Manifold Splines with Single Extraordinary Point

Xianfeng Gu [a], Ying He [b,*], Miao Jin [a], Feng Luo [c], Hong Qin [a], Shing-Tung Yau [d]

[a] *Computer Science Department, Stony Brook University, NY, USA.*

[b] *School of Computer Engineering, Nanyang Technological University, Singapore.*

[c] *Mathematics Department, Rutgers University, NJ, USA.*

[d] *Mathematics Department, Harvard University, MA, USA.*

Abstract

This paper develops a novel computational technique to define and construct manifold splines with only one singular point by employing the rigorous mathematical theory of Ricci flow. The central idea and new computational paradigm of manifold splines are to systematically extend the algorithmic pipeline of spline surface construction from any planar domain to arbitrary topology. As a result, manifold splines can unify planar spline representations as their special cases. Despite its earlier success, the existing manifold spline framework is plagued by the topology-dependent, large number of singular points (i.e., $|2g-2|$ for any genus-g surface), where the analysis of surface behaviors such as continuity remains extremely difficult. The unique theoretical contribution of this paper is that we devise new mathematical tools so that manifold splines can now be constructed with only one singular point, reaching their theoretic lower bound of singularity for real-world applications. Our new algorithm is founded upon the concept of discrete Ricci flow and associated techniques. First, Ricci flow is employed to compute a special metric of any manifold domain (serving as a parametric domain for manifold splines), such that the metric becomes flat everywhere except at one point. Then, the metric naturally induces an affine atlas covering the entire manifold except this singular point. Finally, manifold splines are defined over this affine atlas. The Ricci flow method is theoretically sound, and practically simple and efficient. We conduct various shape experiments and our new theoretical and algorithmic results alleviate the modeling difficulty of manifold splines, and hence, promote the widespread use of manifold splines in surface and solid modeling, geometric design, and reverse engineering.

Key words: Manifold splines, affine structure, discrete Ricci flow, extraordinary point, metric, differential geometry
PACS: Computational Geometry, Object Modeling, Geometric algorithms

1. Introduction and Motivation

1.1. *Problem Statement*

Despite many algorithmic and theoretical advances in solid modeling and shape computing in most recent years, one fundamental objective of our research community is always striving to develop novel modeling, design, and simulation schemes that are capable of accurately representing complicated real-world objects in a compact manner, and facilitating rapid computation of their desirable properties both globally and locally such as differential properties, smoothness requirements, and topological validity. Strongly inspired by the recent development of subdivision surfaces and manifold splines, our current research goal in this paper is to further advance the state of the knowledge in manifold splines. At the theoretic level, we devise manifold splines with only one singular point through the mathematical rigor of Ricci flow and relevant computational techniques. At the application level, we design a brand new algorithmic pipeline that enables all the computational elements towards the widespread use of manifold splines (especially the new, improved scheme with single extraordinary point) in solid

* Corresponding author. Tel: +65 65141008; Fax: +65 67926559. Address: 50 Nanyang Avenue, BLK N4, Nanyang Technological University, Singapore, 639798.

Email addresses: gu@cs.sunysb.edu (Xianfeng Gu), yhe@ntu.edu.sg (Ying He), mjin@cs.sunysb.edu (Miao Jin), fengluo@math.rutgers.edu (Feng Luo), qin@cs.sunysb.edu (Hong Qin), yau@math.harvard.edu (Shing-Tung Yau).

modeling, shape design, and reverse engineering.

1.2. *Manifold Splines*

For the perspectives of solid modeling, engineering design, finite element simulation, and scientific computation, elegant geometric properties such as high-order continuity and the ease of computing all the desirable properties rapidly are always mandatary for the development of novel shape representations. Therefore, it is not surprising to see that spline-centric polar forms [31] are becoming the most popular computational tools in geometric modeling and shape design. Essentially, the methodology of polar forms naturally gives rise to parameterization-centered, piecewise polynomials defined on any planar parameter domain for the effective modeling and accurate computing of smooth spline surfaces.

However, examining all the real-world applications, we observe that the most natural shapes are manifolds with complicated topologies and arbitrarily detailed geometric configurations, which can not be completely covered by a single open surface defined in one coordinate system (note that, it does not matter if the parametric surface is a polynomial or a non-polynomial, this fundamental principle remains the same). Instead, a manifold might be covered by a family of *coordinate charts*, each coordinate chart covers only a portion of the manifold. Different charts may overlap with each other, a *coordinate transition function* transforms from one coordinate system to the other. If we follow the algorithmic procedure of polar forms and other relevant computational techniques in a principled way, we can easily realize that conventional splines (defined over any open domain) can not be transferred over the manifold directly.

In order to model a manifold using piecewise polynomials, current approaches will segment the manifold to many patches, define a single coordinate system over each patch, such that each patch can be modelled by a spline patch. Finally, any generic approach will glue/abut all the spline patches together by adjusting the control points and the knots along their common boundaries. This whole process is mainly performed manually, and it requires the users' skill and mathematical sophistication, and is tedious and error-prone.

It is highly desirable to design splines defined over manifolds directly, such that different spline patches can be automatically glued together with high continuity, and the modelling process requires neither segmentation nor patching. Pioneering work has been done by Grimm and Hughes [7], which can model splines on arbitrary surfaces. Recently, Ying and Zorin [37] introduced a general method by constructing a conformal atlas. In both methods, smooth functions are defined on each chart and blended together to form a function coherently defined over the entire manifold. The methods are flexible for all manifolds with arbitrary topologies. The functions are with any degree of desirable continuity without any singularity. The primary drawbacks of these methods are that surfaces constructed this way are no-longer polynomials and their computation expenses are relatively high in comparison with conventional spline surfaces.

Most recently, the manifold splines proposed by Gu, He, and Qin [11] offer a different approach to manifold domain construction. The main advantage for manifold splines is that on each local chart, the functions are all piecewise polynomials in common use, currently available spline surfaces in commercial software packages can easily serve as building blocks towards the effective design of complicated models of arbitrary topology, and hence, the evaluation and all the necessary computational procedures are both efficient and robust. Furthermore, existing algorithms and softwares for conventional spline surfaces can all be easily adopted for use in applications of manifold surfaces. Nevertheless, certain drawbacks still remain: there must be singularities for general manifolds except tori. In [11], they discovered that the existence of the manifold splines is equivalent to the existence of a special atlas of the underlying manifold domain, whose transition functions are all affine among themselves, denoted as *affine atlas*. Unfortunately, it is impossible to find an affine atlas to cover arbitrary closed surfaces except tori. There must be singularities for the atlas which can not be covered by any chart within its collection set. Moreover, they proved that the minimal number of singularities equals to one without developing any practical algorithm. So, how to lower the number of singular points remains elusive, and how to devise new algorithms with a minimum number of singular points for practical applications remains extremely technically challenging.

Using existing popular techniques of polar forms and Euclidean metrics, the manifold splines in the neighborhood of singularities appear to be extremely difficult to construct, unstable, and error-prone. In addition, the mapping distortion from the surface to the affine atlas significantly affects the quality of the final spline surface. The distortions are intrinsically determined by the singularities for the affine atlas. Therefore, it is highly desirable for users to be able to control the position and the number of the singularities. For open surface cases, however, it is ideal to push the singularities away from the surfaces. In this paper, we demonstrate that Discrete Ricci flow is a powerful theoretic and computational tool for constructing affine atlas with full control of singularities, and specifically, being capable of minimizing the number of singularities to its theoretical lower bound (which is at most one for closed surfaces and zero for all open surfaces).

1.3. *Intrinsic Shape Space*

In reality, surfaces are typically acquired via modern scanning devices, and they are initially approximated by a set of points and/or triangular meshes. We shall consider the triangle-mesh approximation of domain manifolds first. In order to find an affine atlas of a triangle mesh, it is sufficient to find a configuration of edge lengths, such that the one-ring neighbor of each vertex is flat. So, any parameterization problem can be

formulated as:

Finding a configuration of edge lengths, such that each vertex has zero discrete Gaussian curvature [1]

One could naturally raise the following much broader questions: given a mesh,

(i) *What are the all possible configurations of edge lengths?*

(ii) *What are the all possible configurations of curvatures on vertices?*

(iii) *What is the relation between edge length configurations and curvature configurations? It is obvious that edge lengths determine curvatures. Can curvatures determine edge lengths?*

The entire space of all possible configurations of edge lengths is denoted as the *metric space*. The entire space of all possible configurations of vertex curvatures is denoted as the *curvature space*. Metric space and curvature space are *intrinsic shape spaces* of the mesh.

The answer to the admissible edge lengths is straightforward: any configuration satisfying triangle inequality is admissible. For admissible curvature configuration, the answer is much more complicated. There are mainly two constraints: topological constraint and combinatorial constraint. The topological constraint is represented as the Gauss-Bonnet formula, the total curvature equals to the $2\pi\chi$, where χ is the Euler number of the mesh. The combinatorial constraint ensures that all angles are between 0 and π, and represented solely by inequalities of curvature and connectivity. The technical details will be discussed in the next Section.

