Optimal Surface Parameterization Using Inverse Curvature Map

Yongliang Yang[3], Junho Kim[1,4], Feng Luo[2], Shimin Hu[3], and Xianfeng Gu[1]
[1]Stony Brook University [2]Rutgers University [3]Tsinghua University [4]Dong-Eui 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 the 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 *differential* 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 computed from the curvature (by integration). 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 that 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 and 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, and 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]–[3]. In theory, *angle* distortion can be eliminated completely by conformal mapping, but it is impossible for conformal mappings to further eliminate *area* distortion completely, except for developable surfaces.

For a given surface, we can define infinitely many different conformal mappings with different area distortions, as shown in Fig. 1. The central problem of the optimal parameterization can be stated as follows:

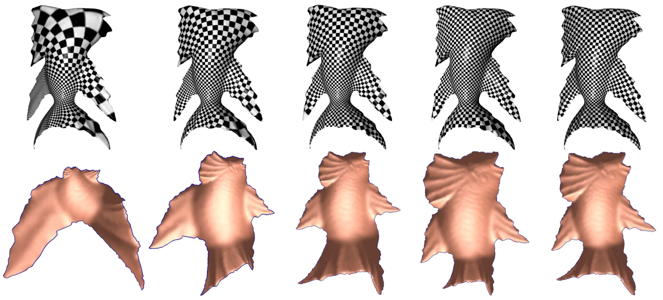


Fig. 1. There are an infinity number of conformal parameterizations for a given surface. We minimize the area distortion within the conformal mappings.

How can we find the best conformal mapping that has the least area distortion?

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 briefly overview the most related works and refer readers to [1]–[3] for wider surveys.

A common approach for parameterization is to minimize a certain energy to control the distortion. Lévy *et al.* [4] defined an energy to approximate the Cauchy-Riemann equation; Desbrun *et al.* [5] optimize Dirichlet energy. Variations of harmonic energies are also optimized using discrete Laplace-Beltrami operators in [6]–[11]. More general energy forms can be found in [12]–[17]. Most linear methods apply a convex Dirichlet-type boundary. Virtual boundaries are applied in [17] and [18] to absorb distortions introduced by the convex boundary conditions. Alternatively, [4] and [5] provide parameterizations which require to fix only a few vertices in the parametric domain. Karni *et al.* [19] discuss the design of geometrically complex boundary conditions with constraints. Zayer *et al.* [20] apply 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) [21] utilizes this property to produce high quality 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. Recently, the method has been improved by several derivative works [22], [23] in terms of speed and robustness.

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 [24]. Circle packings and circle patterns replace infinitesimal circles with finite circles. In the limit of refinement the continuous conformal maps are recovered [25]. Collins and Stephenson [26] have implemented circle packing in their software *CirclePack* which only considers combinatorics. The connection between circle packing and smooth surface Ricci flow [27] was discovered in [28]. The discrete Ricci flow method was introduced in [29] for hyperbolic parameterization.

Kharevych *et al*. [30] provided conformal parameterizations for arbitrary genus types by applying circle patterns based on the variational principle in Bobenko and Springborn [31]. The method in [30] supports very flexible boundary conditions ranging from free boundaries to control of the boundary shape via prescribed curvatures. They can further reduce the distortion by incorporating manually selected cone singularities. Ben-Chen *et al*. [32] introduced a conformal parameterization which automatically determines the locations and target curvatures of the cone singularities.

Our work differs from the previous work in the following aspects. Based on a rigorous theoretic proof, our method can produce a discrete conformal parameterization with least area distortion among all possible parameterizations. Similar to [30], our method is applicable for meshes with general topologies. Furthermore, the method can be extended for optimizing more general energies with constraints on curvatures and area distortions. (Fig. 8 shows parameterizations with special curvature constraints, such that all boundary curvatures are constant.)

In this paper, we will explain our theoretical results and algorithmic implementations with circle packings. Since circle packing and circle patterns are equivalent in theory [33], [34], our results can also be explained with the setting of circle patterns (see 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 [5], [20] 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 is centered at a vertex as shown in Fig. 3. 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:

$$K : \{\textit{configuration of radii}\} \to \{\textit{configuration of curvatures}\}.$$

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, which we call 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, and 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. Computing the optimal circle packing metric (Section III-D, Section III-E)
5. Isometric embedding (Section III-F)

The theoretical results of inverse curvature map are explained in Section II. Each step of the algorithm pipeline 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 proofs are presented in Appendix.

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, $[v_i, v_j]$ (or e_{ij}), and $[v_i, v_j, v_k]$ (or f_{ijk}), respectively. A mesh M embedded in $\mathbb{R}^3$ 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 to each vertex v_i a circle with radius γ_i. On edge e_{ij}, the two circles with radii γ_i and γ_j intersect at an angle of ϕ_{ij}, as shown in Fig. 3.

