IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS

Discrete Surface Ricci Flow

Miao Jin[1], Junho Kim[1,3], Feng Luo[2], and Xianfeng Gu[1]
[1]Stony Brook University [2]Rutgers University [3]Dong-Eui University

***Abstract*— This work introduces a unified framework for discrete surface Ricci flow algorithms, including spherical, Euclidean, and hyperbolic Ricci flows, which can design Riemannian metrics on surfaces with arbitrary topologies by user-defined Gaussian curvatures. Furthermore, the target metrics are conformal (angle-preserving) to the original metrics.**

Ricci flow conformally deforms the Riemannian metric on a surface according to its induced curvature, such that the curvature evolves like a heat diffusion process. Eventually, the curvature becomes the user defined curvature.

Discrete Ricci flow algorithms are based on a variational framework. Given a mesh, all possible metrics form a linear space, and all possible curvatures form a convex polytope. The Ricci energy is defined on the metric space, which reaches its minimum at the desired metric. The Ricci flow is the negative gradient flow of the Ricci energy. Furthermore, the Ricci energy can be optimized using Newton's method more efficiently.

Discrete Ricci flow algorithms are rigorous and efficient. Our experimental results demonstrate the efficiency, accuracy and flexibility of the algorithms. They have the potential for a wide range of applications in graphics, geometric modeling, and medical imaging. We demonstrate their practical values by global surface parameterizations.

***Index Terms*— Discrete Ricci flow, Riemannian metric, uniformization, global conformal parameterizations, manifold**

I. Introduction

Ricci flow is a curvature flow method, which has been applied to the proof of the Poincaré conjecture on three dimensional manifolds [1]–[3]. Ricci flow was introduced by Richard Hamilton for general Riemannian manifolds in his seminal work [4] in 1982.

a) Physical Intuition: Ricci flow has a simple physical intuition. Given a surface with a Riemannian metric, the metric induces the Gaussian curvature function. If the metric is changed, then the Gaussian curvature will be changed accordingly. We deform the metric in the following way: at each point, we locally scale the metric, such that the scaling factor is proportional to the curvature at the point. After the deformation, the curvature will be changed. We repeat the deformation process, then both the metric and the curvature will evolve, such that the curvature evolution is like a heat diffusion process. Eventually, the Gaussian curvature function is constant everywhere. If the surface is closed and simply connected, then the surface becomes a sphere eventually. (The analogy of this process for three dimensional manifolds is the basic idea of the proof of the Poincaré conjecture.)

b) Motivations: Surface Ricci flow is a powerful tool to design a Riemannian metric, such that the metric induces the user-defined Gaussian curvature function on the surface, and is conformal (i.e.,angle-preserving) to the original metric. Many applications in engineering fields can be formulated as finding some metrics with desired properties, where Ricci flow can be directly utilized.

In graphics, a surface parametrization is commonly used, which refers to the process of mapping a surface to another canonical domain. If the domain is planar, then it is equivalent to finding a Riemannian metric, which induces 0 Gaussian curvature everywhere. (Such a metric is called a flat metric.)

In digital geometry processing, if such a parameterization is known, then any signal (e.g. texture) on the surface can be defined on the parametric domain. Complicated processing tasks on surfaces can be transferred to easier ones on the parametric domains, such as texturing [5] and re-meshing [6].

In computer-aided geometric modeling, a flat metric is helpful for constructing manifold splines, whose parametric domains are manifolds with arbitrary topologies instead of planar domains. In order to build a manifold spline, a special atlas of the domain manifold needs to be found, such that all local coordinate transition maps are affine. One way to construct such an atlas is to find a flat metric. Details of the manifold theory and the construction of an affine atlas can be found in [7].

In the medical imaging field, conformal brain mapping has been widely used, which maps the human brain cortical surfaces to the unit sphere to facilitate registration, fusion, and comparison. This is equivalent to finding a Riemannian metric on the brain cortical surface, such that the induced Gaussian curvature is a constant $+1$ everywhere.

c) Brief History: For engineering applications, Ricci flow theory on smooth surfaces needs to be generalized to discrete algorithms on piecewise linear meshes.

Ricci flow on surfaces was first introduced by Hamilton in [8]. A circle packing algorithm was introduced by Thurston in [9]. Chow and Luo discovered their intrinsic relations and laid down the theoretic foundation for discrete Ricci flow in [10], where the existence and convergence of the discrete Ricci flow were established. However, that work only considers the combinatorial structures of triangular meshes, and the algorithm is not efficient in practice. Because the discrete Ricci flow algorithm is the gradient descent algorithm, it has been improved by using Newton's method in our later works on discrete hyperbolic Ricci flow [11] and discrete Euclidean Ricci flow [12].

In this work, we introduce the discrete spherical Ricci flow based on Newton's method for the first time to fulfill our goal of using Ricci flow to handle all sorts of surfaces with different topologies. We provide a simple and unified framework for all three types of discrete Ricci flow algorithms, which is a complete system to design Riemannian metrics with user-defined Gaussian curvatures (if there is no user defined curvature, the target curvature is set to be constant), which are conformal to the original induced Euclidean metric.

d) Organization: The paper is organized as follows: Previous works are briefly surveyed in Section II. Section III introduces the most important concepts in differential geometry which are crucial for designing our algorithms. Section IV presents the

Digital Object Indentifier 10.1109/TVCG.2008.57

theory of discrete surface Ricci flow. Specific algorithms are given in Section V. The performance of the algorithms is analyzed and one application of the algorithms in computer graphics is demonstrated in Section VI. Finally, the conclusion and future work are given in Section VII.

II. Previous Work

There are many applications in graphics and geometric modeling directly related to computing a desired Riemannian metric with user-defined Gaussian curvatures on a given surface as discussed in Section I. Before demonstrating the effectiveness of discrete Ricci flow for these applications, we first give a brief survey of current works in these areas.

A lot of research has been done on mesh parameterization due to its usefulness in computer graphics applications. The surveys of [13], [14] provide excellent reviews on various kinds of mesh parameterization techniques. Here we only review the most related ones. For more details, we refer readers to [13], [14]. Among previous research on mesh parameterization, conformal mesh parameterization is distinguished by its angle preserving nature, and many researchers developed algorithms of conformality mappings [5], [15]–[19]. Lévy *et al*. [5] applied the Cauchy-Riemann equation for mesh parameterization and provided successful results on the constrained 2D parameterizations with free boundaries. Desbrun *et al*. [17] minimized the Dirichlet energy defined on triangle meshes for computing conformal parameterization. Angle based flattening methods [16], [18] approximate conformal mappings directly by minimizing angle distortions. Gu and Yau in [19] computed the conformal structure using Hodge theory. A flat metric of the given surface is induced by computing the holomorphic 1-form with a genus-related number of singularities to obtain a globally smooth parameterization. Gortler *et al*. [20] used discrete 1-forms for mesh parameterization. Their approach provided good results for mesh parameterization with several holes, but they cannot control the curvatures on the boundaries. Ray *et al*. [21] used the holomorphic 1-form to follow the principal curvatures on manifolds and computed a quad-dominated parameterization for arbitrary models. Tong *et al*. [22] utilized harmonic 1-forms for designing quadrangulations. Fisher *et al*. [23] used discrete 1-forms for tangential vector fields design.

Circle packing was first studied by Koebe in 1928. Thurston reinvented it in 1978 [9] in order to find hyperbolic metrics on 3-manifolds. A practical software system for circle packing with an improved algorithm can be found in [24], which considers the combinatorial structure of the triangulation only. A different approach, which generalizes the tangential circle pattern, was proposed by Bobenko-Springborn in [25], [26]. They used the notion of angle structures first introduced by Colin de Verdiére [27] in 1991. The work in [27] is seminal, and first introduced the variational approach to circle packing theory. The practical implementation of [25] was carried out in [28], and was used for surface parameterization on Euclidean geometry. The difference between the work of [28] and ours is as follows.

The circle pattern and the discrete Ricci flow are the Legendre dual to each other (the concept of Legenendre dual is explained in details [29]). Therefore, the convexity of circle pattern energy is equivalent to the convexity of the discrete Ricci energy, and the convergence of one is equivalent to the other. The approach in [25], [28] uses angles as variables and ours uses edge lengths. However, our approach works for circle packing where circles could have intersection angles, which cannot be covered by [25], [28]. The practical application of circle patterns has been provided only for the Euclidean case in [28]. Circle pattern has defined curvatures on both vertices and faces, while discrete Ricci flow has only vertex curvatures. In contrast, our work provides the discrete theories as well as the applications for generally topological surfaces under Euclidean, spherical, and hyperbolic parametrization domains.

For surfaces with spherical topologies there exist several methods for parameterizing the surface onto the sphere. This approach is used for texture mapping in [30], [31] and for conformal brain mappings in [32], [33].

Affine structure is another common geometric structure on arbitrary surfaces, applied for constructing manifold splines on general surfaces in [7], where the affine structures are induced by holomorphic differentials computed using the algorithms in [19], [34].

Hyperbolic structure was applied in [35], [36] for the topological design of surfaces, where the high genus surfaces were represented as quotient spaces of the Poincaré disk over Fuchsian group actions, which results in a single domain, global parameterization for surfaces with handles. In [37], Grimm and Hughes defined parameterizations for high genus surfaces and constructed functions on them. Wallner and Pottmann introduced the concept of spline orbifold in [38], which defined splines on three canonical parameter domains: the sphere, the plane, and the Poincaré disk. However, in their methods, only the combinatorial structure of the mesh is used, while the induced Euclidean metric is not considered. For many applications in graphics, such as texture mapping, shape analysis, and spline constructions, original geometry information is highly desirable.

III. Theoretical Background on Continuous Surfaces

In this section, we introduce the theory of Ricci flow in the continuous setting, which will be generalized to the discrete setting in Section IV. Some basic concepts in differential geometry can be found in Appendix VIII-B.

For readers who are more interested in implementation details, this section can be skipped. For those who are more interested in inventing new schemes to approximate Ricci flow or to improve the existing algorithms, they can find inspiration from this section.

Conformal deformation Let S be a surface embedded in $\mathbb{R}^3$. S has a Riemannian metric induced from the Euclidean metric of $\mathbb{R}^3$, denoted by $\mathbf{g}$. Suppose $u : S \to \mathbb{R}$ is a scalar function defined on S. It can be verified that $\bar{\mathbf{g}} = e^{2u}\mathbf{g}$ is also a Riemannian metric on S. Furthermore, angles measured by $\mathbf{g}$ are equal to those measured by $\bar{\mathbf{g}}$. Therefore, we say $\bar{\mathbf{g}}$ is a *conformal deformation* from $\mathbf{g}$.