The answer to the third question has fundamental importance, it is the main focus of this research work. It is easy to compute curvature using edge lengths, but the inverse is much more complicated. Intuitively speaking, the Gaussian curvature is a map from the metric space to the curvature space, the mapping in general is not injective. However, one can select a subspace of the whole metric space, such that any two metrics in the subspace are conformally equivalent. Restricted on this subspace, Gauss curvature map is a homeomorphism, namely, any curvature configuration uniquely determines an edge length configuration in this subspace.

In practice, one can specify the target curvature and deform the edge length according to the difference between the current curvature and the target curvature. It is guaranteed that the curvature configuration of the final mesh will reach the target one. This kind of deformation process driven by Gaussian curvature is the so called *Ricci flow*.

[1] Discrete Gaussian curvature is defined as the difference between 2π and the summation of all angles adjacent to the vertex (see Section 3.2).

1.4. *Ricci Flow*

Ricci flow was first introduced in differential geometry by Hamilton in [13]. It has solid theoretic foundations. By nature, it is a constructive geometric tool and can be easily implemented, therefore, it has a great potential for real-world applications.

The fundamental idea of Ricci flow is rather simple. We can deform the surface driven by its curvature to the desired shape. Suppose S is a closed surface with Riemannian metric $\mathbf{g}$, and u is a function on S, then $e^{2u}\mathbf{g}$ is another metric on S conformal to $\mathbf{g}$. Ricci flow is explicitly defined as

$$\frac{du(t)}{dt} = \bar{K} - K(t), \tag{1}$$

where the area preserving constraint is explicitly formulated as,

$$\int_S dA = \int_S e^{2u} dA, \tag{2}$$

and $K(t)$ is the Gaussian curvature induced by the metric $e^{2u(t)}g$, and $\bar{K}$ is a constant

$$\bar{K} = \frac{2\pi\chi(S)}{\int_S dA}.$$

It has been proven that Ricci flow converges to the uniform metric that induces constant Gaussian curvature $\bar{K}$ on the surface, $K(\infty) \to \bar{K}$. Furthermore, Ricci flow converges to the final stable solution exponentially fast, for a given surface S, there exist two positive constants c_1, c_2 determined by the geometry of S, such that,

$$|K(t) - K(\infty)| < c_1 e^{-c_2 t}.$$

Ricci flow has many promising properties, which make it very valuable for real-world applications,

– Ricci flow offers the freedom to traverse the intrinsic shape space (all the admissible configurations of edge lengths) by driving the surface to deform to all possible shapes as long as the Gaussian curvature of the target shape is known.

– The deformation induced by Ricci flow is conformal.

– Ricci flow deforms the surface to a single solution and converges to the solution exponentially fast.

– Ricci flow can be formulated as a variational problem. The energy is a convex function, therefore has a single global optimum. Ricci flow is the negative gradient flow of the energy, and can be further speed up using Newton's method.

In our current research, the fundamental motivation for us to use Ricci flow is its computational power to compute the affine atlas of a mesh with any desired number of singularities, especially with only one singularity.

1.5. *Contributions*

In this paper, we devise a novel algorithm to construct manifold splines with only one singular point for closed surfaces and no

singular point for open surfaces, reaching its lower bound in theory. The algorithm is uniquely founded up on a mathematically rigorous tool in differential geometry, namely, Ricci flow. Key contributions of this paper include:

(i) We formulate the intrinsic space of a mesh: the metric space (i.e., all admissible configurations of edge lengths) and the curvature space (i.e., all admissible configurations of vertex curvatures). We point out the topological constraints and the combinatorial constraints for the metric spaces. We re-define the general surface parameterization problem as equivalence to finding flat metrics with any user-assigned singularities.

(ii) We articulate our new computational method to construct an affine atlas with any pre-determined singularities using Ricci flow. The affine atlases serve as the key and necessary elements for constructing manifold splines, especially, for manifold splines with only one singular point.

(iii) We offer a theoretically rigorous, practically simple and computationally efficient tool, Ricci flow, to solve geometric and solid modeling problems. In its discrete setting, given the Gaussian curvature on each vertex, Ricci flow will be employed to compute the configuration of edge lengths.

2. Background Review

This section briefly reviews previous work on splines, Ricci flow, parameterization, and circle packing.

Manifold Splines. Pioneering work has been done earlier by Grimm and Hughes [7], which can model splines on arbitrary surfaces. Recently, Ying and Zorin [37] introduced a general method by constructing a conformal atlas. The function basis in their constructions are smooth and without singularities, however, they are not polynomials, requiring the necessary data exchange between polynomial-based spline surfaces and their special-purpose functions for surface design.

Manifold splines defined by piecewise polynomials over manifold domains of arbitrary topology were first rigorously formulated in [11], which unifies the conventional spline surfaces based on polar forms and the subdivision surfaces of arbitrary topology. In their work, it is proven that a manifold admits a manifold splines based on polar forms if and only if it has an affine atlas. The topological obstruction for the existence of the affine atlas is the Euler class. By removing only one point, any oriented surface has an affine atlas.

Ricci Flow. Ricci flow on surface is introduced by Hamilton in [13], which will conformally deform the metric of a surface to a canonical metric with constant Gaussian curvature. For a closed genus-zero surface, Ricci flow will change the metric to the spherical metric with constant positive Gaussian curvature; for a genus-one closed surface, the solution to Ricci flow is the planar metric with zero Gaussian curvature; for a high genus closed surface, the solution to Ricci flow is the hyperbolic metric with constant negative Gaussian curvature. The analogue of Ricci flow in the discrete, combinatorial setting is first studied in [3]. It is proven that combinatorial Ricci flow will deform the metric of a triangle mesh to metrics with constant vertex curvatures. Recently, Jin *et al.* applied discrete Ricci flow to compute the hyperbolic and real projective structure of surfaces [21].

Circle Packing and Circle Pattern. Circle packing and circle pattern are used for approximating conformal deformations. Circle packing is first introduced by Thurston in [36], where he designed an algorithm to find the circle packing of a graph by adjusting the radii at vertices one at a time. Stephenson et al. developed practical algorithms in [35]. Circle pattern is introduced in [1] and applied for surface parameterizations in [22], which is closely related to circle packing. Instead of using circles centered at each vertex, this method uses the circum-circles of triangles. Comparing with circle pattern, the theoretic framework of Ricci flow is much simpler and clearer. Furthermore, the implementation of the Ricci flow is much easier in practice.

Global Surface Parametrization. Affine atlas can be computed using surface parametrization algorithms. In the literature, there exist many parameterization methods using a variety of distortion metrics. For a thorough survey, we refer the readers to the excellent work of Floater and Hormann [5,34]. We shall focus on the most related work, especially global parameterization methods.

Gu and Yau computed the conformal structure based on Hodge theory in [9]. The method computes the holomorphic 1-form basis, and induces a flat metric with $2g-2$ singularities. Ni et al. extracted the topological structure using harmonic morse function, the vector fields are holomorphic 1-forms, and induced a flat metric with more singularities [24]. Recently, Ray et al. [29] computed the global conformal parameterization also using holomorphic 1-form, but specifically tailored the parameterization to follow the principle curvature lines.

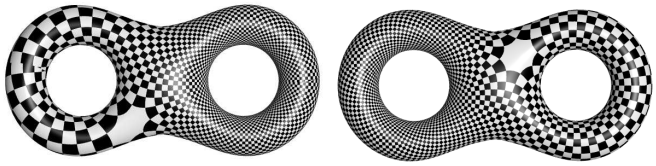
It may be note that, all current parameterization methods will introduce **multiple** singularities due to the topological obstruction. **The method to be developed in this paper is capable of reducing the number of singularity points to its theoretic lower bound (which is one)**.

3. Global Surface Parameterization Using Discrete Ricci Flow

Conventional *local* surface parameterization refers to the process of mapping a simply connected surface patch to a planar region. In contrast, a *global* surface parameterization maps the whole surface to the plane $\mathbb{R}^2$, the unit sphere $\mathbb{S}^2$ or the hyperbolic space $\mathbb{H}^2$ periodically. The global surface parameterization problem could be formulated in a precise and general way as deforming the given surface to satisfy the prescribed curvatures. By deforming the surface, we mean finding a different Riemannian metric (the first fundamental form). If conformal-

ity is required, then the new metric should be conformal to the original metric. Mathematically, suppose $\mathbf{g}$ is the original metric, then the metric conformal to $\mathbf{g}$ has the form $e^{2u}\mathbf{g}$, where u is the function defined on the surface. Then global surface parameterization is to solve function u by the prescribed curvature.

In the following, we assume the surface is an oriented 2-manifold, represented by a two dimensional simplicial complex (i.e., triangular mesh) $M = (V, E, F)$, where V is the set of all vertices, E is the set of all non-oriented edges, and F the set of all faces. We use $v_i, i = 1, 2, \cdots, n$ to denote its vertices, e_{ij} to denote an oriented edge from v_i to v_j, f_{ijk} to denote an oriented face, v_i, v_j, v_k are sorted counterclock-wisely.



(a). front side (b). back side

Fig. 2. **Affine atlas induced by a global conformal surface parameterization.** The affine atlas is illustrated by texture mapping of a checkerboard pattern. There are $2g-2$ singularities centered at the white octagons.

