<<003>>

Team Member： Bella Gu

School： International School of Boston

City： Cambridge, MA

Supervisor： David Xianfeng Gu

Title： Geometric Compression Based on Ricci Flow and Monge-Ampère Equation

Title： Geometric Compression Based on Ricci Flow and Monge-Ampère Equation

Abstract： Geometric compression plays a fundamental role in virtual reality and augmented reality (VR/AR) applications. Dense meshes are re-sampled and re-tessellated to reduce the complexity. This process is called remeshing. In this work, we propose a novel remeshing algorithm based on both angle-preserving parameterization, and measurecontrollable parameterization. The conformal parameterization is carried out by discrete surface Ricci flow method, the measure-controllable parameterization is obtained by an optimal mass transportation map. The sampling is performed on the measurecontrollable parameterization domain, the triangulation is computed on the conformal parameterization domain using Delaunay refinement algorithm. This method gives the user full control of sampling distribution, and produces meshes with curvature measure convergence. The meshing result can emphasize the region of interests, is curvature sensitive. Experimental results demonstrate the efficiency and efficacy of the proposed method.

Abstract

Geometric compression plays a fundamental role in virtual reality and augmented reality (VR/AR) applications. Dense meshes are re-sampled and re-tessellated to reduce the complexity. This process is called remeshing. In this work, we propose a novel remeshing algorithm based on both angle-preserving parameterization, and measurecontrollable parameterization. The conformal parameterization is carried out by discrete surface Ricci flow method, the measure-controllable parameterization is obtained by an optimal mass transportation map. The sampling is performed on the measurecontrollable parameterization domain, the triangulation is computed on the conformal parameterization domain using Delaunay refinement algorithm. This method gives the user full control of sampling distribution, and produces meshes with curvature measure convergence. The meshing result can emphasize the region of interests, is curvature sensitive. Experimental results demonstrate the efficiency and efficacy of the proposed method.

1. Introduction

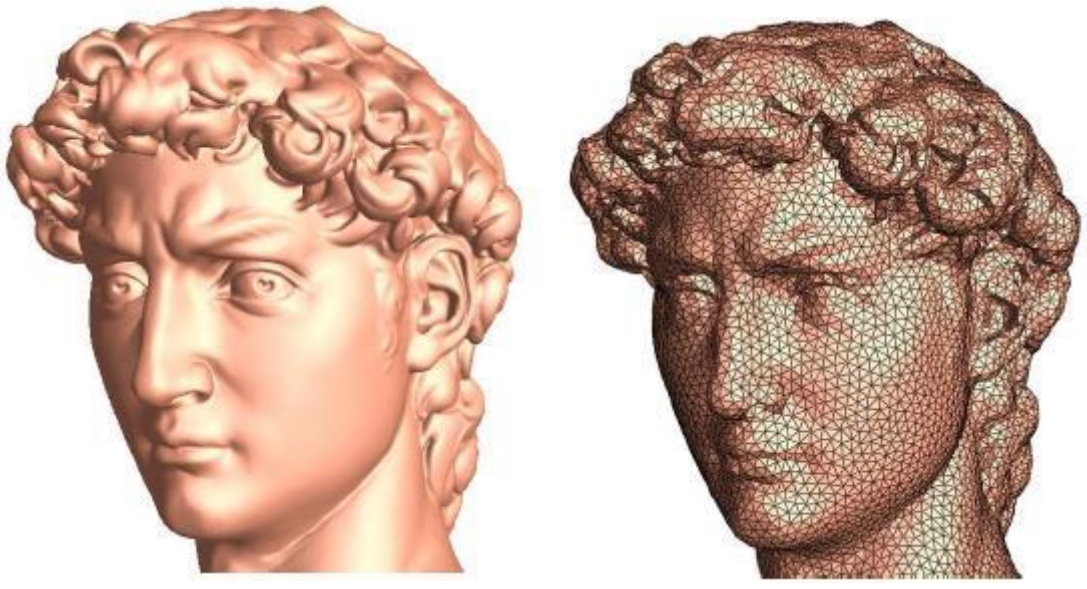
1.1. Central Task

<<003>>

Recent years have witnessed the rapid development of Virtual Reality/Augmented Reality (VR/AR) technologies, which have great potentials to be applied in education, medicine, entertainment, manufacture, finance and many fields in real life. Although VR/AR technologies are reshaping the whole human society, they are facing great challenges.

VR/AR applications create virtual world using geometric methods, mixed with reality. In the virtual world, all the objects are modeled as surfaces with complicated topologies and geometries, and represented as polygonal surfaces, usually *triangle meshes*. A triangle mesh is obtained by triangulating a smooth surface, then approximate each cell by a Euclidean triangle. In order to express the complexity and the subtlety of the geometric features, a triangle mesh has thousands to millions of verticies/faces. Comparing to conventional image/video data, the geometric data requires much larger storage.

Unfortunately, most VR/AR applications are executed on mobile devices, such as cell phones, which have limited computation power, small storage and generally low wireless bandwidth. The fundamental conflict between the complex geometric representation and the limited resource on mobile device, hence geometric compression techniques are mandatary for VR/AR applications.



(a) A Smooth Surface (b) A Triangle Mesh

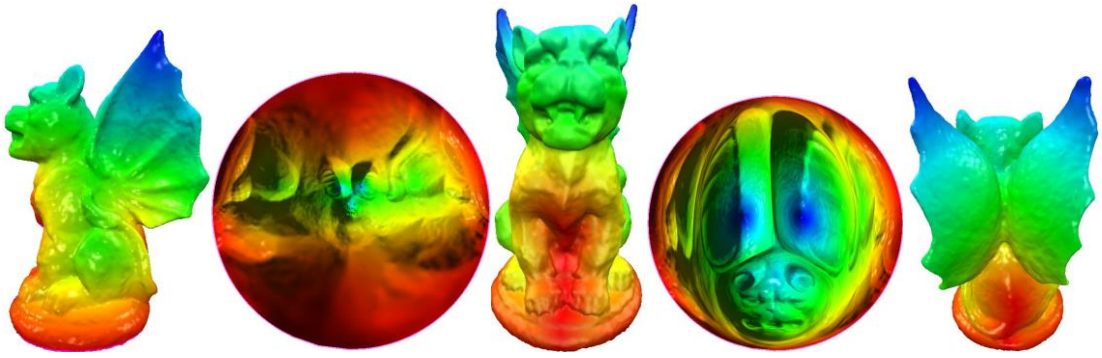
Figure 1: Smooth surfaces are approximated by polygonal surfaces in VR/AR.

As shown in Figure 1, frame (a) shows a C^2 smooth surface, Michelangelo's Kind David head, frame (b) shows a piecewise linear polygonal surface (a triangle mesh). The triangle mesh is

obtained by sampling on the smooth surface, triangulate the sample points, replace each cell by a Euclidean triangle. Evidently, the way of sampling and triangulation will determine approximation accuracy completely. The central goal of the current work is as follows:

Problem 1.1 (Surface Sampling and Tessellation). *How to find rigorous and efficient algorithms for sampling and triangulation to reduce the storage/telecommunication cost and the approximation error?*

In practice, surfaces in real life are acquired by 3D scanning techniques. A surface is sampled with extremely high density. Then the dense point cloud is triangulated to form a high resolution triangle mesh. The high resolution mesh is re-sampled and re-tessellated to reduce the complexity for the geometric compression purpose, this process is called remeshing in the literature.



(a) side view (b) CFP parameterization(c) front view (d) APP parameterization(e) back view

Figure 2: Conformal parameterization (CFP) and area preserving parameterization (APP) for Gargoyle model.

1.2. Key Ideas

There are two main approaches for remeshing: one way is to compute the sampling and triangulation on the 3D mesh directly; the other way is to parameterize the mesh onto a planar domain, and carry out the sampling and triangulation on the 2D domain. The second approach is simpler and more efficient, especially many mature 2D triangulation algorithms can be applied directly. Hence, in the current work, we adopt the remeshing approach based on parameterizations.

Two main issues need to be carefully handled: sampling and triangulation. In our current framework, we address those using different parameterizations: area-preserving parameterization based on optimal mass transportation for sampling, angle-preserving parameterization for

triangulation. The method is based on the following two key observations: the area-preserving parameterization converts the uniform sampling distribution on the surface to the uniform sampling distribution on the parameter domain; the angle-preserving parameterization converts the geodesic Delaunay triangulation on the surface to the Euclidean Delaunay triangulation on the parameter domain. By combining these two key observations, we convert the surface remeshing problem to the planar remeshing problem via two parameterizations.

Fig.2 shows the angle-preserving and area-preserving parameterizations of a gargoyle model.

1.3. Area-Preserving Parameterization

Given a surface S embedded in the three dimensional Euclidean space $\mathbb{E}^3$, it has the induced Euclidean metric g. We can compute a parameterization $\varphi : S \to \mathbb{D}$, where $\mathbb{D}$ is a planar domain. If for any measurable subset of the surface $B \subset S$, the area of B equals to that of $B \subset S$, then the parameterization B is $\varphi(B)$. Area-preserving parameterization can be carried out using optimal mass transportation (OMT) theory.

