

# Mesh Parameterization : a Viewpoint from Constant Mean Curvature Surfaces

Hui Zhao<sup>1</sup> <sup>†</sup> Kehua Su<sup>2</sup> <sup>‡</sup> Chenchen Li<sup>2</sup> Boyu Zhang<sup>5</sup> <sup>§</sup> Shirao Liu<sup>1</sup> Lei Yang<sup>1</sup> Na Lei<sup>4</sup> Steven J. Gortler<sup>5</sup> <sup>¶</sup> Xianfeng Gu<sup>3</sup> <sup>||</sup>

<sup>1</sup> Capital Normal University, China <sup>2</sup> Wuhan University, China

<sup>3</sup> State University of New York at Stony Brook, USA <sup>4</sup> Dalian University of Technology, China <sup>5</sup> Harvard University, USA

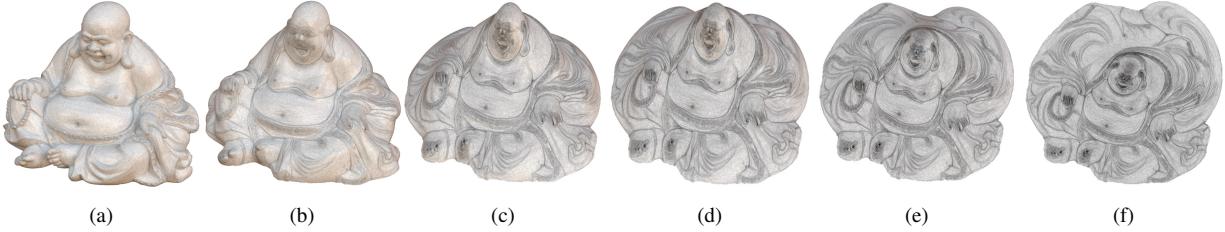


Figure 1: The Buddha model deforms to a planar mesh.

## Abstract

We present a unified mesh parameterization algorithm for both planar and spheric domains based on mesh deformation. Unlike previous methods, our approach can produce intermediate frames from the original to target meshes. We derive and define a novel geometric flow: **unit normal flow (UNF)** and prove that if unit normal flow converges, it will deform a surface to a constant mean curvature (CMC) surface, such as plane and sphere. Our method works by deforming meshes of disk topology to planes, meshes of spheric topology to spheres. Our algorithm is robust, efficient, simple to implement and scales to models with millions of faces. To demonstrate the robustness and effectiveness of our method, we apply it to hundreds of models of varying complexity. Our experiments show that our algorithm compares favourably to other state-of-the-art mesh parameterization methods and the mapping results have few of flips. The unit normal flow we propose also suggests a potential direction for creating CMC surfaces.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modelling—Curve, surface, solid, and object representations

**Keywords:** mesh parameterization, deformation, constant mean curvature, rotation, unit normal flow

## 1. Introduction

In this paper, we present a simple and novel algorithm of planar and spheric mesh parameterization. Our methodology is different from previous ones: we are not computing a direct embedding of a mesh onto planar or spheric domain, instead we deform it towards planar and spheric shape. Our method unifies planar and spheric mesh

parameterization into a single framework, which consists of the iterations of two steps: average of face normals and surface deformation. For meshes of disk or sphere topology, they will converge to planar or spheric shapes automatically under the iterations. Figure 1, 2 and 3 demonstrate the deformations and their planar and spheric parameterizations respectively. Our approach produces locally injective mappings in practice, although we do not prove it theoretically.

The design methodology of major parameterization algorithms in research community obtains the mapping by minimizing some kinds of energies, which are often based on the measurements of angle or area distortions. The conformal and area-preserving parameterizations are computed independently and separately. Then

<sup>†</sup> e-mail:alanzhaohui@qq.com

<sup>‡</sup> e-mail:skh@whu.edu.cn

<sup>§</sup> e-mail:bzhang@math.harvard.edu

<sup>¶</sup> e-mail:sig@cs.harvard.edu

<sup>||</sup> e-mail:gu@cs.stonybrook.edu

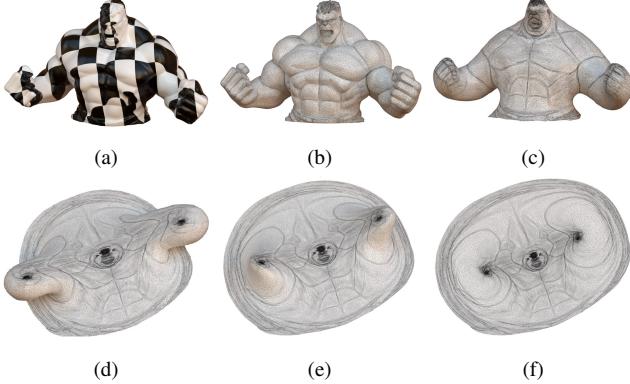


Figure 2: (a) is the texturing; (b) is the original mesh; (c),(d),(e) are the intermediate deforming frames; (f) is the final planar mapping.

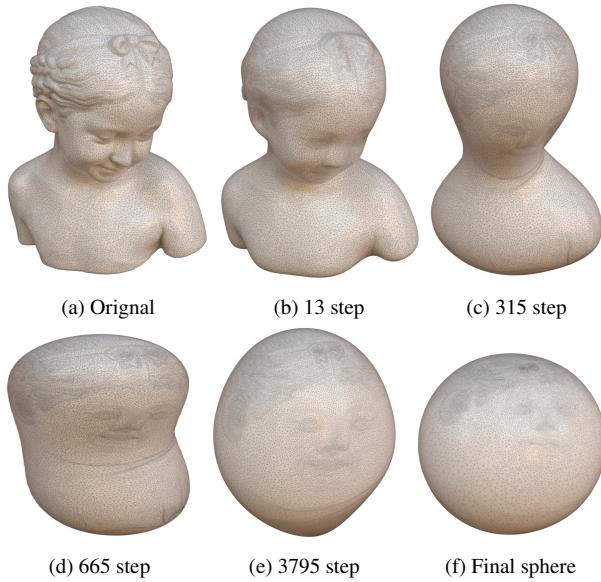


Figure 3: The deformation of bimba model and its spheric parameterization.

the mixed result can be achieved by combining them. Although conformal mapping can preserve angles, it results in huge local area change, and the area-preserving one makes shapes distorted. The ideal solution for texturing application should have both smaller area and angle distortions simultaneously. How to balance these two kinds of distortion is a complicated problem. Intuitively the balance should be affected by the shape of a surface locally and globally.

Firstly we compute the new normal of every point by averaging the normals of its neighbours. Secondly we reconstruct a surface which fits the current normals. The iterations of these two steps emerge a heat-like geometric flow on surfaces. We call it **unit normal flow(UNF)**. Honestly, in this paper, we do not give a proof of convergence of unit normal flow. What we prove is, on a smooth surface, if the flow converges, it will produce a CMC surface, such

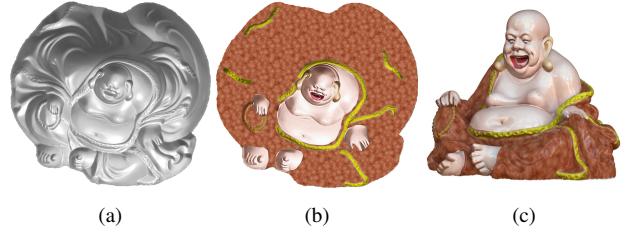


Figure 4: The texturing of the buddha model.

as a plane or a sphere. This observation guides the design of our algorithm. And our experiments on hundreds of discrete meshes suggest that, as far as sphere and plane are concerned, the flow probably will converge.

To be a CMC surface, the mean curvature has the same value on every point of the surface. Given a non-CMC surface, our unit normal flow deforms it to a CMC surface gradually. We can argue that every step of unit normal flow is a smoothing operation. This kind of smoothing is very different from mean curvature and Willmore flows. It is well known that mean curvature flow has singularity [CMI12]. Therefore it needs to be modified to avoid the singularities in every step of its flow [KSBC12]. However our algorithm based on unit normal flow has no any singularity in our experiments, although we do not prove it mathematically.

Mean curvature and Willmore flows are widely used in the application of smoothing meshes. They produce successively smoother approximations of a given piece of geometry by reducing a fairing energy. Such flows have far-ranging applications in fair surface design, inpainting, denoising, and biological modeling. They are also the central object in mathematical problems such as the Willmore conjecture [PS87]. Mean curvature flow and Willmore flow are the minimization of the certain energy, However it is not clear for us that the unit normal flow is based on the certain energy.

The most important application of planar parameterization is texturing meshes. We render the planar mesh 1f with the normals in the corresponding original 3D mesh, such as shown in Figure 4a; then we draw textures on the rendered image 4b; finally the texturing 3D mesh is exhibited in Figure 4c. In our attached video, we also show the special effect of *two dimensional foil* described in the science-fiction books [Liu14, Liu15, Mor17].

In summary, our contributions in this paper are: 1) defining a novel geometric flow on surfaces: unit normal flow; 2) establishing and deriving the relationship between unit normal flow(UNF) and constant mean curvature(CMC) surfaces; 3) proposing a robust, simple-to-implement algorithm to discretizing and approximating the non-linear UNF; 4) applying the algorithm in the application of planar and spheric mesh parameterization, and our method has a special feature of mapping the selected partial parts of meshes onto a plane and keeping left parts unchanged.

The success of our method results from a judiciously-chosen change of variables: instead of computing positions directly, we work with a quantity called unit normal field. As far as we know, our method is the first one which obtains a mesh parameteriza-

tion by a viewpoint of CMC surfaces, and the UNF also appears in mathematics and graphics community for the first time.

## 2. Related Work

Parameterization of triangle meshes is a problem with a long history in computer graphics. Due to the abundance of literature on mesh parameterization, we will focus on the most relevant approaches to ours. For in-depth surveys, we refer to [FH05, SPR\*07].

Fixing the boundary of a topological disk mesh to a convex shape, the linear method [Flo03] produces bijective parameterization, but with huge angle and area distortion. The free-boundary linear conformal methods [Flo03, DMA02, LPRM02, BCGB08, MTAD08] are not guaranteed to have a bijective map. Nonlinear methods which minimizing the energies of conformal or isometric distortion [SdS01, CPSS10, SS15, APL14, LZ14, FLG15] can not process large modes. In contrast, every step of our method solves a linear system. The local/global approach [LZX\*08] is similar to ours. However their iterations works on two dimensions and produces a lot of flipped faces, instead our method iteratively deforms meshes in 3D space and has no flips in practice. The recent method [LYNF18] require an initial bijective mapping as the starting step.

Some of injective [APL14, AL13, SKPSH13, FLG15, SS15, RPPSH17, LGZ\*16] and bijective [JSP17, SS15] mapping algorithms work either through a flip-less initialization or by enforcing a low bounded distortion. These methods need complicated and non-linear computation to guarantee injective or bijective. Our approach produces locally injective mappings automatically in most cases. For a few of complicated shapes, our mappings are injective. However with a simple interactive initialization, we can adjust these injective mappings to be bijective.