3.1. *Discrete Conformal Metrics*

The central task is to approximate Ricci flow (1) in the discrete mesh setting. Continuous Ricci flow conformally deforms a surface.

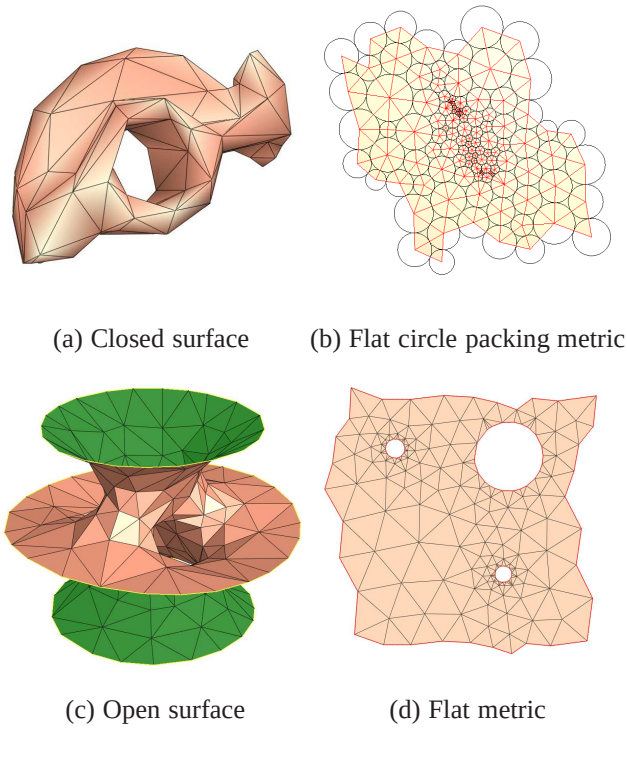
Figure 1 illustrates an important observation for continuous conformal mappings: they transform infinitesimal circles to infinitesimal circles, and preserve the intersection angles among the circles. Based on this property, Thurston introduced the circle packing metric in early eighties [36]: a circle with the radius γ_i is associated with each vertex v_i. For an each edge e_{ij}, two circles intersect at the angle Φ_{ij}, called edge weight. The edge length of e_{ij} is determined by γ_i, γ_j and Φ_{ij},

$$l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\Phi_{ij}}. \tag{3}$$

It can be shown that for any face f_{ijk} with vertex radii $\{\gamma_i, \gamma_j, \gamma_k\}$ and edge weights $\{\Phi_{ij}, \Phi_{jk}, \Phi_{ki}\}$, if edge weights are acute angles, then the edge lengths $\{l_{ij}, l_{jk}, l_{ki}\}$ satisfy the triangle inequality,

$$l_{ij} + l_{jk} > l_{ki}.$$

We use $\Gamma : V \to \mathbb{R}^+$ to denote the vertex radii, $\Phi : E \to [0, \frac{\pi}{2}]$ for the edge weights, then a circle packing metric is represented as (M, Γ, Φ).



(a) Closed surface (b) Flat circle packing metric

(c) Open surface (d) Flat metric

(e) Universal covering space

Fig. 3. **Computing the affine structures for genus one surfaces using discrete Ricci flow.** The right column (b) and (d) shows the embedded fundamental domain. The last row shows the universal covering space.

Two circle packing metrics (M, Γ_1, Φ_1) and (M, Γ_2, Φ_2) are conformal to each other, if $\Phi_1 \equiv \Phi_2$. Namely, a discrete conformal mapping will change the vertex radii only and preserve the intersection angles. Figure 4 and Figure 5 illustrate the circle packing metric.

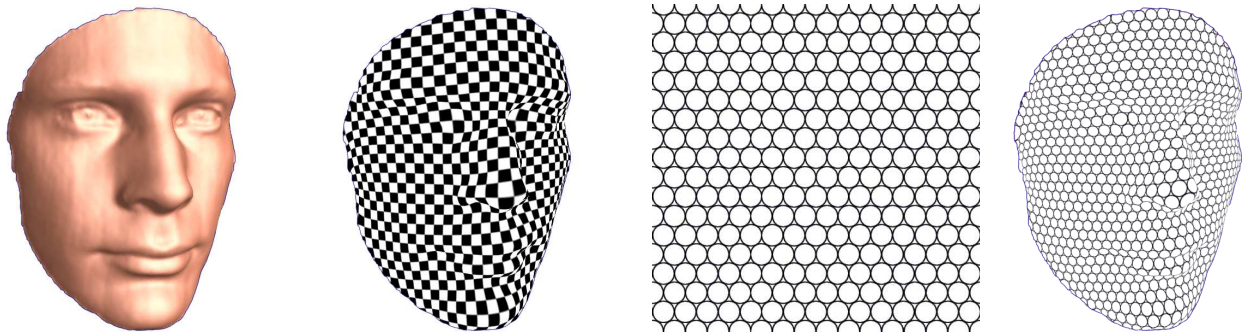


Fig. 1. **Circle packing for a surface.** Conformal mappings transform the infinitesimal circles on the texture plane to the infinitesimal circles on the surface.

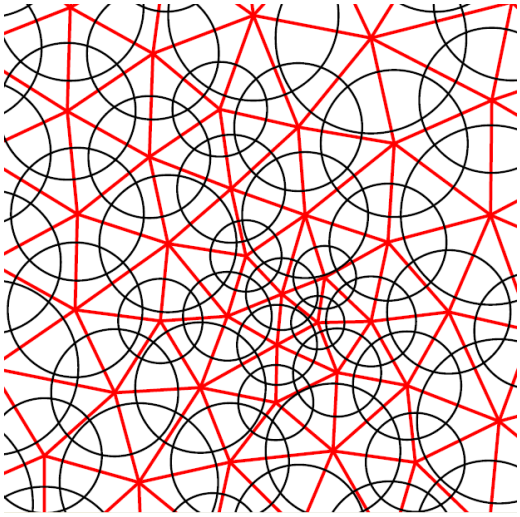


Fig. 4. **Close-up view of the flat circle packing metric using Ricci flow.**

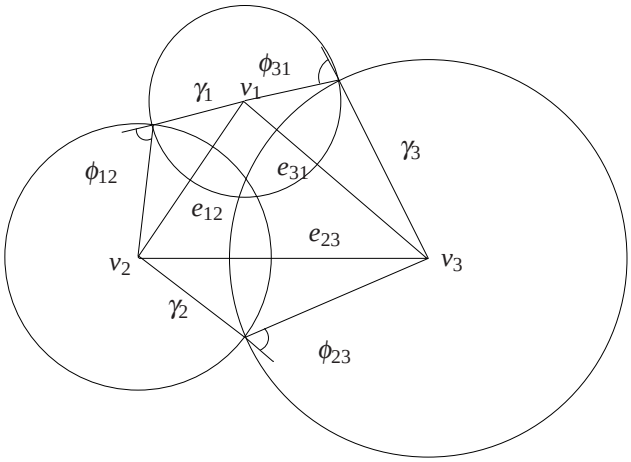


Fig. 5. **Circle packing metric for a triangle.** Triangle $[v_1, v_2, v_3]$ has vertices v_1, v_2 and v_3, edges e_{12}, e_{23} and e_{31}. Three circles centered at v_1,v_2, v_3, with radii γ_1, γ_2 and γ_3 intersect one another, the intersection angles are Φ_{12},Φ_{23} and Φ_{31}, which are the weights associated with the edges. The edge lengths of the triangle are determined by γ_i and Φ_{ij} by the cosine law.

3.2. *Discrete Curvature*

Given a discrete metric (M, Φ, Γ), suppose f_{ijk} is a face, the angle of vertex v_i in f_{ijk} is denoted as θ_i^{jk}, then the discrete

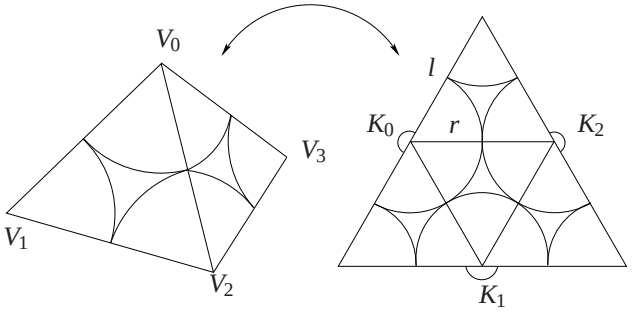


Fig. 6. **Circle packing metric and curvature.** For a canonical tetrahedron, the edges lengths equal to $l = 1.0$, the radii on all the vertices equal to $r = 0.5$. The curvature on each vertex equals to $K_i = \pi$. The weights of all edges Φ equals to 0.

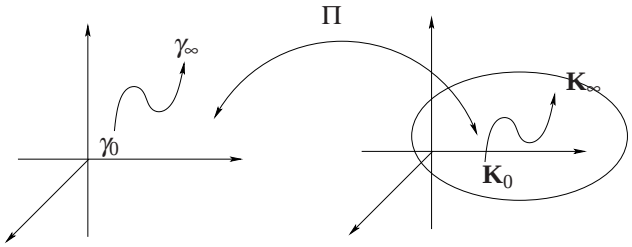


Fig. 7. **Gaussian curvature is a homeomorphism between the circle packing metric space based on (M, Φ) and the curvature space, the inverse map can be computed using Ricci flow.** We start from the known metric γ_0 and the known curvature $\mathbf{k}_0$, then flow to the target curvature $\mathbf{K}_\infty$ using Ricci flow, then the metric will flow to the corresponding metric $\gamma_\infty = \Pi^{-1}(\mathbf{K}_\infty)$.