A conformal deformation maps infinitesimal circles to infinitesimal circles and preserves the intersection angles among the infinitesimal circles. In Fig. 1, we illustrate this property by approximating infinitesimal circles by finite circles. We put a regular circle packing pattern on the texture and map the texture to the surface using a conformal parameterization, where all the circles on the texture still look like circles on the surface, and all

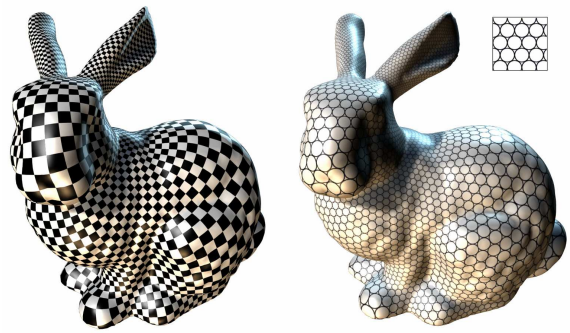
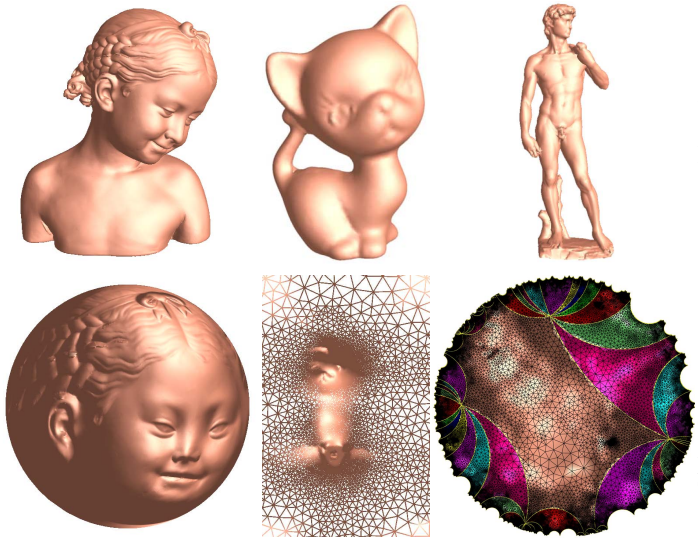


Fig. 1. **Properties of Conformal Mapping**: Conformal mappings transform infinitesimal circles to infinitesimal circles and preserve the intersection angles among the circles. Here, infinitesimal circles are approximated by finite ones. Notice that a circle in the texture appears in a scaled one in the texture mapping result. Also, the angles in the checkerboard pattern preserved in the texture mapping result.



(a) $\chi > 0,\ \mathbb{S}^2$ (b) $\chi = 0,\ \mathbb{E}^2$ (c) $\chi < 0,\ \mathbb{H}^2$

Fig. 2. **Uniformization Theorem**: Each surface in $\mathbb{R}^3$ admits a uniformization metric, which is conformal to the original metric and induces constant Gaussian curvature; the constant is one of $\{+1,0,-1\}$ depending on the Euler characteristic number χ of the surface. Its universal covering space with the uniformization metric can be isometrically embedded onto one of three canonical spaces: sphere, plane, or hyperbolic space. Here, we shows the parameterizations computed by using discrete spherical, Euclidean, and hyperbolic Ricci flows, respectively.

the tangency relations among the circles are preserved.

When the Riemannian metric is conformally deformed, curvatures will also be changed accordingly. Suppose $\mathbf{g}$ is changed to $\bar{\mathbf{g}} = e^{2u}\mathbf{g}$. Then, the Gaussian curvature will become

$$\bar{K} = e^{-2u}(-\Delta_{\mathbf{g}} u + K), \tag{1}$$

where $\Delta_{\mathbf{g}}$ is the Laplacian-Beltrami operator under the original metric $\mathbf{g}$. The geodesic curvature will become

$$\bar{k} = e^{-u}(\partial_{\mathbf{r}} u + k), \tag{2}$$

where $\mathbf{r}$ is the tangent vector orthogonal to the boundary. According to the Gauss-Bonnet theorem, the total curvature is still $2\pi\chi(S)$, where $\chi(S)$ is the Euler characteristic number of S.

Uniformization Theorem Given a surface S with a Riemannian metric $\mathbf{g}$, there exist an infinite number of metrics conformal to $\mathbf{g}$. The following uniformization theorem states that, among all of the conformal metrics, there exists a unique representative, which induces constant curvature. Moreover, the constant will be one of $\{+1,0,-1\}$.

Theorem 1 (Uniformization Theorem): Let $(S,\mathbf{g})$ be a compact 2-dimensional surface with a Riemannian metric $\mathbf{g}$, then there is a metric $\bar{\mathbf{g}}$ conformal to $\mathbf{g}$ with constant Gaussian curvature everywhere; the constant is one of $\{+1,0,-1\}$.

We call such a metric the *uniformization metric* of S. According to the Gauss-Bonnet theorem (Eq. 13), the sign of the constant Gaussian curvature must match the sign of the Euler number of the surface: $+1$ for $\chi(S) > 0$, 0 for $\chi(S) = 0$, and -1 for $\chi(S) < 0$.

Therefore, we can embed the universal covering space of any closed surface using its uniformization metric onto one of the three canonical surfaces: the *sphere* $\mathbb{S}^2$ for genus zero surfaces with positive Euler numbers, the *plane* $\mathbb{E}^2$ for genus one surfaces with zero Euler number, and the *hyperbolic space* $\mathbb{H}^2$ for high genus surfaces with negative Euler numbers (see Fig. 2). Accordingly, we can say that surfaces with positive Euler number admit spherical geometry; surfaces with zero Euler number admit Euclidean geometry; and surfaces with negative Euler number admit hyperbolic geometry.

Smooth Surface Ricci Flow Suppose S is a smooth surface with a Riemannian metric $\mathbf{g}$. The Ricci flow deforms the metric $\mathbf{g}(t)$ according to the Gaussian curvature $K(t)$ (induced by itself), where t is the time parameter

$$\frac{dg_{ij}(t)}{dt} = -2K(t)g_{ij}(t). \tag{3}$$

There is an analogy between the Ricci flow and the heat diffusion process. Suppose $T(t)$ is a temperature field on the surface. The heat diffusion equation is $dT(t)/dt = -\Delta_{\mathbf{g}} T(t)$, where $\Delta_{\mathbf{g}}$ is the Laplace-Beltrami operator induced by the surface metric. The temperature field becomes more and more uniform with the increase of t, and it will become constant eventually.

In a physical sense, the curvature evolution induced by the Ricci flow is exactly the same as heat diffusion on the surface, as follows:

$$\frac{dK(t)}{dt} = -\Delta_{\mathbf{g}(t)} K(t), \tag{4}$$

where $\Delta_{\mathbf{g}(t)}$ is the Laplace-Beltrami operator induced by the metric $\mathbf{g}(t)$. If we replace the metric in Eq. 3 with $g(t) = e^{2u(t)}g(0)$, then the Ricci flow can be simplified as

$$\frac{du(t)}{dt} = -2K(t), \tag{5}$$

which states that the metric should change according to the curvature.

The following theorems postulate that the Ricci flow defined in Eq. 3 is convergent and leads to a conformal uniformization metric. For surfaces with non-positive Euler numbers, Hamilton proved the convergence of Ricci flow in [8]:

Theorem 2 (Hamilton 1988): For a closed surface of non-positive Euler characteristic, if the total area of the surface is preserved during the flow, the Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere.

It is much more difficult to prove the convergence of Ricci flow on surfaces with positive Euler numbers. The following result was

proven by Chow in [39]:

Theorem 3 (Chow 1991): For a closed surface of positive Euler characteristic, if the total area of the surface is preserved during the flow, the Ricci flow will converge to a metric such that the Gaussian curvature is constant everywhere.

The corresponding metric $\mathbf{g}(\infty)$ is the *uniformization metric*. Moreover, at any time t, the metric $\mathbf{g}(t)$ is conformal to the original metric $\mathbf{g}(0)$.

The Ricci flow can be easily modified to compute a metric with a *user-defined* curvature $\bar{K}$ as the following,

$$\frac{du(t)}{dt} = 2(\bar{K} - K). \tag{6}$$

With this modification, the solution metric $\mathbf{g}(\infty)$ can be computed, which induces the curvature $\bar{K}$.

IV. Theoretical Background on Discrete Surfaces

In engineering fields, smooth surfaces are often approximated by simplicial complexes (triangle meshes). Major concepts, such as metrics, curvature, and conformal deformation in the continuous setting can be generalized to the discrete setting. We denote a triangle mesh as Σ, a vertex set as V, an edge set as E, and a face set as F. e_{ij} represents the edge connecting vertices v_i and v_j, and f_{ijk} denotes the face formed by v_i, v_j, and v_k.

Background Geometry In graphics, it is always assumed that a mesh Σ is embedded in the three dimensional Euclidean space $\mathbb{R}^3$, and therefore each face is Euclidean. In this case, we say the mesh is with Euclidean background geometry (see Fig. 2(a)). The angles and edge lengths of each face satisfy the Euclidean cosine law.

Similarly, if we assume that a mesh is embedded in the three dimensional sphere $\mathbb{S}^2$, then each face is a spherical triangle. We say the mesh is with spherical background geometry (see Fig. 2(b)). The angles and the edge lengths of each face satisfy the spherical cosine law.

Furthermore, if we assume that a mesh is embedded in the three dimensional hyperbolic space $\mathbb{H}^2$, then all faces are hyperbolic triangles. We say the mesh is with hyperbolic background geometry (see Fig. 2(c)). The angles and the edge lengths of each face satisfy the hyperbolic cosine law.

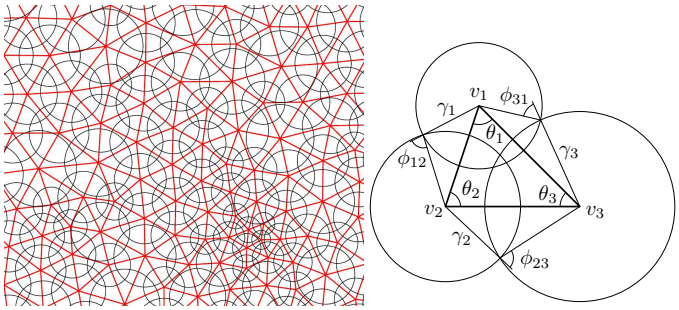
In the following discussion, we will explicitly specify the background geometry for a mesh when it is needed. Otherwise, the concept or the algorithm is appropriate for all kinds of the background geometries.

Discrete Riemannian Metric A Riemannian metric on a mesh Σ is a piecewise constant metric with cone singularities. A metric on a mesh with Euclidean metric is a Euclidean metric with cone singularities. Each vertex is a cone singularity. Similarly, a metric on a mesh with spherical background geometry is a spherical metric with cone singularities; a metric on a mesh with hyperbolic background geometry is a hyperbolic metric with cone singularities.

The edge lengths of a mesh Σ are sufficient to define this Riemannian metric,

$$l : E \to \mathbb{R}^+,$$

as long as, for each face f_{ijk}, the edge lengths satisfy the triangle inequality: $l_{ij} + l_{jk} > l_{ki}$.