The planar mapping algorithms in [JKLG08, Luo04, ZLG\*18, SSP08] are based on Ricci flow, Calabi flow, Yamabe flow respectively. These methods work on metric space and are intrinsic, in contrast, our algorithm is extrinsic and can produce in-between embedding. OMT based area-preserved algorithms are proposed in [SCL\*16, SCQ\*16, SCL\*17], and the method in [YLZG18] can balance the area-preserved and angle-preserved mapping, however these methods need to solve a non-linear Monge-Ampere equation.

For spheric parameterizations, the most related algorithms are modified mean curvature flow(cMCF) [KSBC12] and conformal Willmore flow [CPS13]. Mean curvature flow updates the positions of surface points gradually by minimizing the gradient of surface embedding or the surface area [KSBC12, Tau95, DMSB99, Cho93, PP93]. They are used in the application of surface smoothing, minimal surfaces. This flow has singularity and only the simple convex surfaces can converge [H\*84]. The cMCF updates the metric of the surface in every step and avoids singularity and is stable. Mean curvature flow and Willmore flow is restricted by time step [OR09]. In [CPS13], they design a special Willmore flow: conformal curvature flow. This flow runs in curvature space. Their "change of variables" methodology is similar to ours. [HFL18] decimates meshes to obtain bijective isometric spheric mappings, in contrast, ours is deformation based. In [WLL14], the authors propose a spherical

parameterization which extends the planar ARAP method, however this algorithm leads to flips in high curvature sections.

The value of mean curvature of CMC surfaces [Kap90, XZ08, Ren15] is constant. Minimal surfaces [XPB06, PP93] are special CMC surfaces whose mean curvature value is zero. Normally CMC surface is constructed by minimizing the surface area under the fixed volume constraints [Bra92, PR02, DH06, Ren15], these methods need solve a non-linear optimization problem. The algorithm [PCL\*12] creates a CMC mesh from an existing one, however it changes the connectivity of the input mesh. Our spheric and planar CMC surface guided method preserve the mesh connectivity.

Gauss map diffusion is discussed in [PKC\*16], normal field filtering were also exploited in [Tau01, YOB02, TWBO02, TWBO03, SRML07, ZFAT11], however they were only used in the application of mesh smoothing and denoising. And we apply the flow to the mesh parameterization. The method in [ZLJW06] maps one source surface directly to another target surface of the same genus without the intermediate planar or spheric domain, but it can not guarantee that all source vertices lie on the target surface.

Poisson system based deformation [YZX\*04] is well-known geometry modelling technique. After the rotations of the triangle faces are known, the triangles can be rotated into the new orientations, then a Poisson system is used to blend the triangles together and reconstructs a new shape. The rotations can be achieved by the interpolation from two corresponding meshes [XZWB06] or from the rotations of the constraint faces [ZRKS05]. For the application of mesh deformation [ZG16], a local/global method is used to compute optimal rotations. In polycube construction [ZLL\*17], the rotations are calculated according to the corresponding polycube face normals. Our algorithm also relies on this kind of methodology, however we compute the rotations by averaging face normals.

## 3. Unit Normal Flow

Our motivation is deforming surfaces by the following criteria: The time derivatives of surface normals should be equal to the Laplacians of the normals. In this section, we define and propose **unit normal flow** mathematically. This flow is different from well-known mean curvature flow [KSBC12], averaged mean curvature flow [XPB06], Willmore flow [BS05, WBH\*07], Ricci flow [JKLG08], surface diffusion flow [SK01, XPB06]. All these kinds of flows can be modelled as geometric partial differential equations(PDE) [XPB06, XZ08]. As far as we know, this definition is the first time to appear in the mathematical and graphic research literatures.

Let  $S$  be a smoothly immersed surface in  $\mathbb{R}^3$ . Let  $g$  be the metric on  $S$  restricted from  $\mathbb{R}^3$ . Let  $n$  be the smooth unit normal vector field on  $S$ . Denote  $\langle \cdot, \cdot \rangle$  as the inner product and  $\Delta_g n$  as the Laplacians of the unit normals. The formal definition of **unit normal flow** is the following:

$$\frac{dn}{dt} = \Delta_g n - \langle \Delta_g n, n \rangle \cdot n. \quad (1)$$

Notice that the norm of  $n$  is preserved under this flow, since  $\frac{d}{dt} \langle n, n \rangle = \langle n, \frac{d}{dt} n \rangle = \langle n, \Delta_g n \rangle - \langle n, \Delta_g n \rangle = 0$

**Lemma 1** If the Laplacians  $\Delta_g n$  of the unit normal field  $n$  is parallel to  $n$ , i.e.,  $\Delta_g n \parallel n$ , then the mean curvature  $H$  of  $S$  is constant.

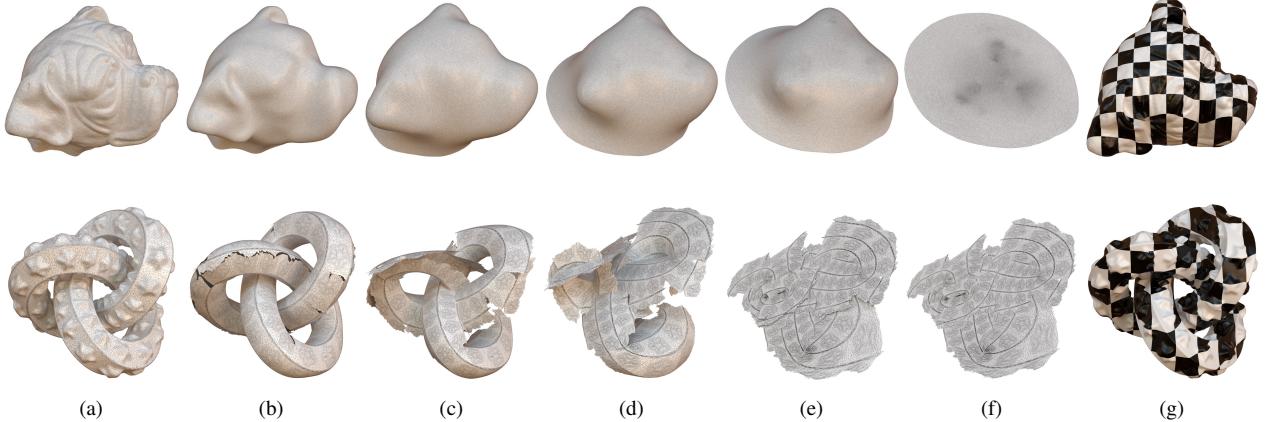


Figure 5: (a) are the original meshes; (b), (c), (d), (e) are intermediate deforming meshes; (f) are the final planar parameterizations and (g) are the texturings. The meshes of doghead, manhead and linkedring have the number of faces of 4318, 98921, 100000 respectively.

*Proof* We prove it with a local calculation.

Use an isothermal coordinate  $r(x_1, x_2)$  for  $S$ . Let  $z = x_1 + ix_2$ . Write

$$\frac{\partial}{\partial z} = \frac{1}{2} \left( \frac{\partial}{\partial x_1} - i \frac{\partial}{\partial x_2} \right),$$

$$\frac{\partial}{\partial \bar{z}} = \frac{1}{2} \left( \frac{\partial}{\partial x_1} + i \frac{\partial}{\partial x_2} \right).$$

Let  $g = \lambda^2((dx_1)^2 + (dx_2)^2)$  be the metric of  $S$  under the isothermal coordinate. Define

$$\Omega = \langle r_{zz}, n \rangle.$$

Here  $r_{zz}$  is a shorthand for  $\frac{\partial^2 r}{\partial z^2}$ .

The vectors  $\{r_z, r_{\bar{z}}, n\}$  form a (compelxified) local frame of  $\mathbb{R}^3$  on  $S$ . The next step is to calculate its equations of motion. Notice that

$$\langle r_z, r_z \rangle = \langle r_{\bar{z}}, r_{\bar{z}} \rangle = 0.$$

Therefore

$$\langle r_{zz}, r_z \rangle = \langle r_{z\bar{z}}, r_z \rangle = \langle r_{z\bar{z}}, r_{\bar{z}} \rangle = \langle r_{\bar{z}\bar{z}}, r_{\bar{z}} \rangle = 0.$$

On the other hand,

$$\langle r_z, r_{\bar{z}} \rangle = \frac{1}{2} \lambda^2.$$

Therefore

$$\langle r_{zz}, r_{\bar{z}} \rangle = \lambda \lambda_z.$$

Notice that the mean curvature

$$\begin{aligned} H &= -\frac{1}{2} \left( \frac{1}{\lambda^2} \langle \frac{\partial r}{\partial x_1}, \frac{\partial n}{\partial x_1} \rangle + \frac{1}{\lambda^2} \langle \frac{\partial r}{\partial x_2}, \frac{\partial n}{\partial x_2} \rangle \right) \\ &= -\frac{2}{\lambda^2} \langle n_z, r_{\bar{z}} \rangle \\ &= \frac{2}{\lambda^2} \langle n, r_{z\bar{z}} \rangle \end{aligned}$$

By the definition of  $\Omega$ ,

$$\langle n_z, r_z \rangle = -\langle n, r_{zz} \rangle = -\Omega.$$

Combining the identities above, one has

$$\begin{cases} r_{zz} = \frac{2}{\lambda} \lambda_z r_z + \Omega n \\ r_{z\bar{z}} = \frac{\lambda^2}{2} H n \\ n_z = -H r_z - \frac{2}{\lambda^2} \Omega r_{\bar{z}} \end{cases}$$

Let  $g$  denote the metric. Notice that for any function  $f$  on  $S$ ,

$$\begin{aligned} \Delta_g f &= \frac{1}{\sqrt{\det g}} \frac{\partial}{\partial x^j} (g^{jk} \sqrt{\det g} \frac{\partial f}{\partial x^k}) \\ &= \frac{4}{\lambda^2} f_{z\bar{z}}. \end{aligned}$$

Therefore

$$\begin{aligned} \Delta_g n // n &\Rightarrow n_{z\bar{z}} // n \\ &\Rightarrow -H_z r_z - \frac{\lambda^2}{2} H^2 n - (\frac{2\Omega}{\lambda^2})_{\bar{z}} r_{\bar{z}} - \frac{2}{\lambda^2} \Omega \left( \frac{2}{\lambda} \lambda_{\bar{z}} r_{\bar{z}} + \Omega n \right) // n \\ &\Rightarrow H_{\bar{z}} = 0 \\ &\Rightarrow H \text{ is a constant.} \end{aligned}$$

□

By the lemma 1, the critical point of unit normal flow equation 1 leads to a CMC surface.