Suppose $\{p_1, p_2, \dots, p_n\} \subset \mathbb{D}$ is a set of samples on the parameter domain, uniformly distributed, then $\{\varphi^{-1}(p_1), \varphi^{-1}(p_2), \dots, \varphi^{-1}(p_n)\} \subset S$ is a set of uniformly distributed samples on the surface, if the parameterization is area-preserving. Namely, area-preserving parameterization preserves uniform sampling distributions.

Furthermore, given any sampling distribution μ on the surface, we can find a parameterization $\varphi : S \to \mathbb{D}$, such that for any measurable subset $B \subset S$, the measure

$$\int_B \mu(p)\, dp = \int_{\varphi(B)} dx \wedge dy$$

Let $\{p_1, p_2, \dots, p_n\} \subset \mathbb{D}$ be a set of uniform samples on the parameter domain, then $\{\varphi^{-1}(p_1), \varphi^{-1}(p_2), \dots, \varphi^{-1}(p_n)\} \subset S$ is a set of samples on the surface, the sampling distribution is μ. Such kind of parameterization φ can be found using Optimal Mass Transportation as well. This means, by using OMT, we can convert any sampling distribution μ on the surface to the uniform sampling distribution on the parameter domain.

In our current work, we use a variational approach to compute the optimal mass transportation

map [1], which induces the area-preserving parameterization. Then we compute the uniform sampling on the parameter domain, and map these samples to the angle-preserving parameterization domain for the triangulation purpose.

1.4. Angle-Preserving Parameterization

Suppose $P = \{p_1, p_2, \dots, p_n\}$ is a set of samplings on the smooth surface $\{S, g\}$, T is a geodesic triangulation using P as vertex set. If P is dense enough, then each triangle in T has a unique circum-geodesic-circle. If each circum-geodesic-circle doesn't contain any fourth sample in P, then T is called a geodesic Delaunay triangulation.

Suppose $\tau : S \to \mathbb{D}$ is an angle-preserving parameterization, $P = \{p_1, p_2, \dots, p_n\}$ is a sampling set on the parameter domain $P \subset \mathbb{D}$. If T is the Delaunay triangulation of P, then T induces the geodesic Delaunay triangulation of the samples $\tau^{-1}(P) = \{\tau^{-1}(p_1), \tau^{-1}(p_2), \dots, \tau^{-1}(p_n)\}$ on the surface, since Delaunay triangulation maximizes the minimal angle and the parameterization is angle-preserving. Namely, angle-preserving parameterization preserves Delaunay triangulations.

In our current work, we use the dynamic discrete Yamabe flow algorithm [2] to compute the angle-preserving parameterization. The algorithm can handle meshes with very low qualities. Then we use Delaunay refinement method to compute the triangulation, which can guarantee the minimal angle is greater than a positive lower bound.

1.5. Curvature Sensitive

In our current framework, the sampling distribution μ can be prescribed by the user completely. This is very valuable for practical purposes because it can achieve remeshing results, which is curvature sensitive.

In practice, it is preferable to allocate more samples on high curvature regions. This can be achieved by assigning the sampling density μ to equal to a linear combination of the surface area-element $g(u,v)du \wedge dv$ and the absolute value of the surface curvature form $|K(u,v)|du \wedge dv$, where g is the determinant of the metric tensor g and K is the Gaussian curvature.

2. Theoretic Background

In this section, we briefly introduce the theoretic foundation of our framework. We refer readers to [2] and [1] for detailed treatments.

2.1. Conformal Mapping

Suppose (S, g) is a simply connected metric surface, with a single boundary. Suppose a diffeomorphic map from to the surface to a planar domain is $\varphi : S \to \mathbb{D}$, which parameterizes the surface. Assume the local parameters of $\mathbb{D}$ is (u, v), the Riemannian metric has the form

$$\mathbf{g}(u, v) = e^{2\lambda(u,v)}(du^2 + dv^2),$$

where $\lambda : S \to \mathbb{R}$ is a smooth function, then φ is called a conformal mapping. A conformal mapping preserves angles and infinitesimal circles.

2.2. Surface Ricci Flow

Suppose (S, g) is a closed surface with a Riemannian metric g. Hamilton developed the surface Ricci flow, which deforms the Riemannian metric proportional to the Gaussian curvature, such that the curvature evolves according to diffusion-reaction equation, and eventually becomes constant everywhere. More explicitly, Hamilton's surface Ricci flow is defined as:

$$\frac{\partial g_{ij}(p,t)}{\partial t} = 2\left(\frac{2\pi\chi(S)}{A(0)} - K(p,t)\right) g_{ij}(p,t),$$

where the metric tensor $g = (g_{ij})$, K is the Gaussian curvature induced by the current metric, $\chi(S)$ is the Euler characteristic number of the surface, $A(0)$ is the initial total area of the surface.

Hamilton and Chow [3] proved that the surface Ricci flow converges to the constant curvature metric, the constant is $\frac{2\pi\chi(S)}{A(0)}$.

In our current work, we use the discrete surface Ricci flow. The existence of the solution to the flow and the uniqueness of the solution have been proved in [2].

2.3. Optimal Mass Transportation

Let Ω be a convex domain in the Euclidean space $\mathbb{R}^n$. Two probability measures μ and ν are given respectively with equal total measures,

$$\int_{\Omega} \mu = \int_{\Omega} \nu$$

A map $T : \Omega \to \Omega$ is measure preserving if for any measurable set $B \subset \Omega$, the following condition holds: $\int_{T^{-1}(B)} \mu = \int_B \nu$

Let $c(x,y)$ be the transportation cost for transporting $x \in \Omega$ to $y \in \Omega$, then the total transportation cost of T is given by:

$$E(T) = \int_{\Omega} c\big(x, T(x)\big) d\mu(x). \quad (1)$$

In 18th century, Monge [4] raised the optimal mass transportation problem: how to find a measure preserving map T,that minimizes the transportation cost in Eqn.1. In 1940's, Kantorovich [5] has introduced the relaxation of Monge's problem and solved it using linear programming. At the end of 1980's, Brenier [6] has proved the following theorem.

Theorem 2.1 (Brenier) *Suppose the transportation cost is the quadratic Euclidean distance, $c(x,y) = |x-y|^2$. Given probabilities measures μ and ν on a convex domain $\Omega \subset \mathbb{R}^n$, then there is a unique optimal transportation map $T: (\Omega, \mu) \to (\Omega, \nu)$, furthermore there is a convex function $f: \Omega \to \mathbb{R}$, unique up to a constant, and the optimal mass transportation map is given by the gradient map $T: x \mapsto \nabla f(x)$.*

Assume the measures μ and ν are smooth, f is with second order smoothness, $f \in C^2(\Omega, \mathbb{R})$, if f is measure-preserving, then it satisfies the Monge-Ampère equation. The two-dimensional Monge- Ampère equation is as follows:

$$det \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix} = \frac{\mu(x_1, x_2)}{\nu \circ \nabla f(x_1, x_2)} . \quad (2)$$

In general, Monge- Ampère equation is highly non-linear, conventional finite element method is incapable of solving this type of partial differential equations. Instead, [1] introduces a discrete method to solve it based on a convex optimization.

2.4. Curvature Convergence

According to the normal cycle theory [7], given a smooth C^2 surface S embedded in $\mathbb{R}^3$, one can construct a sequence triangle meshes M_k, such that

1. All the mesh vertices on M_k's are sampled from the smooth surface,

2. All the corner angles on the meshes have a uniform positive lower bound $c > 0$

3. The maximal edge length of M_k is ε_k, $\lim_{k\to\infty} \varepsilon_k = 0$

then the discrete Gaussian curvature measure and mean curvature measure converge to their smooth counter parts.

In our current framework, the triangulations are computed on the angle-preserving parameter domain using Delaunay refinement algorithm, by Chew's theoretic result in [8], the minimal corner angles are no less than 30 degree. The samples are computed on the area-preserving parameterization domains, therefore ε_k can be guaranteed to converge to 0. Hence, in theory, the meshes generated by our current pipeline has the curvature measure convergence property.

3. Computational Algorithm

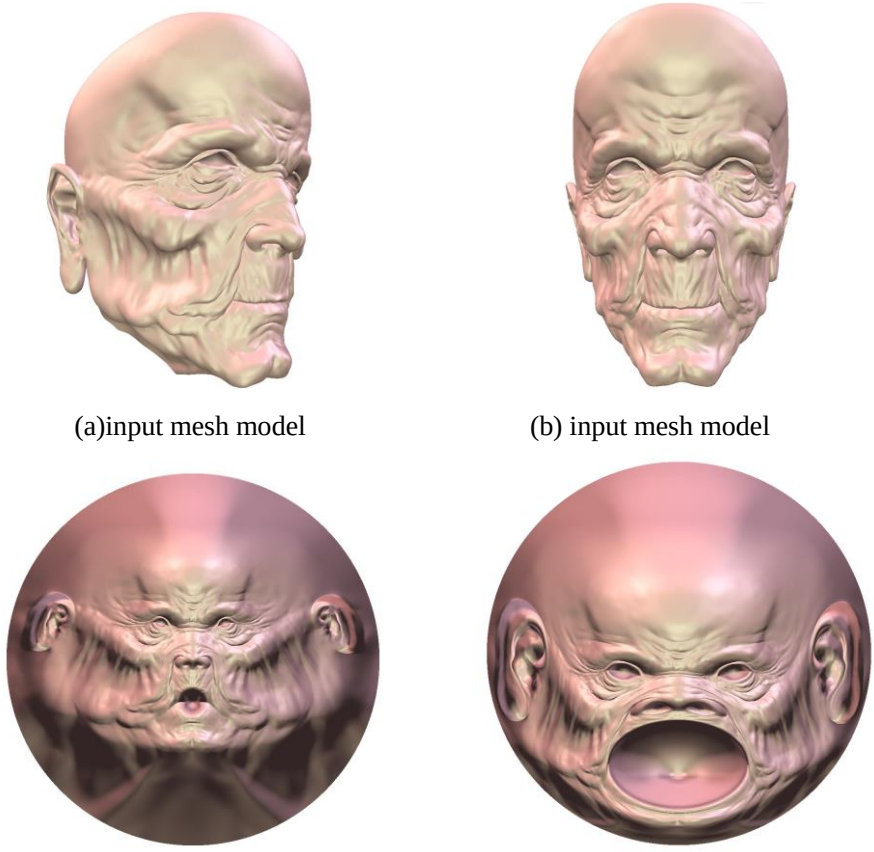
In this section, we explain all the algorithms in details.

