Optimal Surface parameterization Using Inverse Curvature Map

Yongliang Yang[3], Junho Kim[1], Feng Luo[2], Shimin Hu[3], and Xianfeng Gu[1]
[1]Stony Brook University [2]Rutgers University [3]Tsinghua University

***Abstract*— Mesh parameterization is a fundamental technique in computer graphics. The major goals during mesh parameterization are to minimize both the angle distortion and the area distortion. *Angle* distortion can be eliminated by use of conformal mapping, in principle. Our paper focuses on solving the problem of finding the best discrete conformal mapping that also minimizes *area* distortion.**

Firstly, we deduce an exact analytical formula to represent area distortion by curvature change in the discrete conformal mapping, giving a dynamic Poisson equation. On a mesh, the vertex curvature is related to edge lengths by the curvature map. Our result shows the map is invertible, i.e. the edge lengths can be uniquely determined from prescribed curvatures under discrete conformal mappings. Furthermore, we give the explicit Jacobi matrix of the inverse curvature map.

Secondly, we formulate the task of computing conformal parameterizations with least area distortions as a constrained nonlinear optimization problem in curvature space. We deduce explicit conditions for the optima.

Thirdly, we give an energy form to measure the area distortions, and show it has a unique global minimum. We use this to design an efficient algorithm, called *free boundary curvature diffusion*, which is guaranteed to converge to the global minimum; it has a natural physical interpretation.

This result proves the common belief that optimal parameterization with least area distortion has a unique solution and can be achieved by free boundary conformal mapping.

Major theoretical results are deduced using symbolic computation, for which proofs are given as Maple source code. Practical algorithms are presented for optimal parameterization based on the inverse curvature map. Comparisons are conducted with existing methods and using different energies. Novel parameterization applications are also introduced. The theoretical framework of the inverse curvature map can be applied to further study discrete conformal mappings.

***Index Terms*— Mesh, Conformal Parameterization, Poisson, Metric, Curvature, Inverse map**

I. Introduction

Surface parameterization is the process of mapping a surface to a planar region, it has broad applications in graphics. Parameterizations introduce distortions between the original surface and its planar image, which can be separated into *angle distortion* and *area distortion* [1] and [2]. In theory, *angle* distortion can be eliminated completely by conformal mapping, but it is impossible for conformal mappings to further eliminate the *area* distortion completely.

For a given surface, we can define infinitely many different conformal mappings which have different area distortion, as shown in Fig. 1. The central problem of the optimal parameterization can be stated as follows:
How can we find the best conformal mapping that have the least area distortion?

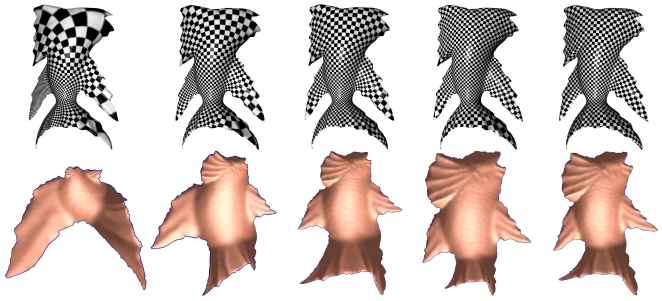


Fig. 1: There are an infinity number of conformal parameterizations of a given surface. We minimize the area distortion within the conformal mappings.
figure

In this paper, we present a set of theoretical tools as well as practical algorithms to tackle this problem.

A. Background

Parameterization methods have become a fundamental tool in graphics, and a significant amount of research has focused on it. Here, we only briefly overview the most related works and refer readers to [1] and [2] for wider surveys.

A common approach for parameterization is to minimize a certain energy to control the distortion. Levy et al. [3] defined an energy to approximate the Cauchy-Riemann equation; Desbrun et al. [4] optimize Dirichlet energy. Variations of harmonic energies are also optimized using discrete Laplace-Beltrami operators in [5]–[10]. More general energy forms can be found in [11]–[16]. Most linear methods apply a convex Dirichlet-type boundary. Virtual boundaries are applied in [17] and [16] to absorb distortions introduced by the convex boundary conditions. Alternatively, [3], [4] apply Neumann boundary conditions by only fixing a few vertices. Karni et al. [18] discuss the design of geometrically complex boundary conditions with constraints. Zayer et al. [19] applies discrete tensorial quasi-harmonic maps to improve the boundary and reduce the distortion.

One of the most prominent characteristics of conformal mapping is that it preserves angles. Angle based flattening method (ABF) [20] utilizes this property to produce high quality pseudo-conformal mappings. They derive the discrete conformal mapping by minimizing the ABF energy which is defined as differences

between the corner angles of faces on the original mesh and their images on the parameter plane. During the process the boundary evolves freely to further reduce the distortion. The method has been improved by applying efficient and robust nonlinear optimization in [21].

Another characteristic of conformal mapping is to map infinitesimal circles to infinitesimal circles and preserve their intersection angles. This inspired the circle packing method in [22]. Circle packings and circle patterns replace infinitesimal circles with finite circles. In the limit of refinement the continuous conformal maps are recovered [23]. Collins and Stephenson [24] have implemented circle packing in their software *CirclePack* which only considers combinatorics. The connection between circle packing and smooth surface Ricci flow [25] was discovered in [26]. The discrete Ricci flow method was introduced in [27] for hyperbolic parameterization.

Circle patterns based on the variational principle in Bobenko and Springborn [28] have been applied for parameterization in [29]. The method supports very flexible boundary conditions ranging from free boundaries to control of the boundary shape via prescribed curvatures. The method can also incorporate cone singularities to further reduce the distortion.

Our work differs from the previous work in the following aspects. To the best of our knowledge, our method is the only one with a rigorous proof that it can achieve a unique global optimum. Similar to [29], our method works for meshes with arbitrary topologies. Our method can compute parameterizations such that the target curvature and target area distortion satisfy some specific equations as shown in Fig. 8. Furthermore, our method can be applied for optimizing arbitrary energies with arbitrary constraints on curvatures and area distortions.

Springborn [30] shows that in theory, circle packing and circle patterns are equivalent. We choose to use circle packing to express our results as it simplifies the theoretical steps and algorithmic implementation. We prove our major theorem in the setting of circle pattern in Appendix B.

B. Overview

Most of the previous works minimize some energy forms which measure both angle distortion and area distortion. In this work, we take the approach similar to those in [4], [19] to separate these two criteria. As shown in Fig. 1, we only minimize the area distortion within the conformal mappings, which eliminate the angle distortions.

We address the angle distortion by using the discrete conformal mappings based on *circle packing*. The given mesh is covered by circles, each of which centered at a vertex as shown in Fig. 2. A circle centered at a vertex is tangent to or intersects with another circle centered at its neighbor vertex. We approximate the conformal mapping by varying the radii while preserving the intersection angles among the circles. (see Section II).

With circle packing, we can establish the mapping from the configuration of radii to the configuration of the curvatures, the so called *curvature map K*:

$$K : \{configuration\ of\ radii\} \rightarrow \{configuration\ of\ curvatures\}.$$

We show that the curvature map is bijective in the conformal mapping. We give an analytical formula for the *inverse curvature map* by explicitly computing its Jacobian, which is revealed as a dynamic Poisson equation (see Section II). Therefore, we can easily compute the radii from the prescribed curvatures.

Discrete conformal parameterization can be treated as finding a configuration of radii such that all curvatures are zeros except those at the boundaries and cone singularities. All curvature configurations corresponding to parameterizations form an affine subspace, we call it as the *admissible curvature space*. Area distortions can be measured by various energy forms defined on the configurations of radii. Optimal conformal parameterization is equivalent to minimizing the specific energy in the admissible curvature space, therefore, it is a nonlinear optimization problem with linear constraints.

Energies with good properties, such as differentiability, unique global minimum, simple forms of gradient and Hessian, are highly preferred in practice. We discovered an energy form that meets all the requirements (see Section III-E). Furthermore, a simple curvature flow algorithm with free boundary conditions is guaranteed to converge to the global minimum.

The pipeline of optimal parameterization system is as follows.
1. Mesh preparation (Section III-A)
2. Computing the initial circle packing metric (Section III-B)
3. Selecting the singular vertex set (Section III-C)
4. Compute the optimal circle packing metric (Section III-D)
5. Isometric embedding (Section III-F)

The theoretical results of inverse curvature map are explained in Section II. Each step of the algorithm pipeline Fig. 3 is elucidated in Section III. The experimental results are demonstrated in Section IV. We conclude our work and point out the future direction in Section V. Detailed theoretical proof in circle packing setting using symbolic computation is in the Appendix A, in circle pattern setting is in the Appendix B.