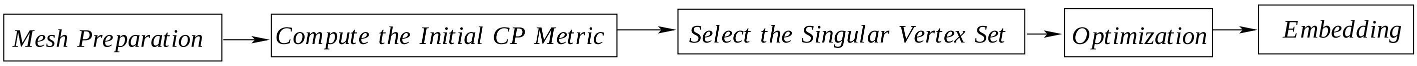


Fig. 2. Algorithm pipeline.

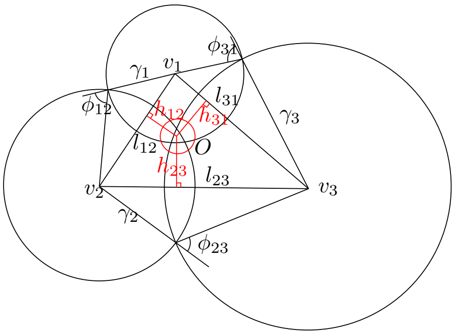


Fig. 3. Circle packing metric.

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 for vertices (i.e., $v_i \mapsto \gamma_i$), and $\Phi : E \to [0,\frac{\pi}{2}]$ is the angle function for edges (i.e., $e_{ij} \mapsto \phi_{ij}$). The discrete metric on an edge e_{ij} is determined by $l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2\gamma_i\gamma_j \cos\phi_{ij}}$.

Now, the edge lengths can be determined by the circle radii γ_i and the intersection angles ϕ_{ij} with the cosine law, depicted in Fig. 3. 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,Γ_1,Φ_1) and (M,Γ_2,Φ_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 defines a hyper-plane in the $\mathbb{R}^n$ that we denote $\Pi_\mathbf{u}$. 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 $\Omega_\mathbf{k} := K(\Pi_\mathbf{u})$ is a convex polytope [28].

The curvature map K from the conformal metric space to the curvature space $K : \Pi_\mathbf{u} \to \Omega_\mathbf{k}$ is bijective; both the map and the inverse map have an infinite degree of smoothness. Furthermore, the curvature map is real 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 $\Pi_\mathbf{u}$ to the curvature space $\Omega_\mathbf{k}$ is a C^∞ diffeomorphism. Furthermore, it is real analytic. The derivative map $dK : T\Pi_\mathbf{u}(\mathbf{u}) \to T\Omega_\mathbf{k}(\mathbf{k})$, satisfies the discrete Poisson equation,

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

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

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

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

A detailed proof can be found in Appendix A. 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, such 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 face, there exists a unique circle perpendicular to all three circles, as the red circle shown in Fig. 3. The center of the circle is the *radial center* (or *power center*). Then the weight for a halfedge equals to $\frac{h_{ij}}{l_{ij}}$, where h_{ij} is the distance from the radial center to the halfedge, l_{ij} is the current length of the halfedge. The edge weight is the sum of those of its halfedge weights and depends on the current curvature (or, equivalently the radii). Therefore the Laplace-Beltrami operator is dynamic. This fact makes the whole theory more complicated.

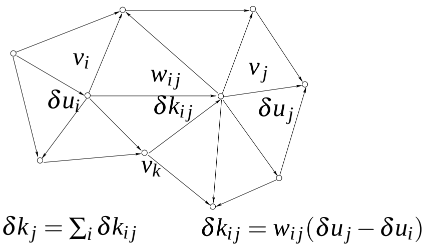

v_i
w_{ij}
v_j
δu_i
δk_{ij}
δu_j
v_k
$\delta k_j = \sum_i \delta k_{ij}$
$\delta k_{ij} = w_{ij}(\delta u_j - \delta u_i)$

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.

Algorithm 1 computes the *inverse curvature map*, which can be applied for conformally parameterizing general meshes directly.

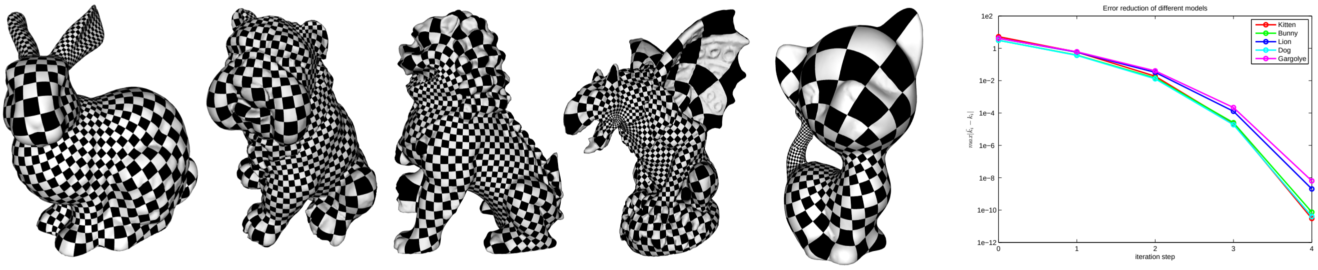


Fig. 5. Discrete conformal parameterizations using *Inverse Curvature Map* ([#v, #f, execution time (sec)] are [20660, 41118, 6.3], [29923, 59417, 11.2], [20224, 40118, 5.8], [20618, 40803, 6.6], and [10219, 20438, 2.5], respectively with Pentium4 2.8GHz with 2GB memory).