3.1. Pipeline

The algorithm pipeline is summarized in Alg.1 and illustrated by Fig.4. The input mesh is a genus zero mesh with a single boundary. The mesh quality of the input mesh could be very low, which doesn't affect our algorithm. The conformal parameterization (CFP) uses the dynamic discrete surface Yamabe flow method [2], which can handle meshes with low meshing qualities. The area-preserving parameterization (APP) is based on the discrete optimal mass transportation using variational approach [1].

The conformal parameterization uses the dynamic discrete surface Yamabe flow method, the details can be found in [2]. In the following, we focus on the optimal transportation map algorithm,

theoretic proofs and more details can be in [1].



(a)input mesh model (b) input mesh model

(c) Conformal parameterization APP domain (d) Area-preserving parameterization CFP domain

Figure 3: Mesh parameterization: frame (a) and (b) shows the input mesh. Frame (c) and (d) show the CFP(Conformal Parameterization) and APP (Area Preserving Parameterization), respectively.

Algorithm 1: Remeshing Algorithm Pipeline.

Input: The input mesh M and the number of samples n

Output: The remeshing result $\widetilde{M}$ with n vertices

1 Compute the Conformal parameterization (CFP);

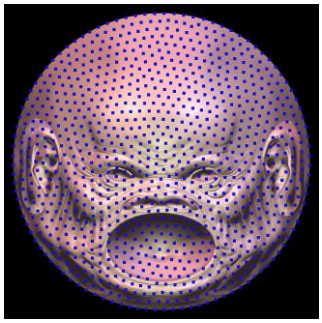
2 Compute the Area preserving parameterization (APP);

3 Sample n points **P** uniformly on APP domain;

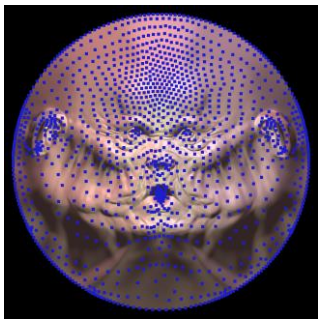
4 Map **P** to CFP domain to get **Q**;

5 Delaunay triangulate **Q** to get a triangulation T;

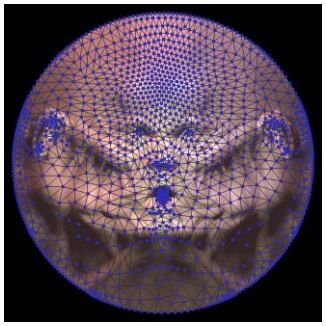
6 Pull back T to the original mesh M to get the final result $\widetilde{M}$;



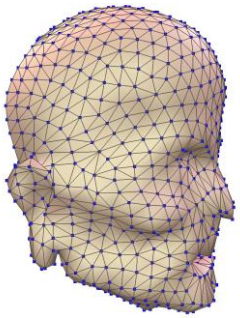
(e) uniform sampling on APP domain



(f) project to CFP domain



(g) triangulate on CFP domain



(h) pull back to get result

Figure 4: Remeshing Algorithm Pipeline: first, we uniformly sample the APP domain, as shown in Frame (e); second, we map the samples onto CFP domain, illustrated by Frame (f); third, we perform the Delaunay Triangulation algorithm on the CFP domain, shown in Frame (g); finally we pull back the triangulation to the original surface, to obtain the remeshing result, illustrated in Frame (h).

3.2. Dynamic Discrete Surface Yamabe Flow

Suppose M is the input mesh with vertex, edge and face sets V, E, F respectively. We use v_i to represent a vertex, $[v_i, v_j]$ the edge connecting v_i and v_j, $[v_i, v_j, v_k]$ the face consisting of the vertices v_i, v_j and v_k.

The edge length of $[v_i, v_j]$ is denoted as l_{ij}, the corner angle at v_i in triangle $[v_i, v_j, v_k]$ is denoted as θ_i^{jk}, θ_i^{jk} can be obtained using cosine law:

$$\theta_i^{jk} = cos^{-1}\frac{l_{ij}^2 + l_{ki}^2 - l_{jk}^2}{2 l_{ij} l_{ki}} \tag{3}$$

The triangulation is Delaunay, if for each edge $[v_i, v_j]$ shared by two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_k]$,

$$\theta_k^{ij} + \theta_l^{ji} \leq \pi.$$

The discrete Gaussian curvature at each vertex is defined as the angle deficit

$$K(v_i)\begin{cases} 2\pi - \sum_{jk}\theta_i^{jk} & v_i \notin \partial M \\ \pi - \sum_{jk}\theta_i^{jk} & v_i \in \partial M \end{cases} \tag{4}$$

It can be easily shown that the discrete Gaussian curvature satisfies the discrete Gauss-Bonnet theorem:

$$\sum_{v_i \in V} K(v_i) = 2\pi\chi(M),$$

where $\chi(\mathrm{M}) = |\mathrm{V}| + |\mathrm{F}| - |\mathrm{E}|$ is the Euler characteristic number of the mesh. The discrete conformal factor is a function defined on the vertex set $\mathrm{u}: \mathrm{V} \to \mathbb{R}$. The edge length is given by

$$l_{ij} = e^{u_i}\beta_{ij}e^{u_j}, \tag{5}$$

where β_{ij} is the initial edge length. Given the target curvature $\overline{\mathrm{K}}: \mathrm{V} \to \mathbb{R}$, such that the target curvature satisfies the discrete Gauss-Bonnet theorem. The discrete Yamabe flow is defined as follows: for each vertex v_i,

$$\frac{\mathrm{du_i}}{\mathrm{dt}} = \overline{\mathrm{K}}(\mathrm{v_i}) - \mathrm{K}(\mathrm{v_i}).$$

Initially, the conformal factor is set to be zero. The edge length induces the curvature K, the flow deforms the conformal factor, changes the edge length, then the curvature in turn. It has been shown that the discrete yambe flow is the gradient flow of the following Yamabe energy

$$E(\mathbf{u}) = \int^{\boldsymbol{u}} \sum_i (\overline{K_\iota} - K_i)du_i,$$

where u is the vector representation of the conformal factors$(\mathrm{u_1}, \mathrm{u_2}, \ldots, \mathrm{u_n})$. The gradient of the Yamabe energy is given by

$$\nabla E(\mathbf{u}) = (\overline{K_1} - K_1, \overline{K_2} - K_2, \ldots \overline{K_n} - K_n)^T. \tag{6}$$

It has been proven in [2] that the Yamabe energy is convex.

Given a triangulation of the vertices of the meshM, we can construct its dual mesh $\overline{\mathrm{M}}$ as follows: for each face $\mathrm{f} \in \mathrm{M}$, its dual is a vertex $\bar{\mathrm{f}} \in \overline{\mathrm{M}}$ which is the circum center of the face; for each edge $\mathrm{e} \in \mathrm{M}$ shared by two faces $\mathrm{f_i}$ and $\mathrm{f_j}$, its dual is an edge $\bar{\mathrm{e}} \in \overline{\mathrm{M}}$ connecting circum centers of $\mathrm{f_i}$ and $\mathrm{f_j}$; for each vertex $\mathrm{v} \in \mathrm{M}$, its dual is a face $\bar{\mathrm{v}} \in \overline{\mathrm{M}}$ consisting of the circum centers of all the neighboring faces. We define the {*edge weight*} as follows: suppose the edge

$[v_i, v_j]$ is shared by two faces $[v_i, v_j, v_k]$ and $[v_j, v_i, v_l]$, then

$$w_{ij} = \cot\theta_k^{ij} + \cot\theta_l^{ji}. \tag{7}$$

For Delaunay mesh, the edge weight is always non-negative. The Hessian matrix for the Yamabe energy can be formulated explicitly as

$$\frac{\partial^2 E(\mathbf{u})}{\partial u_i \partial u_j} = \begin{cases} -w_{ij} & v_i \sim v_j, i \neq j \\ 0 & v_i \nsim v_j, i \neq j \\ \sum_k w_{ik} & i = j \end{cases} \tag{8}$$

From the Hessian formula, it is obvious that if the mesh is Delaunay, then on the hyperplane $\sum_i u_i = 0$, the Hessian matrix is positive definite, therefore the Yamabe energy is strictly convex. The solution is the unique global minimal point.