Honestly, we must point out that we do not prove the singularity, existences, uniqueness and convergence of the solution of the UNF of surfaces in this paper. They are still open problems. Instead we prove this mathematics fact: "when the unit normal flow converges to a stable surface, the surface is a constant mean curvature surface." Our argument is that when these requirements are satisfied, then the final state of the flow is a CMC. Experimentally we demonstrate that the flows converge on most of discrete meshes of disk and spheric topologies.

#### 4. Our algorithms

Motived by the above mathematics lemma 1 for unit normal flow and the following Hopf theorem, we design a simple, easy, elegant, but practical algorithm for planar and spheric mesh parameterization.

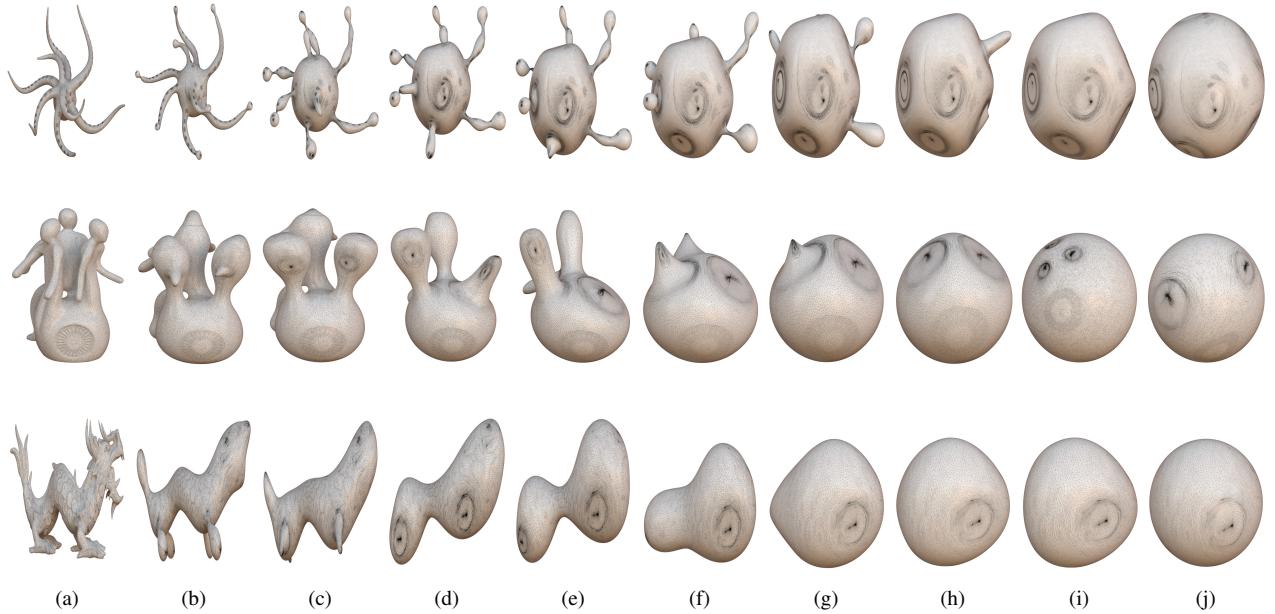


Figure 6: (a) are the original meshes; (b),(e),(c),(f),(g),(h) are the intermediate deforming meshes; (i) are the convergent shapes; (j) are the final normalized spherical parameterizations. Octopus, memento and dragon have the faces of 497236, 99932, 105298 respectively.

**Theorem 1 (Hopf)** Every closed immersed constant mean curvature surface of genus 0 in  $\mathbb{R}^3$  is a round sphere.

The unit normal flow we defined in Equation 1 is a highly nonlinear equation in position variables. It is hard to solve numerically in this format. We propose and use the "change of variable" method which also be applied in [CPS13]. Our algorithms do not compute the positions directly. Instead we solve the current unit normal field firstly, then the geometry of the surface is reconstructed from them.

Equation 1 is defined on continues smooth surface, we need to discretize it on triangle meshes. We approximate smooth unit normal field  $n$  by discrete face normal field. Our unit normal flow updates face normals instead of vertex normals in every iteration.

The key point is the discretization of the Laplace operator. In graphics community, the well-known cotangent Laplace operator [PP93] is for functions defined on vertice of meshes. Therefore it can not be used for our face normals. In this paper, we propose to use a simple method to approximate the Laplacian operator of normal functions on faces by the following formula:

$$\Delta_d n_i(t) = \sum_{j \in \text{Neighbour}(i)} n_j(t) - n_i(t) \approx \Delta_g n(t) \quad (2)$$

Where  $n_j(t)$  denote the unit normal of face  $i$  at time  $t$ ,  $\Delta_d n_i$  represents the normal under the discrete Laplace operator, the  $\text{Neighbour}(i)$  denotes the neighbours of the face  $i$ , which includes  $i$  itself. Then the new normal at time  $t + 1$  is computed by the follows:

$$n_i(t+1) = \Delta_d n_i(t) + n_i(t) = \sum_{j \in \text{Neighbour}(i)} n_j(t) \quad (3)$$

Our flow is defined on **unit** face normals. Even though face normals

$n_i$  are unit,  $\Delta_d n_i$  can not be guaranteed to be unit. We need normalize it. In practice, the faces in  $\text{Neighbour}(i)$  are not constrained to be one-ring, they could be  $k$ -ring neighbours.

If we use the area-weighted average, then the flow will fail. The reason is that the centre of the area-weighted average maybe not fall inside the face. For example, in Fig. 7, the average normal of neighbours of red face is not located inside the red face. In Fig. 9, we demonstrate the success of the simple average over the area-weighted average.

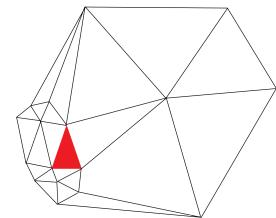


Figure 7

Figure 8: The area-weighted average

After the new face normals are computed in every step, we rotate all triangle faces from their old orientations to the current ones. However the triangles are rotated independently, the result triangle soup is not a valid mesh. We use the Poisson system based method [ZLL\*17, ZG16] to reconstruct the triangles into a unified mesh. This step can be thought as solving a system of the unknown positions from the known normal variables.

Let  $S$  be an original surface and  $S'$  be its deformed surface embedded in 3-dimension. And denote a 3-vector  $x_v$  be the position

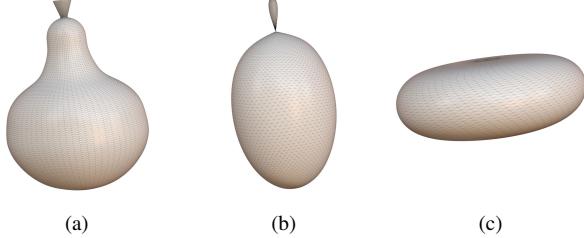


Figure 9: (a) is the original spherical pear mesh; (b) is the deformed result by area-weighted average; (c) is the deformed result by the simple average at the same number iteration with (b).

associated with vertex  $v$  of  $S$ , and a 3-vector  $x'_v$  with vertex  $v$  of  $S'$ . On every triangle of the mesh, we define one rotation matrix variable referred to as  $R(t)$ . Let  $he_{vw}$  represents the half edge from the vertex  $v$  to  $w$ . We denote the angle of the corner opposite to the half edge  $he_{vw}$  in its triangle with  $a_{vw}$ . Finally  $R(t_{vw})$  represents the  $3 \times 3$  rotation matrix associated with the triangle face whose the half edge is  $he_{vw}$ .

After fixing a vertex, we can obtain the unknown position variables  $x'$  by solving a single linear system as the following:

$$\begin{aligned} & \sum_{w \in N(v)} [\cot(a_{vw}) + \cot(a_{wv})] (x'_v - x'_w) \\ &= \sum_{w \in N(v)} [\cot(a_{vw})R(t_{vw}) + \cot(a_{wv})R(t_{wv})] (x_v - x_w). \end{aligned} \quad (4)$$

By defining the 3-vector at vertex  $v$  as:

$$b_v := \sum_{w \in N(v)} [\cot(a_{vw})R(t_{vw}) + \cot(a_{wv})R(t_{wv})] (x_v - x_w), \quad (5)$$

we can change the above system into matrix format as:

$$Lx' = b, \quad (6)$$

where  $L$  is the  $n$ -by- $n$  Laplacian matrix,  $x'$  and  $b$  are n-vectors of 3-vectors.

Given two unit normals, the rotation matrix  $R(t)$  between them can be computed by the algorithm of Rodrigues' rotation formula. We solve the equation 6 by fixing the position of one vertex.

In summary, our algorithm consists of two steps: in the first step, we average the unit face normals; in the second step, we deform or reconstruct the surface by the constraints of the current unit face normals. After the two steps, we get a new mesh which is smoother than previous one. These two steps are iteratively calculated until the flow converges and the shape of the mesh does not change anymore. In every iteration step  $i$ , we recompute the rotation matrix  $R_i(t)$  by averaging the face normals. With the new  $R_i(t)$ , we update the  $L_i$  and  $b_i$ . The whole iteration system is as the following:

$$L_i x'_i = b_i. \quad (7)$$

For small meshes, our algorithm converges quickly in a few of iteration; but for large ones, it needs thousands of steps. In planar case, we speed up it by computing the average normal of all triangle faces and add it to the average results when updating the normals,

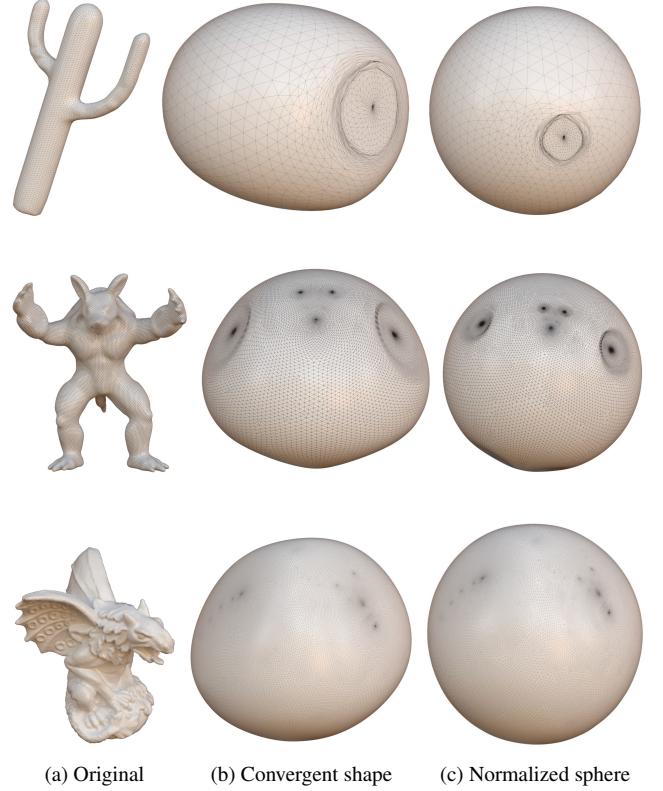


Figure 10: The convergent and normalization of the spheric deformations. (b) are the convergent shapes of unit normal flow and (c) are the normalized spheres.

