Geometric Modelling, 2005, pp.157–186.

[2] A. Sheffer, E. Praun, and K. Rose, *Mesh parameterization methods and their applications*, Found. Trends Comput. Graph. Vis., 2 (2006), pp. 105–171.

[3] K. Hormann, B. Lévy, and A. Sheffer, *Mesh parameterization: Theory and practice*, Proceeding of ACM SIGGRAPH 2007 courses, 2007, pp. 1–122.

[4] M. Desbrun, M. Meyer, and P. Alliez, *Intrinsic parameterizations of surface meshes*, Comput. Graph. Forum, 21 (2002), pp. 209–218.

[5] G. Zou, J. Hu, X. Gu, and J. Hua, *Authalic parameterization of general surfaces using Lie advection*, IEEE Trans. Vis. Comput. Graph., 17 (2011), pp. 2005–2014.

[6] X. Zhao, Z. Su, X. D. Gu, A. Kaufman, J. Sun, J. Gao, and F. Luo, *Area-preservation mapping using optimal mass transport*, IEEE Trans. Vis. Comput., 19 (2013), pp. 2838–2847.

[7] K. Su, L. Cui, K. Qian, N. Lei, J. Zhang, M. Zhang, and X. D. Gu, *Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation*, Comput. Aided Geom. Des., 46 (2016), pp. 76–91.

[8] G. P. T. Choi and C. H. Rycroft, *Density-equalizing maps for simply connected open surfaces*, SIAM J. Imaging Sci., 11 (2018), pp. 1134–1178.

[9] G. P. T. Choi, B. Chiu, and C. H. Rycroft, *Area-preserving mapping of 3D carotid ultrasound images using density-equalizing reference map*, IEEE Trans. Biomed. Eng., 2020, pp. 1–11.

[10] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau, *A novel stretch energy minimization algorithm for equiareal parameterizations*, J. Sci. Comput., 78 (2019), pp. 1353–1386.

[11] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*, ACM Trans. Graph., 21 (2002), pp. 362–371.

[12] F. Luo, *Combinatorial Yamabe flow on surfaces*, Commun. Contemp. Math., 6 (2004), pp. 765–780.

[13] A. Sheffer and E. de Sturler, *Parameterization of faceted surfaces for meshing using angle-based flattening*, Engineering with Computers, 17 (2001), pp. 326–337.

[14] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, *ABF++: Fast and robust angle based flattening*, ACM Trans. Graph., 24 (2005), pp. 311–330.

[15] R. Zayer, B. Lévy, and H.-P. Seidel, *Linear angle based parameterization*, Eurographics Symposium on Geometry Processing, 2007, pp. 135–141.

[16] L. Kharevych, B. Springborn, and P. Schröder, *Discrete conformal mappings via circle patterns*, ACM Trans. Graph., 25 (2006), pp. 412–438.

[17] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun, *Spectral conformal parameterization*, Comput. Graph. Forum, 27 (2008), pp. 1487–1494.

[18] B. Springborn, P. Schröder, and U. Pinkall, *Conformal equivalence of triangle meshes*, ACM Trans. Graph., 27 (2008), article no. 77.

[19] M. Jin, J. Kim, F. Luo, and X. Gu, *Discrete surface Ricci flow*, IEEE Trans. Vis. Comput., 14 (2008), pp. 2838–2847.

[20] Y.-L. Yang, R. Guo, F. Luo, S.-M. Hu, and X. Gu, *Generalized discrete Ricci flow*, Comput. Graph. Forum, 28 (2009), pp. 2005–2014.

[21] M. Zhang, R. Guo, W. Zeng, F. Luo, S.-T. Yau, and X. Gu, *The unified discrete surface Ricci flow*, Graph. Models, 76 (2014), pp. 321–339.

[22] P. T. Choi and L. M. Lui, *Fast disk conformal parameterization of simply-connected open surfaces*, J. Sci.