Given any target curvature $\bar{K}$, satisfying the discrete Gauss-Bonnet theorem, one can use Yamabe flow to find the desired edge length. During the Yamabe flow, it may happen that some triangles are degenerated, therefore the flow has to terminated, the solution cannot be obtained. In order to guarantee the existence of the solution we add one constraint to the flow: during the flow, the triangulation can be modified to be Delaunay in all the time. At each time, the mesh is composed by gluing many Euclidean triangles, in generic cases, there is a unique Delaunay triangulation under this piecewise Euclidean metric, which can be obtained by simple edge swapping algorithm. Therefore, the dynamic Yamabe flow algorithm can be summarized in Alg.2.

Algorithm 2: Dynamic Discrete Surface Yamabe Flow.

Input: The input mesh M and the target curvature $\bar{K}$, threshold ε

Output: The edge length which realizes the target curvature

1 Compute the initial edge lengths $\{\beta_{ij}\}$;

2 Initialize the conformal factor to be zeros;

3 **while** *true* **do**;

4 Compute the edge lengths using Eqn.5;

5 Update the triangulation to be Delaunay by edge swapping;

6 Compute the corner angles using Eqn.3;

7 Compute the edge weights using Eqn.7;

8 Compute the vertex curvature using Eqn.4;

9 **if** $\forall|\overline{K_i} - K_i(\mathbf{h})| < \varepsilon$ **then**

10 **Break;**

11 **end**

12 Compute the gradient of the Yamabe energy using Eqn.6;

13 Compute the Hessian of the Yamabe energy using Eqn.8;

14 Solve the linear system $Hess(\mathbf{u})\delta\mathbf{u} = \nabla E(\mathbf{u})$;

15 $\mathbf{u} \leftarrow \mathbf{u} + \delta\mathbf{u}$;

16 **end**

17 **return** the *edge length* $\{l_{ij}\}$;

The dynamic Yamabe flow can handle meshes with low qualities, the existence of the solution has been proven in [2]. In our current work, we set the target curvature of the interior vertices to be zero everywhere, the target curvatures of the boundary vertices to be constant. After obtaining the target edge length, we can flatten the whole mesh face by face, such that the input simply connected mesh is mapped onto a planar convex domain.

3.3. Optimal Mass Transportation Map

In the current work, the source domain Ω is the canonical convex domain in $\mathbb{R}^2$, the target is a set of discrete points $Y = \{q_1, q_2, \dots, q_k\}$ which densely samples Ω. The source measure on Ω is the uniform measure $\mu = 1$ everywhere. The target measure on Y is prescribed by the user, $\nu = \{\nu_1, \nu_2, \dots, \nu_k\}$, such that $\sum_{i=1}^{k} \nu_i$ equals to the total area of Ω.

For each target point $q_i \in Y$, we construct a hyperplane in $\mathbb{R}^3$, $\pi_i(h, p) \coloneqq \langle q_i, p\rangle + h_i, i = 1,2, \dots, k$. Then we compute the upper envelope of these hyper-planes.

3.3.1. Power Voronoi Diagram and Power Delaunay Triangulation

For each hyperplane $\pi_i(h)$, we construct a dual point $\pi_i^*(h) \in \mathbb{R}^3$ as follows: assume the coordinates of $q_i \in \mathbb{R}^2$ are (x_i, y_i), then the dual point is $\pi_i^*(h) = (x_i, y_i - h_i)$, $i = 1, 2, \dots, k$.Then we compute the convex hull of $\{\pi_1^*(h), \pi_2^*(h), \dots, \pi_k^*(h)\}$ using incremental convex hull algorithm as described in [9], and denote the resulting convex hull as $\mathfrak{C}(h)$. The boundary faces

of $\mathfrak{C}(h)$, whose normals are pointing downwards, form the lower part of the convex hull. We project the lower part of the convex hull $\mathfrak{C}(h)$ to produce the power Delaunay triangulation of the point set Y, denoted as $\mathfrak{T}(h)$.

The upper envelope of the hyperplanes $\{\pi_i(h)\}$ is denoted as $\mathfrak{E}(h)$, which is the dual to the lower part of the convex hull $\mathfrak{C}(h)$. We project the upper envelope onto the (x,y)-plane to obtain the power Voronoi diagram of the plane, each power Voronoi cell intersects Ω to obtain the power Voronoi cell decomposition of Ω, $\Omega = \cup W_i(h)$, denoted as $\mathfrak{V}(h)$

In fact, the upper envelope $\mathfrak{E}(h)$ is exactly the graph of the convex function

$$G(\mathbf{h},\mathbf{p}) = \max_{1\le i\le k}\{\pi_i(\mathbf{h},\mathbf{p})\}, \tag{9}$$

the power Voronoi diagram is the polyhedral partition of Ω by the gradient map of $G(\mathbf{h})$, $p \mapsto \nabla G(\mathbf{h})$.

3.3.2. Volume Energy

Let the area of each cell $W_i(\mathbf{h})$ in the power Voronoi cell decomposition $\mathfrak{V}(\mathbf{h})$ be denoted as $w_i(\mathbf{h})$.We define the *admissible space* of the height vector as follows:

$$H := \{\mathbf{h} | \textstyle\sum_{i=1}^{k} h_i = 0, \forall 1 \le i \le k, w_i(\mathbf{h}) > 0\}. \tag{10}$$

It can be proven that the admissible space is convex, details can be found in [1].

Furthermore, we define the volume energy on the admissible space as follows:

$$E(\mathbf{h}) = \int^{\mathbf{h}} \textstyle\sum_{i=1}^{k} (\nu_i - w_i(\eta)) d\eta_i. \tag{11}$$

The gradient of the energy is the difference between the target measure and the current cell area

$$\nabla E(\mathbf{h}) = (\nu_1 - w_1(\mathbf{h}), \nu_2 - w_2(\mathbf{h}), \dots, \nu_k - w_k(\mathbf{h}))^T. \tag{12}$$

We define the edge weight of the Power voronoi cell decomposition as $\mathfrak{V}(h)$ as follows: suppose $W_i(\mathbf{h})$ and $W_j(\mathbf{h})$ two adjacent cells, intersecting at the edge $e_{ij}(\mathbf{h})$

$$\lambda_{ij}(\mathbf{h}) = \frac{|e_{ij}(\mathbf{h})|}{|p_i - p_j|} \tag{13}$$

The Hessian matrix of the volume energy is given by

$$\frac{\partial^2 E(\mathbf{h})}{\partial h_i \partial h_j} = \begin{cases} -\lambda_{ij}(\mathbf{h}) & W_i(\mathbf{h}) \sim W_j(\mathbf{h}) \quad i \ne j \\ 0 & W_i(\mathbf{h}) \nsim W_j(\mathbf{h}) \quad i \ne j \\ \sum_k \lambda_{ik}(\mathbf{h}) & i = j \end{cases} \tag{14}$$

Because the edge weight is always positive, so the volume energy is positive definite in the admissible space. The global maximizer of volume energy gives the power Voronoi cell decomposition $\mathfrak{V}(\mathbf{h})$, the area of each cell $\mathrm{W_i}(\mathbf{h})$ equals to the desired measure ν_i. Furthermore, the mapping $\mathrm{W_i}(\mathbf{h}) \mapsto \mathrm{q_i}$ is the gradient map of the convex function $\mathrm{G}(\mathbf{h})$, according to Brenier theorem, this mapping is the optimal mass transportation map.

Optimal Transportation Map In our current setting, the discrete point set Y is contained in the unit disk Ω. The initial height vector is set as follows:$h_i = \frac{1}{2}\langle \mathbf{q}_i, \mathbf{q}_i \rangle$, $i = 1,2,\dots,k$. The initial power Delaunay triangulation $\mathfrak{T}(\mathbf{h})$ is the traditional Delaunay triangulation, the power Voronoi cell decomposition of the unit disk is the traditional Voronoi cell decomposition.

At each step, we compute the power Delaunay triangulation $\mathfrak{T}(\mathbf{h})$ and the power Voronoi cell decomposition $\mathfrak{V}(\mathbf{h})$. The gradient of the volume energy in Eqn.11 is given in Eqn.12, the Hessian of the volume energy is given by Eqn.14. Then we solve the following linear equation

$$\nabla E(\mathbf{h}) = \left(\frac{\partial^2 E(\mathbf{h})}{\partial h_i \partial h_j}\right) \delta\mathbf{h} \tag{15}$$

with the linear constraint $\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{h_i} = 0$, the solution exists and is unique. Then we can update the height vector by using Newton's method $\mathbf{h} \leftarrow \mathbf{h} + \lambda(\delta\mathbf{h})$, where λ is the step length parameter. In theory, the step length parameter should be chosen such that the height vector is kept inside the admissible space H (Eqn.10), namely, in the power Voronoi cell decomposition $\mathfrak{V}(\mathbf{h})$, each cell $\mathrm{W_i}(\mathbf{h})$ is non-empty. In practice, in the middle of the optimization, we allow h to exceed the admissible space H. The convexity of the volume energy automatically guides the height vector to return to the admissible space. The details of the algorithm can be found in Alg.3.