II. Inverse Discrete Curvature Map

In this section, we introduce the inverse curvature map, which is the key ingredient of our optimal parameterization.

The discussion is based on general triangular meshes with arbitrary topologies. We denote a mesh by $M = \{V, E, F\}$. A vertex, an edge and a face are denoted as v_i, $[v_i, v_j]$, $[v_i, v_j, v_k]$, respectively. A mesh M embedded in $\mathbf{R}^3$ has a naturally induced Euclidean metric, which is determined by each edge length. The vertex curvatures are defined as follows. For an interior vertex, the curvature equals 2π minus the sum of angles between edges at the vertex, whereas for a boundary vertex, it is π minus this sum. The discrete Gauss-Bonnet theorem states that the total curvature is $2\pi\chi(M)$, where $\chi(M)$ is the Euler number of the mesh.

A. Circle Packing Metric

Given a triangular mesh, we associate each vertex v_i with a circle with radius γ_i. On edge e_{ij}, the two circles intersect at the angle of ϕ_{ij}, as shown in Fig. 2.

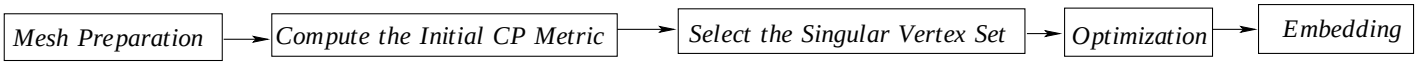


Fig. 3: Algorithm pipeline.

figure

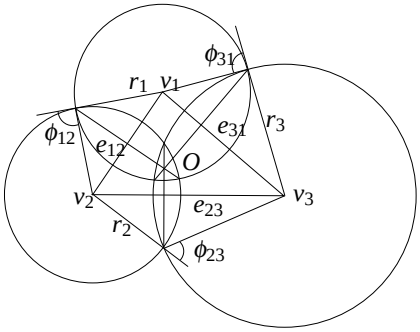


Fig. 2: Circle packing metric.

figure

Definition 2.1 (Circle Packing Metric): A circle packing metric for a mesh M is defined as (M,Γ,Φ), where M represents the triangulation, $\Gamma : V \to \mathbb{R}^+$ is the circle radius function, $\Phi : E \to [0, \frac{\pi}{2}]$ is the edge angle function. The discrete metric is determined by $l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\phi_{ij}}$.

Now, the mesh edge lengths can be determined by the circle radii and the intersection angles with the cosine laws, depicted in Fig. 2. Since the edge lengths determine the angles on each face, the circle radii determine the vertex curvatures. We designate the mapping from the configuration of radii to the configuration of the vertex curvatures as the *curvature map.*

B. Inverse Curvature Map

Two circle packing metrics of the same mesh M, $\{M,\Gamma_1,\Phi_1\}$ and $\{M,\Gamma_2,\Phi_2\}$, are *conformal* to each other, if Φ_1 equals Φ_2. Each conformal equivalence class of circle packing metrics forms a space which we call a *conformal discrete metric space*, denoted by U. Upon fixing the edge angles ϕ_{ij}, a discrete circle packing metric can be represented by a vector $\mathbf{u} = (u_1, \cdots, u_n)$, where $u_i = \log\gamma_i$, $u_i \in (-\infty, +\infty)$, and n is the number of vertices. Each conformal discrete metric space is homeomorphic to $\mathbb{R}^n$. Because scaling does not affect the curvature, we normalize the conformal metrics by requiring $\sum_i u_i = 0$, which is a hyper-plane in the $\mathbb{R}^n$ which we denote Π_u. The discrete curvature K maps each $\mathbf{u}$ to a curvature function $\mathbf{k} = (k_1, k_2, \cdots, k_n)$, and the image of $\Omega_k := K(\Pi_u)$ is a convex polytope [26].

The curvature map K from the conformal metric space to the curvature space $K : \Pi_u \to \Omega_k$ is bijective; both the map and the inverse map have an infinite degree of smoothness. Furthermore, the curvature map is real and analytic (so it can be represented as the summation of an infinite series.)

Theorem 2.2 (Inverse Curvature Map): The curvature map K from a conformal class of circle packing metrics Π_u to the curvature space Ω_k is a C^∞ diffeomorphism, furthermore, it is real and analytic.

The derivative map $dK : T\Pi_u(\mathbf{u}) \to T\Omega_k(\mathbf{k})$, satisfies the discrete Poisson equation,

$$d\mathbf{k} = \Delta(\mathbf{u})d\mathbf{u}, \tag{1}$$

where $T\Pi_u(\mathbf{u})$ is the tangent space of Π_u at the point $\mathbf{u}$, $T\Omega_k(\mathbf{k})$ is the tangent space of Ω_k at the point $\mathbf{k}$, and $\Delta(\mathbf{u})$ is a positive definite matrix when restricted to $T\Pi_u(\mathbf{u})$.

Therefore, the curvature map and the inverse curvature map can be represented as

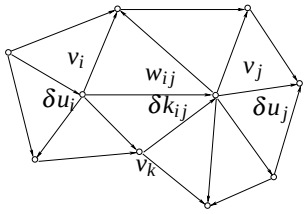
$$\mathbf{k}_1 - \mathbf{k}_0 = \int_{\mathbf{u}_0}^{\mathbf{u}_1} \Delta(\mu)d\mu, \quad \mathbf{u}_1 - \mathbf{u}_0 = \int_{\mathbf{k}_0}^{\mathbf{k}_1} \Delta(\xi)^{-1}d\xi. \tag{2}$$

A detailed proof can be found in the Appendix. Here we give an intuitive picture using a *differential network flow model* as shown in Fig. 4. We treat the mesh as a network. Curvature flows along the edges when vertex radii change. Suppose v_i and v_j are two adjacent vertices, that the logarithms of the radii change by δu_i and δu_j respectively, and the conductivity (weight) for the edge is $w_{ij} > 0$, which depends on the current vertex radii.

Then the curvature flux from v_i to v_j along the edge is $\delta k_{ij} = w_{ij}(\delta u_j - \delta u_i)$. Each vertex has several edges connected to it, so the net edge curvature flux equals the overall curvature change at the vertex, $\delta k_j = \sum_i \delta k_{ij}$. Therefore, the Laplace matrix has an explicit form: $\Delta = (d_{ij})$,

$$d_{ij} = \begin{cases} -w_{ij} & i \neq j, [v_i, v_j] \in E \\ \sum_k w_{ik} & i = j \\ 0 & i \neq j, [v_i, v_j] \notin E \end{cases} \tag{3}$$

We now explain the geometric meaning of the edge weight. On each edge, two circles intersect and share a common chord (or a common tangent). For each face, three common chords intersect at one point, *the center of the face* as shown in figure 2. Then the weight for a halfedge equals the ratio of the distance from the center to the halfedge and the current length of the halfedge. The edge weight is the sum of those of its halfedge weights. The edge weight depends on the current curvature (or, equivalently the radii), and therefore the Laplace-Beltrami operator is dynamic. This fact makes the whole theory more complicated.



$$\delta k_j = \sum_i \delta k_{ij} \qquad \delta k_{ij} = w_{ij}(du_j - du_i)$$

Fig. 4: Differential network curvature flow model. The curvature flux along edge δk_{ij} is driven by the gradient of $\delta\mathbf{u}$. The change of curvature at a vertex δk_i equals the divergence of the curvature flow.

figure

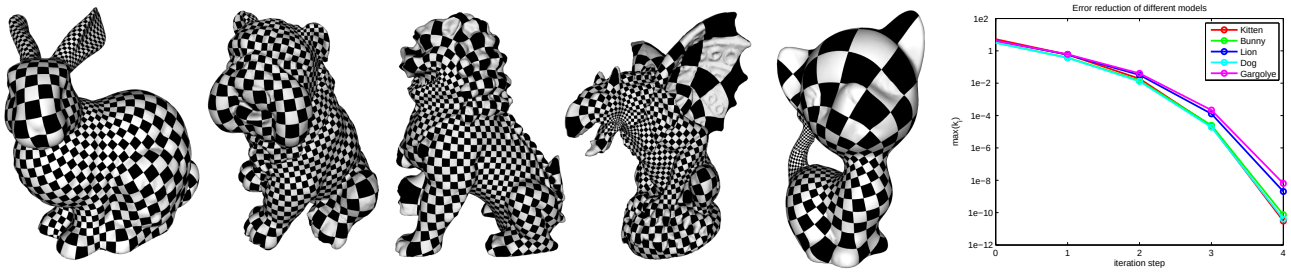


Fig. 5: Discrete conformal parameterizations using *Inverse Curvature Map*. ([#v, #f, execution time (sec)] are (20660, 41118, 6.3), (29923, 59417, 11.2), (20224, 40118, 5.8), (20618, 40803, 6.6), and (10219, 20438, 2.5), respectively with Pentium4 2.8GHz with 2GB memory).
figure