Fig. 5 demonstrates some parameterization results using this algorithm.

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

```
**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 $\mathbf{k}$
  $\mathbf{u} \Leftarrow \mathbf{u}_0$, where $\mathbf{u}_0$ 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
  $\bar{\mathbf{u}} \Leftarrow \mathbf{u}$
```

C. Relation with Discrete Ricci Flow

Inverse curvature map can be also deduced from the theory of discrete Ricci flow [28]. The fact that curvature map $K : \mathbf{u} \to \mathbf{k}$ is invertible is proven in the following way.

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

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

From discrete Ricci flow theory, $E_{Ric}(\mathbf{u})$ 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, using the steepest descent method

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

which 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 the key step for a nonlinear optimization algorithm.

III. OPTIMAL SURFACE PARAMETERIZATION

This section explains the algorithm pipeline for the optimization system as shown in Fig. 2, each subsection corresponds to one step respectively.

A. Mesh Preparation (optional)

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) to ensure the uniqueness of the solution. If the input mesh has too many obtuse angles and skinny triangles, we remesh it using the algorithms described in [35], [36] to improve the mesh quality.

B. Computing the initial Circle Packing Metric

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 $[v_i, v_j, v_k] \in F$ do
    for all corners $c_i$, related to $v_i$ in each face do
      $\gamma(c_i) \leftarrow \frac{l_{ki} + l_{ij} - l_{jk}}{2}$
    end for
  end for
  for all vertex $v_i \in V$ do
    $\gamma_i \leftarrow max\{\gamma(c) | c$ is attached to $v_i\}$
  end for
  for all edges $[v_i, v_j] \in E$ 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, which we call a *singular vertex set*. For example, if a mesh has boundaries, all of the boundary vertices are in the singular vertex set in general.

Moreover, we can set several interior vertices as singular ones in order to minimize the area distortions. Several effective methods have been introduced to select singular vertices, including manual selection [30], vector field analysis [37], and spectral analysis [32].

We select the singular vertex set in step 3 of algorithm pipeline in Fig. 2. Intuitively, we first let the curvature be uniformly distributed on all vertices, then measure the area distortion, and finally 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

```
$\overline{k}_i \leftarrow 0$ for $v_i \in \partial M$,
$\overline{k}_i \leftarrow \frac{2\pi\chi(M)}{n}$ for $v_i \notin \partial M$, where $n$ is number of interior vertices
$\overline{\mathbf{u}} \leftarrow K^{-1}(\overline{\mathbf{k}})$ using Algorithm 1
$S \Leftarrow \partial M \cup \{$local minima of $\overline{\mathbf{u}}\}$
```

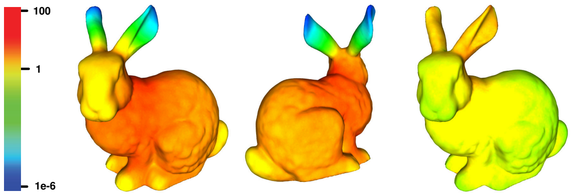


Fig. 6. Singularity selection process: (a) and (b) depict the area distortion of the bunny model without any singular vertex. We select critical points in ear tips and the point between the roots of the ears. (c) shows the area distortion after computing the inverse curvature map with the selected singular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

D. Compute the Optimal Circle Packing Metric

Now, we explain the process to compute the optimal circle packing metrics in step 4 of the algorithm pipeline in Fig. 2. 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, they are equivalent. We decide to define the energy in curvature space because this is a convex affine subspace. If the energy is convex, only one optimum exists. Therefore, it is easy to handle both theoretically and practically.

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

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

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

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 $\Pi_{\mathbf{k}}$:

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

Basically, we compute the gradient of $E(\mathbf{k})$ w.r.t $\mathbf{k}$, denoted as $\nabla_{\mathbf{k}} E$, and project the gradient to the affine subspace $\Pi_{\mathbf{k}}$. While the projected gradient is not equal to zero, $\mathbf{k}$ is updated along its direction. Namely, at a critical point of $E(\mathbf{k})$, $\nabla_{\mathbf{k}} E$ is orthogonal to the admissible curvature space.

This procedure can find one local minimum but does not guarantee that neither all minima, nor the global minimum can be found. In next subsection, we design a special energy with a unique minimum, in order that this algorithm can reach the global minimum.