as shown in equation 8. The bigger the parameter  $weight$ , the faster the convergence. However for the meshes of bad quality, the big  $weight$  will lead to flipped faces. In practice, we set it equal to 0.02. Surprisingly, this simple method decreases our iteration number to dozen for large meshes. The experimental details are exhibited in next section.

$$\Delta_d n_i = \sum_{j \in Neighbor(i)} n_j + weight \times \sum_{k \in AllFaces} n_k \quad (8)$$

After the convergence of the flow, we need do some post-processing. For planar parameterization, the convergent mesh possibly does not fall on a perfect plane, we rotate it and align the average normal of all triangle faces to Z direction, then project it onto the XY plane to obtain their texturing coordinates. For spheric mapping, the convergent shapes are not perfect spheres too, we need normalize the shapes, as exhibited in Figure 10. We first compute the mean center position  $C$  of all vertices; then the average distance  $R$  between all vertices and the mean center  $C$  is calculated; finally

the normalized position  $p'_i$  of the vertex  $i$  is computed by the following formula:

$$p'_i = (p_i - C) \times \left( \frac{R}{\|p_i - C\|} \right) \quad (9)$$

## 5. Experiments

To demonstrate the efficiency and robustness of our algorithm, we apply our algorithm to hundreds of challenging meshes of disk and spheric topology, summarize its performance and compare with other methods. All the demonstrations shown in this paper are local injective. We ran out experiments on a 12-core Xeon clocked at 2.7GHz, using the Eigen solver [GJ\*10] for the linear system solve.

Table 1: Five kinds of distortions.

Symmetric Dirichlet	$\sum_{t=1}^n (\sigma_{1,t}^2 + \sigma_{2,t}^{-2})$
Conformal AMIPS 2D	$\sum_{t=1}^n \frac{\sigma_{1,t}^2 + \sigma_{2,t}^2}{\sigma_{1,t} \sigma_{2,t}}$
ASAP	$\sum_{t=1}^n A_t (\sigma_{1,t} - \sigma_{2,t})^2$
ARAP	$\sum_{t=1}^n A_t [(\sigma_{1,t} - 1)^2 + (\sigma_{2,t} - 1)^2]$
Green-Lagrange	$\sum_{t=1}^n A_t [(\sigma_{1,t}^2 - 1)^2 + (\sigma_{2,t}^2 - 1)^2]$

In graphic literature, there are several different definitions of mapping distortion [LZX\*08, RPPSH17, FLG15, ZMT05]. As-Rigid-As-Possible (ARAP) and symmetric Dirichlet(SD) are isometric measurements. And conformal ones include As-Similar-As-Possible (ASAP), conformal AMIPS (cAMIPS) and Green-Lagrange(GL). Denote  $\sigma_1, \sigma_2$  as the two singular values of the triangle mapping,  $A_t$  as the area of triangle  $t$ , then these five kinds of measurements are summarized in Table 1:

Besides these five kinds of distortions, we also calculate the change ratios of areas, edges and angles. We do not have a mathematical proof on the mapping quality of our algorithm. Therefore we use the extensive experimental results to justify the robustness, applicability and performance of our method.

### 5.1. The ring size of neighbours.

The size of the neighbouring area in our algorithm affects the convergent speed. The number of iterations decreases quickly when the ring size enlarges. The relationship is demonstrated in Figure 13.

However, when the number of rings is bigger than one threshold value, the mapping will not be flip-less. This value is not fixed and depends on the shape and size of the mesh. We list the statistic data of iterations for several meshes with varying representative size in Table 2.

The mapping quality does not change dramatically with the ring size, as shown in Figure 14. Therefore we have a wide and stable spectrum for the ring size parameter, this fact makes our algorithms practical and do not need to adjust the parameter for every mesh.

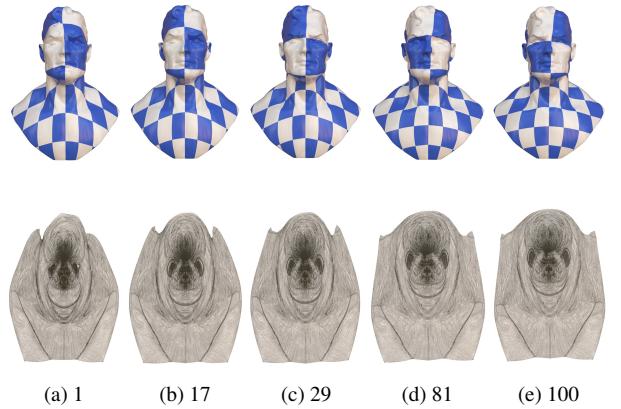


Figure 11: The planar mapping and texturing of the same model by the ring size of 1, 17, 29, 81 and 100 respectively.

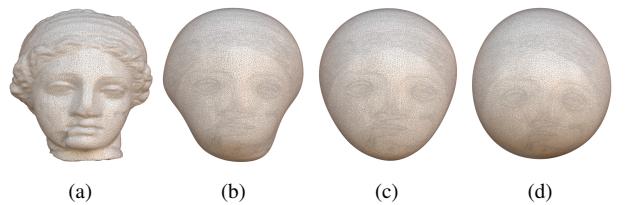


Figure 12: (a) is the original model; (b),(c),(d) are the intermediate deforming meshes at the iteration step of 200 for the ring size of 3, 10 and 20 respectively.

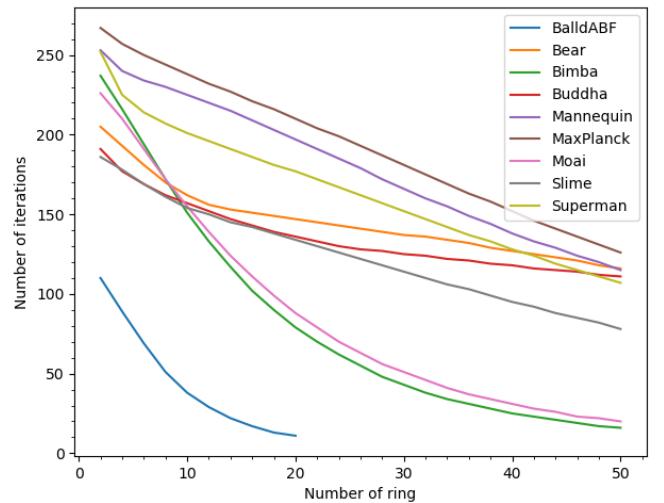


Figure 13: The number of iterations decreases when the ring size gets bigger.

According to our extensive experiments on hundreds of meshes, the above parameter works successfully for most of them. Furthermore we can always obtain flip-less parameterization by shrinking the neighbourhood in practice if the parameter fails in a few cases. Figure 12 and 11 exhibit the effects of ring size for the meshes of spheric and disk topology respectively. In most of cases, disk

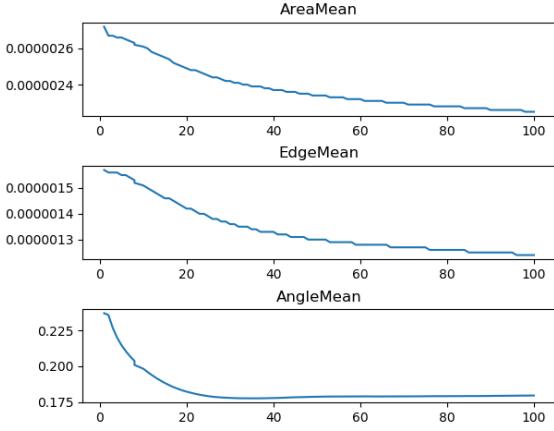


Figure 14: The mapping qualities of different ring sizes of the superman mesh.

Model	#Faces	Iter(2)	Iter(10)	Iter(24)	Iter(50)
BallABF	1k	110	38	F	F
Oni	3K	1481	97	F	F
Hand	7k	262	103	F	F
Bimba	11k	237	151	62	16
Moai	16k	226	155	70	20
MaxPlanck	84k	267	238	199	126
Slime	103k	186	154	126	78
Mannequin	109K	253	225	185	115
Superman	190k	252	201	167	107
Bear	296k	205	162	143	116
Buddha	471k	191	157	130	111

Table 2: The number of iterations in case of ring size of 2, 10, 24 and 50; F=Flip.

meshes can converge by dozens of iterations and spheric ones needs thousands of steps. We recommend the following empirical values for the ring size:

$$ring = \begin{cases} 5 & faces < 1000 \\ 10 & faces < 10000 \\ 20 & faces \geq 100000 \end{cases}$$

## 5.2. Disk topology

We demonstrate our mapping and corresponding texturing of disk-topology meshes of single boundary in Figure 5 and multi-boundary in Figure 15. The intermediate deforming frames are also exhibited in Figure 1, 2.

In our experiments, the normals of the boundary face are not fixed. Therefore the unfolded result of our algorithm is affected by the initial normals of boundary faces. When the boundary of a mesh is small and tight, possibly it can not unfold towards a plane driven by its natural initial face normals. The mesh in Figure 16a will deform to the non-planar shape of Figure 16f after 500

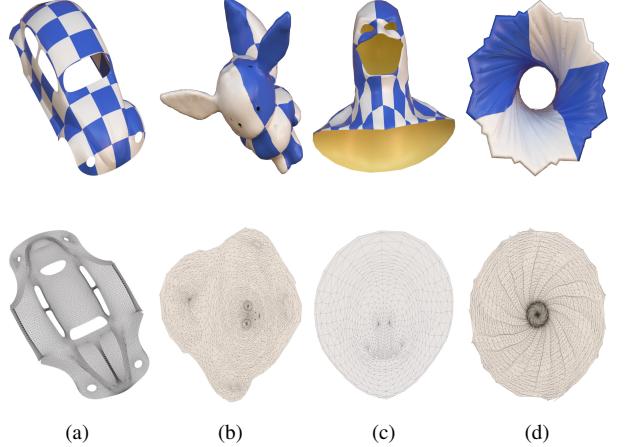


Figure 15: The planar mapping and texturing of four multi-boundary meshes.

hundreds of iterations through Figure 16d and 16e with its natural boundary face normals. We solve the problem by assigning it a set of specific boundary normals to pull faces apart, such that it is able to stretch to a plane in Figure 16c whose corresponding texturing is shown in Figure 16b.