Gaussian curvature K_i at an interior vertex v_i is defined as

$$K_i = 2\pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, v_i \notin \partial M, \tag{4}$$

the discrete Gaussian curvature for an boundary vertex v_i is defined as

$$K_i = \pi - \sum_{f_{ijk} \in F} \theta_i^{jk}, v_i \in \partial M. \tag{5}$$

Figure 6 demonstrates the circle packing metric for a tetrahedron surface, where all the edge weights are zeros, all the vertex radii are 0.5, and all the vertex curvatures are π.

The Gaussian curvature at each vertex could be arbitrary, but the total curvature is confined by the topology of the surface. This is indicated by the Gauss-Bonnet theorem.

Theorem 1 (Gauss-Bonnet) *Suppose M is a mesh, the total discrete Gaussian curvature equals the product of 2π and its Euler number,*

$$\sum K_i = 2\pi\chi. \tag{6}$$

Furthermore, for any discrete metric (M,Φ,Γ), $\Phi : E \to [0,\frac{\phi}{2}]$ and any proper subset I of vertices V,

$$\sum_{i\in I} K_i(v) > -\sum_{(e,v)\in Lk(I)} (\pi - \Phi(e)) + 2\pi\chi(F_I), \tag{7}$$

where F_I is the set of all faces in M whose vertices are in I, $Lk(I)$ is the link of I being the set of pairs (e,v) of an edge e and a vertex v so that (1) the end points of e are not in I and (2) the vertex v is in I and (3) e and v form a triangle.

The following theorem is fundamental that the map between the vertex radii Γ and the discrete curvature K is a homeomorphism, detailed proof can be found in [3].

Theorem 2 *If a discrete metric (M,Γ,Φ) induces discrete curvature K, then K satisfies the Gauss-Bonnet Equation (6) and the set of all linear inequalities (7). If M and Φ are given, K satisfies (6) and the set of all linear inequalities (7), then there exists a Γ unique up to scaling, such that K is induced by the metric (M,Γ,Φ).*

Global surface parameterization problem can be re-formulated as follows:

Global surface parameterization is to find a special metric, such that the curvatures are zero almost everywhere except at several singularities.

For example, conventional global conformal surface parameterization is to compute a special metric on the mesh, such that at $|2g-2|$ singularities, the curvatures equal to -2π. The singularities are determined by the geometry of the surface, as shown in Figure 2. Ricci flow method allows the user to freely assign singularities for global parameterizations, as long as the target curvature satisfies the conditions in Theorem 2. Figure 7 illustrates the relation between circle packing metric space and curvature space.

3.3. *Discrete Ricci Flow*

One can assign discrete Gaussian curvature $\bar{K}$ for a weighted mesh (M,Φ) as long as $\bar{K}$ satisfies the conditions in Theorem 2. Discrete Ricci flow is able to solve the vertex radii Γ. We use $e^u\Gamma$ to denote the conformal metric with vertex radius $e^{u_i}\gamma_i$ at vertex i. Similar to the continuous Ricci flow (1),

Definition 3 (Discrete Ricci flow) *The discrete Ricci flow is defined as*

$$\frac{du_i}{dt} = (\bar{K_i} - K_i), \tag{8}$$

where $\bar{K_i}$ is the desired discrete Gaussian curvature at vertex v_i, under the constraint $\sum u_i = 0$ (equivalent to the area-preserving constraint).

Similar to continuous Ricci flow, it is proven that discrete Ricci flow also converges to this stable solution exponentially fast.

Definition 4 (convergence) *The solution to (8) is called* convergent *if*

(i) $\lim_{t\to\infty} K_i(t) = \bar{K_i}$ *exists for all i,*

(ii) $\lim_{t\to\infty} \gamma_i(t) = \bar{\gamma_i} \in \mathbb{R}^+$ *exists for all i.*

A convergent solution is called convergent exponentially fast *if there are positive constants c_1, c_2, so that for all time $t \geq 0$,*

$$|K_i(t) - \bar{K_i}| \leq c_1 e^{-c_2 t},$$

and

$$|\gamma_i(t) - \bar{\gamma_i}| \leq c_1 e^{-c_2 t}.$$

The following theorem states that discrete Ricci flow is guaranteed to converge exponentially [3].

Theorem 5 *Suppose (M,Φ) is a closed weighted mesh. Given any initial circle-packing metric based on the weighted mesh, the solution to the discrete Ricci flow (8) in the Euclidean geometry with the given initial value exists all the time and converges exponentially fast. The solution converges to the metric $\Pi^{-1}(\bar{K})$.*

3.4. *Conformality*

In practice, it is highly desirable for the deformation to be conformal, namely, angle preserving. A conformal map transforms an infinitesimal circle to an infinitesimal circle, as shown in Figure 1. Therefore conformal mapping only changes the radii γ in the circle packing metric (M,Φ,Γ), and preserves the intersection angles Φ among the circles. It can be proven that continuous conformal mapping can be approximated with arbitrary accuracy by discrete maps using circle packing [30].

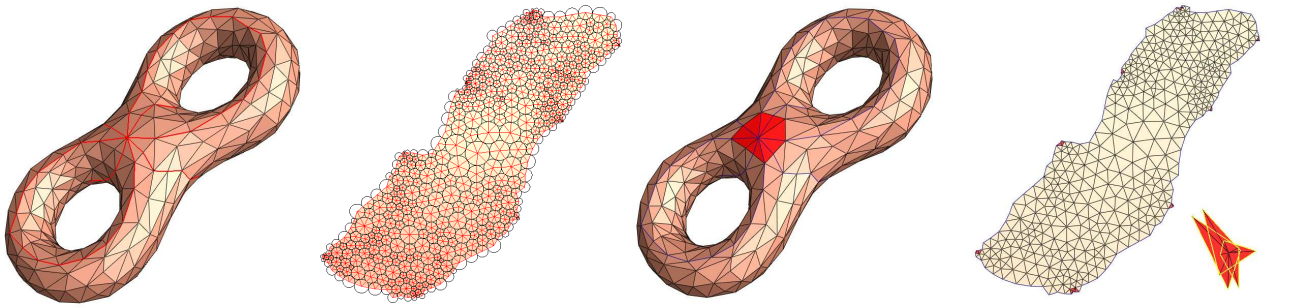
In graphics applications, the meshes are embedded in $\mathbb{R}^3$, the metrics are induced from that of $\mathbb{R}^3$. We can find the optimal weight Φ with initial circle radii Γ, such that the circle packing metric (M,Φ,Γ) is as close as possible to the Euclidean metric in the least square sense. Namely, we want to find (M,Φ,Γ) by minimizing the following functional

$$min_{\Gamma,\Phi} \sum_{e_{ij}\in E} |l_{ij} - \bar{l_{ij}}|^2, \tag{9}$$

where $\bar{l_{ij}}$ is the edge length of e_{ij} in $\mathbb{R}^3$.

4. Affine Atlas Construction

In this section, we detail our algorithm of constructing the affine atlas by employing Ricci flow. The entire pipeline of the algorithm is illustrated in Figure 9.

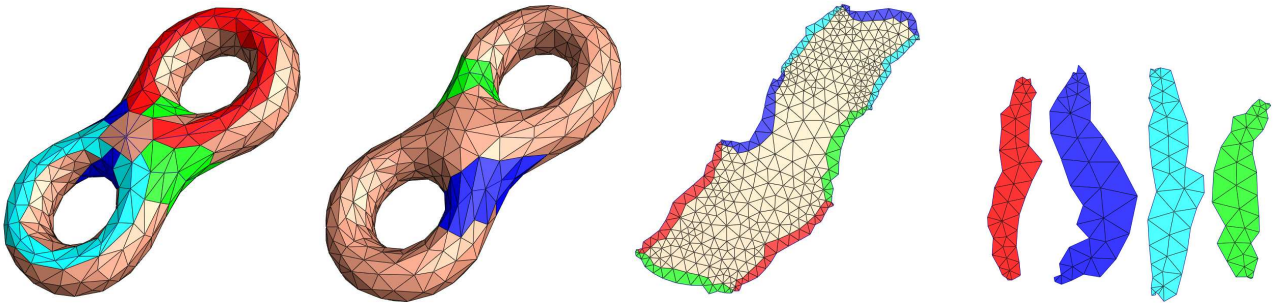


(a) Canonical homology basis passing the singular vertex

(b) Flat circle packing metric

(c) One ring neighbor of the singularity

(d) Central chart and the one ring neighbor of the singularity



(e) Open Covering front view

(f) Open covering back view

(g) The central chart

(h) Other charts covering the cut graph

Fig. 8. **Affine atlas automatically acquired by using Ricci Flow.** First, the user selects one singular point as shown in (a). Then a cut graph is labelled either manually or automatically as the dark curves in (a), where the cut graph is a set of canonical homology basis curves passing through the singular vertex. Second, the flat circle packing metric is computed using Ricci flow, illustrated in (b). The flat metric induces an planar embedding. The entire surface is sliced open along the cut graph to form a topological disk (i.e. fundamental domain). The interior of the fundamental domain is bijectively mapped to the plane. The mapping of the one-ring neighbor of the singular vertex is not 1 to 1, but $2g-1$ to 1 shown in (d). Other charts covering the cut graph are constructed as shown in (e) and (f), and their overlapping relation with the central chart is shown in (g) and (h) by encoded colors, respectively.