Algorithm 3: Discrete Optimal Mass Transportation Map

Input: A convex domain $\Omega \subset \mathbb{R}^2$ and a set of discrete points $Y = \{\mathbf{q}_1, \cdots, \mathbf{q}_\mathrm{n}\}$,

discrete target measure $\nu = \{\nu_1, \dots, \nu_\mathrm{n}\}$, such that $\sum_i \nu_i = Area(\Omega)$

Output: A partition of Ω, $\Omega = \cup_i W_i$, such that $W_i \mapsto \mathbf{q}_i$ is the optimal mass transportation map.

1 Translate and scale Y, such that $Y \subset \Omega$

2 Initialize the height vector $\mathbf{h}$, such that $\mathrm{h_i} \leftarrow 1/2\,\langle \mathbf{q}_i, \mathbf{q}_i \rangle$

3 **while** *true* **do**;

4 **for** $i \leftarrow 1$ to k **do**

5 Construct the plane $\pi_i(\mathbf{h}) \coloneqq \langle \mathbf{q}_i, \mathbf{p} \rangle + h_i$

6 Compute the dual point of the plane $\pi_i^*(\mathbf{h})$

7 **end**

8 Construct the convex hull $\mathfrak{C}(\mathbf{h})$ of the dual points $\{\pi_i^*(\mathbf{h})\}$

9 Compute the dual of the convex hull to obtain the upper envelope $\mathfrak{E}(\mathbf{h})$ of the planes $\{\pi_i(\mathbf{h})\}$

10 Project $\mathfrak{C}(\mathbf{h})$ to obtain the power Delaunay triangulation $\mathfrak{T}(\mathbf{h})$ of Y

11 Project $\mathfrak{E}(\mathbf{h})$ to obtain the power Voronoi cell decomposition $\mathfrak{V}(\mathbf{h})$ of Ω

12 **for** i $\leftarrow 1$ to k **do**

13 Compute the area of $W_i(\mathbf{h})$, denoted as $w_i(\mathbf{h})$

14 **end**

15 Construct the gradient Eqn.12

16 Construct the Hessian matrix Eqn.14

17 Solve the linear equation $\mathrm{Hess}(\mathbf{h})\delta\mathbf{h} = \nabla E(\mathbf{h})$

18 $\lambda \leftarrow 1$

19 Compute the power Voronoi diagram $\mathfrak{A}(\mathbf{h} + \lambda\delta\mathbf{h})$ of Ω

20 **while** $\exists w_i(\mathbf{h} + \lambda(\delta\mathbf{h}))$ is *empty* **do**

21 $\lambda \leftarrow 1/2\,\lambda$

22 Compute the power Voronoi diagram $\mathfrak{A}(\mathbf{h} + \lambda\delta\mathbf{h})$ of Ω

23 **end**

24 $\mathbf{h} \leftarrow \mathbf{h} + \lambda(\delta\mathbf{h})$

25 **if** $\forall |w_i(\mathbf{h}) - \nu_i| < \varepsilon$ **then**

26 **Break**

27 **end**

28 **end**

29 **return** *the mapping* $\{W_i(\mathbf{h}) \mapsto \mathbf{q}_i, i = 1,2,\cdots,k\}$

3.4. Measure Controllable Parameterization

<<003>>

In our current framework, given a triangle mesh M, which is of genus zero and with a single boundary, we first scale the mesh, such that its total area equals to π, then compute a conformal map from the mesh to the planar unit disk, $\varphi: M \to \mathbb{D}$. For each vertex$v_i \in M$, we define the target measure ν_i as follows:

$$\nu_i := \frac{1}{3}\sum_{jk} A([v_i, v_j, v_k]), \tag{16}$$

where $A(\cdot)$ is the area of the triangle in $\mathbb{R}^3$. Then we compute an optimal mass transportation map from the unit disk with the uniform measure to the discrete point set $\{\varphi(\nu_1), \varphi(\nu_2), \dots, \varphi(\nu_n)\}$ with discrete measures$\{\nu_1, \nu_2, \dots, \nu_n\}$. That means we find a Power voronoi cell decomposition of $\mathbb{D}$, $\mathbb{D} = \cup W_i$, and each cell W_i is mapped to $\varphi(\nu_i)$. Suppose the mass center of each W_i is c_i, then the piecewise linear mapping from the original mesh to the disk, $\nu_i \mapsto c_i$, gives the discrete area-preserving parameterization. By varying the target measure $\{\nu_i\}$, we can achieve measure-controllable parameterization.

4. Experiment

We implemented proposed algorithms in Matlab and C++. All the experiments are carried out on a Windows laptop with 2.3GHz dual core CPU and 8GB memory. We report our results in the following subsections, which demonstrate that our algorithm allows users to control the sampling distribution and produce high quality meshes.

4.1. Different Sampling and Triangulation Strategies on APP and CFP Domains

In this subsection, we compare the remeshing results produced by different sampling/triangulation strategies on APP and CFP domains, which is summarized in Fig.5. We have tested four combinations, both sampling and triangulation on CFP, both sampling and triangulation on APP, sampling on CFP/triangulation on APP, and sampling on APP/triangulation on CFP. From the experiment, we can see that the last strategy outperforms all the others.

The number of sample points on the gargoyle model is 5k. The same mesh is illustrated in (top row), on the CFP domain (2nd row) and on the APP domain (3rd row).

4.1.1. Sampling/Triangulation on CFP

The first column in Fig.5. shows the remeshing result obtained by both sampling and triangulating on the CFP domain.

As shown in the CFP domain frame (e), the triangulation is both uniform and well-shaped (triangles are close to equilateral). Frame (a) shows the mesh in three dimensional Euclidean space, the triangulation is well-shaped, but highly non-uniform.

4.1.2. Sampling/Triangulation on APP

The second column in Fig.5. shows the remeshing result obtained by both sampling and triangulating on the APP domain. The result mesh in (b) is not well-shaped but uniform.

4.1.3. Sampling on CFP/Triangulation on APP

The third column in Fig.5. shows the remeshing result obtained by sampling on CFP domain and triangulating on APP domain. The result mesh in (c) neither well-shaped nor uniform. This strategy gives the worst remeshing result.

4.1.4. Sampling on APP/Triangulation on CFP

The fourth column in Fig.5. shows the remeshing result obtained by sampling on APP domain and triangulating on CFP domain. The result mesh in (d) both well-shaped and uniform. This strategy gives the best remeshing result.

This experiment shows our key insights: the area-preserving parameterization preserves the uniform sampling distribution; the angle-preserving parameterization preserves the Delaunay property of the triangulation.

4.2. Curvature Sensitive Remeshing

In practice, it is highly desirable to allocate denser samples in the regions with higher curvatures, and sparser samples in the flatter areas. This requires our remeshing algorithm to be sensitive to the curvature.

We can easily achieve this by modifying the target measure by adding the absolute value of the

<<003>>

discrete Gaussian curvature to Eqn.16:

$$\mathrm{v_i} \coloneqq \frac{1}{3}\sum_{jk} A\big([v_i, v_j, v_k]\big) + |K(v_i)|, \tag{17}$$

which we call area-curvature target measure.