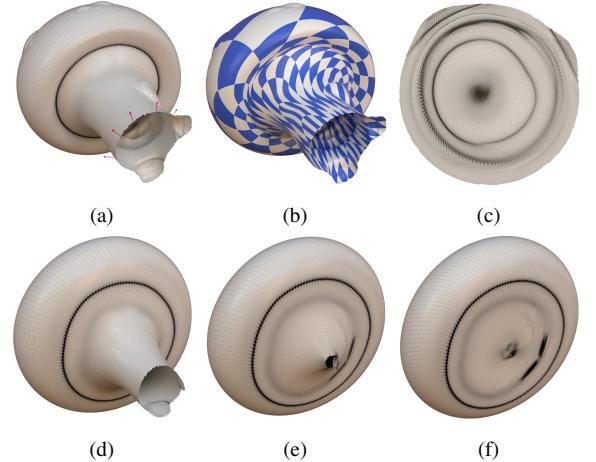


Figure 16: The red arrows represent the natural face normals on the boundary of the original mesh (a) and the green ones are the normals we assigned; (d) (e) and(f) are the deforming meshes by red normals; (c) is the planar mesh deformed by the green normals; The texturing is exhibited in (b).

Our experiments produce bijective mappings for most of meshes with the natural initial face normals. In some complicated cases, the natural unfolded result may be injective, such as in Figure 17b. By adjusting its boundary normals manually and interactively, we can turn this injective mapping to be bijective, as shown in Figure 17d.

**The options of Laplacian matrix.** On smooth surfaces, the Laplacian operator is dynamic and depends on the current deforma-

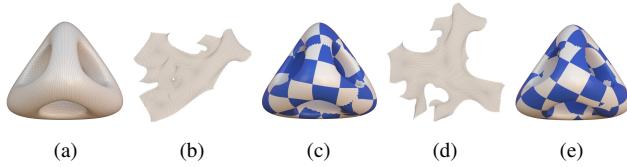


Figure 17: (a) is the original mesh cut; (b), (c) are the injective mapping and its texturing; (d), (e) are the bijective mapping and its texturing.

ing surface. In our setting, we approximate it with the Laplacian matrix of the initial mesh and keep it unchanged. Another alternative is updating the Laplacian matrix of the equation (6) in every iteration by the current geometry of the deforming mesh. The two kinds of mapping and the distortions are exhibited in Figure 18 and Table 3. We concludes that the updated Laplacian matrix preserves the area and edge length better, and the fixed one produces better conformal mapping results.

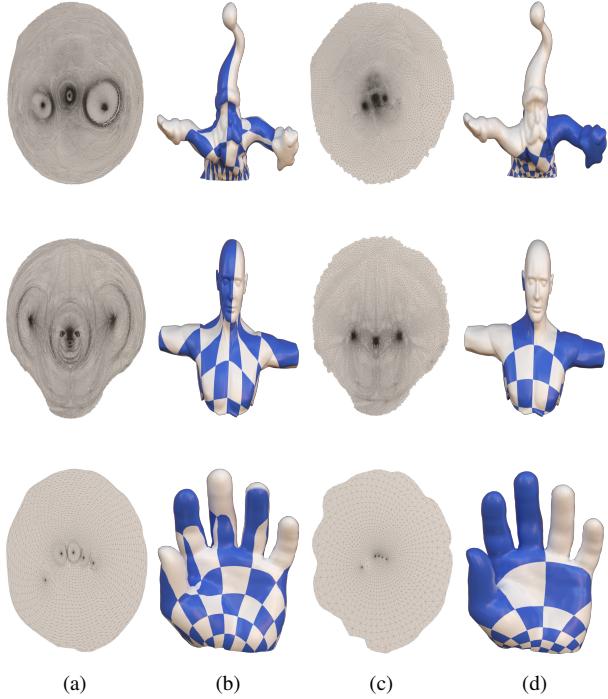


Figure 18: (a) and (b) are the planar mappings and corresponding texturings by updating Laplacian matrix; (c) and (d) are by fixing Laplacian matrix.

**The partial parameterization.** Our deformation based method has a special feature, which can deform a part of mesh to planar and keeps the left parts unchanged, as shown in Figure 27. As far as we know, this is the first approach which can mix two pieces of 3D and 2D mesh together seamlessly.

**The comparisons.** As the same as some recent developed algorithms [FL16, SPSH\*17], we also compare all kinds of mapping

distortions of our algorithm with the ones in [RPPSH17], which is the standard benchmark of bijective parameterizations.

For a multi-core implementation, we report the running times of some of our experimental meshes of disk topology in Table 4. The number of iterations and timing are the same magnitude as the [RPPSH17]. On most of meshes, our approach converges on dozens or one hundred of iterations. Our algorithm is also scalable on meshes of huge size as far as the linear system solver can process it.

Due to the scope of the paper, we demonstrate the mapping and texturing of our method and [RPPSH17] for several models in Figure 19 and list the distortions in Table 6. The more demos are provided in the supplemental file. We observe that our method is less isometric than SLIM.

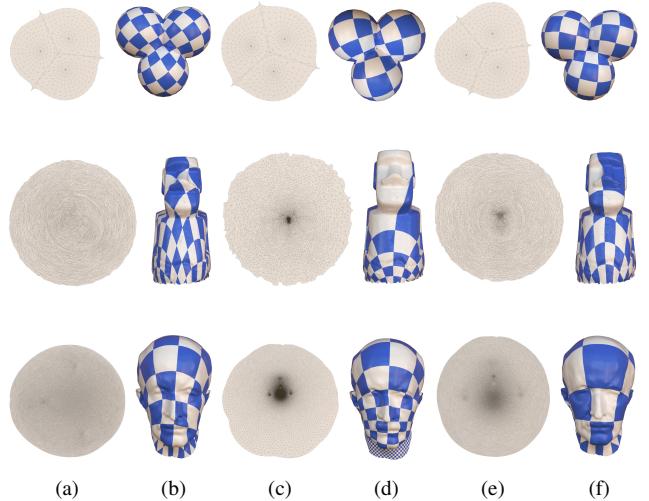


Figure 19: (a) and (b) are the parameterization and texturing of SLIM [RPPSH17]; (c), (d) are of the results of our unit normal flow by fixed Laplacian matrix; (e), (f) are the ones by updated Laplacian matrix.

### 5.3. Spheric topology

In Figure 6, we demonstrates the deformation and parameterization of several meshes of spheric topology. More exhibitions are shown in the supplemental and attached video. Unlike the planar one, the spheric unit normal flow converges slowly and needs thousands of iterations for most meshes.

The method of conformal Willmore flow [CPS13] needs the meshes without obtuse angles, it is highly strict requirement. But our method and cMCF can work on most of meshes. We exhibit the deformations of our method, cMCF [KSBC12] and conformal Willmore flow [CPS13] on the same bunny model of 28576 faces in Figure 26. The measurements are summarized in the Table 5. We notice that our method demonstrates better edge preserving and isometric performance than cMCF and conformal Willmore flow.

In Figure 25, Table 7 and the supplemental file, we show the

Table 3: The distortions by the fixed and updated Laplacian matrices; (Fix) denotes the fixed Laplacian matrix.

Model	AreaSum	EdgeSum	AngleSum	Symm Dirichilet	Conformal AMIPS2D	ASAP	ARAP	Green Lagrange
Buddha	0.30	0.20	$2.72 \times 10^5$	$2.35 \times 10^6$	$1.04 \times 10^6$	0.22	0.21	1.11
Buddha(Fix)	0.56	0.33	$1.45 \times 10^4$	$4.72 \times 10^6$	$9.41 \times 10^5$	$5.36 \times 10^{-4}$	0.32	1.17
Hand	0.89	0.63	$6.76 \times 10^3$	$1.89 \times 10^6$	$2.41 \times 10^4$	1.06	1.34	16.54
Hand(Fix)	1.38	1.02	$5.36 \times 10^3$	$1.28 \times 10^{11}$	$1.36 \times 10^4$	0.01	1.67	11.70
Santa	0.84	0.63	$1.57 \times 10^5$	$6.89 \times 10^7$	$5.42 \times 10^5$	2.10	2.19	79.93
Santa(Fix)	1.43	1.06	$1.38 \times 10^5$	$2.96 \times 10^{27}$	$2.74 \times 10^9$	$9.00 \times 10^{-4}$	1.74	13.82
Slime	0.35	0.29	$5.18 \times 10^4$	$5.59 \times 10^6$	$2.27 \times 10^5$	0.24	0.24	1.34
Slime(Fix)	0.66	0.43	$2.47 \times 10^3$	$1.81 \times 10^6$	$2.06 \times 10^5$	$3.34 \times 10^{-4}$	0.39	1.41
Torso	0.68	0.55	$8.55 \times 10^4$	$6.44 \times 10^6$	$2.91 \times 10^5$	1.46	1.47	28.11
Torso(Fix)	1.17	0.87	$9.38 \times 10^3$	$1.25 \times 10^6$	$1.87 \times 10^5$	$9.55 \times 10^{-4}$	1.31	8.07

Table 4: Timing of our planar mapping algorithm.

Model	Faces	Iteration	Time(s)	Vertices
Bimba	11k	26	1.089	6k
Beetle	39k	14	1.777	20k
MaxPlanck	85k	29	8.458	42k
Gargoyle	99k	57	18.178	50k
Bunny	100k	45	15.375	50k
NicoloDaUzzano	100k	17	5.599	50k
Mannequin	109k	27	9.119	55k
Surperman	190k	25	15.144	96k
Bear	296k	21	19.31	148k
Buddha	471k	20	30.657	236k

Table 5: The distortions of our method(UNF), cMCF [KSBC12], and conformal Willmore flow [CPS13].

	cMCF	Willmore	our method
AreaMean	$2.08 \times 10^{-5}$	$2.09 \times 10^{-5}$	$2.38 \times 10^{-5}$
EdgeMean	$1.01 \times 10^{-5}$	$1.01 \times 10^{-5}$	$9.56 \times 10^{-6}$
AngleMean	0.04	0.14	0.12
Symmetric Dirichlet	$4.18 \times 10^9$	$3.7 \times 10^9$	$5.79 \times 10^7$
Conformal AMIPS 2D	$5.72 \times 10^4$	$6.03 \times 10^4$	$6.14 \times 10^4$
ASAP	$7.31 \times 10^{-4}$	0.10	0.04
ARAP	0.46	0.52	0.48
Green Lagrange	1.07	1.62	1.77

spheric mappings and distortions of our method, cMCF and harmonic approach [GWC\*04]. In all fourteen meshes, harmonic approach can not obtain bijective parameterization; the mappings flip in only one model by our algorithm, six models by cMCF. We observe that our method is more conformal than harmonic approach.