Algorithm 4 Optimal Discrete Conformal Parameterization

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

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

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

Proof The proof is based on the KKT theorem [38]. If $\mathbf{k}$ is an optimum of $E(\mathbf{k})$, then $\nabla_{\mathbf{k}} E \perp \Pi_{\mathbf{k}}$. Suppose v_i is a non-singular 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 non-singular vertices v_i and $d_j = 1$ for singular vertices v_j. Therefore

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

where μ is a real number. □

The common energy forms used in the literature are:

1) Angle Based Flattening energy defined in [21]: 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 [5]: this energy measures the ratio between the original face area and the face area on the parameter plane,

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

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 is a vector with 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.

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

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

The gradient of an energy E w.r.t. $\mathbf{u}$ is $\nabla_{\mathbf{u}}E$, which is related to $\nabla_{\mathbf{k}}E$ by

$$\nabla_{\mathbf{k}}E = \Delta^{-1}(\mathbf{u})\nabla_{\mathbf{u}}E.$$

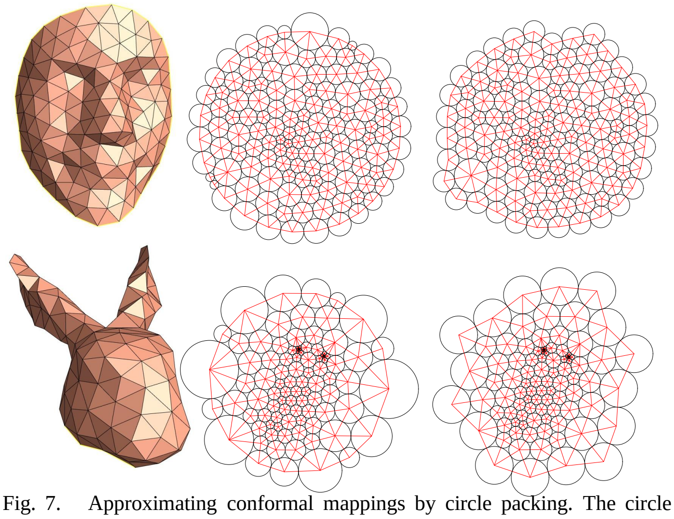


Fig. 7. Approximating conformal mappings by circle packing. The circle radii are changed while tangency relations are preserved. The second column shows a circular boundary condition; the third column shows a free boundary condition. For the free boundary condition, all circle radii on the boundary vertices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

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 a unique global minimum or not. Secondly, their gradient has a complicated form, leading to an expensive computation using the projected gradient method.

In this part, we introduce a novel energy, called *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 efficiently be computed with methods other than the projected gradient method.

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

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

where $\mathbf{u}_0$ is the initial metric with the curvature $\mathbf{k}_0$ defined with the initial mesh.

The integration measures the uniformity of the area distortion function, (i.e., conformal factor). The one-form $\mathbf{u}^T d\mathbf{k}$ is a closed one-form, and therefore the integration is path independent. We can choose an arbitrary path from $\mathbf{k}_0$ to $\mathbf{k}$ in the curvature space.

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

$$E_{u^2}(\mathbf{u}) = \int_{\mathbf{u}_0}^{\mathbf{u}} (\boldsymbol{\mu} - \mathbf{u}_0)^T I d\boldsymbol{\mu},$$