The algorithm for computing the discrete conformal metric $\bar{\mathbf{u}}$ for the prescribed curvature $\bar{\mathbf{k}}$ is as follows: Algorithm 1 can be

Algorithm 1 Inverse Curvature Map $\bar{\mathbf{u}} = K^{-1}(\bar{\mathbf{k}})$

Compute the initial circle packing metric $\{M, \Gamma, \Phi\}$
Compute initial curvature $\mathbf{k}$
$\mathbf{u} \Leftarrow \mathbf{u}_0$, $\mathbf{u}_0$ is the initial circle packing metric.
while $|\bar{\mathbf{k}} - \mathbf{k}| > \varepsilon$ **do**
  Compute $w_{ij}(\mathbf{u})$ to form the Laplace matrix $\Delta(\mathbf{u})$.
  $d\mathbf{u} \Leftarrow \Delta(\mathbf{u})^{-1}(\bar{\mathbf{k}} - \mathbf{k})$
  $\mathbf{u} \Leftarrow \mathbf{u} + d\mathbf{u}$
  $\mathbf{k} \Leftarrow K(\mathbf{u})$
end while
$\bar{\mathbf{u}} \Leftarrow \mathbf{u}$

applied for conformally parameterizing general meshes directly. Fig. 5 demonstrates some parameterization results using this algorithm.

C. Relation with Discrete Ricci Flow

Inverse curvature map can be also deduced from the theory of discrete Ricci flow [26]. The fact that curvature map $K : \mathbf{u} \to \mathbf{k}$ is invertible is proven in the following way.

Suppose $\mathbf{u}_0$ is the initial metric with the curvature $\mathbf{k}_0$, suppose $\bar{\mathbf{k}}$ is the prescribed curvature, the corresponding metric is $\bar{\mathbf{u}}$ then we can define the following *discrete Ricci energy*

$$E_{Ric}(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} (\bar{\mathbf{k}} - \mathbf{k})^T d\mu. \tag{4}$$

From discrete Ricci flow theory, the $E(\mathbf{u})$ is convex in the sub-affine space $\sum u_i = 0$, and $\bar{\mathbf{u}}$ is the unique global minimal point. The target metric $\bar{\mathbf{u}}$ can be obtained by minimizing the energy, the steepest descent method

$$\frac{d\mathbf{u}}{dt} = -\nabla E_{Ric}(\mathbf{u}) = -(\bar{\mathbf{k}} - \mathbf{k}),$$

is the discrete Ricci flow. Therefore, the curvature map K is invertible.

In order to compute the optimal parameterization, we need an explicit form of the Jacobi matrix of the curvature map, which is related to the Hessian matrix of $E_{Ric}(\mathbf{u})$. This is key for nonlinear optimization algorithm.

III. Optimal Surface Parameterization

This section explains the algorithm pipeline for the optimization system as shown in Fig. 3, each subsection corresponds to one step respectively.

A. Mesh Preparation

In practice, the initial circle packing metric requires all the edge angles to be acute, so that the Jacobi matrix of the curvature map is positive definite (equivalently, the Ricci energy Eq. (4) is convex). In order to meet this requirement, we remesh the input models using the algorithms described in [31] and [32] for getting the faces close to equilateral triangles.

B. Computing the initial Circle Packing Metric

Then, we use a simple method for the initial circle packing metric, described in Algorithm 2.

Algorithm 2 Compute the initial Circle Packing Metric

for all faces $f \in F$ **do**
  for all corners $c_i \in f$ **do**
    $\gamma(c_i) \leftarrow \frac{l_{i+1} + l_{i+2} - l_i}{2}$
  end for
end for
for all vertex $v_j \in V$ **do**
  $\gamma_j \leftarrow max\{\gamma(c) | c \text{ is attached to } v_j\}$
end for
for all edges $e = [v_i, v_j] \in E$ **do**
  $\phi_{ij} = \min\{\cos^{-1} \frac{l_{ij}^2 - \gamma_i^2 - \gamma_j^2}{2\gamma_i\gamma_j}, \frac{\pi}{2}\}$, where l_{ij} is the Euclidean edge length
end for

In experiments, this algorithm guarantees to get acute corner angles, and the initial circle packing metrics are very closed to the induced Euclidean metric of the mesh.

C. Selecting Singular Vertex Set

In order to reduce the area distortion, it is very helpful to concentrate curvatures on a subset of vertices, we call them a *singular vertex set*. For example, in general, if a mesh has

boundaries, all of the boundary vertices are in the singular vertex set.

In step 4 of our algorithm pipeline Fig. 3, we select the singular vertex set using algorithm 3, which is similar to the one used in the circle pattern work [29]. Intuitively, we let the curvature be uniformly distributed on all vertices, then we measure the area distortion, and pick the critical points of the area distortion function. The algorithm is described in Algorithm 3. Fig. 6 demonstrates the algorithm using the Stanford Bunny model.

Algorithm 3 Compute the Singular Vertex Set

$\bar{k_i} \leftarrow 0, v_i \in \partial M$,
$\bar{k_i} \leftarrow \frac{2\pi\chi(M)}{n}, v_i \notin \partial M$, n is number of interior vertices
$\bar{u} \leftarrow K^{-1}(\bar{\mathbf{k}})$ using algorithm 1
$S \Leftarrow \partial M \cup \{\text{local minima of } \bar{\mathbf{u}}\}$

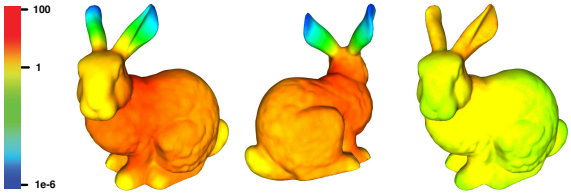


Fig. 6: Singularity selection process: (a) and (b) depict the area distortion of the bunny model without any singular vertex. We select critical points in ear tips and the point between the roots of the ears. (c) shows the area distortion after computing the inverse curvature map with the selected singular vertices. The uniformity of the area distortion is greatly improved.
figure

D. Compute the optimal Circle Packing Metric

In the following, we explain the process to compute the optimal circle packing metrics in step 4 in the algorithm pipeline in Fig. 3.

There are two methods for the optimization, the projected gradient method described in Section III-D.1 and the free boundary curvature diffusion method described in Section III-E.

1) Projected Gradient Method for Optimization: We can define various energy forms to measure the area distortion. The energy can be defined either in the conformal metric space or in the curvature space. In terms of computational complexity, two methods are equivalent. We decide to define the energy in curvature space, the main reason is that the curvature space is a convex affine subspace, if there energy is convex, then there exists only one optimum, it is easy to handle both theoretically and practically.

The possible solutions must be a valid parameterization, namely, all curvatures are zeros except the singularities.

Definition 3.1 (Admissible curvature space): Given a mesh M, the vertices are divided to two sets S and N. S represents the singular vertices, N represents regular vertices. The admissible curvature space is an affine subspace defined as the intersection of the following hyper-planes

$$\Pi_k := \bigcap_{v_i \in N} \{k_i = 0\} \bigcap \{\sum_{v_j \in S} k_j = 2\pi\chi(M)\} \bigcap \Omega_k \qquad (5)$$

Then the optimal parameterization problem is equivalent to optimizing some energy form $E(\mathbf{k})$ in the admissible curvature space.

Optimal Parameterization Problem: Compute the minima of the energy $E(\mathbf{k})$ in the admissible curvature space Π_k:

$$min_{\mathbf{k}} E(\mathbf{k}), s.t. \mathbf{k} \in \Pi_k.$$

Basically, we compute the gradient of $E(\mathbf{k})$ w.r.t $\mathbf{k}$, denoted as $\nabla_k E$, and project the gradient to the affine subspace Π_k, and update $\mathbf{k}$ along the projected gradient direction, until the projected gradient equals to zero. Namely, at a critical point of $E(\mathbf{k})$, $\nabla_k E$ is orthogonal to the admissible curvature space. This procedure can find one local minimum and does not guarantee we find all the minima, neither the global minimum. In next subsection, we design a special energy with unique minimum, therefore this algorithm can reach the global minimum.