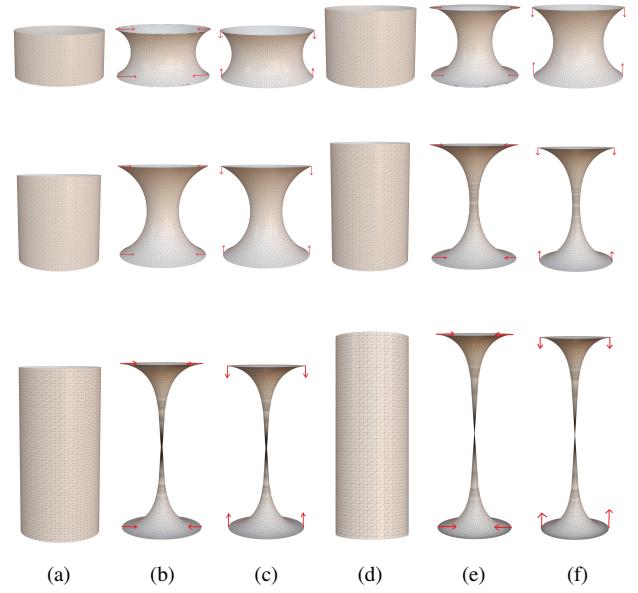


Figure 20: The CMC-like surfaces generated by unit normal flow. The radii and heights of the cylinders are (10,10),(10,15),(10,20),(10,30),(10,40),(10,50) respectively.

#### 5.4. CMC-like surfaces

Plane and sphere are special and simple constant mean curvature surfaces. Our approximation and discretization of unit normal flow work successfully on them. For other kinds of CMC surfaces, our algorithm can also drive the flow to deform the corresponding discrete meshes. However, the convergent shapes are not CMC surfaces in exact mathematical sense. We call them CMC-like surfaces.

In Figure 20, we demonstrate the convergent shapes of the cylinders of a set of different radii and heights, constrained by two sets of the different boundary face normals, under our unit normal flow. In this experiment, the positions of the boundary vertex of the cylinders are fixed. The red arrows are the representatives of the first set

of the boundary face normals; the green arrows are from the second set. The convergent shapes are catenoid-like surfaces, however the radii and heights we uses does not satisfy the exact mathematical formula of catenoids.

The cylinders with different top and bottom radii are exhibited in Figure 24. In Figure 21, we deform a half-sphere and a unit disk to the different CMC-like surfaces under varying face normals constraints. Unduloid-like surfaces are demonstrated in Figure 22 and in them, all the boundary face normals are towards the centres. More demons are shown in Figure 23.

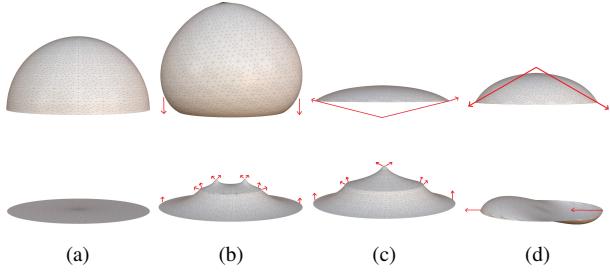


Figure 21: The half-sphere and unit disk (a) are assigned three different set of boundary face normals; (b), (c), (d) show their corresponding convergent shapes.

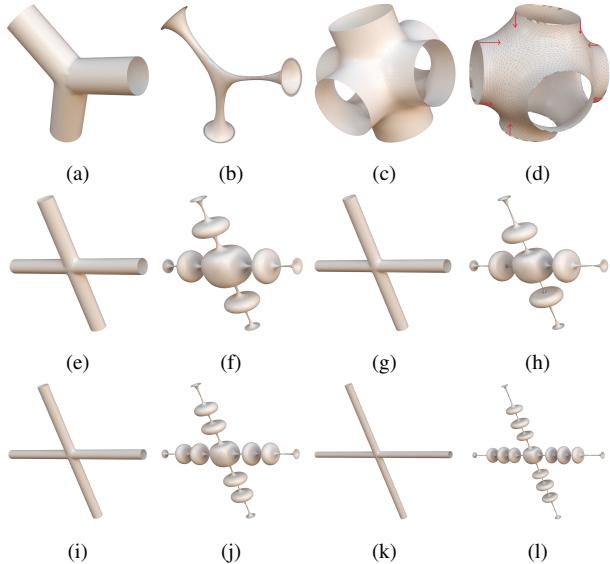


Figure 22: The first and third columns are the original meshes, the second and fourth columns are the convergent CMC-like surfaces. The radii and heights in (e),(g),(i),(k) are (10,120),(10,160),(10,200),(10,300).

On one hand, CMC-like surfaces suggest and give us hint that unit normal flow could be mathematically convergent on smooth surfaces for all CMC surfaces. On the another hand, How to design a more accurate discrete unit normal flow for other kinds of CMC surfaces is a challenging problem and our future works.

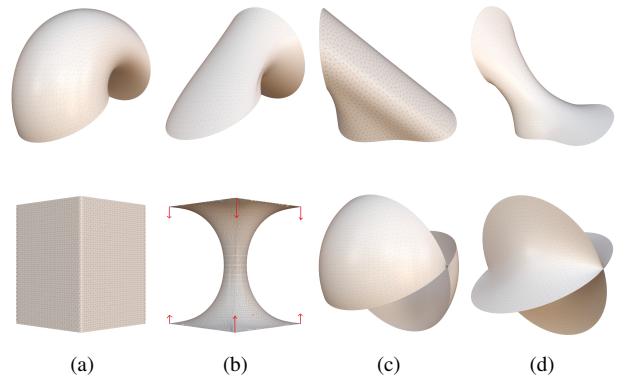


Figure 23: (a),(c) are the original meshes; (b),(d) are the convergent CMC-like surfaces.

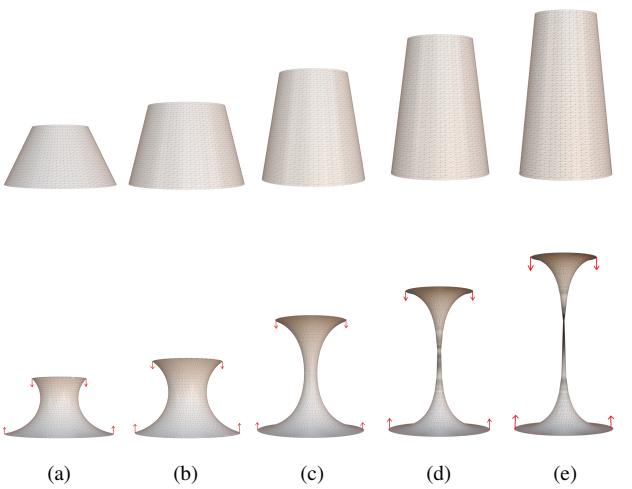


Figure 24: The CMC-like surfaces generated by unit normal flow. The top and bottom radii and heights of the cylinders are (10,20,20),(10,15,20),(10,15,30),(10,15,40),(10,15,50) respectively.

## 6. Conclusion and future work

We propose a special unit normal flow(UNF) to deform surfaces. This flow averages the normals of a smooth surface, and reconstruct the geometry to fit the smoothed normals. We define the mathematical equation of unit normal flow, and prove that the convergent surface has constant mean curvature if the flow is stable and converges. We also present an approximation method on discrete meshes and apply it to the applications of planar and spheric mesh parameterization. Our algorithm provides locally injective mapping and it outperforms many state-of-art methods.

There are still some important works left for future. The reconstruction step is affected and will fail if the quality of meshes is extremely bad. The convergence, singularity, existences and uniqueness of the unit normal flow are waiting to be proved. It is also a great challenge to design an efficient, stable and accurate discrete

Model	Method	Area Mean	Edge Mean	Angle Mean	Symm Dirichlet	Conformal AMIPS	ASAP	ARAP	Green Lagrange
Bimba(11k)	SLIM	$4.76 \times 10^{-5}$	$2.65 \times 10^{-6}$	0.42	$8.51 \times 10^4$	$3.48 \times 10^4$	1.47	1.12	13.76
	UNF(Fix)	$9.89 \times 10^{-5}$	$5.65 \times 10^{-5}$	0.03	$5.71 \times 10^6$	$2.26 \times 10^4$	$1.45 \times 10^{-3}$	1.23	3.90
	UNF	$6.21 \times 10^{-5}$	$3.26 \times 10^{-5}$	0.31	$1.58 \times 10^5$	$3.06 \times 10^4$	1.19	1.25	17.71
Bunny(100k)	SLIM	$3.85 \times 10^{-6}$	$2.46 \times 10^{-6}$	0.47	$7.67 \times 10^5$	$3.24 \times 10^5$	1.31	0.90	9.19
	UNF(Fix)	$1.05 \times 10^{-5}$	$4.58 \times 10^{-6}$	0.09	$1.21 \times 10^{11}$	$2.00 \times 10^5$	$6.68 \times 10^{-4}$	0.96	4.13
	UNF	$5.50 \times 10^{-7}$	$2.69 \times 10^{-6}$	0.35	$9.24 \times 10^6$	$2.93 \times 10^5$	0.80	0.78	6.71
Buddha(471k)	SLIM	$5.1 \times 10^{-7}$	$2.6 \times 10^{-7}$	0.20	$2.22 \times 10^6$	$1.04 \times 10^6$	0.22	0.17	0.81
	UNF(Fix)	$1.18 \times 10^{-6}$	$4.6 \times 10^{-7}$	0.01	$4.68 \times 10^6$	$9.41 \times 10^5$	$4.94 \times 10^{-4}$	0.31	1.06
	UNF	$6.2 \times 10^{-7}$	$2.8 \times 10^{-7}$	0.18	$2.32 \times 10^6$	$1.03 \times 10^6$	0.19	0.19	0.92

Table 6: The distortions of our unit normal flow (UNF) method and SLIM [RPPSH17]. UNF and UNF(fix) denote the updated and Fixed Laplacian matrice respectively.

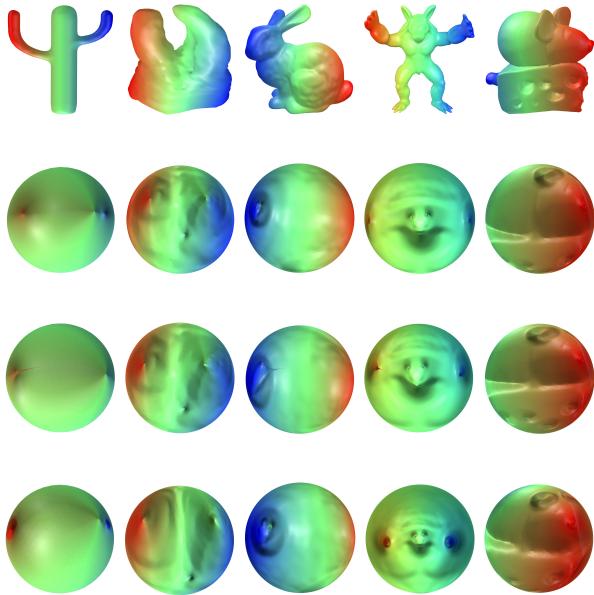


Figure 25: The spherical mapping of cMCF(the second row), harmonic(the thrid row) and UNF methods(the fourth row).

algorithm to construct other types of constant mean curvature surfaces besides planes and spheres.