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

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

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 $max_{v_i \in N}|k_i| > \varepsilon$ do
  for all $v_i \in N$ do
    $du_i \Leftarrow -k_i$
    $u_i \leftarrow u_i + \lambda du_i$
  end for
  $c \leftarrow \frac{\sum_{v_i \in V} u_i}{|V|}$
  for all vertex $v_j \in V$ do
    $u_i \leftarrow u_i - c$
  end for
end while
```

Intuitively, Algorithm 5 sets $\frac{d\mathbf{u}}{dt} = -\mathbf{k}$. According to 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 the entropy increases in the heat diffusion process, we name this energy as curvature entropy.

We show that when the algorithm terminates, the deformation of the singular vertices is 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 At the beginning, the $u_i - u_i^0$ terms are zero for all boundary vertices. At each normalization step, $u_i - u_i^0$ changes 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 the circle radii of the boundary vertices are equal (this is because the radii for all vertices in the initial circle packing metric are equal). This result is consistent with theorem 3.2, i.e. 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 a unique global minimum point in admissible curvature space. The free boundary optimization algorithm leads to the global minimum.

Proof In order to show that the energy is well defined, we need to show that 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 also symmetric.

We can directly compute the gradient of $E_{EN}(\mathbf{k})$, $\nabla_{\mathbf{k}} E_{EN}(\mathbf{k}) = \mathbf{u} - \mathbf{u}_0$. The necessary condition of the optimum 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, 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, implying that the energy has a unique global minimum on it.

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

Comparing to the projected gradient algorithm in Section III-D.1, the curvature diffusion algorithm doesn't need to solve the Poisson equation, it is simple, direct and easy to implement. With free boundary conditions, we can apply the curvature diffusion algorithm directly.

F. Embedding

The final step in our algorithm pipeline in Fig. 2 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 it to an open topological disk. Several algorithms [39], [40] 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 the whole mesh is flattened.

For large scale meshes, the propagated errors accumulate, so instead, we use a method similar to [22]. According to lemma 5.4, the planar embedding $f : M \to \mathbb{R}^2$ is a harmonic map $\Delta f = 0$, where the Laplace-Beltrami operator is determined by the final circle packing metric. We first fix the parametric positions of two vertices, then form a linear system to approximate the parameter positions for other vertices 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 [4], ABF++ [22], and circle patterns [30]. For fair comparisons, we tested the previous methods with the codes which are available on the websites of the original authors.

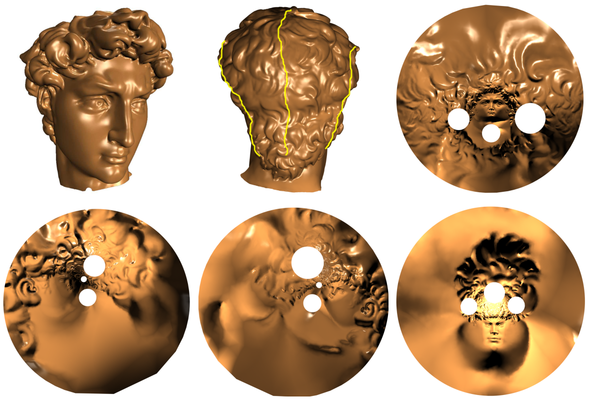


Fig. 8. Mesh parameterization with the inverse curvature map for a topologically complicated model: The David head model has four boundaries, as shown in (a) and (b). Four different configurations are depicted in (c)-(f), each of which has one outer boundary and three inner boundaries in the parametric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

Table I summarizes the statistics of our experiments on several models as shown in Fig. 9, Fig. 10, Fig. 11. Angle distortions are measured with three different energy forms: conformality [4], L2 shear [22], and squared sum of angle differences [22]. As shown in Table I, all of the methods minimize the angle distortion and hold the conformality well.