(a) (b)

Fig. 3. **Circle Packing Metric** (a) Flat circle packing metric (b) Circle packing metric on a triangle.

Discrete Gaussian Curvature The discrete Gaussian curvature K_i on a vertex $v_i \in \Sigma$ can be computed from the angle deficit,

$$K_i = \begin{cases} 2\pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, & v_i \notin \partial\Sigma \\ \pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, & v_i \in \partial\Sigma \end{cases} \tag{7}$$

where θ_i^{jk} represents the corner angle attached to vertex v_i in the face f_{ijk}, and $\partial\Sigma$ represents the boundary of the mesh. The discrete Gaussian curvatures are determined by the discrete metrics.

Discrete Gauss-Bonnet Theorem The Gauss-Bonnet theorem (Eq. 13) states that the total curvature is a topological invariant. It still holds on meshes as follows.

$$\sum_{v_i \in V} K_i + \lambda \sum_{f_i \in F} A_i = 2\pi\chi(M), \tag{8}$$

where A_i denotes the area of face f_i, and λ represents the constant curvature for the background geometry; $+1$ for the spherical geometry, 0 for the Euclidean geometry, and -1 for the hyperbolic geometry.

Discrete Conformal Deformation Conformal metric deformations preserves infinitesimal circles and the intersection angles among them. The discrete conformal metric deformation of metrics uses circles with finite radii to approximate the infinitesimal circles.

The concept of the circle packing metric was introduced by Thurston in [9] as shown in Fig. 3. Let Γ be a function defined on the vertices, $\Gamma : V \to \mathbb{R}^+$, which assigns a radius γ_i to the vertex v_i. Similarly, let Φ be a function defined on the edges, $\Phi : E \to [0, \frac{\pi}{2}]$, which assigns an acute angle $\Phi(e_{ij})$ to each edge e_{ij} and is called a *weight* function on the edges. The pair of vertex radius function and edge weight function on a mesh Σ, (Γ, Φ), is called a *circle packing metric* of Σ.

Fig. 3 illustrates the circle packing metrics. Each vertex v_i has a circle whose radius is γ_i. For each edge e_{ij}, the intersection angle ϕ_{ij} is defined by the two circles of v_i and v_j, which either intersect or are tangent.

Two circle packing metrics (Γ_1, Φ_1) and (Γ_2, Φ_2) on the same mesh are *conformally equivalent* if $\Phi_1 \equiv \Phi_2$. A *conformal deformation* of a circle packing metric only modifies the vertex radii and preserves the intersection angles on the edges.

Admissible Curvature Space A mesh Σ with edge weight Φ is called a *weighted mesh*, which is denoted as (Σ, Φ). In the following, we want to clarify the spaces of all possible circle packing metrics and all possible curvatures of a weighted mesh.

Let the vertex set be $V = \{v_1, v_2, \cdots, v_n\}$, and the radii be $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$. Let u_i be

$$u_i = \begin{cases} \log \gamma_i & \mathbb{E}^2 \\ \log \tanh \frac{\gamma_i}{2} & \mathbb{H}^2 \\ \log \tan \frac{\gamma_i}{2} & \mathbb{S}^2 \end{cases} \tag{9}$$

where $\mathbb{E}^2$, $\mathbb{H}^2$, and $\mathbb{S}^2$ indicate the background geometry of the mesh. We represent a circle packing metric on (Σ, Φ) by a vector $\mathbf{u} = (u_1, u_2, \cdots, u_n)^T$. Similarly, we represent the Gaussian curvatures at mesh vertices by the curvature vector $\mathbf{k} = (K_1, K_2, \cdots, K_n)^T$. All the possible $\mathbf{u}$'s form the *admissible metric space*, and all the possible $\mathbf{k}$'s form the *admissible curvature space*.

According to the Gauss-Bonnet theory (Eq. 8), the total curvature must be $2\pi\chi(\Sigma)$, and therefore the curvature space is $n-1$ dimensional. We add one linear constraint to the metric vector $\mathbf{u}$, $\sum u_i = 0$, for the normalized metric. As a result, the metric space is also $n-1$ dimensional. If all the intersection angles are acute, then the edge lengths induced by a circle packing satisfy the triangle inequality. There is no further constraint on $\mathbf{u}$. Therefore, the admissible metric space is simply $\mathbb{R}^{n-1}$.

A curvature vector $\mathbf{k}$ is *admissible* if there exists a metric vector $\mathbf{u}$, which induces $\mathbf{k}$. The admissible curvature space of a weighted mesh (Σ, Φ) is a convex polytope, specified by the following theorem. The detailed proof can be found in [10].

Theorem 4: Suppose (Σ, Φ) is a weighted mesh with Euclidean background geometry, I is a proper subset of vertices, F_I is the set of faces whose vertices are in I and the link set $Lk(I)$ is formed by faces (e, v), where e is an edge and v is the third vertex in the face,

$$Lk(I) = \{(e, v) | e \cap I = \emptyset, v \in I\},$$

then a curvature vector $\mathbf{k}$ is admissible if and only if

$$\sum_{v_i \in I} K_i > -\sum_{(e,v) \in Lk(I)} (\pi - \phi(e)) + 2\pi\chi(F_I).$$

The admissible curvature spaces for weighted meshes with hyperbolic or spherical background geometries are more complicated. We refer readers to [29] for detailed discussions.

Discrete Surface Ricci Flow Suppose (Σ, Φ) is a weighted mesh with an initial circle packing metric. The discrete Ricci flow is defined as follows.

$$\frac{du_i(t)}{dt} = (\bar{K}_i - K_i), \tag{10}$$

where $\bar{\mathbf{k}} = (\bar{K}_1, \bar{K}_2, \cdots, \bar{K}_n)^T$ is the user defined target curvature. Discrete Ricci flow is in the exact same form as the smooth Ricci flow (Eq. 6), which deforms the circle packing metric according to the Gaussian curvature, as in Eq. 10.

Discrete Ricci flow can be formulated in the variational setting, namely, it is a negative gradient flow of a special energy form. Let (Σ, Φ) be a weighted mesh with spherical (Euclidean or hyperbolic) background geometry. For two arbitrary vertices v_i and v_j, the following symmetric relation holds:

$$\frac{\partial K_i}{\partial u_j} = \frac{\partial K_j}{\partial u_i}.$$

Let $\omega = \sum_{i=1}^n K_i du_i$ be a differential one-form [40]. The symmetric relation guarantees that the one-form is closed (curl free) in the metric space.

$$d\omega = \sum_{i,j} (\frac{\partial K_i}{\partial u_j} - \frac{\partial K_j}{\partial u_i}) du_i \wedge du_j = 0.$$

By Stokes theorem, the following integration is path independent,

$$f(\mathbf{u}) = \int_{\mathbf{u_0}}^{\mathbf{u}} \sum_{i=1}^n (\bar{K}_i - K_i) du_i, \tag{11}$$

where $\mathbf{u_0}$ is an arbitrary initial metric. Therefore, the above integration is well defined, so called the *discrete Ricci energy*. The discrete Ricci flow is the negative gradient flow of the discrete Ricci energy. The discrete metric which induces $\bar{\mathbf{k}}$ is the minimizer of the energy.

Computing the desired metric with user-defined curvature $\bar{\mathbf{k}}$ is equivalent to minimizing the discrete Ricci energy. For the Euclidean (or hyperbolic) case, the discrete Ricci energy (see Eq. 11) has been proven to be strictly convex (namely, its Hessian is positive definite) in [10]. The global minimum uniquely exists, corresponding to the metric $\bar{\mathbf{u}}$, which induces $\bar{\mathbf{k}}$. The discrete Ricci flow converges to this global minimum.

Theorem 5 (Chow & Luo: Euclidean Ricci Energy): The Euclidean Ricci energy $f(\mathbf{u})$ on the space of the normalized metric $\sum u_i = 0$ is strictly convex.

Theorem 6 (Chow & Luo: Hyperbolic Ricci Energy): The hyperbolic Ricci energy is strictly convex.

Although the spherical Ricci energy is not strictly convex, the desired metric $\bar{\mathbf{u}}$ is still a critical point of the energy. In our experiments, the solution can be reached using Newton's method.

V. Algorithm

In this section, we explain the algorithms in detail. It requires some knowledge from hyperbolic geometry and algebraic topology. We briefly introduce the most related concepts in Appendices VIII-C and VIII-A. We refer readers to [41] and [42] for further details.

The unified pipeline for all kinds of the discrete Ricci flow algorithms is as follows:

1) Determine the target curvature and the background geometry;
2) Compute the initial circle packing metric;
3) Optimize the Ricci energy using both gradient descent and Newton's methods;
4) Compute the layout using the result metric.

Step 1. Determine the Target Curvature and the Background Geometry

The user is free to define the target curvatures for different applications, while obeying the Gauss-Bonnet theorem in Eq. 8. In other words, the user can distribute the curvatures arbitrarily across the vertices, in a way that the sum of target curvatures is constant.

For example, for constructing manifold splines (see [7] and [12] for details), it is desirable to obtain a flat metric with a minimal number of cone singularities. One can concentrate all the curvatures at a single vertex and make everywhere else flat. In this case, the background geometry of the mesh is chosen to be Euclidean and the curvature for the selected vertex is set to $2\pi\chi(\Sigma)$. The curvature at all other vertices is set to zero.

For the application of surface classification using conformal structures (see [43] for details), no cone singularities are allowed. All of the curvatures must be evenly distributed over the whole surface. In this case, the target curvature is zero for all vertices and the background geometry is hyperbolic for high genus meshes.

Step 2. Compute the Initial Circle Packing Metric

In this step, the initial circle packing metric (Γ, Φ) is computed. This metric should approximate the original Euclidean metric as much as possible. In practice, we use the following simple heuristic procedure, which gives satisfactory results.

1) For each face f_{ijk}, compute a radius for the vertex v_i:
$$\gamma_i^{jk} = \frac{l_{ki} + l_{ij} - l_{jk}}{2},$$
where l_{ij}, l_{jk}, l_{ki} are the lengths of the edges e_{ij}, e_{jk}, e_{ki}, respectively.
2) For each vertex v_i, we approximate the radius γ_i by averaging the radii from the faces adjacent to v_i:
$$\gamma_i = \frac{1}{m} \sum_{f_{ijk} \in F} \gamma_i^{jk},$$
where m is the number of the adjacent faces to vertex v_i.
3) For each edge e_{ij}, we compute its edge weight ϕ_{ij} from γ_i, γ_j using a cosine law, which depends on the background geometry:
$$\begin{array}{rcll} l_{ij}^2 & = & \gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\phi_{ij}, & \mathbb{E}^2 \\ \cosh l_{ij} & = & \cosh\gamma_i \cosh\gamma_j + \sinh\gamma_i \sinh\gamma_j \cos\phi_{ij}, & \mathbb{H}^2 \\ \cos l_{ij} & = & \cos\gamma_i \cos\gamma_j - \sin\gamma_i \sin\gamma_j \cos\phi_{ij}, & \mathbb{S}^2 \end{array} \quad (12)$$
If the edge weight is greater than $\frac{\pi}{2}$, we take $\frac{\pi}{2}$ as its value.