### Acknowledgements

We would like to thank anonymous reviewers for their insightful feedbacks, valuable comments, and suggestions. Some pictures are rendered by Mitsuba [Jak10]. All 3D models are from the AIM@SHAPE shape repository and Thingi10K repository. Thanks MeshDGP [Zha16] framework for the implementation reference. The project is partially supported by NSFC No. 61772379, 61772105, 61720106005, NSF DMS-1737812, NSF DMS-1418255, AFOSR FA9550-14-1-0193, NIH 1R01LM012434 and National ST Major Project of China (2018ZX10301201), the

National Natural Science Foundation of China (11772047) and Key international collaborating Project from National Natural Science Foundation of China (11620101001).

### References

- [AL13] AIGERMAN N., LIPMAN Y.: Injective and bounded distortion mappings in 3d. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 106. [3](#)
- [APL14] AIGERMAN N., PORANNE R., LIPMAN Y.: Lifted bijections for low distortion surface mappings. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 69. [3](#)
- [BCGB08] BEN-CHEN M., GOTSMAN C., BUNIN G.: Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 449–458. [3](#)
- [Bra92] BRAKKE K. A.: The surface evolver. *Experimental mathematics* 1, 2 (1992), 141–165. [3](#)
- [BS05] BOBENKO A. I., SCHRÖDER P.: Discrete willmore flow. In *ACM SIGGRAPH 2005 Courses* (2005), ACM, p. 5. [3](#)
- [Cho93] CHOPP D. L.: Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics* 106, 1 (1993), 77–91. [3](#)
- [CM112] COLDING T. H., MINICOZZI II W. P.: Generic mean curvature flow ii; generic singularities. *Annals of mathematics* 175, 2 (2012), 755–833. [2](#)
- [CPS13] CRANE K., PINKALL U., SCHRÖDER P.: Robust fairing via conformal curvature flow. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 61. [3, 5, 9, 10, 13](#)
- [CPSS10] CHAO I., PINKALL U., SANAN P., SCHRÖDER P.: A simple geometric model for elastic deformations. In *ACM transactions on graphics (TOG)* (2010), vol. 29, ACM, p. 38. [3](#)
- [DH06] DZIUK G., HUTCHINSON J. E.: Finite element approximations to surfaces of prescribed variable mean curvature. *Numerische Mathematik* 102, 4 (2006), 611–648. [3](#)
- [DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. In *Computer graphics forum* (2002), vol. 21, Wiley Online Library, pp. 209–218. [3](#)
- [DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 317–324. [3](#)
- [FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*. Springer, 2005, pp. 157–186. [3](#)

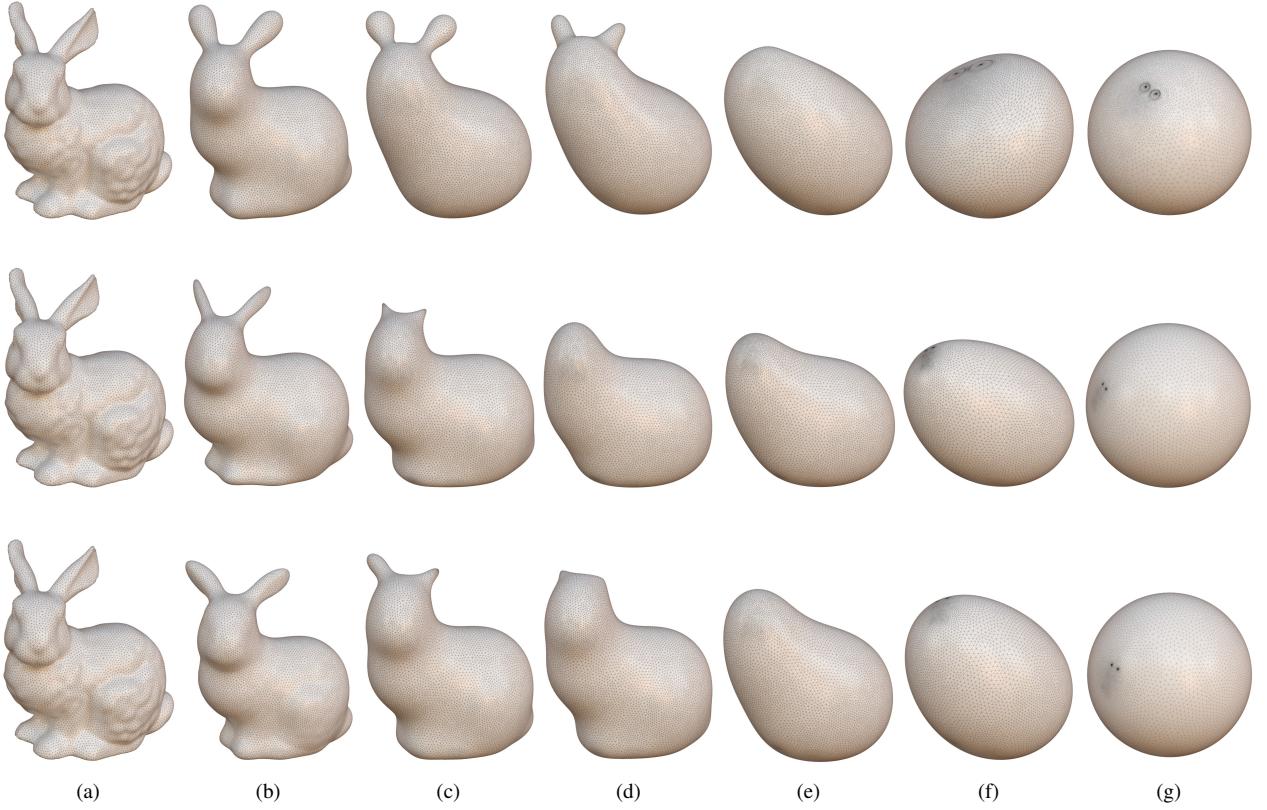


Figure 26: The deforming comparision of our method (the first row), cMCF [KSBC12] (the second row) and conformal Willmore flow [CPS13] (the third row).

Model	Method	Flip	Area Mean	Edge Mean	Angle Mean	Symm Dirich	Conformal AMIPS2D	ASAP	ARAP	Green Lagrange
flow(29k)	UNF	0	$2.38 \times 10^{-5}$	$9.56 \times 10^{-6}$	0.12	$5.79 \times 10^7$	$6.14 \times 10^4$	0.04	0.48	1.77
	cMCF	0	$2.09 \times 10^{-5}$	$1.01 \times 10^{-5}$	0.04	$4.22 \times 10^9$	$5.72 \times 10^4$	$8.03 \times 10^{-4}$	0.46	1.08
	Harmonic	1584	$2.21 \times 10^{-5}$	$8.60 \times 10^{-6}$	0.27	$2.31 \times 10^{10}$	$3.60 \times 10^5$	0.14	0.45	1.87
armadillo(60k)	UNF	0	$1.95 \times 10^{-5}$	$8.66 \times 10^{-6}$	0.23	$9.81 \times 10^6$	$1.43 \times 10^5$	0.10	1.20	6.00
	cMCF	0	$1.90 \times 10^{-5}$	$1.04 \times 10^{-5}$	0.13	$1.56E+12$	$1.21 \times 10^5$	$7.09 \times 10^{-4}$	1.24	3.77
	Harmonic	7897	$1.82 \times 10^{-5}$	$8.99 \times 10^{-6}$	0.50	$1.35E+12$	$1.74 \times 10^6$	0.11	1.19	5.62
bimba(100k)	UNF	0	$1.16 \times 10^{-5}$	$3.88 \times 10^{-6}$	0.23	$1.29 \times 10^6$	$2.29 \times 10^5$	0.56	1.39	34.92
	cMCF	0	$1.26 \times 10^{-5}$	$4.53 \times 10^{-6}$	0.02	$2.23 \times 10^6$	$2.00 \times 10^5$	$2.87 \times 10^{-3}$	1.14	9.10
	Harmonic	0	$9.50 \times 10^{-6}$	$3.74 \times 10^{-6}$	0.42	$6.16 \times 10^6$	$4.94 \times 10^5$	1.99	2.02	56.14

Table 7: The distortions of our method(UNF), cMCF [KSBC12] and harmonic spheric mapping [GWC\*04].

- [FL16] FU X.-M., LIU Y.: Computing inversion-free mappings by simplex assembly. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 216. 9
- [FLG15] FU X.-M., LIU Y., GUO B.: Computing locally injective mappings by advanced mips. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 71. 3, 7
- [Flo03] FLOATER M.: One-to-one piecewise linear mappings over triangulations. *Mathematics of Computation* 72, 242 (2003), 685–696. 3
- [GJ\*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. <http://eigen.tuxfamily.org>, 2010. 7
- [GWC\*04] GU X., WANG Y., CHAN T. F., THOMPSON P. M., YAU S.-T.: Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging* 23, 8 (2004), 949–958. 10, 13
- [H\*84] HUISKEN G., ET AL.: *Flow by mean curvature of convex surfaces into spheres*. Australian National University, Centre for Mathematical Analysis, 1984. 3
- [HFL18] HU X., FU X.-M., LIU L.: Advanced hierarchical spherical parameterizations. *IEEE transactions on visualization and computer graphics* 24, 6 (2018), 1930–1941. 3
- [Jak10] JAKOB W.: Mitsuba renderer, 2010. <http://www.mitsuba-renderer.org>. 12
- [JKLG08] JIN M., KIM J., LUO F., GU X.: Discrete surface ricci flow. *Visualization and Computer Graphics, IEEE Transactions on* 14, 5 (2008), 1030–1043. 3