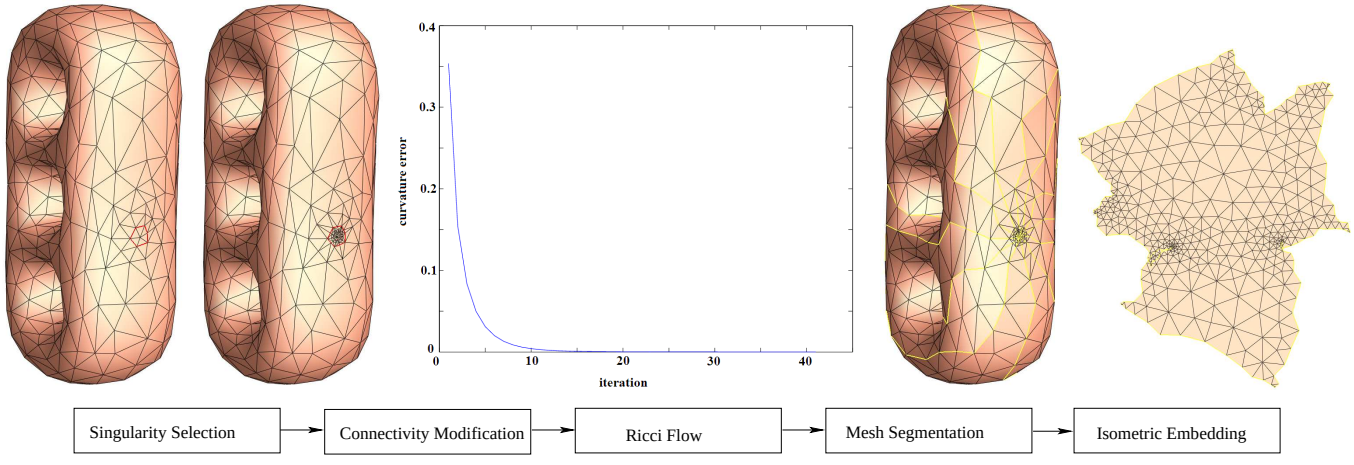


Fig. 9. **The pipeline to compute the affine atlas using discrete Ricci flow.**

Step 1: Selecting Singularities

We can select the singular vertices $\{v_1, v_2, \cdots, v_k\}, k \geq 0$ anywhere on the mesh arbitrarily, then we assign the target curvature of the singular vertices such that

$$\sum_{i=1}^{k} \overline{K}(v_i) = 2\pi\chi,$$

where χ is the Euler number of the surface; the target curvature of other vertices are zero. Note that, there are several special cases that must be addressed.

– If the surface is a closed genus one mesh, then no singular

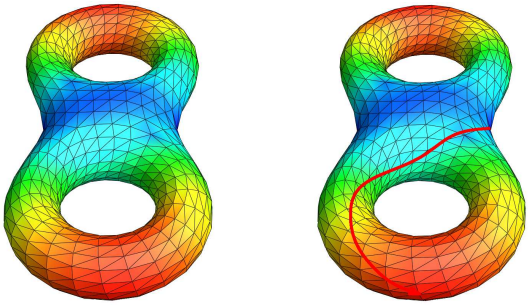


Fig. 10. **(a) The distribution of the area distortion is color encoded. (b) The path of the arrow illustrate the rough behavior of the error distribution for the eight model.**

vertex is needed.

– For a high genus mesh, we can select only one singular vertex and concentrate all curvatures on it.

– If the mesh is open, we can assign the target curvatures for all the interior vertices to be zero and assign the target curvatures for boundary vertices such that the total boundary curvatures equal to $2\pi\chi$. By this way, all the non-zero curvature will be pushed to the boundary.

Ricci flow only changes the vertex radii, therefore, the resulting metric is conformal to the original one, and no angle distortion will be introduced. But the area distortion is unavoidable. The uniformity of the parameterization varies drastically depending on the choice of singularities. Our selection is based on the minimal area distortion among all possible cases. For each case, we set one vertex as the single singularity.

The area distortion error is measured with the following equation:

$$\frac{\sum_{i=1}^{n}(-\log(S_i)-\log(s_i))^2}{n}$$

where S_i is the i-th triangle area in surface, s_i is its area in parametrization domain, and n is the total number of triangles. Figure 10(a) shows the distribution of the area distortion, blue area with low distortion, and red area with high distortion. In figure 10(b), the path of the arrow illustrates the rough behavior of the error distribution for the eight model.

When we sort the vertices with the area distortion errors, we may find the best position to put the singularity. Figure 11 gives five different cases of setting singularity. From the left to the right, the area distortion is increasing, with the left most being the best parameterized and the right most being the worst parameterized.

Step 2: Modify Local Connectivity around the Singular Vertex

In order to determine the desired flat metric, the combinatorial constraints for the curvature (7) have to be satisfied. If both the initial curvature configuration and the target curvature configuration satisfy the constraints, any intermediate curvature configuration during Ricci flow will satisfy the constraints. Thus, it is enough to only consider the target curvature. If some singularities have high target curvature concentration, we need to modify the local connectivity in their neighborhoods.

In practice, we replace the combinatorial constraints Eq. 7 by a stronger one which is independent of the edge weight Φ,

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

We modify the connectivity around the extraordinary point and make the sampling in the neighborhood of the singularity much denser. We increase the connectivity of the singularity no less than four times of genus, and vertices in the neighborhood have valence of about 6. This can be summarized as follows:

(i) The valence of a singular vertex v is no less than $4-\frac{2\bar{K}(v)}{\pi}$.

(ii) For all the vertices in the first n ring neighbor of the singular vertex, their valences are no less than 6, n is a small integer. We set $n=3$ in our experiments in this paper.

Figure 15 demonstrates the step of the connectivity modification around the extraordinary point. This step can be easily done using the edge split operation in half edge data structure.

Step 3: Ricci Flow

In order to compute the flat metric, we use Ricci flow to drive the mesh to deform in this order:

(i) Set the initial value $u_i = 0$ for each vertex.

(ii) Assign the weight for each edge and the radii for each vertex by minimizing the energy,

$$min_{\Gamma,\Phi}\sum_{e_{ij}\in E}|l_{ij}-\bar{l}_{ij}|^2, \tag{10}$$

where $\bar{l}_{ij}$ is the edge length of e_{ij} in $\mathbb{R}^3$. Namely, the discrete metric (M,Φ,Γ) is consistent with the induced Euclidean metric on M.

(iii) Update the vertex radius γ_i by $e^{u_i}\gamma_i$. Compute current edge length l_{ij}'s using Equation (3), corner angles, and discrete curvatures for each vertex using Equation (4) or (5).

(iv) Update vertex radii,

$$u_i += \varepsilon\times(\bar{K}_i - K_i), \tag{11}$$

where ε is a carefully selected step length. Note that if the step length is too small, the convergence is very slow. $\varepsilon = 0.1$ is used in the experiments in this paper.

(v) Normalize u_i, such that the summation of all u_i's equals to zero.

(vi) Check the deviation between K_i and $\bar{K}_i$, if the error is less than a predetermined threshold, the algorithm terminates. Otherwise, goto Step 2.

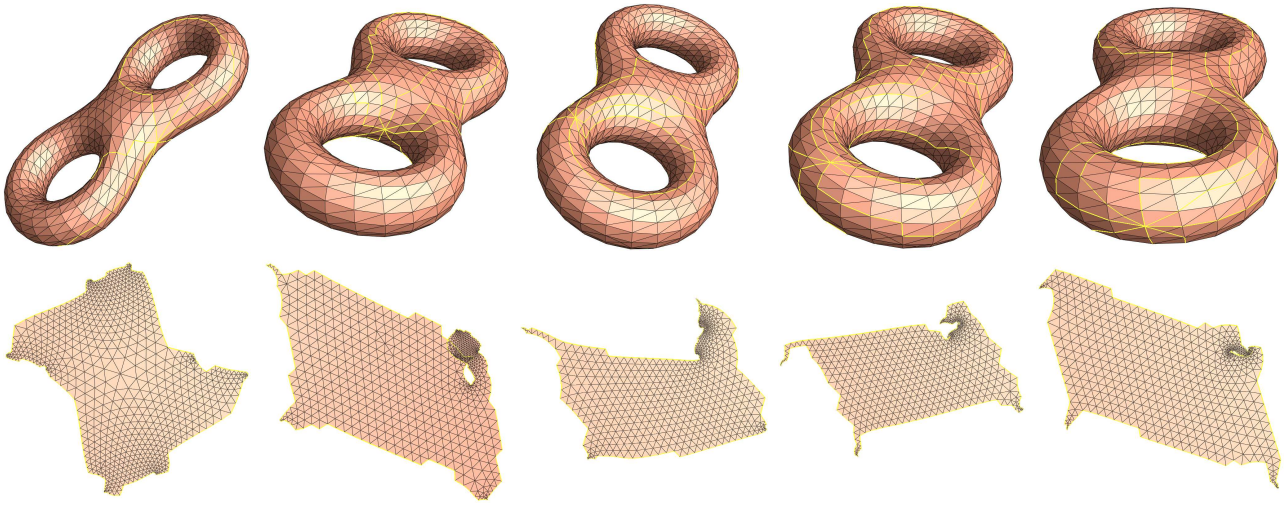


Fig. 11. **The first row gives the different positions of singularities on the same model; The second row shows their corresponding flat metric.** From left to right, the area distortion increases.