Area distortions are measured with two different energy forms: L2 stretch in [13] and Log area distortion Eq. (8). Our optimization approaches provide the best results for the complicated models, such as the horse and the camel. ABF++ produces small area distortions in many cases, but it may fall into a local minimum (see Fig. 11). The circle patterns provide comparable good results with all the tested models.

$$\text{Log area distortion} = \sum_f \left(\log \frac{s_f(\mathbf{k})}{s_f(\mathbf{k}_0)} \right)^2 \tag{8}$$

The efficiency of both Algorithm 1 and Algorithm 4 greatly depends on solving the Poisson equation $d\mathbf{k} = \Delta(\mathbf{u})d\mathbf{u}$. Since the Laplace matrix Δ is positive definite when restricted on $T\Pi_{\mathbf{u}}(\mathbf{u})$, we use the conjugate gradient method to solve the linear system, which is efficient in terms of time and storage. 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 method (Algorithm 4). This is because the latter needs to compute inverse curvature map for each iteration step, which is very time consuming when the energy is close to the minimum. In the experiment for the horse model (Fig.11) with $30k$ vertices, the ICM entropy method takes several minutes, whereas the curvature diffusion method only takes 21s to get the similar result. For other energy forms (ABF, Area Distortion U2), their gradients are more complicated than one of curvature entropy energy, thus they are computationally more expensive. 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. (6) 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, called the curvature entropy, is investigated. It has a unique global minimum and can be optimized using curvature diffusion algorithm with free boundaries. Our experiments on complex meshes support our theoretical discoveries.

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

Although we solved the optimal parametrization problem for a given singular vertex set, it remains a challenging problem to determine the optimal vertex set. The common belief for choosing singular vertices is to pick the critical points of the area distortion function, as our algorithm in Section III-C does. 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), the Natural Science Foundation of China (Project Number 60673004, 60628202), and the National High Technology Research and Development Program of China (Project Number 2007AA01Z336). Feng Luo is partially supported by the NSF (DMS 0625935). Junho Kim is partially supported by the IITA&MIC scholarship program.

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 omit the calculation results returned by Maple.

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 (see Fig. 3), called the *radial center* of three circles.

For a triangle with the circle packing metric, the following equations hold (the proof can be found in [24]):

$$\frac{\partial \theta_i}{\partial u_j} = \frac{h_k}{l_k} \tag{9}$$

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

Also we have (the proof can be found in [26], [28]):

$$\frac{\partial \theta_i}{\partial u_j} = \frac{\partial \theta_j}{\partial u_i}. \tag{10}$$

Lemma 5.1: For a triangle with the circle packing metric, the derivative of θ_i satisfies:

$$d\theta_i = -\frac{h_k}{l_k}(du_i - du_j) - \frac{h_j}{l_j}(du_i - du_k). \tag{11}$$

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. 10, $\partial\theta_i/\partial u_i = -\partial\theta_i/\partial u_j - \partial\theta_i/\partial u_k$. Therefore,

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

□

Now, we are ready to prove the main theorem,