Algorithm 4 Optimal Discrete Conformal Parameterization

Randomly select a $\mathbf{k} \in \Pi_k$
repeat
  $\mathbf{u} \leftarrow K^{-1}(\mathbf{k})$
  Compute the gradient $\nabla_u E$
  $\nabla_k E \Leftarrow \Delta(\mathbf{u})^{-1} \nabla_u E$
  for all $v_i \in N$ **do**
    $\nabla_k E \Leftarrow \nabla_k E - < \nabla_k E, \mathbf{e}_i > \mathbf{e}_i$
  end for
  $\nabla_k E \leftarrow \nabla_k E - < \nabla_k E, \mathbf{d} > \frac{\mathbf{d}}{|\mathbf{d}|^2}$
  $\mathbf{k} \leftarrow \mathbf{k} - \lambda \nabla_k E$
until $|\nabla_k E| < \varepsilon$

where $\mathbf{d}$ is a vector, where $d_i = 0$ for normal vertices and $d_j = 1$ for singular vertices.

Theorem 3.2: Suppose $\mathbf{k}$ is an interior point of Π_k, also an optimum for a energy form $E(\mathbf{u})$, then all the components of $\nabla_k E$ corresponding to the singular vertices are equal.

Proof If $\mathbf{k}$ is an optima of $E(\mathbf{k})$, then $\nabla_k E \perp \Pi_k$. Suppose v_i is a normal vertex, the normal to the hyperplane $\{k_i = 0\}$ is $\mathbf{e}_i$, the normal to the plane $\{\sum_{v_j \in S} k_j = 2\pi\chi(M)\}$ is $\mathbf{d}$, where $d_i = 0$ for normal vertices v_i and $d_j = 1$ for singular vertices. Therefore

$$\nabla_k E = \sum_{v_i \in N} \lambda_i \mathbf{e}_i + \mu \mathbf{d},$$

where μ is a real number. □

The common energy forms used in the literature are:

1) Angle Based Flattening energy defined in [20]: this energy measures the differences between the original and the target angles at all the corners, (a corner is determined by a face and one vertex adjacent to it).

$$E_{ABF}(\mathbf{k}) = \sum_{\theta} (\theta(\mathbf{k}) - \theta(\mathbf{k}_0))^2,$$

where $\theta(\mathbf{k}_0)$'s are the original corner angles.
2) Area distortion energy defined in [4]: this energy measure the ratio between the original face area and the face area on the parameter plane,

$$E_{AD}(\mathbf{k}) = \sum_{f} (\frac{s_f(\mathbf{k})}{s_f(\mathbf{k}_0)} - 1)^2, \qquad (6)$$

where $s_f(\mathbf{k}_0), s_f(\mathbf{k})$ are the areas of the face f under the original metric and the target metric. This energy is the most direct measurement for area distortion.

3) u-square energy: this energy is just the norm of $\mathbf{u}$, which are the logarithms of the change of circle radii, because the mean of $\mathbf{u}$ is zero, this energy can be treated as the variance of $\mathbf{u}$, which is a measurement of the uniformity of u_i.

$$E_{u^2}(\mathbf{k}) = |\mathbf{u}(\mathbf{k}) - \mathbf{u}(\mathbf{k}_0)|^2, \tag{7}$$

where $u(\mathbf{k}_0)$ is the initial circle packing metric.

The gradient of a energy E w.r.t. $\mathbf{u}$ is $\nabla_u E$, which is related to $\nabla_k E$ by

$$\nabla_k E = \Delta^{-1}(\mathbf{u})\nabla_u E.$$

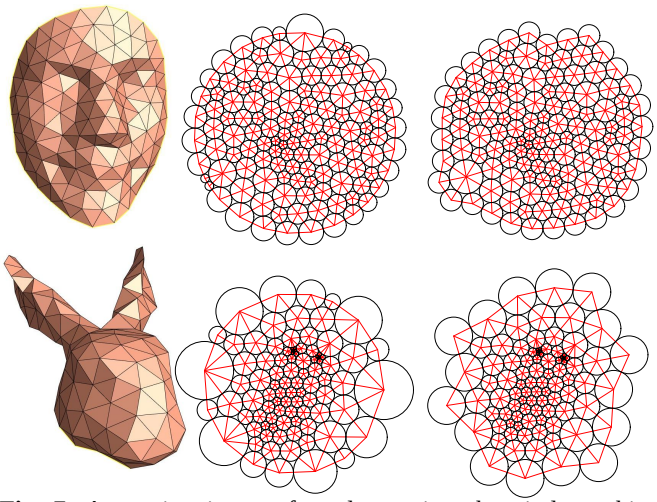


Fig. 7: Approximating conformal mappings by circle packing. The circle radii are changed while tangency relations are preserved. The second column shows a circular boundary condition; the third column shows a free boundary condition. For the free boundary condition, all circle radii on the boundary vertices are equal.
figure

E. Curvature diffusion with free boundary conditions

The energies introduced in the above are not satisfactory in practice. Firstly, it is unclear whether they have unique global minimum; secondly, their gradient has complicated form, therefore it is expensive to compute using the projected gradient method.

In this part, we introduce a novel energy, *curvature entropy*, which overcomes the shortcomings of other energies. It has a unique global minimum; its gradient has the simplest form $\mathbf{u}$; it can be computed efficiently with methods other than projected gradient method.

Definition 3.3 (Curvature Entropy Energy): The entropy energy is

$$E_{EN}(\mathbf{k}) = \int_{\mathbf{k}_0}^{\mathbf{k}} (\mathbf{u} - \mathbf{u}_0)^T d\mathbf{k}.$$

The curvature entropy energy is the Legendre dual of discrete Ricci energy

$$E_{Ric}(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} (\mathbf{k} - \mathbf{k}_0)^T d\mathbf{u}.$$

The curvature entropy energy is closely related to the u-square energy defined in 7,

$$E_{u^2}(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} (\mathbf{u} - \mathbf{u}_0)^T Id\mathbf{u},$$

where I is the identity matrix. If we replace I by $\Delta(\mathbf{u})^{-1}$ in the above formulae, we will get the curvature entropy energy

$$E_{EN}(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} (\mathbf{u} - \mathbf{u}_0)^T \Delta(\mathbf{u})^{-1} d\mathbf{u} = \int_{\mathbf{k}_0}^{\mathbf{k}} (\mathbf{u} - \mathbf{u}_0)^T d\mathbf{k}.$$

Both $\Delta^{-1}(\mathbf{u})$ and I are positive definite. Therefore u-square energy and curvature entropy energy are equivalent for the purpose of measuring the uniformity of the area distortion $\mathbf{u}$.

This energy can be directly optimized using the projected gradient method. The gradient of the curvature entropy is very simple,

$$\nabla_{\mathbf{k}} E_{EN}(\mathbf{u}) = \mathbf{u} - \mathbf{u}_0.$$

It can also be minimized by the following *curvature diffusion* method.

Algorithm 5 Curvature Diffusion with Free Boundaries

```
while max_{v_i ∈ N} |k_i| > ε do
  for all v_i ∈ N do
    du_i ⇐ −k_i
    u_i ← u_i + λ du_i
  end for
  c ← (Σ_{v_i ∈ V} u_i) / |V|
  for all vertex v_j ∈ V do
    u_i ← u_i − c
  end for
end while
```

Intuitively, the algorithm sets $\frac{d\mathbf{u}}{dt} = -\mathbf{k}$, then according to the Eq. (1), the curvature will evolve like a heat diffusion, $\frac{d\mathbf{k}}{dt} = -\Delta\mathbf{k}$. The singular vertices absorb all the curvature flux, and the whole surface deforms to be flat in the most natural way. Because in the heat diffusion process, the entropy increases, we name this energy as curvature entropy.

We show that when the algorithm terminates, the deformation on the singular vertices are uniform.

Lemma 3.4: Suppose $\mathbf{u}$ is the solution of the free boundary curvature diffusion algorithm, then $u_i - u_i^0 \equiv const, \forall v_i \in S$.

Proof For all boundary vertices, at the beginning $u_i - u_i^0$'s are zeros. At each normalization step, $u_i - u_i^0$ change by the same amount. Therefore, $u_i - u_i^0$'s are always equal. □
The third column in Fig. 7 demonstrates this fact that all of the circle radii of boundary vertices are equal (this is because the radii for all vertices in the initial circle packing metric are equal). This result is consistent to theorem 3.2, the gradient of the curvature entropy is $\mathbf{u}$, and this algorithm leads to a solution where all u_i's are equal on the singular vertices. Therefore, this algorithm minimizes the entropy in a different approach.