The algorithm will converge exponentially fast. In practice, the step length might be time-varying in order to improve the efficiency.

Step 4: Segmentation

Next, in order to construct the affine atlas, an open covering of the mesh needs to be built. The basic idea is to find a set of curves G such that the mesh M can be sliced open along the curves and form a topological disk. Such kind of curves form the *cut graph* as introduced in the work on geometry images [11].

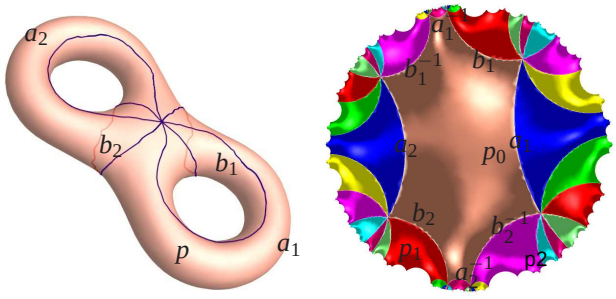


Fig. 12. **A genus two surface with a set of canonical fundamental group generators $\{a_1,b_1,a_2,b_2\}$ is shown on the left. A finite portion of its universal covering space is shown on the right.** Different fundamental domains are drawn in different colors. The boundary of each fundamental domain is the preimage of $a_1b_1a_1^{-1}b_1^{-1}a_2b_2a_2^{-1}b_2^{-1}$. The points $\{p_0,p_1,p_2\}$ are the primages of p on the surface.

If there is only one singularity p_0, the cut graph can be constructed using a set of canonical homology basis passing through the singularity as introduced in [2]. The cut graph has one node and $2g$ edges. The edges can be labelled as $a_1,b_1,a_2,b_2,\cdots,a_g,b_g$ (see Figure 12).

Then the mesh is cut open along the cut graph to form a big chart $\bar{M}$. The boundary of $\bar{M}$ has canonical form

$$\partial\bar{M} = a_1b_1a_1^{-1}b_1^{-1}a_2b_2a_2^{-1}b_2^{-1}\cdots a_gb_ga_g^{-1}b_g^{-1}.$$

Each edge starts and ends at the singularity p_0. We cover each edge s_k by a chart

$$U_k = \cup_{v_i\in s_k} N_i, v_i \neq p_0, N_i = \cup f_{ijk},$$

where N_i represents the one ring neighbor of vertex v_i. The algorithm for computing an open covering of M is as follows:

(i) Compute a cut graph G using a canonical homology basis.

(ii) Slice the mesh along the cut graph to form a topological disk $\bar{M}$.

(iii) For edges of the cut graph, compute the union of one ring neighbors of all its interior vertices.

(iv) The open covering of M is formed by $\bar{M}$ and U_k,

$$M/\{p_0\} \subset \bar{M}\cup_k U_k.$$

Step 5: Planar Embedding

Because the curvature for each vertex is zero, the faces can be flattened one by one on the plane. The following algorithm describes the details on how to flatten an open set $U \subset M/\{p_0\}$. Let the desired parameterization is $\tau : U \to \mathbb{R}^2$,

(i) Label all faces in U as non-processed ones. Meanwhile, label all vertices in U as non-processed.

(ii) Select randomly a face $f_0 = [V_0,V_1,V_2]$ from U, label f_0 as processed, label all its vertices as processed. Assign $\tau(v_0) = (0,0)$ and $\tau(v_1) = (l_{01},0)$. Compute $\tau(v_2)$ such that

$$|\tau(v_2)-\tau(v_0)| = l_{01}, |\tau(v_2)-\tau(v_1)| = l_{12}, \qquad (12)$$

and

$$(\tau(v_1)-\tau(v_0))\times(\tau(v_2)-\tau(v_0))\cdot n > 0, \qquad (13)$$

(iii) Find all faces in U sharing an edge with f_0, insert them to a face queue Q.

(iv) If Q is empty, simply terminate. Otherwise, fetch the first face $f = [v_0, v_1, v_2]$ from Q, label f as processed.

(v) If all vertices of f have been processed, go to Step 3. Otherwise, there must be only one vertex which has not been processed, assume it is v_2, label v_2 as processed.

(vi) Compute $\tau(v_2)$, such that both distance condition (12) and orientation condition (13) are satisfied.

(vii) Find all neighboring faces sharing an edge with f and they are not yet to be processed, add them to Q. Go to Step (iii).

In order to reduce the accumulation error, the parameterization can be further improved by minimizing the following functional,

$$\min_{\tau} \sum_{e_{ij}} (|\tau(v_i) - \tau(v_j)|^2 - l_{ij}^2)^2.$$

The purpose of the above functional is to find a valid embedding such that the distortion between the edge length in the parametric domain and the Ricci flow output is minimal. In practice, this step is usually unnecessary if singularities are carefully chosen to spread out the surface and the curvatures for each of them is not extremely high.

Figure 3, 8, and 16 demonstrate the affine atlas for surfaces from genus one, two and three, respectively.

5. Manifold Spline Construction

After the affine atlas are constructed in the previous section, this section first briefly summarizes the theory of manifold splines, then presents our experimental results.

Suppose M is a mesh with the one ring neighbors of the singular vertices removed. $\{(U_i, \tau_i)\}$ is an affine atlas, where U_i is a topological disk comprised by a set of faces of M, $\tau_i : U_i \to \mathbb{R}^2$ maps U_i onto the plane, namely, (U_i, τ_i) forms a local coordinate chart. The chart transition functions $\tau_{ij} : \tau_i(U_i \cap U_j) \to \tau_j(U_i \cap U_j)$ is a rigid-body motion in $\mathbb{R}^2$.

A manifold spline is defined on the mesh $\mathbf{F} : M \to \mathbb{R}^3$, such that

– The local representations of manifold splines, $\mathbf{F} \circ \tau_i^{-1} : \tau_i(U_i) \to \mathbb{R}^3$, are commonly used spline schemes with planar parameter domain.

– The evaluation of manifold splines is independent of the choice of local parameter charts,

$$\mathbf{F} \circ \tau_i^{-1} = \mathbf{F} \circ \tau_j^{-1} \circ \tau_{ij}$$

In our current implementation for this paper, we use triangular B-splines [**?**], because it has no restrictions on the connectivity of the mesh and it can represent any polynomials defined over planar [14], spherical [**?**] and manifold domain [11]. We have implemented our own version based on a generic half edge mesh library as in [20], while adding the edge lengths, vertex radii, and curvature as the new attributes for the underlying mesh.

In our prototype software system, we have tested several meshes of genus from zero to three. In this paper, we choose manifold triangular B-spline because of its flexibility in domain construction. This method can be also applied to other manifold splines, such as T-splines [18] and Powell-Sabin splines [16].

Given a domain manifold M, a manifold triangular B-spline surface is defined as follows:

$$\mathbf{F}(\mathbf{u}) = \sum_{I} \sum_{|\beta|=n} \mathbf{c}_{I,\beta} N_{I,\beta}(\tau_I(\mathbf{u})),\ \mathbf{u} \in M,$$

where I is the triangle index and $\beta = (\beta_0, \beta_1, \beta_2)$ is the 3-tuple to label the control points and knots. The algorithm for constructing manifold triangular B-spline is as follows:

(i) The initial control points $\mathbf{c}_{I,\beta}$ are chosen by uniformly subdivided the domain manifold M according to the user-specified degree n. Each domain triangle is associated with $(n+1)(n+2)/2$ control points.

(ii) To compute the optimal control points $\tilde{\mathbf{c}}_{I,\beta}$, we solve the following linear constrained least square problem:

$$\min_{\tilde{\mathbf{c}}} \sum_{I} \sum_{|\beta|=n} \|\tilde{\mathbf{c}}_{I,\beta} - \mathbf{c}_{I,\beta}\|^2 \qquad (14)$$

$$\text{subject to } \tilde{\mathbf{c}}_{I,\beta} = f^J(V_\beta^I), \forall I, \forall \beta, |\beta| = n, \beta_2 \le r$$

where $I = \triangle(v_0, v_1, v_2)$ and $J = \triangle(v_3, v_1, v_0)$ are adjacent triangles with common edge $\{v_0, v_1\}$, $V_\beta^I = \{\mathbf{t}_{0,0}^I, \ldots, \mathbf{t}_{0,\beta_0-1}^I, \ldots, \mathbf{t}_{2,0}^I, \ldots, \mathbf{t}_{2,\beta_2-1}^I\}$ and $\mathbf{t}_{i,j}^I$ are the knots for triangle I.