Theorem 5.2 (Inverse Curvature Map): The curvature map K from a conformal class of circle packing metrics $\Pi_{\mathbf{u}}$ to the curvature space $\Omega_{\mathbf{k}}$ is a C^∞ diffeomorphism. Furthermore, it is real analytic.

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

$$d\mathbf{k} = \Delta(\mathbf{u})d\mathbf{u}, \tag{12}$$

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

Proof We consider the one ring neighborhood of a vertex v_i. Let an adjacent face be $[v_i, v_j, v_k]$, where θ_i^{jk} denotes the angle at v_i within the face. Then from the definition of discrete curvature and from Eqn. 11 in the lemma 5.1,

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. 9. 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

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.

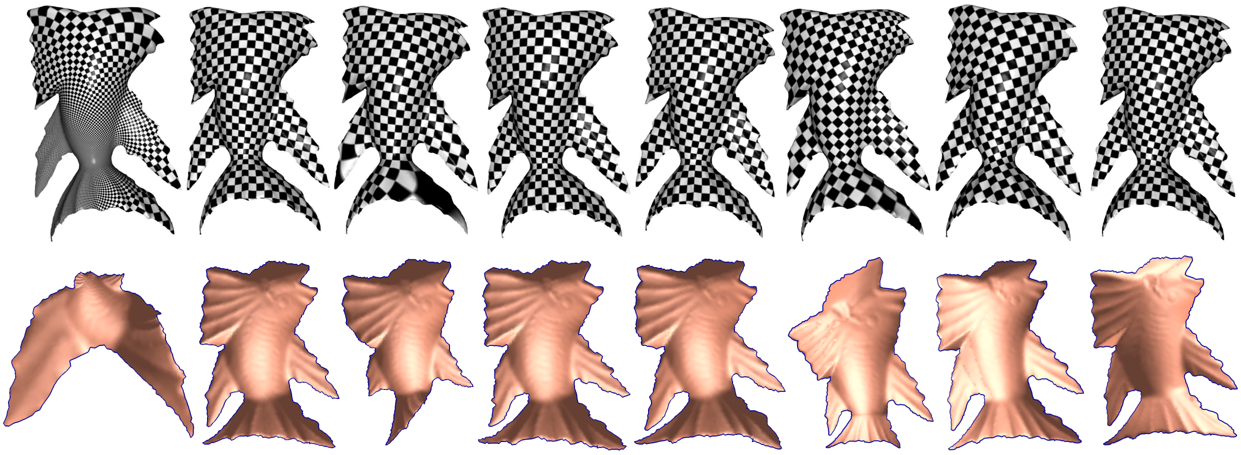


Fig. 9. Comparison of different parameterization with the Woodfish model: (a) initial, (b)-(e) optimizing ABF, AD, entropy, and U2, respectively, (f)-(h) LSCM, circle patterns, and ABF++, repectively

is equal to $w_{ij} = \frac{\partial \theta_i^{jk}}{\partial u_j} + \frac{\partial \theta_i^{jl}}{\partial u_j}$. In [24], Thurston gave a geometric proof to show that $\frac{\partial \theta_i}{\partial r_j}$ is positive if the radial 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 Jacobi 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.

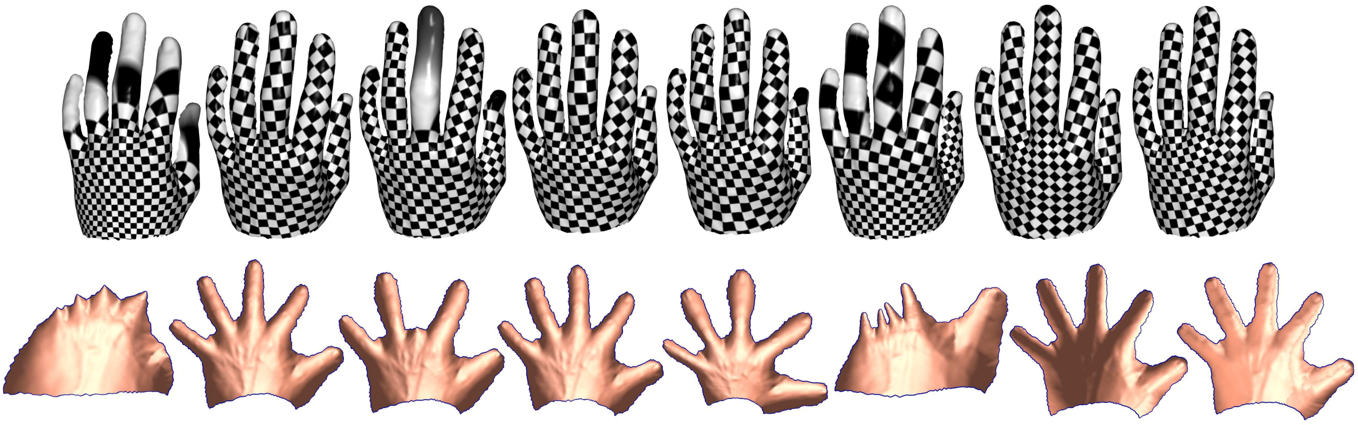


Fig. 10. Comparison of different parameterization with the Oliver's hand model: (a) initial, (b)-(e) optimizing ABF, AD, entropy, and U2, respectively, (f)-(h) LSCM, circle patterns, and ABF++, respectively

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 Jacobi matrix is invertible. According to the inverse function theorem, the curvature map $K : \Pi_{\mathbf{u}} \to \Omega_{\mathbf{k}}$ is invertible.

By direct computation, the Jacobi 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 elements are elementary functions of the u_i. Hence, the map is real analytic. □

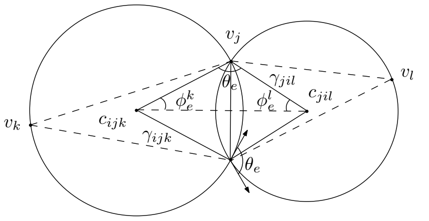


Fig. 12. Circle pattern metric

Appendix B: Inverse Curvature Map Theorem in Circle Pattern Setting

We used the notations in [30] for the following explanation. The configuration of circle pattern is shown in Fig. 12, 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

$$\phi_e^k = tan^{-1}\frac{\sin\theta_e}{e^x - \cos\theta_e}, \tag{13}$$

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

$$\Phi_t = 2\pi - \sum_{e \in t} 2\phi_e^t \tag{14}$$

Theorem 5.3 (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, 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 Eqn. (9) and (10), 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 5.1, the following holds

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

then from equation 14, 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 (Algorithm 4) and curvature diffuse algorithm (Algorithm 5) can be directly translated to the circle pattern setting.

Appendix C: Embedding Induced by Inverse Curvature Map

In this section, we give the formal proof to show that the embedding of the mesh induced by the Inverse Curvature Map is harmonic. This builds the intrinsic connection to conventional harmonic maps.

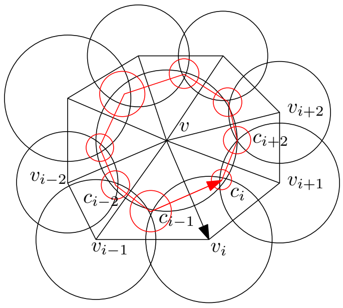


Fig. 13. The embedding induced by Inverse Curvature Map is harmonic.

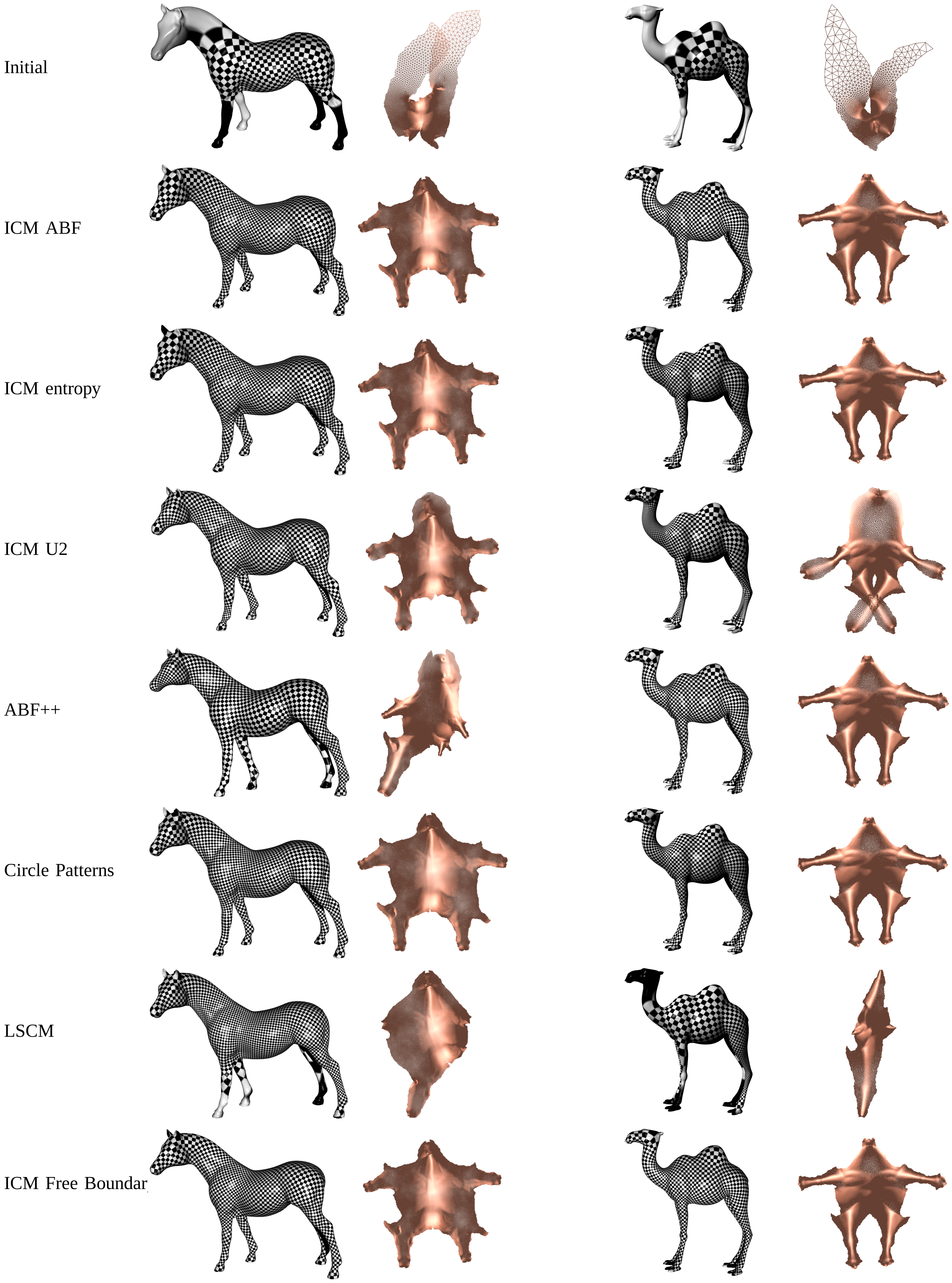


Fig. 11. Comparison

Lemma 5.4: Suppose M is a mesh with a circle packing metric, which induces zero Gaussian curvature everywhere, $f : M \to \mathbb{R}^2$ is the isometric embedding of M onto the plane. Then the embedding f is harmonic, namely

$$\Delta f = 0.$$