Theorem 3.5: The curvature entropy energy is well defined (namely, the value is independent of the choice of the integration path)and has unique global minima point in admissible curvature

space. The free boundary optimization algorithm leads to the global minima.

Proof In order to show the energy is well defined, we need to show the 1-form $\sum(u_i - u_i^0)dk_i$ is closed. Namely, the matrix $(\frac{du_i}{dk_j})$ is symmetric. Δ is symmetric, therefore $\Delta^{-1} = (\frac{du_i}{dk_j})$ is symmetric also.

We can directly compute the gradient of $E_{EN}(\mathbf{k})$, $\nabla_k E_{EN}(\mathbf{k}) = \mathbf{u} - \mathbf{u}_0$. The necessary condition of the optima point is $u_i - u_i^0 = const$ for all singular vertices.

We further compute the Hessian matrix of E_{EN}, which is directly $\Delta^{-1}(\mathbf{k})$. Because Δ is positive definite, therefore Δ^{-1} is positive definite. Therefore E_{EN} is a convex energy. On the other hand, the admissible space of $\mathbf{k}$ is a convex affine space, the energy has unique global minima on it.

The free boundary optimization algorithm can reach one critical point of E_{EN} and E_{EN} has only one unique critical point, therefore, the free boundary curvature optimization algorithm can reach the global minima. □

Comparing to the projected gradient algorithm Section III-D.1, the curvature diffusion algorithm doesn't need to solve the Poisson equation, it is simple and direct, and easy to implement. In the algorithm pipeline Fig. 3, *step 4* can apply the curvature diffusion algorithm directly.

F. Embedding

The final step in our algorithm pipeline Fig. 3 is to isometrically embed the mesh on the plane using the circle packing metric obtained from the optimization.

We first compute a cut on the mesh to slice the mesh to an open topological disk. Several algorithms [33], [34] can be applied directly. Then we embed the open mesh isometrically onto the plane using the optimal circle packing metric. For meshes with less than $30k$ faces, we select a face near to the center of the mesh as the root face and directly embed it, then flatten the faces adjacent to it. We propagate the embedding face by face until we have flattened the whole mesh. For large scale meshes, the propagated errors accumulate, so instead, we use a method similar to [21]. Basically, we form a linear system using the final circling packing metric, and approximate the parameter positions for each vertex in the least square sense. The result was always found to be a valid embedding for all of our experiments.

IV. Implementations and Experimental Results

In this section, we give the experimental results of our algorithms and compare our methods with the state-of-the-art techniques including LSCM [3], ABF++ [21], and circle patterns [29]. For fair comparison, we tested the previous methods with the codes which are available in the websites of the original authors.

Table I summarize the statistics of our experiments on several models as shown in Fig. 9, Fig. 10, Fig. 11. The angle distortion and area distortion are judged by various energies respectively. We measure the angle distortions with the three different measurements; conformality [3], L2 shear [21], and squared sum of angle differences [21]. As shown in Table I, all of the methods minimize the angle distortion and hold the conformality well.

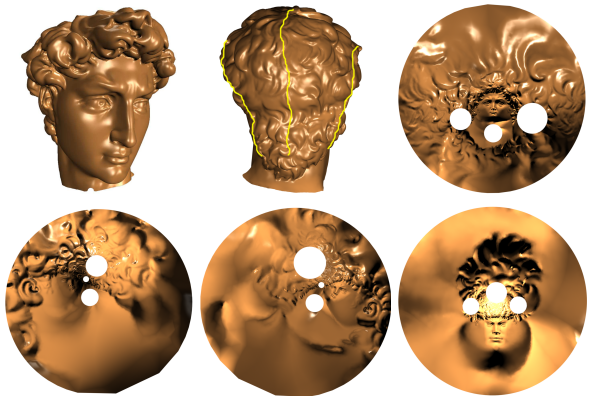


Fig. 8: Mesh parameterization with the inverse curvature map for a topologically complicated model: The David head model to have four boundaries, as shown in (a) and (b). Four different configurations are depicted in (c)-(f), each of which has one outer boundary and three inner boundaries in the parametric space. For each case, the parameterization is obtained by using the inverse curvature map by specifying the sum of target curvatures for an outer boundary as 2π and the sum of target curvatures for each inner boundary as -2π and $k_i e^{-u_i}$ is constant for boundary vertices.
figure

The area distortions are measured with two different approaches. Our optimization approaches provide the best results for the complicated models, such as the horse and camel. ABF++ gives the small area distortions in many cases, however it can fall into the local minimum in the case of a complicated model (see Fig. 11). The circle patterns provide comparable good results with all tested models.

$$\text{Log area distortion} = \sum_f \left(\log \frac{s_f(\mathbf{k})}{s_f(\mathbf{k}_0)} \right)^2$$

For both Algorithm 1 and Algorithm 4, the efficiency is greatly dependent on solving the Poisson equation $d\mathbf{k} = \Delta(\mathbf{u})d\mathbf{u}$. This is a linear sparse system whose rank is determined by the number of mesh vertices. As the Laplace matrix Δ is positive definite when constrained on $T\Pi_u(\mathbf{u})$, we use conjugate gradient method to solve the linear system, which is time and space efficient. As shown in Fig. 5, our parameterization with inverse curvature map is comparable to those of ABF++ and circle pattern methods. The error deduction is very fast as in the convergence chart. Most models can be parameterized within 4 steps in Algorithm 1.

For optimal parameterization, the free boundary curvature diffusion method (Algorithm 5) is much more efficient than the projected gradient descent type method (Algorithm 4). This is because the latter need to compute inverse curvature map for each iteration step, this is very time consuming when the energy function is tiny deceased near the minima. The curvature diffusion method can get to the minima much more directly. Take the 30k vertices horse model (Fig.11) as example: the ICM entropy method cost several minutes, whereas the curvature diffusion

method only takes 21s to get the comparable result. For other energy (ABF, Area Distortion U2), the gradient form is more complicated than curvature entropy energy, thus cost more time. Experiments show that the optimizations of curvature entropy energy and ABF energy lead to parameterizations with higher qualities than other energies. The area distortion energy Eq. (5) has the local minima.

V. Conclusion and Future Work

In this work, we introduce a set of rigorous theoretical tools and practical algorithms to solve the optimal conformal paramaterization problem.

Inverse curvature map represents the exact analytical relation between area distortion and curvature as a dynamic Poisson system. This enables us to find the conformal parameterization with the least area distortion using nonlinear optimization techniques with linear constraints. The explicit conditions for the optima are deduced from the variational principle. A special energy form to measure the area distortion is investigated, the so-called curvature entropy, which has a unique global minimum and can be optimized using curvature diffusion algorithm with free boundaries. Our experiments on complicated meshes support our theoretical discoveries.

The inverse curvature map theorem is deduced for meshes with Euclidean geometry, namely, the mesh is formed by gluing Euclidean triangles. We believe the inverse curvature map holds for meshes with hyperbolic and spherical geometry, and leads to novel hyperbolic and spherical parameterization algorithms. Hyperbolic and spherical parameterizations play important roles for shape analysis and geometric modeling.

Although we solved the optimal parametrization problem for given singular vertex set, it remains challenging problem to determine the optimal singular vertex set. The common belief for choosing singular vertices is to pick the critical points of the area distortion function, as our algorithm Section III-C. In the future, we will apply our theoretic tools to continue the exploration along this direction.

Acknowledgements

The authors are grateful to Professor Tom Sederberg and Professor Ralph Martin for stimulating discussion. Special thanks to Professor Zhongxiao Jia in Tsinghua for his insightful feedback of the numerical computing problems. The authors thank Yukun Lai and Qianyi Zhou for their help for remeshing. The models used in this paper are the courtesy of Stanford University and AIM@SHAPE shape repository. This work is partially supported by the National Science Foundation CCF-0448339, DMS-0626223, DMS-0528363, National Basic Research Project of China (Project Number 2006CB303102) and the Natural Science Foundation of China (Project Number 60673004, 60628202).

Appendix A: Proof of Inverse Curvature Map Theorem

In this appendix, we give a detailed proof of the main theorem in this paper. The proof is based on analytic geometry. We have used Maple to derive various formulae. In order to save space, we only give the Maple source code and omit the calculation results returned by Maple.