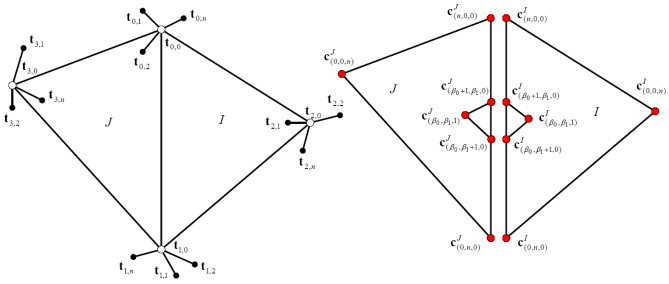


Fig. 13. **The linear constraints in the least square problem Eq 14 for a cubic manifold triangular B-spline.** (a) shows two adjacent triangles $I = \triangle(v_0, v_1, v_2)$ and $J = \triangle(v_3, v_1, v_0)$ and the knot configurations. (b) highlights six control points, three for each triangle. Note that the constraint for case $r = 0$ is equivalent to the shared control points, i.e., $\mathbf{c}^J_{\beta_0+1,\beta_1,0} = \mathbf{c}^I_{\beta_0+1,\beta_1,0}$ and $\mathbf{c}^J_{\beta_0,\beta_1+1,0} = \mathbf{c}^I_{\beta_0,\beta_1+1,0}$. The constraint for case $r = 1$ requires that the highlighted six control points to be coplanar.

Note that the initial manifold triangular B-spline surfaces acquired by step 1 usually have very bad curvature distribution, especially along the edges of the domain triangles. The purpose of step 2 is to fair the spline surface by modifying the control points. In the objective function Eq (14), we minimize the squared distance between the control points of the original and the new spline surface, which implies that the minimal

change of the shape. In the constraints, we use an integer r, $0 \leq r \leq n-1$, to control the fairness of the spline surface. The bigger the value r, the more faired surface we obtain. In our experiments, we can get visually pleasing surfaces with $r = 1$ for cubic splines or $r = 2$ for splines of degree 5 or above. Figure 13 illustrates the case $r = 1$. For the detailed information about spline fairing, please refer to [17].

Fig. 14. **Examples of manifold splines with various extraordinary points.** Rabbit: genus zero, two boundaries, no singularity; Vase: genus two, one singularity; Cup: genus two, one boundary, no singularity.

Figure 8 shows the flat circle packing metric of a genus two surface and its affine atlas. Figure 9 demonstrates the process of using Ricci flow on how to compute the affine atlas. The sculpture surfaces in Figure 16 is of genus three with different resolutions. The singular vertex and the cut graph are explicitly shown in this figure. The affine atlas are also highlighted in the figure.

All the examples of manifold triangular *B*-splines are shown in Figure 14, 17 and 18. Table 1 shows the statistics of the test cases. As shown in this table, our algorithms for constructing the affine atlas and manifold triangular *B*-splines are extremely fast, i.e., within only a few seconds.

Note that genus 1 surfaces, such as Rockerarm (closed) and Hypersheet (open), do not have singularities. There is no singularity for the cup model (genus 2 open surface) as all the non-zero curvatures are pushed to the boundary. For the Rabbit model, we introduce two cuts, one on the top, the other on the bottom, and then apply the double covering to convert it into a closed genus one surface. Thus, no singularity exists for the Rabbit model. For the Bunny model, we specify the singularity on the head. For other genus 0 surfaces with large extrusion parts, such as Horse and Camel, we choose multiple singularities on their legs to reduce the area distortion of the affine atlas. For the high genus closed models, such as Eight, Vase and 3-hole torus, only a single extraordinary point is specified. As we remove the extraordinary points and their one-ring neighbors in spline construction, there are holes in the resulted splines. In the postprocessing step, we use planar triangular *B*-splines to fill these holes with G^1 continuity along the boundaries.

6. Discussion

This section discusses several implementation related issues.

Conformal structure vs affine structure. It is proven that conformal structure induces affine but not vice versa [11]. Therefore, conformal structure is in some sense stronger than affine structure. In [11], Gu *et al.* showed a method to construct manifold triangular *B*-splines using conformal structure. In fact, manifold splines are solely defined using affine structure, since all the popular planar splines such as NURBS, Bèzier splines, triangular *B*-splines are parametric affine invariant. Thus, from the spline construction point of view, one can totally ignore the conformal constraints. The affine atlas shown in Figure 9 is computed by ignoring the conformal constraints, i.e., we simply skip Step 3 (ii) and assign the initial radius of each vertex to be 1. Therefore, the angle Φ_{ij} in the circle packing metric is zero for each edge e_{ij}. Since discrete Ricci flow preserves the Φ_{ij}, the triangles in the isometric embedding tend to be equilateral triangles. We should also point out that remeshing of the input domain mesh is not necessary except the valance of the singular point needs to be increased.

Number of singularities. The intrinsic connection between manifold splines and polar forms results from affine structure and affine atlas. To make these geometric structures computational tractable in shape modeling applications, we resort to the powerful tool of global parameterization over arbitrary manifold domain. It is known that global parameterization often suffers from severe area distortion. The quality of the global parameterization is determined by many factors, such as the connectivity of the mesh, the weights on edges, the positions and curvatures of the singularities. It is technically challenging on how to optimize these factors towards the quality improvement of global parameterization.

Section 4 presents a brutal force method to locate the singularity such that the area distortion of the affine atlas is minimal. Usually, the more number of singularities, the less area distortion in the affine atlas. In practice, the singular points should be chosen on parts with large extrusion, e.g., the feet of the horse model in 17.

Besides this work which minimizes the *number of singularities*, we developed polycube T-splines which aims to minimizes the *area distortion* in affine atlas [38]. However, the price to pay for the lower area distortion is the significantly larger number

Table 1
Statistics of test examples. g, genus of domain manifold M; N_f, # of faces in M; N_b, # of boundaries in M; N_s, # of singularities; N_c, # of control points; T_{ricci}, time for computing the discrete Ricci flow and isometric embedding (Step 3 and 5 in Section 4); T_{spline}, time for the spline construction; n, degree of splines. Note that time measures in seconds.

Object	g	N_f	N_b	N_s	N_c	T_{ricci}	T_{spline}	n	Continuity
Bunny	0	621	3	1	2827	6s	3s	3	C^2
Rabbit	0	1038	2	0	4698	12s	8s	3	C^2
Horse	0	4002	5	2	18074	26s	18s	3	C^2
Camel	0	2958	5	1	13380	18s	12s	3	C^2
Rockerarm	1	614	0	0	7675	2s	5s	3	C^2
Hypersheet	1	300	3	0	1446	5s	2s	3	C^2
Eight	2	806	0	1	3644	4s	3s	3	C^2
Vase	2	1480	0	1	6666	15s	10s	3	C^2
Cup	2	1929	1	0	15515	20s	18s	4	C^3
3-hole torus	3	878	0	1	3955	8s	4s	3	C^2

of extraordinary points. Therefore, in real applications, it is the user's call to make the balance

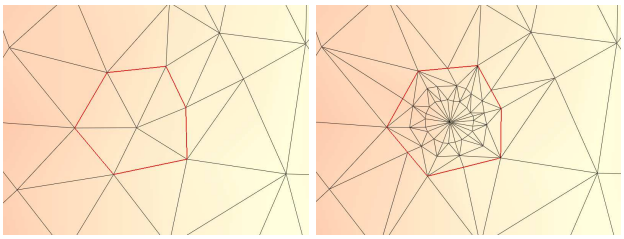
7. Conclusion and Future Work

This paper has developed an efficient and rigorous algorithm for constructing a manifold spline surface of complicated topology and complex geometry with single extraordinary point, which has already reached the theoretic lower bound of the number of singularities. The uniqueness of this construction algorithm for manifold splines is that, it is solely based on a simple and powerful computational tool: Ricci flow. From the mathematical point of view, Ricci flow has substantial relevance to the curvature flow method in differential geometry. For example, Ricci flow can conformally deform the metric to induce any prescribed curvature.

Current manifold splines are essentially founded upon the natural integration of the affine atlas for domain manifold and polar forms used to define conventional spline surfaces over any planar domain. Due to their topological obstruction, general high genus surfaces admit neither a flat metric nor an affine atlas. Therefore, ideally the most natural spline solutions for high genus manifolds should not depend on the affine structure. In the future, we shall investigate different spline schemes which are not based on the affine structure of the underlying surface domain.

8. Acknowledgements

The authors would like to thank Junho Kim for his help on preparing Figures 10 and 11. Xianfeng Gu and Miao Jin are partially supported by NSF CCF-0448399, DMS-0528363, DMS-0626223, IIS-0713145, and NSF China 60628202. Ying He is partially supported by NTU/SUG 19/06. Hong Qin is partially supported by NSF IIS-0326388, IIS-0601168, and IIS-0710819.