If the initial mesh has too many obtuse angles and the requirement for the conformality is very high, we can use an extra re-meshing step to improve the triangulation quality.

Step 3. Optimize Ricci Energy

In the following we introduce two methods to optimize the Ricci energy; one is the gradient descent method and another is Newton's method.

3.1 Gradient Descent

The Ricci energy can be optimized using the gradient descent method, which is the direct analogy of the smooth Ricci flow. Note that during the computation the vertex radii Γ vary over time while the edge weights Φ are fixed. This reflects the fact that conformal metric deformation preserves angles.

1) Compute edge lengthes l_{ij} from the current vertices radii γ_i and γ_j and the fixed edge weight ϕ_{ij}, using the cosine law (Eq. 12) for the background geometry.
2) Compute the corner angles θ_i^{jk} in each face f_{ijk} from the current edge lengths by using the cosine law according to the background geometry.
3) Compute the discrete Gaussian curvature K_i of each vertex v_i by using Eq. 7.
4) Update u_i of each vertex v_i by using Eq. 11, as follows.
$$u_i = u_i + \varepsilon(\bar{K}_i - K_i),$$
where $\bar{K}_i$ is the target Gaussian curvature. In our experiments, ε is no greater than 0.05.
5) Normalize the metrics. Let $s = \sum u_i$, then
$$u_i = u_i - \frac{s}{n},$$
where n is the total number of vertices.
6) Update the radius γ_i of each vertex v_i, using u_i and Eq. 9.
7) Repeats the steps from 1 through 5, until the maximal curvature error falls below a threshold delta.
$$\max |\bar{K}_i - K_i| < \delta,$$
where δ is a user-specified error tolerance. In our experiments, we set it to $1e-6$, which is good enough for the later embedding procedure. For meshes with $30k$ faces, no folding or overlapping has been found.

3.2 Newton's Method

As described in Section IV, Ricci flow is the negative gradient flow of the discrete Ricci energy in Eq. 11. We can further improve the convergence speed by using Newton's method.

The key to Newton's method is to compute the Hessian matrix. Different Ricci flows have different Hessian matrices according to their background geometries. The Hessian matrix for Euclidean Ricci energy is explained here. We refer readers to Appendix VIII-D for the other cases.

The elements in the Hessian matrix are $\partial K_i / \partial u_j$. From the circle packing metric, the edge lengths and the corner angles can be derived from Eq. 12; then the discrete Gauss curvatures can be obtained from Eq. 7. Combining them, we obtain the relationship between the Gaussian curvature K_i and γ_i, γ_j, γ_k as follows.
$$K_i = 2\pi - \sum_{f_{ijk} \in F} \cos^{-1}\left(\gamma_i^2 + \gamma_i\gamma_j\cos\phi_{ij} + \gamma_k\gamma_i\cos\phi_{ki} - \gamma_j\gamma_k\cos\phi_{jk}\right)$$
From the above equation, we can deduce $\partial K_i / \partial u_i$ and $\partial K_i / \partial u_j$, as follows.
$$\frac{\partial K_i}{\partial u_i} = \frac{\partial \gamma_i}{\partial u_i}\frac{\partial K_i}{\partial \gamma_i} = \gamma_i \sum_{f_{ijk} \in F} \frac{AD - BC}{A\sqrt{A^2 - B^2}},$$
$$\frac{\partial K_i}{\partial u_j} = \frac{\partial \gamma_j}{\partial u_j}\frac{\partial K_i}{\partial \gamma_j} = \gamma_j \sum_{f_{ijk} \in F} \frac{AF - BE}{A\sqrt{A^2 - B^2}},$$
where
$$\begin{array}{rcl} A & = & 2l_{ij}l_{ki} \\ B & = & l_{ij}^2 + l_{ki}^2 - l_{jk}^2 \\ C & = & 2(\gamma_i + \gamma_j\cos\phi_{ij})\frac{l_{ki}}{l_{ij}} + 2(\gamma_i + \gamma_k\cos\phi_{ki})\frac{l_{ij}}{l_{ki}} \\ D & = & 2\left(2\gamma_i + \gamma_j\cos\phi_{ij} + \gamma_k\cos\phi_{ki}\right) \\ E & = & 2(\gamma_j + \gamma_i\cos\phi_{ij})\frac{l_{ki}}{l_{ij}} \\ F & = & 2\left(\gamma_i\cos\phi_{ij} - \gamma_k\cos\phi_{jk}\right) \end{array}$$

Step 4. Compute the Layouts

In this step, we flatten the mesh with the target metric onto one of the canonical domains: the plane $\mathbb{E}^2$, the sphere $\mathbb{S}^2$, or the hyperbolic space $\mathbb{H}^2$. The algorithms in this step involve several topological concepts, such as fundamental domain, canonical fundamental group basis, universal covering space, etc. We refer readers to Appendix VIII-A for details.

The following is the unified pipeline for computing the layout:

1) Flatten a seed face.
2) Flatten a fundamental domain.
3) Flatten the universal covering space.

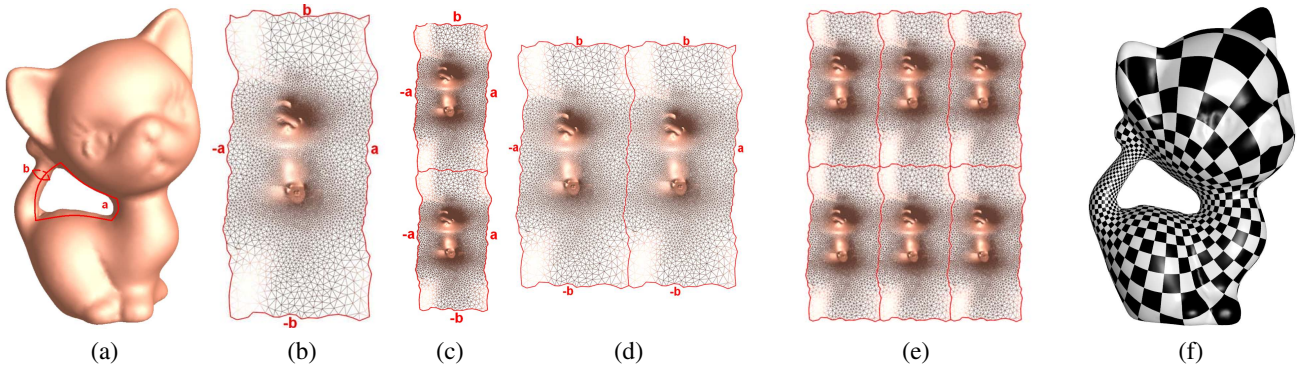


Fig. 4. **Euclidean Ricci flow** (a) Genus one kitten model marked with a set of canonical fundamental group generators a and b. (b) A fundamental domain is conformally flattened onto the plane, marked with four sides $aba^{-1}b^{-1}$. (c) One translation moves the side b to b^{-1}. (d) The other translation moves the side a to a^{-1}. (e) The layout of the universal covering space of the kitten mesh on the plane, which tiles the plane. (f) The conformal parameterization is used for the texture mapping purpose. A checkerboard texture is placed over the parameterization in b). The conformality can be verified from the fact that all the corner angles of the checkers are preserved.

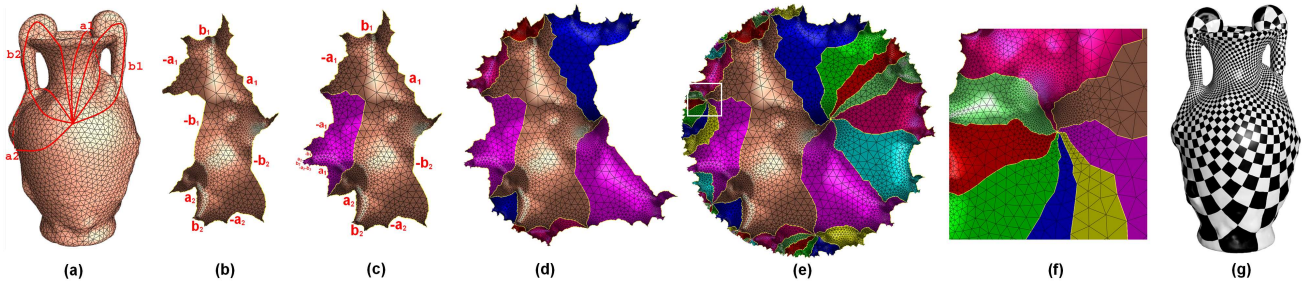


Fig. 5. **Hyperbolic Ricci flow** (a) Genus two vase model marked with a set of canonical fundamental group generators which cut the surface into a topological disk with eight sides: a_1, b_1, a_1^{-1}, b_1^{-1}, a_2, b_2, a_2^{-1}, b_2^{-1}. (b) The fundamental domain is conformally flattened onto the Poincaré disk with marked sides. (c) A Möbius transformation moves the side b_1 to b_1^{-1}. (d) Eight copies of the fundamental domain are glued coherently by eight Möbius transformations. (e) A finite portion of the universal covering space is flattened onto the Poincaré disk. (f) Zoom in on a region on the universal covering space, where eight fundamental domains join together. No seams or overlapping can be found. (g) Conformal parameterization induced by the hyperbolic flattening. The corners angle of checkers are well-preserved.

4.1 Flatten a Seed Face

We randomly select a seed face f_{012}, and compute the parametric positions of the vertices v_0, v_1, and v_2 using the edge lengths of f_{012}. For meshes with the Euclidean geometry, the positions of the three vertices are set as:

$$\tau(v_0) = (0,0), \tau(v_1) = (l_{01}, 0), \tau(v_2) = l_{02}(\cos\theta_0^{12}, \sin\theta_0^{12});$$

in the hyperbolic case, the positions are set as $\tau(v_0) = (0,0)$:

$$\tau(v_1) = \frac{e^{l_{01}}-1}{e^{l_{01}}+1}(1,0), \tau(v_2) = \frac{e^{l_{02}}-1}{e^{l_{02}}+1}(\cos\theta_0^{12}, \sin\theta_0^{12});$$

in the spherical case, the positions are set as $\tau(v_0) = (0,0,1)$:

$$\tau(v_1) = (0, 0.5\times l_{01}\times\sqrt{4-(l_{01})^2}, 1-0.5\times(l_{01})^2),$$

$$\begin{aligned}\tau(v_2)[2] &= 1-0.5\times(l_{02})^2\\ \tau(v_2)[1] &= \frac{1-0.5\times(l_{12})^2-\tau(v_2)[2]\times\tau(v_1)[2]}{\tau(v_1)[1]}\\ \tau(v_2)[0] &= -\sqrt{1-(\tau(v_2)[1])^2-(\tau(v_2)[2])^2}.\end{aligned}$$

Then we put faces adjacent to the seed face into a queue.