Lemma 5.1: For a Euclidean triangle $[v_1, v_2, v_3]$ with a circle packing metric with radii $\gamma_1, \gamma_2, \gamma_3$, and intersection angles $\phi_{12}, \phi_{23}, \phi_{31}$, the three common chords intersect in one point O, which we call the center.

Proof We first compute the three edge lengths, l_i, l_j, l_k are the edge lengths opposite to vertices v_i, v_j, v_k respectively:

```
> l_k:= sqrt(r_i^2+r_j^2+2*r_i*r_j*phi_ij);
> l_i:= sqrt(r_j^2+r_k^2+2*r_j*r_k*phi_jk);
> l_j:= sqrt(r_k^2+r_i^2+2*r_k*r_i*phi_ki);
```

We next compute the inner angle at v_i, θ_i

```
>theta_i:= arccos((l_j^2+l_k^2-l_i^2)/2*l_j*l_k);
>theta_j:= arccos((l_k^2+l_i^2-l_j^2)/2*l_k*l_i);
>theta_k:= arccos((l_i^2+l_j^2-l_k^2)/2*l_i*l_j);
```

Let the coordinates of the three vertices v_i, v_j, v_k be $(x_i, y_i), (x_j, y_j)$ and (x_k, y_k),

```
>x_i :=0   ; y_i:=0;
>x_j :=l_k ; y_j:=0;
>x_k :=lj*cos(theta_i); y_k:=l_j*sin(theta_i);
```

The three circles are

```
>c_i:= (x-x_i)^2+(y-y_i)^2-r_i^2;
>c_j:= (x-x_j)^2+(y-y_j)^2-r_j^2;
>c_k:= (x-x_k)^2+(y-y_k)^2-r_k^2;
```

The equations of the three common chords are

```
>s_ij:= simplify(c_i-c_j);
>s_jk:= simplify(c_j-c_k);
>s_ki:= simplify(c_k-c_i);
```

We first compute the intersection point (x_o, y_o) of the common chords s_{ij} and s_{ki},

```
>x_o:=solve(s_ij,x);
>y_o:=solve(subs({x=xo},s_ik),y);
```

Finally, we verify (x_o, y_o) is also on s_{jk},

```
>simplify(subs({x=x_o,y=y_o},s_jk));
```

The result is zero, which means the three common chords intersect at one point. □

Lemma 5.2: For a triangle with the circle packing metric, the following equations hold:

$$\frac{\partial \theta_i}{\partial u_j} = \frac{h_k}{l_k} \tag{8}$$

where $u_j = \log \gamma_j$, and h_k is the distance from the center O to the edge e_k.

Proof We continue the symbolic computation from the proof of the last lemma. Verification is straightforward: $\partial \theta_i / \partial u_j = \partial \theta_i / \partial \gamma_j \gamma_j$, and h equals y_o from the above proof:

```
> simplify( diff(theta_i,r_j)*r_j - y_o );
```

The result is zero, so $\partial \theta_i / \partial u_j$ equals the ratio of the distance from the center to the edge, to the edge length. □

Lemma 5.3: For a triangle with the circle packing metric, the following equations hold:

$$\frac{\partial \theta_i}{\partial u_j} = \frac{\partial \theta_j}{\partial u_i}. \tag{9}$$

Model	Methods	conformality	Angle distortion	L2 shear	L2 stretch	log area distortion
camel	LSCM	0.00002	0.00203	0.0511	185.8090	11.84310
#v: 20775	ABF++	0.00015	0.00135	0.0453	5.1039	0.74653
#f: 40384	CirclePatterns	0.00008	0.00145	0.0472	8.5819	0.70627
	Free Boundary	0.00032	0.00397	0.0699	5.1496	0.68069
	Optimize U2	0.00125	0.00374	0.0691	5.4908	1.92130
	Optimize entropy	0.00019	0.00291	0.0613	5.4141	0.74996
	Optimize ABF	0.00020	0.00278	0.0596	5.4812	0.73377
horse	LSCM	0.00017	0.00081	0.0310	16.8601	7.45106
#v: 31400	ABF++	0.00005	0.00047	0.0256	1.5570	0.86409
#f: 61588	CirclePatterns	0.00005	0.00046	0.0262	1.6924	0.40167
	Free Boundary	0.00034	0.00195	0.0502	1.6968	0.40979
	Optimize U2	0.00051	0.00167	0.0465	1.3928	0.36636
	Optimize entropy	0.00028	0.00161	0.0458	1.7169	0.42591
	Optimize ABF	0.00027	0.00156	0.0454	1.6462	0.40199
oliverhand	LSCM	0.00024	0.00034	0.0216	3.2283	3.97385
#v: 5660	ABF++	0.00007	0.00018	0.0162	1.0274	0.05258
#f: 10782	CirclePatterns	0.00012	0.00038	0.0228	1.0278	0.05314
	Free Boundary	0.00117	0.00353	0.0641	1.0383	0.06702
	Optimize U2	0.00077	0.00263	0.0569	1.0766	0.13460
	Optimize entropy	0.00095	0.00264	0.0584	1.2530	0.39787
	Optimize ABF	0.00081	0.00249	0.0543	1.0363	0.06675
	Optimize AD	0.00104	0.00294	0.0587	57.0863	6.40391
woodfish	LSCM	0.00022	0.00036	0.0209	3.2104	3.33530
#v: 4457	ABF++	0.00008	0.00021	0.0152	1.0126	0.02515
#f: 8449	CirclePatterns	0.00008	0.00028	0.0192	1.0135	0.02689
	Free Boundary	0.00162	0.00330	0.0664	1.0171	0.02832
	Optimize U2	0.00108	0.00227	0.0571	1.0217	0.03755
	Optimize entropy	0.00158	0.00315	0.0649	1.0167	0.02803
	Optimize ABF	0.00128	0.00215	0.0555	1.0167	0.03056
	Optimize AD	0.00066	0.00267	0.0620	3.8291	2.09491

TABLE I: Comparison of different conformal parameterization methods.

table

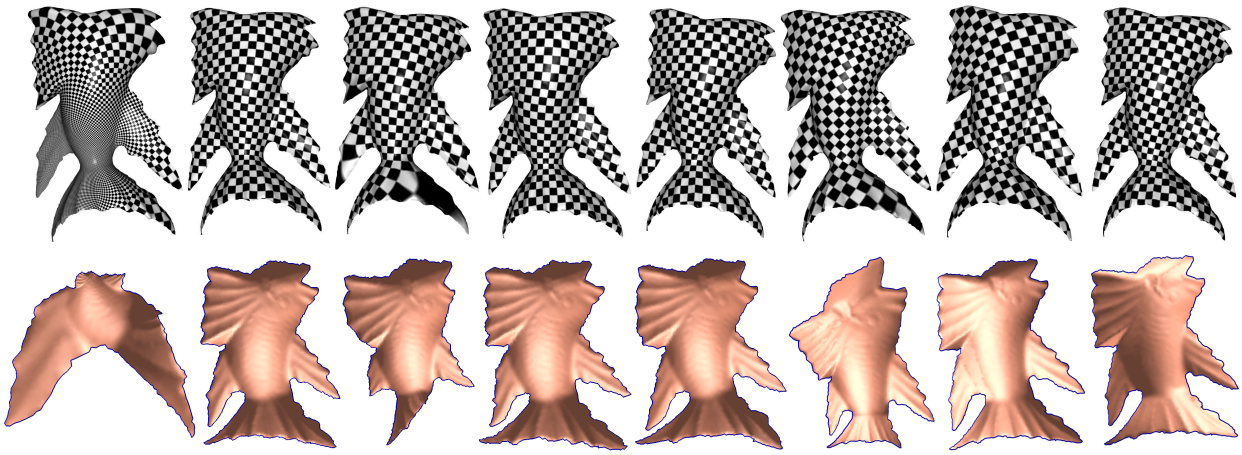


Fig. 9: Comparison of different parameterization with the Woodfish model: (a) initial, (b)-(e) optimizing ABF, AD, entropy, and U2, respectively, (f)-(h) LSCM, circle patterns, and ABF++, repectively
figure

The original proofs of this lemma can be found in [26] and [24]. For the sake of completeness, we give the following proof similar to that in [26].