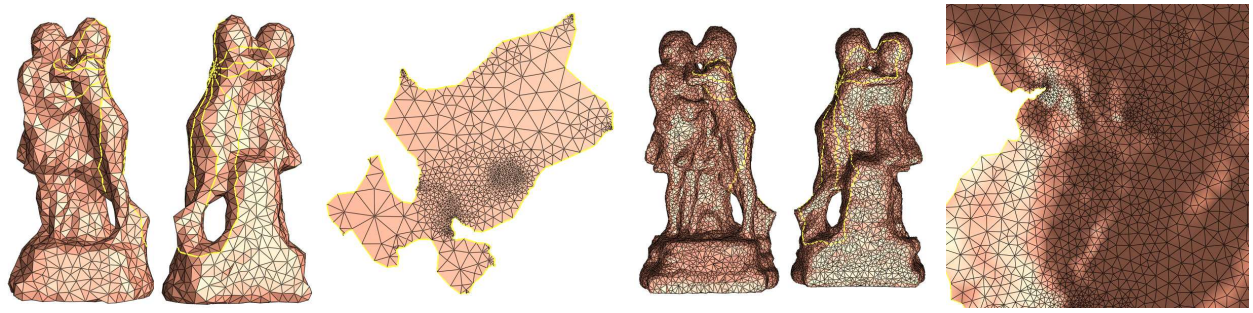


(a) original connectivity (b) Modified connectivity

Fig. 15. **Modify local connectivity around the extraordinary point.** (a) The extraordinary point and its one-ring neighbors are marked in red. (b) Modify local connectivity to satisfy the combinatorial constraint Equation (7).

References

[1] Alexander I. Bobenko, Boris A. Springborn. Variational principles for circle patterns and Koebe's theorem. Trans. Amer. Math. Soc. 356, 659-689, 2004.
[2] Christopher Carner, Miao Jin, Xianfeng Gu, Hong Qin. Topology-driven Surface Mappings with Robust Feature Alignment. Proceedings of IEEE Visualization, pp.543-550, 2005.
[3] Bennett Chow, Feng Luo. Combinatorial Ricci flows on surfaces. Journal of Differential Geometry, 63(1):97-129, 2003.
[4] W. Dahmen, C.A. Micchelli, H.-P. Seidel. Blossoming begets B-spline bases built bettery by B-patches. Mathematics of Computation, 59(199):97-115, 1992.
[5] M.S. Floater, K. Hormann. Surface Parameterization: a Tutorial and Survey. Advances in Multiresolution for Geometric Modelling, 157-186, 2005.
[6] Craig Gotsman, Xianfeng Gu, Alla Sheffer. Fundamentals of spherical parameterization for 3D meshes. ACM Trans. Graph., 22(3):358-363, 2003.
[7] Cindy Grimm, John F. Hughes. Modeling surfaces of arbitrary topology using manifolds. Proceedings of SIGGRAPH, 359-368, 1995.
[8] Xianfeng Gu, Steven J. Gortler, Hugues Hoppe. Geometry images. Proceedings of SIGGRAPH, pp.355-361, 2002.
[9] Xianfeng Gu, Shing-Tung Yau. Global conformal surface parameterization. In SGP. 127-137, 2003.



(a) Sculpture surface with 2000 faces. (b) The central chart of (a) (c) Sculpture surface with 10000 faces. (d) The close-up view of the central chart.

Fig. 16. **Affine atlas using Ricci Flow for a genus** 3 **surface.** The sculpture surface is of genus 3 with different resolutions, respectively, one is with $2,000$ faces as shown in (a), the other one is with $10,000$ faces as shown in (c). A vertex is selected as the singularity, and a set of canonical homology basis curves passing through the singularity are drawn as the yellow curves in (a) and (c), respectively. By using Ricci flow, flat metrics on the meshes are computed such that all curvature are concentrated on the singularities. The flattened fundamental domains are shown in (b) and (d), respectively.

[10] Xianfeng Gu, Yalin Wang, Tony F. Chan, P. M. Thompson, Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. IEEE Transactions on Medical Imaging, 23(8):945–958, 2004.

[11] Xianfeng Gu, Ying He, Hong Qin. Manifold splines. Graphical Models, Vol.68, No.3, pp.237-254, 2006.

[12] Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro, Michael Halle. Conformal Surface Parameterization for Texture Mapping. IEEE Trans. Vis. Comput. Graph., 6(2):181-189, 2000.

[13] Richard S. Hamilton. The Ricci flow on surfaces. Mathematics and general relativity (Santa Cruz, CA, 1986). Contemp. Math., 71, 237–262, 1998.

[14] Ying He, Hong Qin. Surface reconstruction with triangular *B*-splines. Proceedings of Geometric Modeling and Processing (GMP '04), 47–58, 2004.

[15] Ying He, Xianfeng Gu, Hong Qin. Rational spherical spline for genus zero shape modeling. Proceedings of IEEE International Conference on Shape Modeling and Applications (SMI '05), 82–91, 2005.

[16] Ying He, Miao Jin, Xianfeng Gu, Hong Qin. A C^1 globally interpolatory spline of arbitrary topology. LNCS, vol. 3752, pp. 295-306, 2005.

[17] Ying He, Xianfeng Gu, Hong Qin. Automatic shape control of triangular *B*-splines of arbitrary topology. Journal of Computer Science and Technology, Vol. 21, No. 2, pp. 232-237, 2006.

[18] Ying He, Kexiang Wang, Hongyu Wang, Xianfeng Gu, Hong Qin. Manifold T-spline. LNCS, vol. 4077, pp. 409-422, 2006.

[19] Michael Henle. A Combinatorial Intorduction to Topology, 1979.

[20] Hugues Hoppe. Efficient implementation of progressive meshes. Computers & Graphics, 22(1):27-36, 1998.

[21] Miao Jin, Feng Luo, Xianfeng Gu. Computing Surface Hyperbolic Structure and Real Projective Structure. ACM Symposium on Solid and Physics Modeling, 2006.

[22] Lilya Kharevych, Boris Springborn, Peter Schröder. Cone Singularities to the Rescue: Mitigating Area Distortion in Discrete Conformal Map. ACM SIGGRAPH/Eurographics Symposium on Geometry Processing, 2005.

[23] Andrei Khodakovsky, Nathan Litke, Peter Schröder. Globally smooth parameterizations with low distortion. ACM Trans. Graph., 22(3):350-357, 2003.

[24] Xinlai Ni, Michael Garland, John C. Hart. Fair morse functions for extracting the topological structure of a surface mesh. ACM Trans. Graph., 23(3):613-622, 2004.

[25] Grisha Perelman. The entropy formula for the Ricci flow and its geometric applications. arXiv.org, November 11, 2002.

[26] Grisha Perelman. Ricci flow with surgery on three-manifolds, arXiv.org, March 10, 2003.

[27] Grisha Perelman. Finite extinction time for the solutions to the Ricci flow on certain three-manifolds. arXiv.org, July 17, 2003.

[28] E. Praun, H. Hoppe. Spherical parameterization and remeshing. ACM Transaction on Graphics, 22(3):340–349, 2003.

[29] Nicolas Ray, Wan Chiu Li, Bruno Levy, Alla Sheffer, Pierre Alliez. Periodic Global Parameterization. ACM Transactions on Graphics, 2006.

[30] Burt Rodin, Dennis Sullivan. The convergence of circle packings to the Riemann mapping. Journal of Differential Geometry, 26(2):349-360, 1987.

[31] Hans-Peter Seidel. An Introduction to Polar Forms. IEEE Comput. Graph. Appl. 13(1):38-46, 1993.

[32] Alla Sheffer, Eric de Sturler, Parameterization of Faceted Surfaces for Meshing using Angle-Based Flattening. Eng. Comput. (Lond.) 17(3):326-337, 2001.

[33] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, Alexander Bogomyakov. ABF++: fast and robust angle based flattening. ACM Trans. Graph. 24(2):311-330, 2005.

[34] Alla Sheffer, Emil Praun, Kenneth Rose. Mesh Parameterization Methods and their Applications. Foundations and Trends® in Computer Graphics and Vision. 2(2), 2006.

[35] Kenneth Stephenson. Introduction to circle packing:the theory of discrete analytic functions. Cambridge University Press. 2005.

[36] Thurston, W. Hyperbolic geometry and 3-manifolds, Low-dimensional topology, London Math. Soc. Lecture Note Ser. 48: 9–25, 1982.

[37] Lexing Ying, Denis Zorin. A simple manifold-based construction of surfaces of arbitrary smoothness. ACM Trans. Graph. 23(3):271-275, 2004.

[38] Hongyu Wang, Ying He, Xin Li, Xianfeng Gu, Hong Qin. Polycube splines. Proceedings of ACM Symposium on Solid and Physical Modeling, pp.241-251, 2007.

Fig. 17. **Multiple singularities.** Within our framework, the user can also specify multiple singularities on the models. The horse model has five boundaries (four on the feet and one on the mouth) and two singularities. The camel model has five boundaries and one singularity. The target geodesic curvature of the boundary vertices is zero. The holes and singularities are filled using minimal surfaces.

(a) Parametric domain (b) The central chart (c) Manifold splines (d) Spline patchwork (e) Control point distribution

Fig. 18. **Examples of manifold triangular *B*-splines.** The affine atlas are computed using Ricci flow under free boundary condition. The transition function is a combination of translation and rotation. The red curves on the spline surfaces (see (d)) highlight the triangular patchwork.