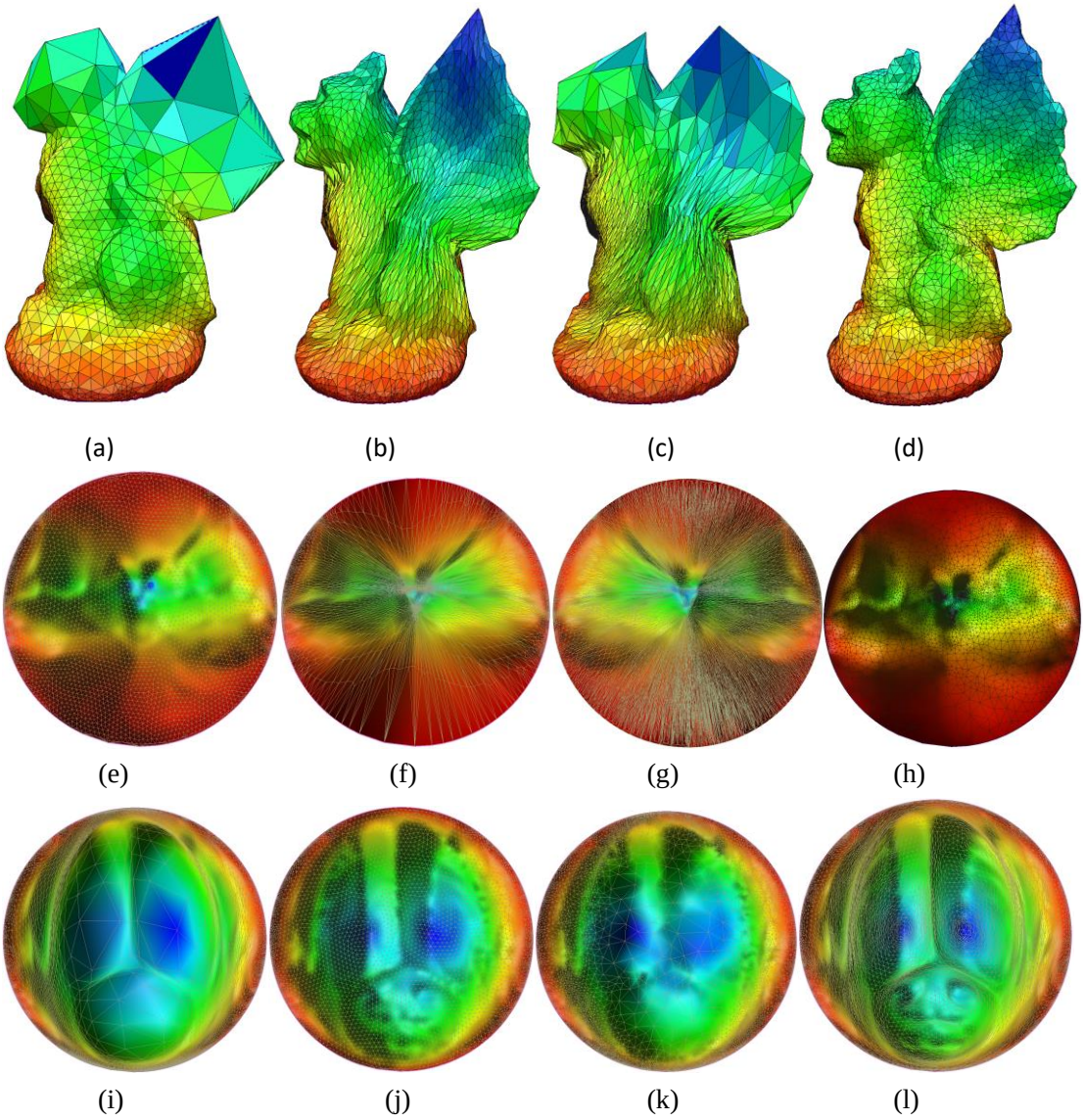
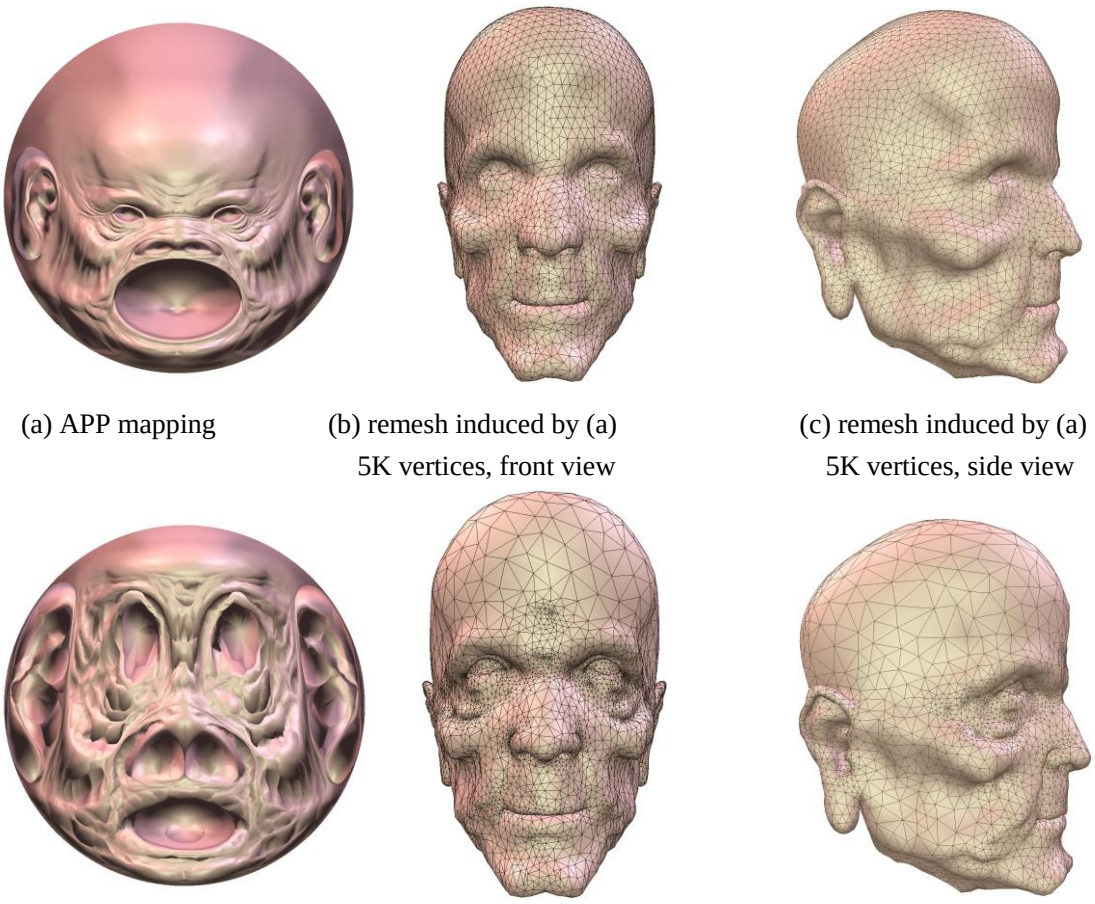


Figure 5: Comparison among different remeshing strategies, by sampling/triangulating on CFP/APP domains. The total number of sample points is 5k. The same mesh is illustrated in the Euclidean space (top row), on the CFP domain (2nd row) and on the APP domain (3rd row).The first column, both sampling and triangulation are computed on the CFP domain; the 2nd column, both sampling and triangulation on APP domain; the 3rd column, sampling on CFP domain, triangulation on APP domain; the last column, sampling on APP domain, triangulation on CFP domain.

Fig. 6 demonstrates our curvature sensitive remeshing result. The original mesh has 140k vertices, whereas the remeshed one has only 5k vertices. The area-preserving parameterization is

<<003>>

shown in (a), and the remeshing result is shown in (b) and (c), where the samples are uniformly sampled with respect to the surface area. In contrast, the measure-controllable parameterization with area-curvature target measure in Eqn.17 is shown in frame (d), the remeshing result based on (d) is shown in (e) and (f). It is obvious that, more samples are allocated for high curvature regions. Furthermore, by carefully examining the eye, nose and ear regions, we can see the curvature sensitive remeshing better preserves the geometric details.



(a) APP mapping

(b) remesh induced by (a) 5K vertices, front view

(c) remesh induced by (a) 5K vertices, side view

(d) parameterization using area-curvature target measure

(e) remesh induced by (d) 5K vertices, front view

(f) remesh induced by (d) 5K vertices, side view

Figure 6: Comparison between remeshing algorithms based on area-preserving parameterization (top row) and the curvature sensitive parameterization (bottom row).

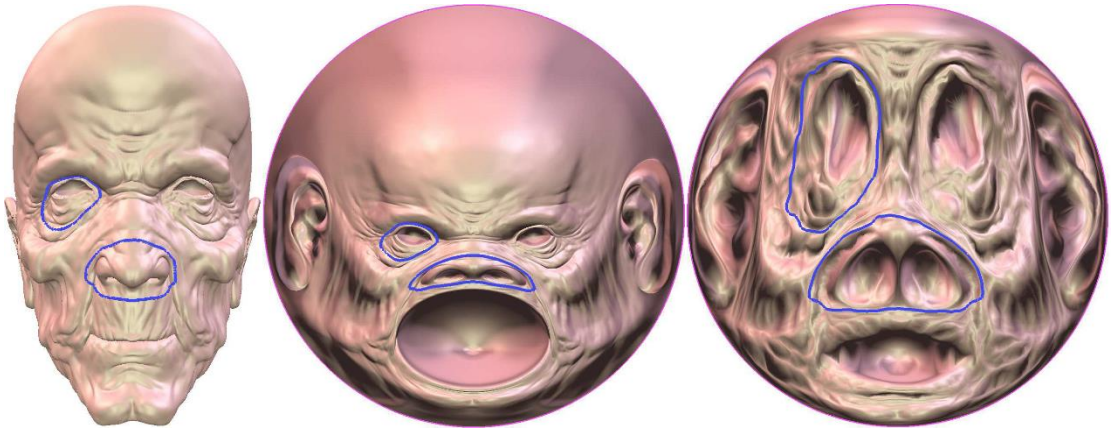


Figure 7: The nose and right eye regions are selected for testing the curvature convergence rate.

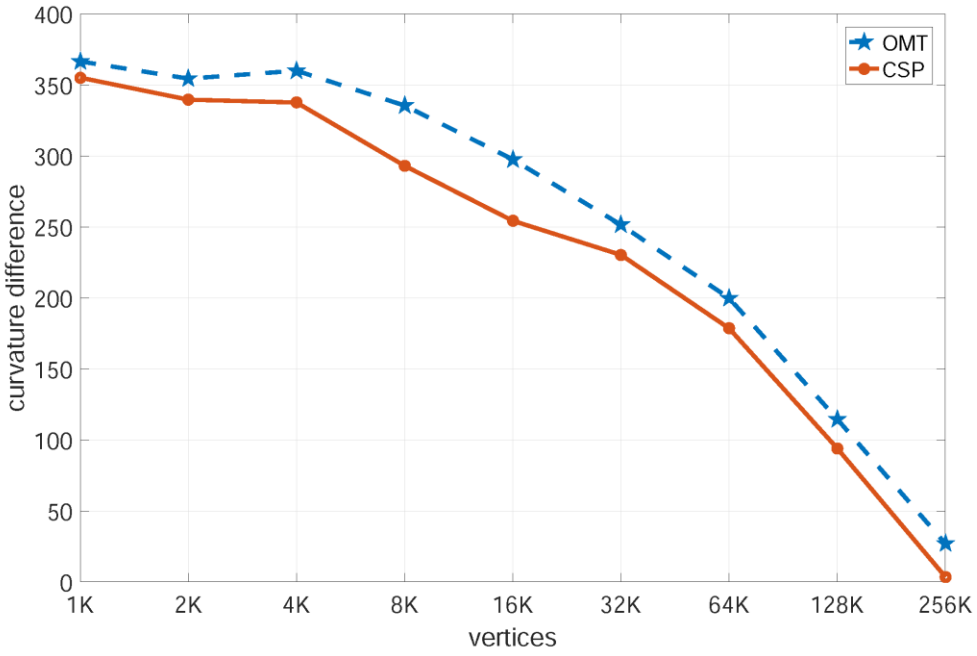


Figure 8: The curvature measure differences on the nose region.