Proof The proof is direct:

```
> d_ij:= diff(theta_i,r_j)*r_j;
> d_ji:= diff(theta_j,r_i)*r_i;
> simplify(d_ij^2-d_ji^2);
```

The result is zero. □

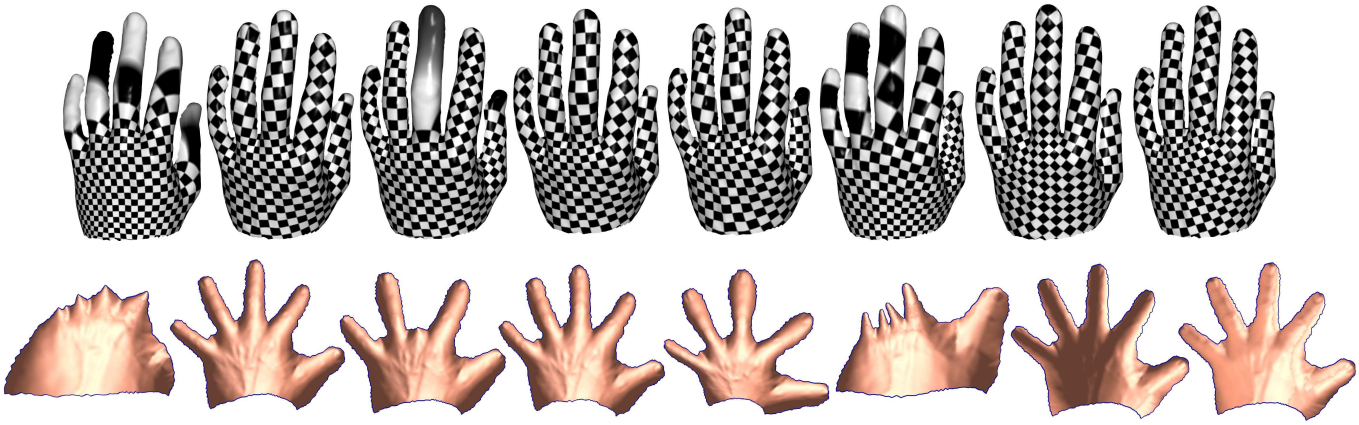


Fig. 10: Comparison of different parameterization with the Oliver's hand model: (a) initial, (b)-(e) optimizing ABF, AD, entropy, and U2, respectively, (f)-(h) LSCM, circle patterns, and ABF++, respectively
figure

Lemma 5.4: For a triangle with the circle packing metric, the derivative of θ_i satisfies:

$$d\theta_i = -\frac{h_k}{l_k}(du_i - du_j) - \frac{h_j}{l_j}(du_i - du_k). \tag{10}$$

Proof Because the face is a Euclidean triangle, $\theta_i + \theta_j + \theta_k = \pi$, and therefore $\partial\theta_i/\partial u_i + \partial\theta_j/\partial u_i + \partial\theta_k/\partial u_i = 0$. Because of symmetry in Eqn. 9, $\partial\theta_i/\partial u_i = -\partial\theta_i/\partial u_j - \partial\theta_i/\partial u_k$. Therefore,

$$\begin{aligned} d\theta_i &= \frac{\partial\theta_i}{\partial u_i}du_i + \frac{\partial\theta_i}{\partial u_j}du_j + \frac{\partial\theta_i}{\partial u_k}du_k \\ &= -\frac{h_k}{l_k}(du_i - du_j) - \frac{h_j}{l_j}(du_i - du_k). \end{aligned}$$

□

Now, we are ready to prove the main theorem,

Theorem 5.5 (Inverse Curvature Map): The curvature map K from a conformal class of circle packing metrics Π_u to the curvature space Ω_k is a C^∞ diffeomorphism, and furthermore, it is real and analytic.

The derivative map $dK : T\Pi_u(\mathbf{u}) \to T\Omega_k(\mathbf{k})$, satisfies the discrete Poisson equation,

$$d\mathbf{k} = \Delta(\mathbf{u})d\mathbf{u}, \tag{11}$$

where $T\Pi_u(\mathbf{u})$ is the tangent space of Π_u at the point $\mathbf{u}$, $T\Omega_k(\mathbf{k})$ is the tangent space of Ω_k at the point $\mathbf{k}$, and $\Delta(\mathbf{u})$ is a positive definite matrix when constrained to $T\Pi_u(\mathbf{u})$.

Proof We consider the one ring neighborhood of a vertex v_i. Let an adjacent face be $[v_i, v_j, v_k]$, where we use θ_i^{jk} to denote the angle at v_i within the face. Then from the definition of discrete curvature and from Eqn. 10 in the lemma 5.4, we get $dk_i = -\sum_{[v_i,v_j,v_k]\in F} d\theta_i^{jk} = \sum_{[v_i,v_j]\in E} w_{ij}(du_i - du_j)$, where w_{ij} is the edge weight as defined in Eqn. 8. If edge $[v_i, v_j]$ is adjacent to two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$, then its weight is equal to $w_{ij} = \frac{\partial\theta_i^{jk}}{\partial u_j} + \frac{\partial\theta_i^{jl}}{\partial u_j}$. In [22], Thurston gave a geometric proof to show that $\frac{\partial\theta_i}{\partial r_j}$ is positive if the center is inside the triangle, which is guaranteed if all edge angles ϕ_{ij} are acute. Therefore all edge weights w_{ij} are positive. In our work, we require all of the edge angles to be acute. The Jacobian matrix in Eqn. 1 has the following characteristics: summation of each row is zero, and only the diagonal elements are positive. Using linear algebra, it can be shown that J has a one dimensional null space, spanned by $\mathbf{t} = (1, \cdots, 1)$, and J is positive definite constrained on the complement space.

In the tangent space of admissible curvature space, because of the Gauss-Bonnet theorem, $d\mathbf{k}$ is orthogonal to $\mathbf{t}$, i.e. $\sum_i dk_i = 0$. In the tangent space of the normalized conformal metric space, because of the normalization condition, $d\mathbf{u}$ is orthogonal to $\mathbf{t}$, i.e. $\sum_i du_i = 0$. Therefore, the Jacobian matrix in invertible. According to the inverse function theorem, the curvature map $K : \Pi_u \to \Omega_k$ is invertible.

By direct computation, the Jacobian matrix is differentiable to infinite degree, and so is its inverse. Therefore the curvature map is a C^∞ diffeomorphism. Furthermore, the explicit formula for the Jacobian shows that its element are elementary functions of the u_i, and therefore the map is real and analytic. □

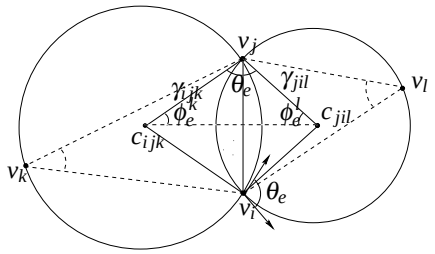


Fig. 12: Circle pattern metric
figure

APPENDIX B: INVERSE CURVATURE MAP THEOREM IN CIRCLE PATTERN SETTING

We used the symbols in [29] for our argument. The configuration of circle pattern is shown in figure V, two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$ sharing an edge $e = [v_i, v_j]$. The face circum-circles centered at c_{ijk} and c_{jil} with radii γ_{ijk} and γ_{jil}, intersecting at an angle θ_e. Let $\rho_{ijk} = log\gamma_{ijk}$, $\rho_{jil} = log\gamma_{jil}$, then