Proof As shown in Fig. 13, v is a vertex on the mesh, $v_0, v_1, \cdots, v_{n-1}$ are the neighbor vertices of v, and c_i is the center of the dual circle on the face $[v, v_i, v_{i+1}]$, which is orthogonal to the three circles centered at the vertices. We embed the one ring neighborhood of v onto the complex plane $\mathbb{C}$, and use the same symbol to represent the complex coordinates of the vertices. Then by definition, the edge weight of edge $[v, v_i]$, denoted as w_i equals to

$$w_i = \frac{|c_{i-1}c_i|}{|vv_i|}, w_i(v_i - v) = \sqrt{-1}(c_i - c_{i-1}).$$

Therefore,

$$\sum_i w_i(v_i - v) = \sum \sqrt{-1}(c_i - c_{i-1}) = 0,$$

where this is equivalent to $\Delta f = 0$. Hence the coordinates of the vertices of M are harmonic functions.

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] K. Hormann, B. Lévy, and A. Sheffer, “Mesh parameterization: Theory and practice,” in *SIGGRAPH ’07: ACM SIGGRAPH 2007 courses*, ACM, 2007.

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

[5] 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.

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

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

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

[9] 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.

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

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

[12] 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.

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

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

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

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

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

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

[25] 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.

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

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

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

[29] 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.

[30] 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.

[31] 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.

[32] M. Ben-Chen, C. Gotsman, and G. Bunin, “Conformal flattening by curvature prescription and metric scaling,” *Computer Graphics Forum (Proc. Eurographics 2008)*, vol. 27, no. 2, p. to appear, 2008.

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

[34] F. Luo, X. Gu, and J. Dai, *Variational Principles for Discrete Surfaces*. High Education Press, 2007.

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

[36] 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.

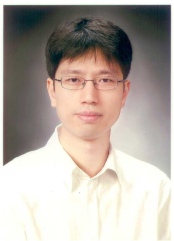
[37] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez, “Periodic global parameterization,” *ACM Transactions on Graphics*, vol. 25, no. 4, pp. 1460–1485, 2006.

[38] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

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

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

Yong-Liang Yang received a bachelor’s degree in computer science from Tsinghua University in 2004. He is a PhD student in the Department of Computer Science and Technology at Tsinghua University. His research interests include computer graphics, geometric modeling and processing.



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

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

Shi-Min Hu obtained his Ph.D. in 1996 from Zhejiang University. He is currently a chair professor of computer science at Tsinghua University, China. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He is on the editorial boards of Computer Aided Design.

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