4.2 Flatten a Fundamental Domain

In this step, we propagate the flattening to the rest of all faces, namely we want to embed a fundamental domain. We call the resulting layout a *fundamental polygon*.

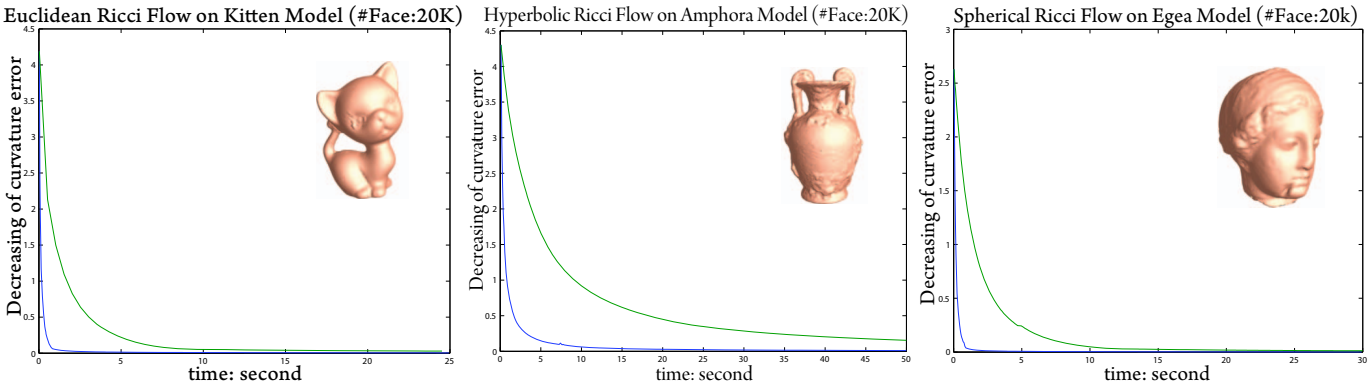
To propagate the flattening, we put all unprocessed faces adjacent to the current face into the queue. We pop a face f_{ijk} from the queue and test whether all its vertices have been set to parametric positions. If so, we continue to pop the next one from the queue as long as the queue is nonempty. Otherwise, suppose that v_i and v_j have been embedded, then $\tau(v_k)$ can be computed as one of the two intersection points between the two circles, $c(\tau(v_i), l_{ki})$ and $c(\tau(v_j), l_{kj})$, satisfying $(\tau(v_j)-\tau(v_i))\times(\tau(v_k)-\tau(v_i)) > 0$. Note that, depending on the background geometry, the circles may be spherical, Euclidean or hyperbolic circles.

By using a stereo-graphic projection, spherical circles can be converted to Euclidean circles; hyperbolic circles in the Poincaré disk coincide with Euclidean circles. Therefore, computing the intersection points between spherical (or hyperbolic) circles boils down to finding intersections between Euclidean circles.

Different choices of the seed faces induce different layouts, which differ by a rigid motion. In the Euclidean case, a rigid motion is a planar translation and rotation; in the hyperbolic case, it is a Möbius transformation; in the spherical case, it is a rotation. Fig. 8 (b), (c) and (d) are the layouts for the same genus two model, shown in Fig. 8(a), with different seed faces marked in red. The layouts in (c) and (d) are transformed to align with the layout in (b) by different Möbius transformations, as shown in Fig. 8(e). The difference of positions of the same vertex among the three aligned layouts is less than $1e-6$.

4.3 Flatten the Universal Covering Space

For the purpose of texture mapping, it is enough to flatten a



(a) Euclidean (b) Hyperbolic (c) Spherical

Fig. 6. **Performance of Ricci flow** The horizontal axis represents time, and the vertical axis represents the maximal curvature error. The blue curves are for the Newton's method; the green curves are for the gradient descent method. The meshes have about $30k$ faces. The tests were carried out on a laptop with 1.7GHz CPU and 1G RAM. All the algorithms are written in C++ on a Windows platform without using any other numerical library.

fundamental domain. For the purpose of constructing a manifold spline (see [7] and [12] for details) or surface classification by conformal equivalence (see [43] for details), we need to flatten a finite portion of the universal covering space.

The universal covering space of a closed mesh with a positive Euler characteristic number is the mesh itself, which can be embedded onto the unit sphere $\mathbb{S}^2$. The universal covering space of a mesh with a zero or negative Euler number can be embedded onto the whole Euclidean plane $\mathbb{E}^2$ or the whole hyperbolic space $\mathbb{H}^2$. The algorithmic pipeline is as follows:

1) Embed a canonical fundamental domain.
2) Compute the deck transformation group generators.
3) Tile the whole canonical domain $\mathbb{R}^2$ or $\mathbb{H}^2$.

The first step is exactly the same as the procedures introduced in 4.2. Figs. 4(b) and 5(b) give the embeddings of the canonical fundamental domains for the genus one kitten model on the plane and the genus two amphora model on the Poincaré disk.

Compute Deck Transformation Group Generators

The embedding of a canonical fundamental domain for a closed genus g surface has $4g$ different sides, which induce $4g$ rigid transformations. These $4g$ rigid motions are the generators of the deck transformation group.

Fig. 4 explains the process for a mesh with a zero Euler number. The surface has two fundamental group generators a and b. The complement of $\{a,b\}$ is a fundamental domain, denoted as D. The boundary of D is $\partial D = aba^{-1}b^{-1}$. Let the flattening map be $\tau : D \to \mathbb{R}^2$. Then the planar image $\tau(D)$ has 4 sides, $\tau(a), \tau(b), \tau(a^{-1}), \tau(b^{-1})$ (see Fig. 4(b)). There exists unique translations $\alpha, \beta : \mathbb{R}^2 \to \mathbb{R}^2$, such that α maps $\tau(a)$ to $\tau(a^{-1})$ and β maps $\tau(b)$ to $\tau(b^{-1})$, as shown in Fig. 4(c) and (d). α, β are the generators of the deck transformation group.

Similarly, Fig. 5 illustrates the process for a mesh with a negative Euler number. Let $\{a_1, b_1, \cdots, a_g, b_g\}$ be a set of canonical fundamental group generators, where g is the genus. The embedding of its canonical fundamental domain in hyperbolic space has $4g$ sides, $\tau(a_1), \tau(b_1), \tau(a_1^{-1}), \tau(b_1^{-1}), ..., \tau(a_g), \tau(b_g), \tau(a_g^{-1}), \tau(b_g^{-1})$ (see Fig. 5(b) in Poincaré disk). There exists unique Möbius transformations α_k, β_k, which map the $\tau(a_k)$ and $\tau(b_k)$ to $\tau(a_k^{-1})$ and $\rho(b_k^{-1})$ respectively, as shown in Fig. 5(c) and (d). The Möbius transformations $\{\alpha_1, \beta_1, \alpha_2, \beta_2, \cdots, \alpha_g, \beta_g\}$ form a set of generators of the deck transformation group.

The following explains the details for computing β_1. Let the starting and ending vertices of the two sides be: $\partial\tau(b_1) = q_0 - p_0$ and $\partial\tau(b_1^{-1}) = p_1 - q_1$, the geodesic distance from p_0 to q_0 equals the geodesic distance from p_1 to q_1 in the Poincaré disk. To align them, we first construct a Möbius transformation τ_0, which maps p_0 to the origin, q_0 to a positive real number, with

$$\tau_0 = e^{-i\theta_0}\frac{z - p_0}{1 - \bar{p}_0 z}, \theta_0 = arg\frac{q_0 - p_0}{1 - \bar{p}_0 q_0}.$$

Similarly, we can construct another Möbius transformation τ_1, which maps p_1 to the origin, and q_1 to a real number, with $\tau_1(q_1)$ equals to $\tau_0(q_0)$. By composing the two, we get the final Möbius transformation $\beta_1 = \tau_1^{-1} \circ \tau_0$, which satisfies $p_1 = \beta_1(p_0)$ and $q_1 = \beta_1(q_0)$, and aligns the two sides together.

Tile the Canonical Domain

Any deck transformation can be produced by composing the generators $\{\alpha_k, \beta_k\}$. Then the whole canonical domain can be tiled by transforming a fundamental polygon by all deck transformations. This induces a flattening of the universal covering space of the mesh onto the canonical domain. Fig. 4(e) shows the flattening of the universal covering space of the kitten mesh onto the whole Euclidean plane. Fig. 5(e) illustrates the layout of the universal covering space of a genus two amphora model onto the whole Poincaré disk.

VI. Experimental Results

Discrete Ricci flow is a powerful tool for computing the desired metrics, which satisfy the user-defined Gaussian curvatures on general surfaces. In the following, we report our experimental results and analyze the performance of the algorithms.

A. Performance

The performance of Ricci energy optimization is analyzed from the following aspects.

Convergence For both the Euclidean and hyperbolic cases, the discrete Ricci energies are convex. Therefore, there exists a unique global minimum. Both gradient descent and Newton's method converge to it stably. For the spherical case, the Ricci energy cannot be theoretically proven to be strictly convex. The desired metric is a critical point of the energy. In our experiments, the

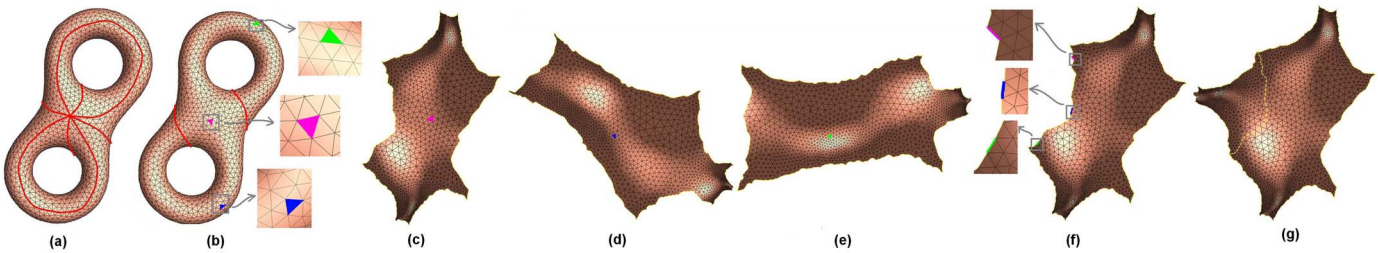


Fig. 8. **Accuracy Testing for Hyperbolic Layout** Frame (a)-(b): a set of fundamental group generators are marked with red on a genus two eight model with $4k$ faces, and three randomly chosen seed faces are marked with red, green, and blue respectively. Frame (c)-(e) show the flattened results on the Poincaré disk using the hyperbolic uniformization metric, with different seed faces. Any two of them only differ a Möbius transformation. Frame (f): randomly chosen Edges marked with different colors can be chosen to compute the same deck transformation shown in (g).