$$\phi_e^k = tan^{-1}\frac{\sin\theta_e}{e^x - \cos\theta_e}, \tag{12}$$

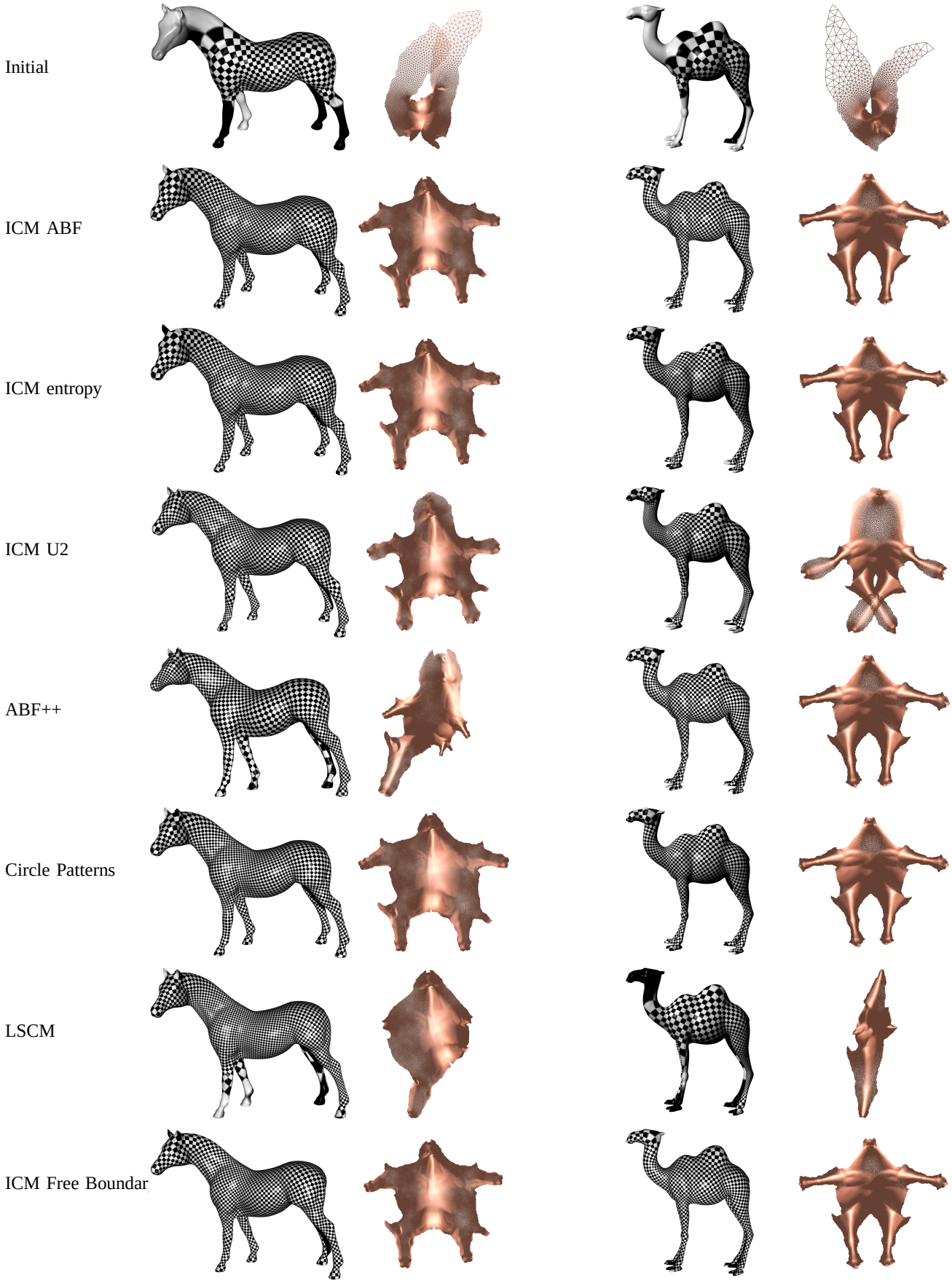


Fig. 11: Comparison figure

where $x = \rho_{ijk} - \rho_{jil}$. Then the curvature of face $t \in T$ is defined as

$$\Phi_t = 2\pi - \sum_{e \in t} 2\phi_e^t \tag{13}$$

Theorem 5.6 (Inverse Curvature Map in Circle Pattern):
Suppose M is a closed mesh with a circle pattern. If the edge weights $\theta_e \in (0, \pi)$ are fixed, then the face curvature map $\Phi : \{\rho\} \to \{\Phi_t\}$ with the constraints $\sum_{t \in T} \rho_t = 0$ is bijective and real analytic.

Proof Similar to lemmas 5.2 and 5.3, direct computation shows

$$\frac{\partial \phi_e^k}{\partial \rho_l} = \frac{\partial \phi_e^l}{\partial \rho_k} = \frac{\sin\theta_e e^x}{(e^x - \cos\theta_e)^2(1 + \frac{\sin^2\theta_e}{(e^x - \cos\theta_e)^2})} > 0, when\ \theta_e \in (0, \pi)$$

Similar to Lemma 10, the following holds

$$d\phi_e^k = -\frac{\partial \phi_e^k}{\partial \rho_l}(d\rho_k - d\rho_l),$$

then from equation 13, the Jacobian map is

$$d\Phi = \Delta(\rho) d\rho$$

where $\Delta(\rho)$ has the same characteristics as Eq. (3), therefore, it is positive definite. □
The projected gradient optimization algorithm 4 and curvature diffuse algorithm 5 can be directly translated to the circle pattern setting.

REFERENCES

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

[2] A. Sheffer, E. Praun, and K. Rose, *Mesh Parameterization Methods and Their Applications*, vol. 2 of *Foundations and Trends in Computer Graphics and Vision*. Now Publisher, 2006.

[3] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least squares conformal maps for automatic texture atlas generation," *SIGGRAPH 2002*, pp. 362–371, 2002.

[4] 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.

[5] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution analysis of arbitrary meshes.," in *SIGGRAPH*, 1995.

[6] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, pp. 15–36, 1993.

[7] M. S. Floater, "Parametrization and smooth approximation of surface triangulations," *Computer Aided Geometric Design*, vol. 14, no. 3, pp. 231–250, 1997.

[8] A. Sheffer and E. de Sturler, "Smoothing an overlay grid to minimize linear distortion in texture mapping," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 874–890, 2002.

[9] X. Gu and S.-T. Yau, "Global conformal parameterization," in *Symposium on Geometry Processing*, pp. 127–137, 2003.

[10] M. S. Floater, "Mean value coordinates," *Computer Aided Geometric Design*, vol. 20, no. 1, pp. 19–27, 2003.

[11] K. Hormann and G. Greiner, "Mips: An effficient global parametrization method.," in *Curve and Surface Design: Saint-Malo 1999*, pp. 153–162, Vanderbilt Unitersity Press, 2000.

[12] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe, "Texture mapping progressive meshes.," in *SIGGRAPH*, pp. 409–416, 2001.

[13] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski, "Bounded-distortion piecewise mesh parameterization.," in *IEEE Visualization*, pp. 355–362, 2002.

[14] P. Degener, J. Meseth, and R. Klein, "An adaptable surface parameterization method.," in *IMR*, pp. 201–213, 2003.

[15] S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel, "A fast and simple stretch-minimizing mesh parameterization.," in *SMI*, pp. 200–208, 2004.

[16] E. Zhang, K. Mischaikow, and G. Turk, "Feature-based surface parameterization and texture mapping," *ACM Transactions on Graphics*, vol. 24, no. 1, pp. 1–27, 2005.

[17] Y. Lee, H. S. Kim, and S. Lee, "Mesh parameterization with a virtual boundary.," *Computers & Graphics*, vol. 26, no. 5, pp. 677–686, 2002.

[18] Z. Karni, C. Gotsman, and S. J. Gortler, "Free-boundary linear parameterization of 3d meshes in the presence of constraints.," in *SMI*, pp. 268–277, 2005.

[19] R. Zayer, C. Rössl, and H.-P. Seidel, "Setting the boundary free: A composite approach to surface parameterization," in *Eurgraphics Symposium on Geometry Processing* (H. Pottmann and M. Desbrun, eds.), pp. 91–100, 2005.

[20] 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.

[21] 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.

[22] W. P. Thurston, *Geometry and Topology of Three-Manifolds*. Princeton lecture notes, 1976.

[23] B. Rodin and D. Sullivan, "The convergence of circle packings to the riemann mapping," *Journal of Differential Geometry*, vol. 26, no. 2, pp. 349–360, 1987.

[24] C. R. Collins and K. Stephenson, "A circle packing algorithm," *Computational Geometry: Theory and Aplications*, vol. 25, pp. 233–256, 2003.

[25] R. S. Hamilton, "The ricci flow on surfaces," *Mathematics and general relativity*, vol. 71, pp. 237–262, 1988.

[26] B. Chow and F. Luo, "Combinatorial ricci flows on surfaces," *Journal Differential Geometry*, vol. 63, no. 1, pp. 97–129, 2003.

[27] 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.

[28] 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.

[29] 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.

[30] B. A. Springborn, *Variational principles for circle patterns*. PhD thesis, Technical University of Berlin, 2003.

[31] A. P. Witkin and P. S. Heckbert, "Using particles to sample and control implicit surfaces.," in *SIGGRAPH*, pp. 269–277, 1994.

[32] Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, J. Wallner, and H. Pottman, "Robust feature classification and editing," *IEEE Transaction on Visualization and Computer Graphics*, vol. 13, pp. 34–45, 2007.

[33] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 355–361, 2002.

[34] A. Sheffer and J. C. Hart, "Seamster: Inconspicuous low-distortion texture seam layout.," in *IEEE Visualization*, 2002.