Comput., 65 (2015), pp. 1065–1090.
[23] T. W. Meng, G. P.-T. Choi, and L. M. Lui, *TEMPO: Feature-endowed Teichmüller extremal mappings of point clouds*, SIAM J. Imaging Sci., 9 (2016), pp. 1922–1962.
[24] G. P. T. Choi, Y. Chen, L. M. Lui, B. Chiu, *Conformal mapping of carotid vessel wall and plaque thickness measured from 3D ultrasound images*, Med. Biol. Eng. Comput., 55 (2017), pp. 2183-2195.
[25] G. P.-T. Choi and L. M. Lui, *A linear formulation for disk conformal parameterization of simply-connected open surfaces*, Adv. Comput. Math., 14 (2018), pp. 87-114.
[26] M.-H. Yueh, W.-W. Lin, C.-T. Wu, and S.-T. Yau, *An efficient energy minimization for conformal parameterizations*, J. Sci. Comput., 73 (2017), pp. 203–227.
[27] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, *Conformal geometry and brain flattening*, Medical image computing and computer-assisted intervention (MICCAI), 1999, pp. 271–278.
[28] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle, *Conformal surface parameterization for texture mapping*, IEEE Trans. Vis. Comput., 6 (2000), pp. 181–189.
[29] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, *Genus zero surface conformal mapping and its application to brain surface mapping*, IEEE Trans. Med. Imaging, 23 (2004), pp. 949–958.
[30] R. Lai, Z. Wen, W. Yin, X. 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.
[31] 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.
[32] G. P.-T. Choi, K. T. Ho, and L. M. Lui, *Spherical conformal parameterization of genus-0 point clouds for meshing*, SIAM J. Imaging Sci., 9 (2016), pp. 1582–1618.
[33] J. E. Hutchinson, *Computing conformal maps and minimal surfaces*, Theoretical and Numerical Aspects of Geometric Variational Problems, 1991, pp. 140–161.
[34] U. Pinkall and K. Polthier, *Computing discrete minimal surfaces and their conjugates*, Exp. Math., 2 (1993), pp. 15–36.
[35] C. J. Bishop, *Conformal welding and koebe's theorem*, Ann. Math., 2007, 613–656.
[36] O. Lehto and K. I. Virtanen, *Quasiconformal mappings in the plane*, Springer-Verlag Berlin Heidelberg, 1973.
[37] V. A. Pfluger, *Ueber die konstruktion riemannscher flachen durch verheftung*, J. Indian Math. Soc., 24 (1960), pp. 401–412. (in German)
[38] R. Kühnau, *Numerische realisierung konformer abbildungen durch interpolation*, ZAMM Z. Angew. Math. Mech., 63 (1983), pp. 631–637. (in German)
[39] D. E. Marshall and J. A. Morrow, *Compositions of slit mappings*, Manuscript, 1987.
[40] D. E. Marshall and S. Rohde, *Convergence of a variant of the zipper algorithm for conformal mapping*, SIAM J. Numer. Anal., 45 (2007), pp. 2577–2609.
[41] F. Gardiner and N. Lakic, *Quasiconformal Teichmüller theory*. American Mathematics Society (2000).
[42] L. M. Lui, Y. Wang, and T. F. Chan, *Solving PDEs on manifolds with global conformal parametrization*. International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, 2005, pp. 307–319.
[43] G. P.-T. Choi, M. H.-Y. Man, and L. M. Lui, *Fast spherical quasiconformal parameterization of genus-0 closed surfaces with application to adaptive remeshing*, Geom. Imag. Comput., 3 (2016), pp. 1–29.
[44] G. P. T. Choi and L. Mahadevan, *Planar morphometrics using Teichmüller maps*, Proc. R. Soc. A, 474 (2018), 20170905.
[45] G. P. T. Choi, H. L. Chan, R. Yong, S. Ranjitkar, A. Brook, G. Townsend, K. Chen, and L. M. Lui, *Tooth morphometry using quasi-conformal theory*, Pattern Recognit., 99 (2020), 107064.
[46] AIM@SHAPE Shape Repository. http://visionair.ge.imati.cnr.it/ontologies/shapes/
[47] The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/
[48] TurboSquid. http://www.turbosquid.com/
[49] Keenan's 3D Model Repository. https://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository/
[50] Bundled CETM Code. https://people.mpi-inf.mpg.de/~chen/bdmorph/
[51] Linear Disk Conformal Parameterization. https://scholar.harvard.edu/choi/software-demos
[52] Disk-Shaped Conformal Parameterization. https://www.mathworks.com/matlabcentral/fileexchange/64258-disk-shaped-conformal-parameterization
[53] Spherical Conformal Map. https://www.mathworks.com/matlabcentral/fileexchange/

65551-spherical-conformal-map
[54] Chebfun. http://www.chebfun.org/
[55] M. Botsch, D. Bommes, and L. Kobbelt, *Efficient linear system solvers for mesh processing*. Mathematics of Surfaces XI, 2005, pp. 62–83.
[56] R. Sawhney and K. Crane, *Boundary first flattening*, ACM Trans. Graph., 37 (2018), 5.
[57] Boundary First Flattening (BFF). https://github.com/GeometryCollective/boundary-first-flattening
[58] I. Koutis, G. L. Miller, and D. Tolliver, *Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing*. Computer Vision and Image Understanding, 115 (2011), pp. 1638–1646.
[59] CMG: Combinatorial Multigrid. http://www.cs.cmu.edu/~jkoutis/cmg.html
[60] I. Koutis and G. L. Miller, *A linear work, $O(n^{1/6})$ time, parallel algorithm for solving planar Laplacians*. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA), 2007, pp. 1002–1011.
[61] R. Peng and D. A. Spielman, *An efficient parallel solver for SDD linear systems*. Proceedings of the forty-sixth annual ACM symposium on Theory of computing (STOC), 2014, pp. 333–342.
[62] R. Kung, Y. T. Lee, R. Peng, S. Sachdeva, and D. A. Spielman, *Sparsified Cholesky and Multigrid Solvers for Connection Laplacians*. Proceedings of the forty-eighth annual ACM symposium on Theory of Computing (STOC), 2016, pp. 842–850.
[63] O. E. Livne and A. Brandt, *Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver*. SIAM J. Sci. Comput. 34 (2012), pp. B499–B522.