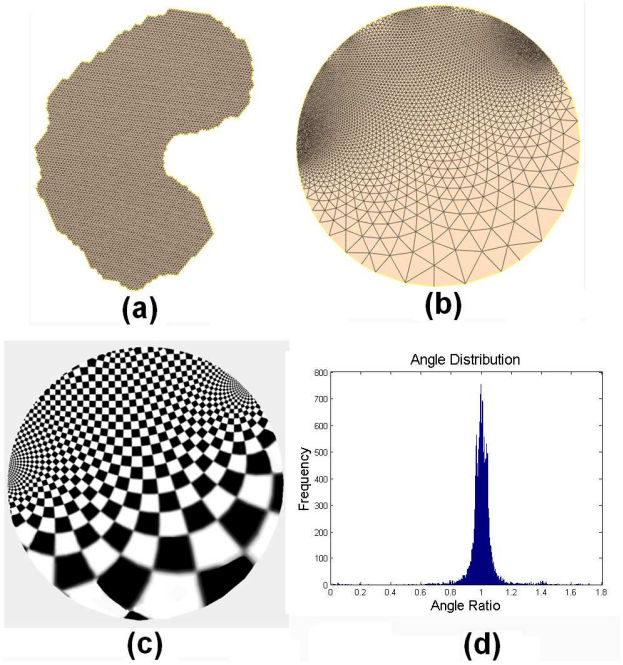


Fig. 7. **Conformality Testing** Discrete Euclidean Ricci flow conformally maps an irregular shape in (a) to a disk in (b). The parameterization is illustrated as the check board texture mapping in (c), where all the corners of the checkers are well-preserved visually. The histogram of the angle ratio, which is defined as the ratio of the original angle value and the new one after mapping, is calculated and shown in (d). The distribution highly concentrates at 1.

Mesh Model	# Face	genus	Bnds	Sing	Type	Time (Sec)
Kitten	20438	1	0	0	$\mathbb{E}^2$	3.53
Horse	12186	0	4	0	$\mathbb{E}^2$	2.365
Horse	12186	0	4	1	$\mathbb{E}^2$	2.528
Eight	4117	2	0	0	$\mathbb{H}^2$	3.89
Amphora	20010	2	0	0	$\mathbb{H}^2$	9.86
Whole David	19638	3	0	0	$\mathbb{H}^2$	13.45
Egea	20226	0	0	0	$\mathbb{S}^2$	3.76

TABLE I

TIME FOR COMPUTING DISCRETE RICCI FLOW.

desired spherical metrics still can be reached efficiently using Newton's method.

Speed Fig. 6 compares the performance of the optimization of Ricci energy using Newton's method (the blue curves) and the gradient descent method (the green curves). It is obvious that Newton's method is much faster than the gradient descent method for optimizing Ricci energy. We implemented our system using C++ on a Windows platform desktop with 3.4GHz CPU Intel Xeon, 4.0G RAM using Newton's method. We report the timings to compute the desired metric in Table I. The speed for the hyperbolic case is the slowest.

Conformality Fig. 7 shows an irregular planar domain in (a) is mapped to a disk in (b). The conformality is shown with checkerboard texture mapping in (c). We compute the ratio between each corner angle in (a) and in (b). The histogram of the ratios is illustrated in (d), and we can see the ratio is highly concentrated around 1.

Accuracy From our experiments, we found it is relatively easier to compute the metric, but it is more challenging to compute the layout due to the accumulation error, especially for the hyperbolic case. In order to improve the accuracy for hyperbolic flattening, we adapt the following methods.

a) Control the target metric by the maximum curvature error. We found the quality of the final embedding is strongly affected by the quality of the final metric. On the other hand, the running time depends on the maximum curvature error also. Through thorough testing, we find a good balance to set the maximum curvature error to be $1e-6$ for a mesh with $30k$ faces, then the embedding result is satisfactory and the computation speed is reasonable.

b) Move all of the computation to the center of the Poincaré disk. Because the area distortion in the Poincaré disk is non-uniform, computations near the boundaries are highly unstable, where the area distortion goes to infinity. We use a Möbius transformation to move each current processing face to the center of the Poincaré disk, where the area distortion is close to 1 and the computation is much more stable.

c) In flattening each face, we avoid using trigonometric functions and use more algebraic functions. For example, in order to set the third vertex position of a triangle, we compute the intersection points of two circles.

d) Instead of flattening face by face for embedding the universal covering space, we compute the deck transformations and flatten fundamental domain by fundamental domain. This greatly reduces the accumulated error.

e) Divide and conquer. If the mesh is big, we partition it into patches and flatten each patch, then use Möbius transformations to glue different patches together. Each edge on the cut determines a Möbius transformation; we take the average of the transformations induced by all the edges on the same cut.

Fig. 5 demonstrates the hyperbolic flattening process. Frame (f)

zooms in on a region where 8 fundamental domains join together, which illustrates the accuracy of our embedding method.

We conduct a special experiment to test the hyperbolic accuracy as shown in Fig. 8. We flatten a genus two mesh in (a) with hyperbolic uniformization metric onto the Poincaré disk. We start from different seed faces marked with different colors and get different layouts of its canonical fundamental domain (its canonical fundamental generators are marked with red in both (a) and (b)), as shown in (c), (d), and (e). By using Möbius transformations given in table II, the layouts in (d) and (e) are transformed to align perfectly with the layout in (c). Then we compute a deck transformation to glue two layouts as shown in (g). The deck transformation is computed three times using different edges on the same side as shown in (f), where three edges are colored as red, green, and blue. Using formulae Eq. 14, each Möbius transformation is represented by an angle θ and a center z_0, listed in table II. The differences among the three angles are less than $1e-6$; the differences among three centers are also less than $1e-6$. This shows the accuracy of our hyperbolic embedding method.

B. Global Conformal Parameterization

A parameterization of a surface can be viewed as a one-to-one mapping from a suitable domain to the surface, typically a planar domain, which has many applications such as texture mapping, surface approximation, scattered data fitting, remeshing etc. Unfortunately, it is impossible to map the whole of a closed surface with non-toroid topology to the $2D$ plane without any angle distortion. Therefore, in order to parameterize a mesh of this type, researchers have resorted to partitioning the mesh into several charts [5] or introducing singularities on the mesh [19], [22], [28]. In these cases, however, heuristic inputs or special algorithms to locate seams or singularities are needed.

Discrete Ricci flow is an effective approach for different types of automatic and seam-free global conformal parameterizations of surfaces with arbitrary topologies. Ricci flow gives us the freedom to allocate the total in different regions of the mesh: on the interior vertices, on the boundary vertices, and on the other points.

Fig. 9 demonstrates the flexibility of this method, where for different purposes, the total curvature is distributed to different regions and using different background geometries. The original surfaces are shown in the first and the last frames, which is a genus zero surface with 3 boundaries.
1) We can allocate all the curvature on the boundaries and make everywhere else be flat, as shown in frame (b). Furthermore, we can make the boundaries be circles. The centers and the radii of the boundary circles are determined by the geometries of the original surface. They can be used as the fingerprint of the surfaces, which are valuable for shape recognition and classification applications in the vision field (see [44] for more details).
2) We can make all the interior vertex curvatures be zeros, and the curvatures on the boundary vertex also be zeros (i.e. the boundaries become geodesics), and set the curvature to be -1 for every point on the mesh, as shown in frame (c). The background geometry is hyperbolic. The geodesic lengths of three boundaries are the fingerprints of the shape, which can be used for shape classification and comparison. (see [43] for more details).
3). We can set one boundary curvature to be zero, and the other two boundaries to be circles. All vertex curvatures are zeros except 4 cone singularities, whose curvatures equal to $\frac{\pi}{2}$, as shown in frame (d). This can be used to construct polycube splines in geometric modeling field, (see [45] for more details). The way we construct the cube-shaped model is as follows: We use discrete Euclidean Ricci flow to conformally map the head model to the unit disk with circular holes as in (b). We then map the cube (without the bottom face) onto the unit disk. By matching the two unit disks, we locate the positions of cone singularities in the head model. We set the target curvatures of cone singularities to be $\frac{\pi}{2}$.

We set the target curvatures on the outer boundary to zero, and the target curvatures of other boundary vertices to be constant. A flat metric satisfying these curvature conditions can be found using discrete Euclidean Ricci flow. Next we find the geodesics from the cone singularities to the outer boundaries under the target metric, which are orthogonal to the outer boundary. This gives us the isometric embedding of the surface with the target curvature in $\mathbb{R}^3$.

Different parameterizations have different numbers of singularities and different area distortions. The followings are some comparisons among different parameterization methods, which demonstrates the versatility of the Ricci flow method.

Boundaries vs. Vertices The bottom of the hoofs of the horse mesh in Fig. 10 are removed as shown in (c), therefore the mesh is a genus zero surface with 4 boundaries, and the total curvature is -4π. We set the total curvature of each boundary to be $-\pi$, and all interior vertex curvatures to be zero. The layout of a fundamental domain with the flat metric is shown in (b), where the body region has a bigger area in the parameter domain, and the leg regions have smaller areas in the parameter domain. Then we concentrate all the curvature at a single vertex, colored as red in (c), the layout is shown in (d). Compared to (b), the leg regions are greatly enlarged, and the body region shrinks greatly.

Boundary vs. Boundary Fig. 11 gives the comparison of different layouts of the same David head model, which is depicted in Fig. 9(a) and (e). Since the mesh is with three boundaries, the total curvature is -2π. Each row shows a periodic layout induced by a flat metric, which allocates the total curvature on a single boundary.

Euclidean vs. Hyperbolic Fig. 12 gives a comparison for the same model using different methods. The right one was conformally parameterized to a Euclidean plane with 4 singularities, where the area distortion is highly non-uniform. The left one was conformally parameterized to the Poincaré disk using discrete

Möbius Trans.	θ	Z_0
M_0	2.917271	$(-0.200690, -0.568453)$
M_1	0.567858	$(0.197063, -0.776343)$
M_2	-2.982119	$(0.187574, 0.951998)$
M_3	-2.982118	$(0.187576, 0.951999)$
M_4	-2.982113	$(0.187578, 0.951990)$

TABLE II

ACCURACY TESTING FOR HYPERBOLIC LAYOUT IN FIG. 8. MÖBIUS TRANSFORMATION M_0 MOVES DOMAIN IN (D) TO (C); M_1 MOVES (E) TO (C); M_2, M_3, AND M_4 ARE DECK TRANSFORMATIONS COMPUTED FROM DIFFERENT EDGES ON THE SAME BOUNDARY SEGMENT.