<<003>>

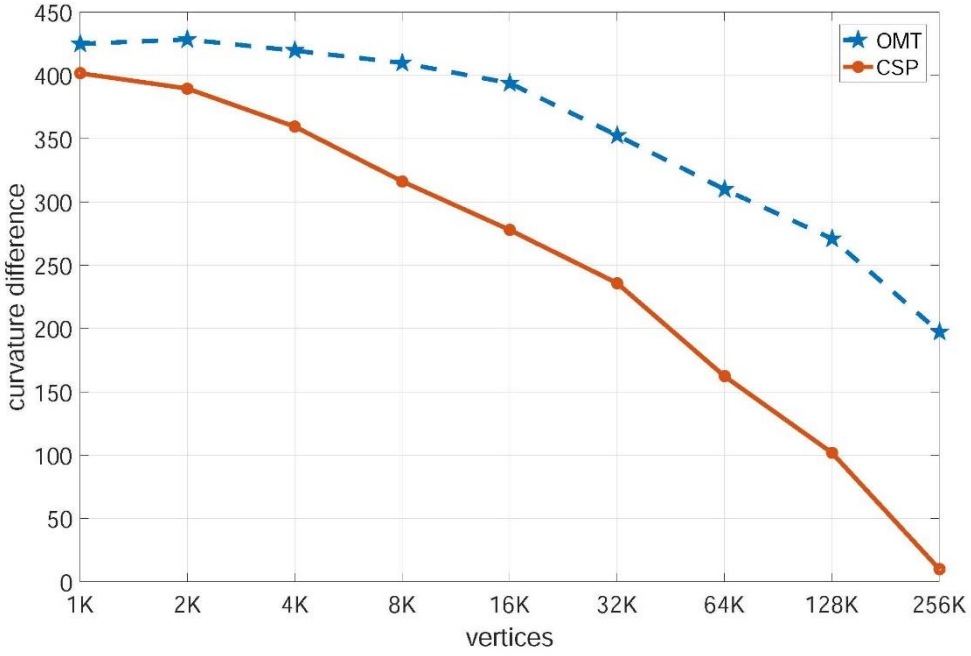


Figure 9: The curvature measure differences on the right eye region.

Figure 7 shows two regions selected for testing the curvature measure convergence rate, the nose region and the right eye region. Figure 8 and 9 show the curvature measure errors for the nose region and eye region respectively. The horizontal axis is the number of sampling vertices, the vertical axis is the curvature measure difference between the original high resolution mesh and the remeshed surface. The blue curve is obtained by sampling on the area preserving parameterization, the red curve is obtained by sampling on the curvature sensitive parameterization. It is obvious that curvature sensitive remeshing algorithm achieves higher curvature measure convergence rate.

4.3. Multi-Resolution Remeshing

In In 2016 summer, our team collected the dynamic facial surfaces using real time high speed and high precision 3D system. About 85 people's faces are acquired, each person has about 800 facial surfaces with different expressions, therefore the total number of facial surfaces is about 70000. The facial database is used to test the algorithm.

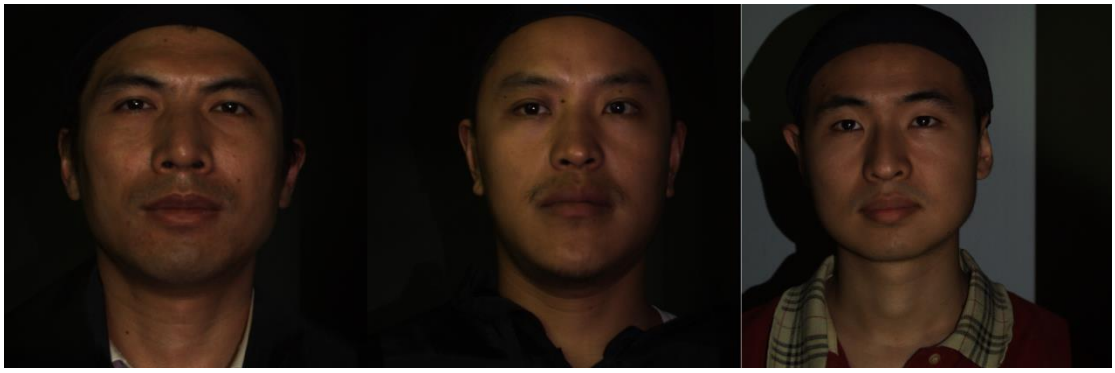


Figure 10: Human facial database, including 85 people, 70000 3D facial surfaces.

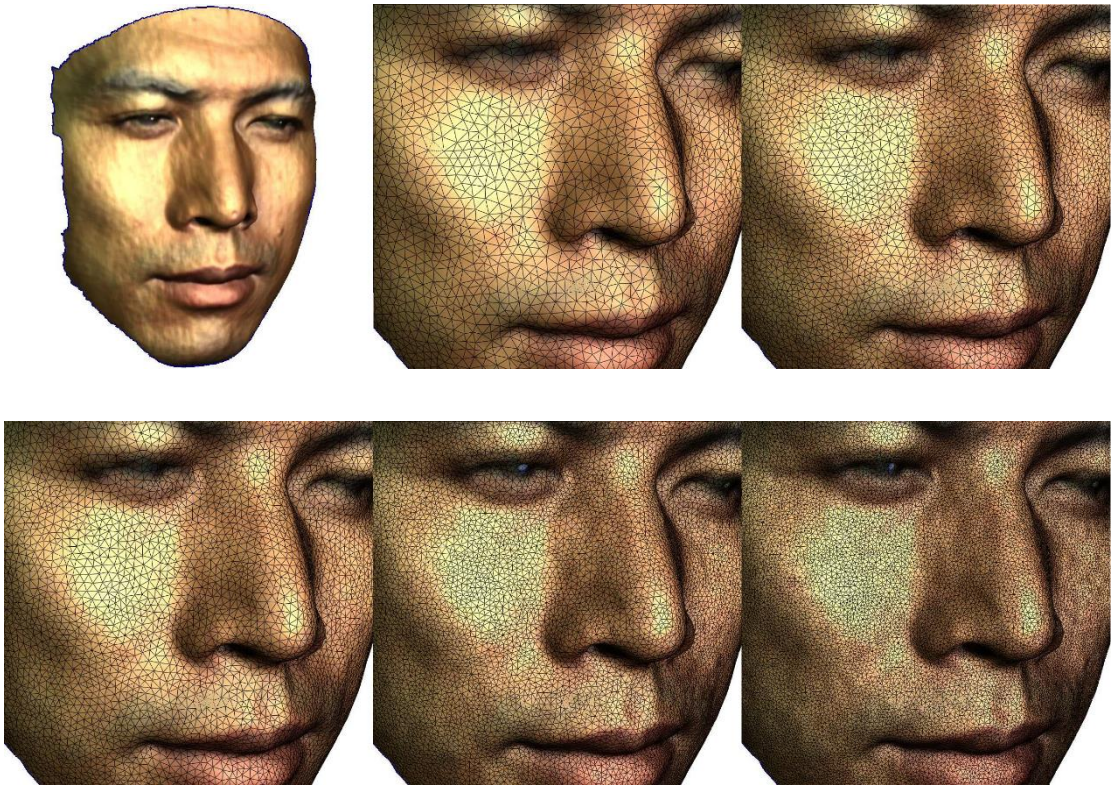


Figure 11: Multiresolution remeshing

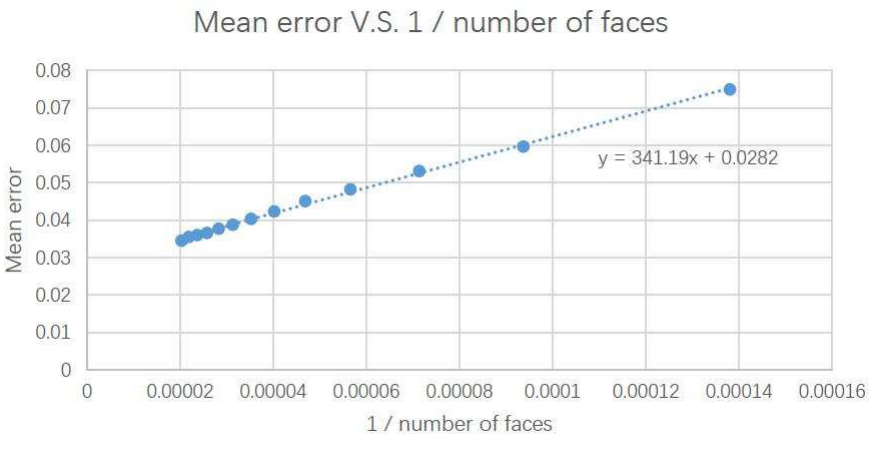
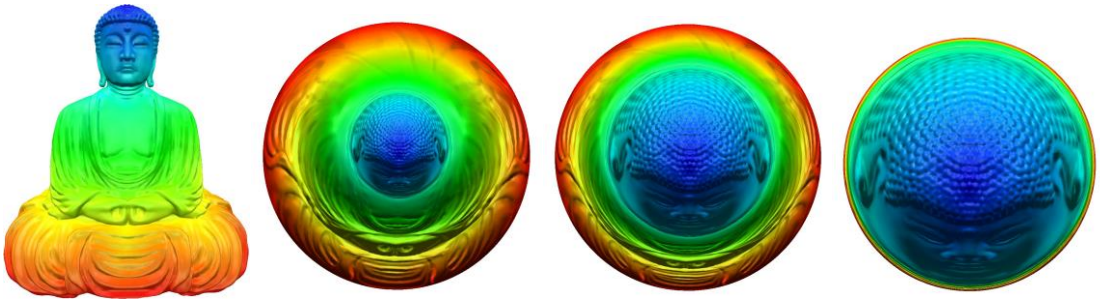


Figure 12. Hausdorff error vs. the inverse of the number of faces on a mesh.