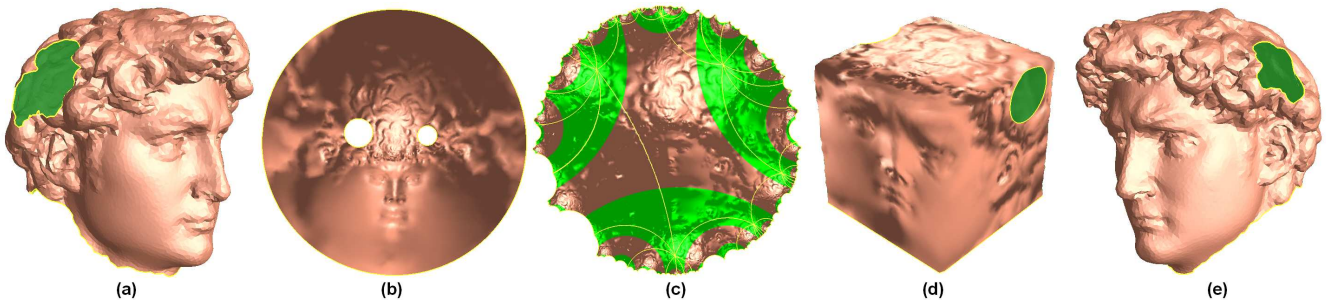


Fig. 9. **Different Global Conformal Parameterizations based on Ricci flow** (a) and (e) Original surface (b) All the curvatures are allocated on the boundaries using Euclidean Ricci flow. (c) Uniformization metric by discrete hyperbolic Ricci flow, boundaries are geodesics. (d) Flat metric with singularities by discrete Euclidean Ricci flow.

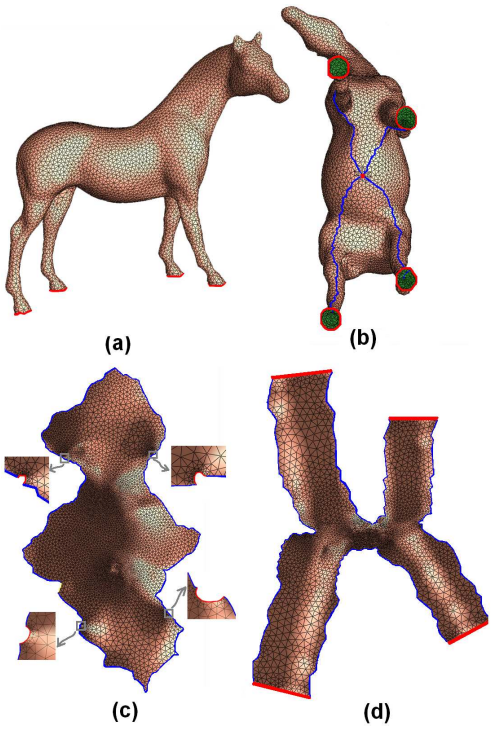


Fig. 10. **Curvature Allocations: Boundary vs. Vertex** (a) (b) A horse model: four boundaries on its hoofs marked with red; one singularity vertex marked with red; the cut graph marked with blue. (c) By pushing all curvatures to boundaries, the horse model is embedded on the plane with the cut graph. (d) By concentrating all curvatures on one singularity vertex, the horse model is embedded on the plane with the same cut graph.

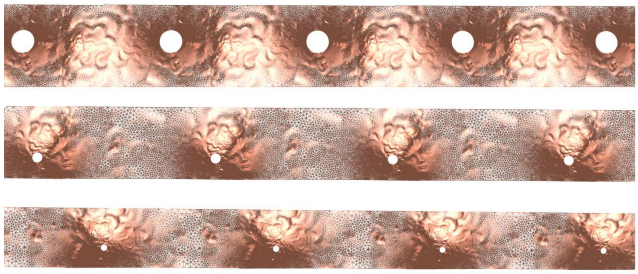


Fig. 11. **Curvature Allocation: Boundary vs. Boundary** The original surface is illustrated in Fig. 9(a) and (e). All the curvatures are allocated on one of the three boundaries. The universal covering space is flattened periodically with the corresponding metrics onto the plane.

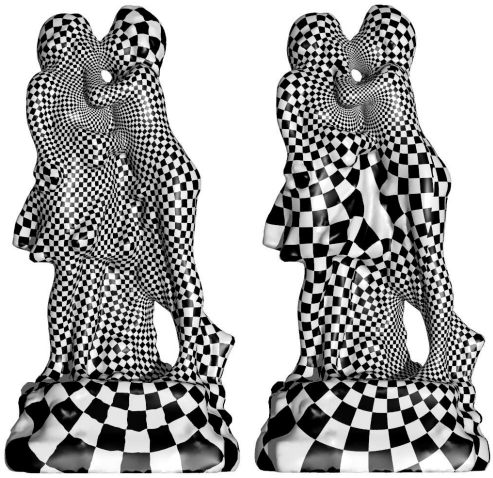


Fig. 12. **Euclidean vs. Hyperbolic** The right one was conformally parameterized to a Euclidean plane with 4 singularities, while the left one was conformally parameterized to the Poincaré disk and is singularity-free.

hyperbolic Ricci flow but without singularity.

VII. Conclusion

This work proposes a unified framework for discrete Ricci flows, generalized from continuous Ricci flow from modern geometry for computing Riemannian metrics with user-defined Gaussian curvatures. The algorithms to compute Riemannian metrics with user-defined Gaussian curvatures for general discrete meshes are presented in detail. Experimental results for spherical, Euclidean and hyperbolic Ricci flows are demonstrated to show the efficiency and accuracy of the algorithms.

In future research work, we will study the intrinsic relations between discrete Ricci flow and discrete holomorphic 1-forms. Algorithms for designing metrics with different conditions will be investigated. We will explore more applications of discrete Ricci flow in graphics, visualization, geometric modeling and medical imaging fields.

VIII. Acknowledgement

The authors would like to thank Prof. Shing-Tung Yau, Prof. Tom Sederberg, and Prof. Hong Qin for the discussions. This work is supported by NSF 0448399 career: Conformal Geometry Applied to Shape Analysis and Geometric Modeling, NSF 0528363: conformal geometry in graphics and visualization, and NSF 0626223: discrete curvature flow. This project is partially supported by NSF funding for Gu, CCF-0448399, DMS-0528363,

IIS-0713145, and for both of Luo and Gu, DMS-0626223. Feng Luo is partially supported by the NSF (DMS 0625935). Junho Kim is partially supported by the IITA&MIC scholarship program.

REFERENCES

[1] G. Perelman, "The entropy formula for the Ricci flow and its geometric applications," Tech. Rep. arXiv.org, November 11 2002.
[2] G. Perelman, "Ricci flow with surgery on three-manifolds," Tech. Rep. arXiv.org, March 10 2003.
[3] G. Perelman, "Finite extinction time for the solutions to the Ricci flow on certain three-manifolds," Tech. Rep. arXiv.org, July 17 2003.
[4] R. S. Hamilton, "Three manifolds with positive Ricci curvature," *Journal of Differential Geometry.*, vol. 17, pp. 255–306, 1982.
[5] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *SIGGRAPH 2002*, pp. 362–371, 2002.
[6] P. Alliez, M. Meyer, and M. Desbrun, "Interactive geometry remeshing," *SIGGRAPH 2002*, pp. 347–354, 2002.
[7] X. Gu, Y. He, and H. Qin, "Manifold splines," *Graphical Models*, vol. 68, no. 3, pp. 237–254, 2006.
[8] R. S. Hamilton, "The Ricci flow on surfaces," *Mathematics and general relativity (Santa Cruz, CA, 1986), Contemp. Math. Amer.Math.Soc. Providence, RI*, vol. 71, 1988.
[9] W. P. Thurston, *Geometry and Topology of Three-Manifolds*. Princeton lecture notes, 1976.
[10] B. Chow and F. Luo, "Combinatorial Ricci flows on surfaces," *Journal Differential Geometry*, vol. 63, no. 1, pp. 97–129, 2003.
[11] M. Jin, F. Luo, and X. Gu, "Computing surface hyperbolic structure and real projective structure," in *ACM Symposium on Solid and Physics Modeling*, pp. 105–116, 2006.
[12] X. Gu, Y. He, M. Jin, F. Luo, H. Qin, and S.-T. Yau, "Manifold splines with single extraordinary point," in *ACM Symposium on Solid and Physics Modeling*, pp. 61–72, 2007.
[13] M. S. Floater and K. Hormann, "Surface parameterization: a tutorial and survey," in *Advances in Multiresolution for Geometric Modelling*, pp. 157–186, Springer, 2005.
[14] A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 2, 2006.
[15] M. S. Floater, "Parametrization and smooth approximation of surface triangulations," *Computer Aided Geometric Design*, vol. 14, no. 3, pp. 231–250, 1997.
[16] A. Sheffer and E. de Sturler, "Parameterization of faced surfaces for meshing using angle based flattening," *Engineering with Computers*, vol. 17, no. 3, pp. 326–337, 2001.
[17] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic parameterizations of surface meshes," *Computer Graphics Forum (Proc. Eurographics 2002)*, vol. 21, no. 3, pp. 209–218, 2002.
[18] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov, "ABF++: Fast and robust angle based flattening," *ACM Transactions on Graphics*, vol. 24, no. 2, pp. 311–330, 2005.
[19] X. Gu and S.-T. Yau, "Global conformal parameterization," in *Symposium on Geometry Processing*, pp. 127–137, 2003.
[20] S. J. Gortler, C. Gotsman, and D. Thurston, "Discrete one-forms on meshes and applications to 3D mesh parameterization," *Computer Aided Geometric Design*, vol. 23, no. 2, pp. 83–112, 2005.
[21] N. Ray, W. C. Li, B. Levy, A. Sheffer, and P. Alliez, "Periodic global parameterization," *ACM Transactions on Graphics*, vol. 25, no. 4, pp. 1460–1485, 2005.
[22] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun, "Designing quadrangulations with discrete harmonic forms," in *Symposium on Geometry Processing*, pp. 201–210, 2006.
[23] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," *ACM Trans. Graph.*, vol. 26, no. 3, p. 56, 2007.
[24] K. Stephenson, *Introduction To Circle Packing*. Cambridge University Press, 2005.
[25] A. I. Bobenko and B. A. Springborn, "Variational principles for circle patterns and koebe's theorem," *Transactions of the American Mathematical Society*, vol. 356, pp. 659–689, 2004.
[26] A. Bobenko and P. Schröoder, "Discrete willmore flow," in *Symposium on Geometry Processing*, pp. 101–110, 2005.
[27] Y. C. de Verdière, "Un principe variationnel pour les empilements de cercles. (french) [a variational principle for circle packings]," *Invent. Math.*, vol. 104, no. 3, pp. 655–669, 1991.
[28] L. Kharevych, B. Springborn, and P. Schröder, "Discrete conformal mappings via circle patterns," *ACM Transactions on Graphics*, vol. 25, no. 2, pp. 412–438, 2006.
[29] F. Luo, X. Gu, and J.Dai, *Variational Principles for Discrete Surfaces*. High Education Press and International Press, 2007.
[30] C. Gotsman, X. Gu, and A. Sheffer, "Fundamentals of spherical parameterization for 3d meshes," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 358–363, 2003.
[31] E. Praun and H. Hoppe, "Spherical parametrization and remeshing," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 340–349, 2003.
[32] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau, "Genus zero surface conformal mapping and its application to brain surface mapping," *IEEE Transactions on Medical Imaging*, vol. 23, no. 8, pp. 949–958, 2004.
[33] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle, "Conformal surface parameterization for texture mapping," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 2, pp. 181–189, 2000.
[34] M. Jin, Y. Wang, S.-T. Yau, and X. Gu, "Optimal global conformal surface parameterization," in *IEEE Visualization 2004*, pp. 267–274, 2004.
[35] H. Ferguson, A. P. Rockwood, and J. Cox, "Topological design of sculptured surfaces," *SIGGRAPH '92*, pp. 149–156, 1992.
[36] H. Ferguson and A. P. Rockwood, "Multiperiodic functions for surface design.," *Computer Aided Geometric Design*, vol. 10, no. 3-4, pp. 315–328, 1993.
[37] C. Grimm and J. F. Hughes, "Parameterizing N-holed tori," in *IMA Conference on the Mathematics of Surfaces*, pp. 14–29, 2003.
[38] J. Wallner and H. Pottmann, "Spline orbifolds," *Curves and Surfaces with Applications in CAGD*, pp. 445–464, 1997.
[39] B. Chow, "The Ricci flow on the 2-sphere," *J. Differential Geom.*, vol. 33, no. 2, pp. 325–334, 1991.
[40] S. H. Weitraub, *Differential Forms: A Complement to Vector Calculus*. Academic Press, 2007.
[41] W. P. Thurston, *Three-Dimensional Geometry and Topology*. Princeton University Press, 1997.
[42] A. Hatcher, *Algebraic Topology*. Camerbirdge University Press, 2006.
[43] M. Jin, F. L. 0002, S.-T. Yau, and X. Gu, "Computing geodesic spectra of surfaces," in *Symposium on Solid and Physical Modeling*, pp. 387–393, 2007.
[44] X. Gu, S. Wang, J. Kim, Y. Zeng, Y. Wang, H. Qin, and D. Samaras, "Ricci flow for 3d shape analysis," *Eleventh IEEE International Conference on Computer Vision (ICCV 2007)*, 2007.
[45] H. Wang, Y. He, X. Li, X. Gu, and H. Qin, "Polycube splines," in *ACM Solid and Physical Modeling*, pp. 241–251, 2007.
[46] C. Carner, M. Jin, X. Gu, and H. Qin, "Topology-driven surface mappings with robust feature alignment," in *IEEE Visualization 2005*, pp. 543–550, 2005.
[47] H. W. Guggenheimer, *Differential Geometry*. Dover Publications, 1977.

APPENDIX

In this section, we briefly list the major concepts in hyperbolic geometry, differential geometry, and algebraic topology, which are necessary for the algorithms. Furthermore, we give detailed formulae for the Hessian matrices of hyperbolic and spherical Ricci energies.

A. Fundamental Group and Universal Covering Space

The closed loops on the surface can be classified by homotopic equivalence. If two closed curves on surface M can deform to each other without leaving the surface, they are *homotopic* to each other. Two closed curves sharing common points can be concatenated to form another loop. This operation defines the multiplication of homotopic classes. All the homotopy classes form the so called *first fundamental group* of M. A collection of curves on the surface is a *cut graph*, if their complement is a topological disk, which is called the *fundamental domain* of the surface.

For genus g closed surface, the fundamental group has $2g$ generators. A set of fundamental group basis $\{a_1,b_1,a_2,b_2,\cdots,a_g,b_g\}$ is *canonical*, if a_i,b_i have one geometric intersection, but a_i,a_j have zero geometric intersection, and a_i,b_j have zero geometric intersection too. Figs. 4(a) and 5(a) show the sets of canonical fundamental group generators for the kitten model with zero Euler number and the amphora model with a negative Euler number. If we slice M along the curves, we can get a disk-like domain with boundary $\{a_1b_1a_1^{-1}b_1^{-1}a_2b_2a_2^{-1}b_2^{-1}\cdots a_gb_ga_g^{-1}b_g^{-1}\}$, which is called the *canonical fundamental domain* of the surface M.

A covering space of M is a surface $\bar{M}$ together with a continuous surjective map $p:\bar{M}\to M$, such that for every $q\in M$ there exists an open neighborhood U of q such that $p^{-1}(U)$ (the inverse image of U under p) is a disjoint union of open sets in $\bar{M}$ each of which is mapped homeomorphically onto U by p. If $\bar{M}$ is simply connected, then $\bar{M}$ is called the *universal covering space* of M. Suppose $\phi:\bar{M}\to\bar{M}$, $p=\phi\circ p$, then ϕ is called a *deck transformation*. A deck transformation maps a fundamental domain to another fundamental domain. All the deck transformations form the so-called *deck transformation group*, which is isomorphic to the fundamental domain. We use the algorithms in [46] to compute the canonical fundamental group generators.

B. Riemannian Metric and Gaussian Curvature

All the concepts and the detailed explanations can be found in [47]. Suppose S is a C^2 smooth surface embedded in $\mathbb{R}^3$ with local parameter (u_1,u_2). Let $\mathbf{r}(u_1,u_2)$ be a point on S and $d\mathbf{r}=\mathbf{r}_1du_1+\mathbf{r}_2du_2$ be the tangent vector defined at that point, where $\mathbf{r}_1,\mathbf{r}_2$ are the partial derivatives of $\mathbf{r}$ with respect to u_1 and u_2, respectively. The *Riemannian metric* or the *first fundamental form* is:

$$<d\mathbf{r},d\mathbf{r}>=\sum<\mathbf{r}_i,\mathbf{r}_j>du_idu_j,\quad i,j=1,2.$$

The Gauss map $G:S\to\mathbb{S}^2$ from the surface S to the unit sphere $\mathbb{S}^2$ maps each point p on the surface to its normal $\mathbf{n}(p)$ on the sphere. The *Gaussian curvature $K(p)$ is defined as the Jacobian of the Gauss map*. Intuitively, it is the ratio between the infinitesimal area of the Gauss image on the Gaussian sphere and the infinitesimal area on the surface.

Let ∂S be the boundary of the surface S, k_g the geodesic curvature, dA the area element, ds the line element, and $\chi(S)$ the Euler characteristic number of S. The total curvature is determined by the topology:

$$\int_S KdA+\int_{\partial S}k_gds=2\pi\chi(S). \tag{13}$$

C. Hyperbolic Geometry

The hyperbolic space model we used in this paper is the Poincaré disk, which is a unit disk on the complex plane, with Riemannian metric

$$ds^2=\frac{4dwd\bar{w}}{(1-\bar{w}w)^2}.$$

In the Poincaré disk, rigid motion is a Möbius transformation,

$$z\to e^{i\theta}\frac{z-z_0}{1-\bar{z}_0z},z_0\in\mathbb{C},\theta\in[0,2\pi);$$

the geodesics are circle arcs which are orthogonal to the unit circle; the hyperbolic circle $(\mathbf{c},r)$ coincides with an Euclidean circle $(\mathbf{C},R)$ with

$$\mathbf{C}=\frac{2-2\mu^2}{1-\mu^2|\mathbf{c}|^2}\mathbf{c},\quad R^2=|\mathbf{C}|^2-\frac{|\mathbf{c}|^2-\mu^2}{1-\mu^2|\mathbf{c}|^2},$$

where $\mu=\frac{e^r-1}{e^r+1}$.

D. Hessian Matrix

Let $c(x),s(x)$ be $\cos x,\sin x$ for the spherical case and $\cosh x,\sinh x$ for the hyperbolic case. Define a function

$$\tau_{ij}=s(\gamma_i)c(\gamma_j)+c(\gamma_i)s(\gamma_j)\cos\phi_{ij},$$

The elements in the Hessian matrix of Ricci energy are:

$$\begin{aligned}\frac{\partial K_i}{\partial u_i}&=s(\gamma_i)\times\sum_{f_{ijk}}\frac{CB-DA}{A\sqrt{A^2-B^2}}\\ \frac{\partial K_i}{\partial u_j}&=s(\gamma_j)\times\sum_{f_{ijk}}\frac{EB-FA}{A\sqrt{A^2-B^2}}\end{aligned}$$

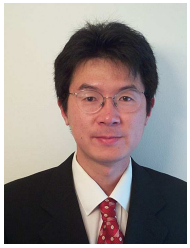
where

$$\begin{aligned}A&=s(l_{ij})s(l_{ki})\\ B&=c(l_{ij})c(l_{ki})-c(l_{jk})\\ C&=\tau_{ij}\cdot s(l_{ki})\cdot\tfrac{c(l_{ij})}{s(l_{ij})}+\tau_{jk}\cdot s(l_{ij})\cdot\tfrac{c(l_{ki})}{s(l_{ki})}\\ D&=\tau_{ij}\cdot c(l_{ki})+\tau_{ki}\cdot c(l_{ij})\\ E&=\tau_{ji}\cdot s(l_{ki})\cdot\tfrac{c(l_{ij})}{s(l_{ij})}\\ F&=\tau_{ji}\cdot c(l_{ki})-\tau_{jk}\end{aligned}$$

Miao Jin received the BS degree from the Computer Sciences Department at Beijing University of Telecommunication and Post in 2000, and the MS degree from the Computer Science Department at the State University of New York at Stony Brook, in 2006. She is a PhD candidate in the Computer Sciences Department at the State University of New York at Stony Brook. Her research interests include surface parameterization, computing surface geometric structures, discrete differential geometry, and geometric modeling. For more information see http://www.cs.sunysb.edu/∼mjin.

Junho Kim received the BS, MS, and Ph.D degrees in computer science and engineering from Pohang University of Science and Technology (POSTECH) in 1998, 2000, and 2005, respectively. In his Ph.D course, he worked at RWTH Aachen University as a BK21 visiting Ph.D student from 2003 to 2004. After getting his Ph.D degree, he worked at Stony Brook University as a research associate from 2005 to 2008. He is currently a full-time lecturer of game engineering at Dong-Eui University, Korea. His research interests include computer graphics, computer vision, geometric modeling, and interactive games.

Feng Luo received his Ph. D. from University of California at San Diego in 1989. He is a professor of Mathematics at Rutgers University. His research interests include geometry and topology on surfaces and 3-manifolds, and computer graphics. For more information, see http:///www.math.rutgers.edu/∼fluo.



Xianfeng Gu received the PhD in computer science from Harvard University in 2003. He is an assistant professor of computer science at Stony Brook University. He won the US National Science Foundation CAREER award in 2004. His research interests are computer graphics, computer vision, and medical imaging. His major works include geometry images, global conformal surface parameterization, manifold splines, and computational conformal geometry. For more information, see http://www.cs.sunysb.edu/∼gu. He is a member of the IEEE and the IEEE Computer Society.