Figure 11 shows our multi-resolution remeshing results for a male 3D facial surface. The number of triangles on each mesh ranges from 5k to 50k. We measure the Hausdorff distance between the original facial mesh obtained by scanning and the remeshed surface, then take the average among all the facial meshes. We conducted the experiments for more than 50 people, and thousands of faces. Figure 12 shows the relation between the inverse of number of triangles and the mean Hausdorff error, which is almost linear. This is consistent with the theoretic result in geometric approximation, that suppose the remeshed surface is with positive lower bound of all the corner angles, the Hausdorff distance is proportional to the square of the edge length [7,10].

4.4. Remeshing Adapted to Region of Interest

Our method can allow user to choose region of interests (ROI), and determines the sampling density distribution. As shown in Figure7, the head region of the Buddha model is chosen as the ROI. By specifying the target measure, the measure-preserving mappings induced by our OMT algorithm magnify the ROI with different factors. By uniformly sampling on the parameter domain, we obtain different remeshing results.



(a) original Buddha model (b) APP (c) ROI based mapping head× 4 (d) ROI based mapping head ×16

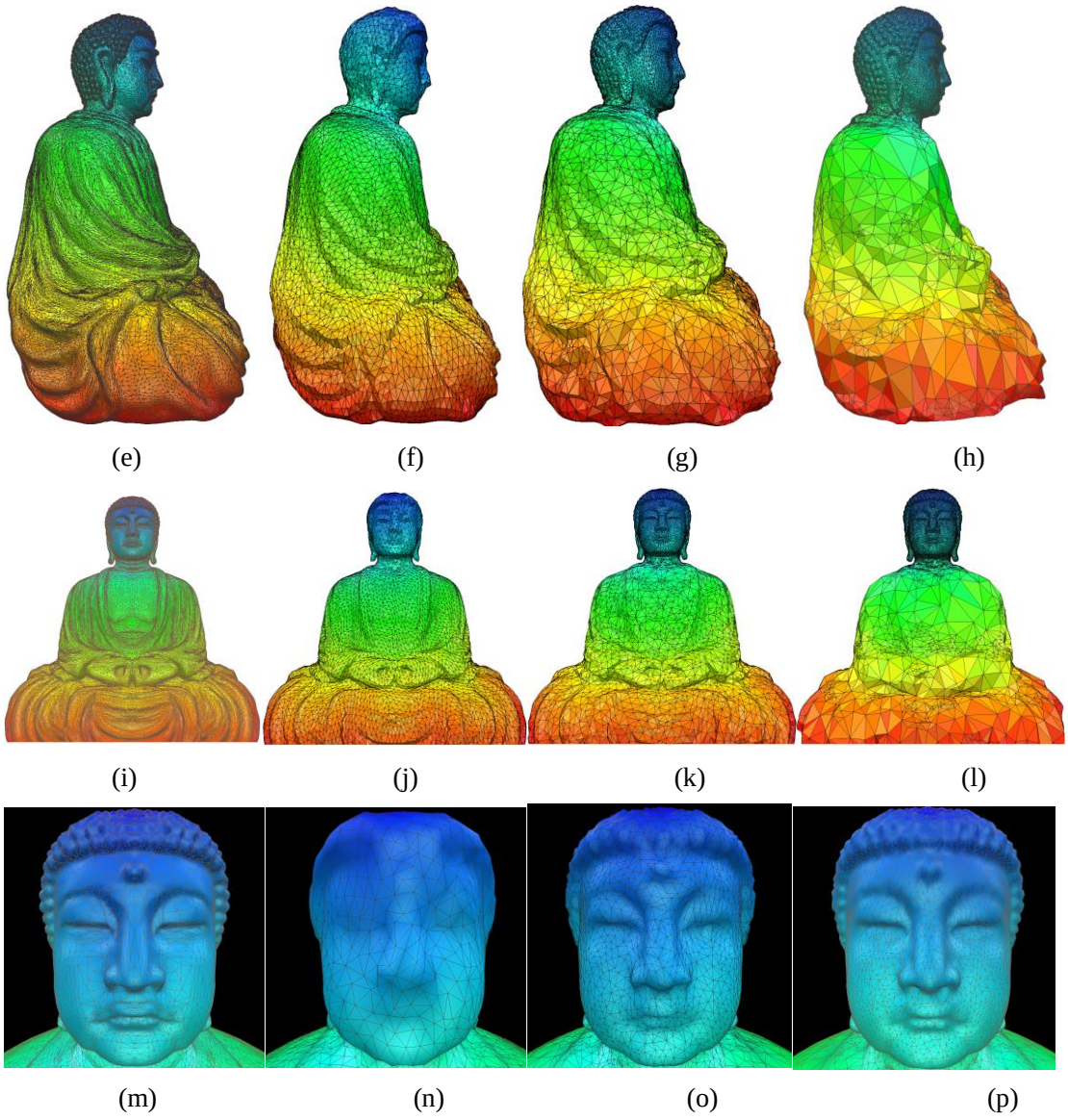


Figure 7: ROI based remeshing for the Buddha model. The input buddha model is illustrated in the 1st column, which has about 700k vertices. The APP based remeshing results are shown in the 2nd column, the ROI based remeshing results are shown in the 3rd and 4th columns. Each remeshed model has about 10k vertices. The head of the buddha is selected as ROI region (in blue). The OMT based parameterizations are shown in the top row, where the ROI is enlarged by different scaling factors (1,4, 16 in (b), (c) and (d) respectively). The remeshing results induced by different sampling distributions are shown in different columns. The 2nd column shows the remeshing result using uniform sampling distribution, the 3rd and 4th column show the results with denser samplings in ROI.

5. Conclusions

This work proposes a novel framework for remeshing based on angle-preserving parameterization and measure-controllable parameterization based on optimal mass transportation. The angle-preserving parameterization preserves Delaunay triangulation, the measure-controllable parameterization converts any sampling distributions on the surface to uniform distribution on the parameter domain. The remeshing method can be curvature sensitive, and emphasize the region of interests. Our experimental results demonstrate the efficiency and efficacy of the proposed method.

Current method cannot handle surfaces with complicated topologies. In the future, we will generalize the proposed remeshing method for surfaces with more complicated topologies.

References:

[1] X. Gu, F. Luo, J. Sun, S.-T. Yau, Variational principles for minkowski type problems, discrete optimal transport, and discrete Monge-Ampère equations, Asian Journal of Mathematics(AJM) 2 (20) (2016) 383–398.

[2] X. Gu, F. Luo, J. Sun, T. Wu, A discrete uniformization theorem for polyhedral surfaces, arXiv:1309.4175.

[3] W. Zeng, X. D. Gu, Ricci flow for shape analysis and surface registration: theories, algorithms and applications, Springer Science & Business Media, 2013.

[4] N. Bonnotte, From knothe's rearrangement to brenier's optimal transport map, SIAM Journal on Mathematical Analysis 45 (1) (2013) 64–87. 300 [5] L. Kantorovich, On a problem of monge, Uspekhi Mat. Nauk. 3 (1948) 225–226.

[6] Y. Brenier, Polar factorization and monotone rearrangement of vector-valued functions, Communications on Pure and Applied Mathematics 44 (4) (1991) 375–417.

[7] H. Li, W. Zeng, J. M. Morvan, L. Chen, X. D. Gu, Surface meshing with curvature convergence, IEEE Transactions on Visualization and Computer Graphics 20 (6) 305 (2014) 919–934.

[8] L. P. Chew, Guaranteed-quality triangular meshes, Tech. rep., DTIC Document (1989).

[9] M. de Berge, O. Cheong, M. van Kreveld, M. Overmars, Computational Geometry: Algorithm and Application, 3rd Edition, Springer-Verlag, 2008.

[10] Huibin Li, Wei Zeng, Jean-marie Morvan, Limin Chen, Surface Meshing with Curvature Convergence, IEEE Transaction on Visualization and Computer Graphics (TVCG), 20(6):919-934, 2014.