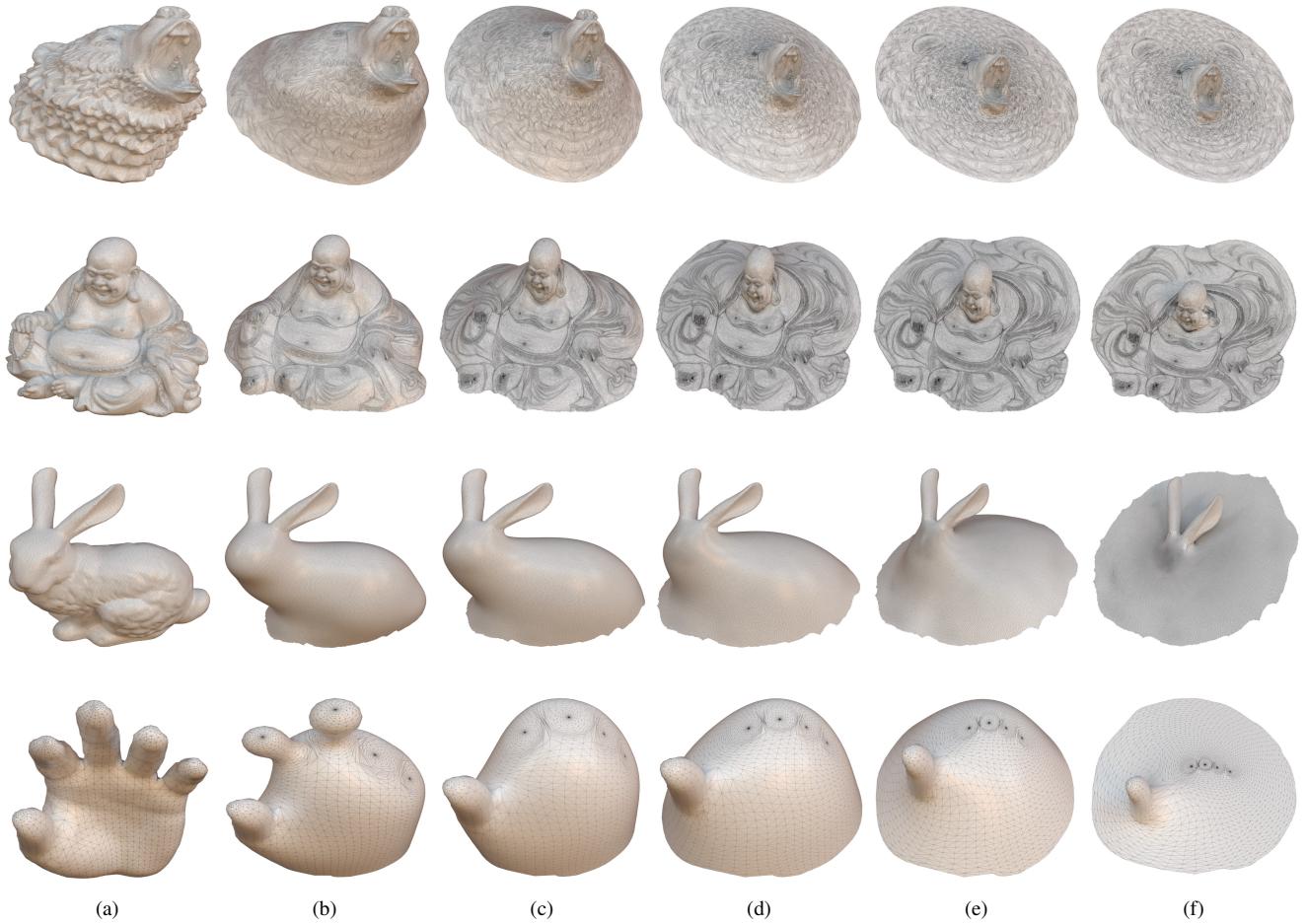


Figure 27: The center faces in the original meshes (a) are selected and kept unchanged in our deformation; (f) exhibits the final partial planar mesh mapping.

- [JSP17] JIANG Z., SCHAEFER S., PANZOZO D.: Simplicial complex augmentation framework for bijective maps. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 186. 3
- [Kap90] KAPOULEAS N.: Complete constant mean curvature surfaces in euclidean three-space. *Annals of Mathematics* 131, 2 (1990), 239–330. 3
- [KSBC12] KAZHDAN M., SOLOMON J., BEN-CHEN M.: Can mean-curvature flow be modified to be non-singular? In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1745–1754. 2, 3, 9, 10, 13
- [LGZ\*16] LIU T., GAO M., ZHU L., SIFAKIS E., KAVAN L.: Fast and robust inversion-free shape manipulation. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 1–11. 3
- [Liu14] LIU C.: *The three-body problem*, vol. 1. Macmillan, 2014. 2
- [Liu15] LIU C.: *The dark forest*, vol. 2. Macmillan, 2015. 2
- [LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. In *ACM transactions on graphics (TOG)* (2002), vol. 21, ACM, pp. 362–371. 3
- [Luo04] LUO F.: Combinatorial Yamabe flow on surfaces. *Communications in Contemporary Mathematics* 6, 05 (2004), 765–780. 3
- [LYNF18] LIU L., YE C., NI R., FU X.-M.: Progressive parameterizations. *ACM Transactions on Graphics (SIGGRAPH)* 37, 4 (2018). 3
- [LZ14] LEVI Z., ZORIN D.: Strict minimizers for geometric optimization. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 185. 3
- [LZX\*08] LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S. J.: A local/global approach to mesh parameterization. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1495–1504. 3, 7
- [Mor17] MORRISON M. A.: Death’s end. *World Literature Today* 91, 1 (2017), 74–75. 2
- [MTAD08] MULLEN P., TONG Y., ALLIEZ P., DESBRUN M.: Spectral conformal parameterization. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1487–1494. 3
- [OR09] OLISCHLÄGER N., RUMPF M.: Two step time discretization of willmore flow. In *Mathematics of Surfaces XIII*. Springer, 2009, pp. 278–292. 3
- [PCL\*12] PAN H., CHOI Y.-K., LIU Y., HU W., DU Q., POLTHIER K., ZHANG C., WANG W.: Robust modeling of constant mean curvature surfaces. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 85. 3
- [PKC\*16] PRADA F., KAZHDAN M., CHUANG M., COLLET A., HOPPE H.: Motion graphs for unstructured textured meshes. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 108. 3
- [PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36. 3, 5

- [PR02] POLTHIER K., ROSSMAN W.: Discrete constant mean curvature surfaces and their index. In *Visualization and Mathematics* (2002), Cite-seer. 3
- [PS87] PINKALL U., STERLING I.: Willmore surfaces. *The Mathematical Intelligencer* 9, 2 (1987), 38–43. 2
- [Ren15] RENKA R. J.: A simple and efficient method for modeling constant mean curvature surfaces. *SIAM Journal on Scientific Computing* 37, 4 (2015), A2076–A2099. 3
- [RPPSH17] RABINOVICH M., PORANNE R., PANZOZZO D., SORKINE-HORNUNG O.: Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 16. 3, 7, 9, 12
- [SCL\*16] SU K., CHEN W., LEI N., CUI L., JIANG J., GU X. D.: Measure controllable volumetric mesh parameterization. *Computer-Aided Design* 78 (2016), 188 – 198. SPM 2016. 3
- [SCL\*17] SU K., CHEN W., LEI N., ZHANG J., QIAN K., GU X.: Volume preserving mesh parameterization based on optimal mass transportation. *Computer-Aided Design* 82 (2017), 42 – 56. Isogeometric Design and Analysis. 3
- [SCQ\*16] SU K., CUI L., QIAN K., LEI N., ZHANG J., ZHANG M., GU X. D.: Area-preserving mesh parameterization for poly-annulus surfaces based on optimal mass transportation. *Computer Aided Geometric Design* 46 (2016), 76 – 91. 3
- [SdS01] SHEFFER A., DE STURLER E.: Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with computers* 17, 3 (2001), 326–337. 3
- [SK01] SCHNEIDER R., KOBBELT L.: Geometric fairing of irregular meshes for free-form surface design. *Computer aided geometric design* 18, 4 (2001), 359–379. 3
- [SKPSH13] SCHÜLLER C., KAVAN L., PANZOZZO D., SORKINE-HORNUNG O.: Locally injective mappings. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 125–135. 3
- [SPR\*07] SHEFFER A., PRAUN E., ROSE K., ET AL.: Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision* 2, 2 (2007), 105–171. 3
- [SPSH\*17] SHTENGEL A., PORANNE R., SORKINE-HORNUNG O., KOVALSKY S. Z., LIPMAN Y.: Geometric optimization via composite majorization. *ACM Trans. Graph* 36, 4 (2017). 9
- [SRML07] SUN X., ROSIN P., MARTIN R., LANGBEIN F.: Fast and effective feature-preserving mesh denoising. *IEEE transactions on visualization and computer graphics* 13, 5 (2007). 3
- [SS15] SMITH J., SCHAEFER S.: Bijective parameterization with free boundaries. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 70. 3
- [SSP08] SPRINGBORN B., SCHRÖDER P., PINKALL U.: Conformal equivalence of triangle meshes. In *ACM Transactions on Graphics (TOG)* (2008), vol. 27, ACM, p. 77. 3
- [Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 351–358. 3
- [Tau01] TAUBIN G.: Linear anisotropic mesh filtering. *Res. Rep. RC2213 IBM* 1, 4 (2001). 3
- [TWBO02] TASDIZEN T., WHITAKER R., BURCHARD P., OSHER S.: Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of the conference on Visualization'02* (2002), IEEE Computer Society, pp. 125–132. 3
- [TWBO03] TASDIZEN T., WHITAKER R., BURCHARD P., OSHER S.: Geometric surface processing via normal maps. *ACM Transactions on Graphics (TOG)* 22, 4 (2003), 1012–1033. 3
- [WBH\*07] WARDETZKY M., BERGOU M., HARMON D., ZORIN D., GRINSPUN E.: Discrete quadratic curvature energies. *Computer Aided Geometric Design* 24, 8 (2007), 499–518. 3
- [WLL14] WANG C., LIU Z., LIU L.: As-rigid-as-possible spherical parameterization. *Graphical Models* 76, 5 (2014), 457–467. 3
- [XPB06] XU G., PAN Q., BAJAJ C. L.: Discrete surface modelling using partial differential equations. *Computer Aided Geometric Design* 23, 2 (2006), 125–145. 3
- [XZ08] XU G., ZHANG Q.: A general framework for surface modeling using geometric partial differential equations. *Computer Aided Geometric Design* 25, 3 (2008), 181–202. 3
- [XZWB06] XU D., ZHANG H., WANG Q., BAO H.: Poisson shape interpolation. *Graphical Models* 68, 3 (2006), 268–281. 3
- [YLZG18] YU X., LEI N., ZHENG X., GU X.: Surface parameterization based on polar factorization. *Journal of Computational and Applied Mathematics* 329 (2018), 24–36. 3
- [YOB02] YAGOU H., OHTAKE Y., BELYAEV A.: Mesh smoothing via mean and median filtering applied to face normals. In *Geometric Modeling and Processing, 2002. Proceedings* (2002), IEEE, pp. 124–131. 3
- [YZX\*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 644–651. 3
- [ZFAT11] ZHENG Y., FU H., AU O. K.-C., TAI C.-L.: Bilateral normal filtering for mesh denoising. *IEEE Transactions on Visualization and Computer Graphics* 17, 10 (2011), 1521–1530. 3
- [ZG16] ZHAO H., GORTLER S. J.: A report on shape deformation with a stretching and bending energy. *CoRR abs/1603.06821* (2016). URL: <http://arxiv.org/abs/1603.06821>. 3, 5
- [Zha16] ZHAO H.: MeshDGP: A C Sharp mesh processing framework, 2016. <http://meshdgp.github.io/>. 12
- [ZLG\*18] ZHAO H., LI X., GE H., LEI N., ZHANG M., WANG X., GU X.: Conformal mesh parameterization using discrete calabi flow. *Computer Aided Geometric Design* (2018). 3
- [ZLJW06] ZHANG L., LIU L., JI Z., WANG G.: Manifold parameterization. In *Advances in Computer Graphics*. Springer, 2006, pp. 160–171. 3
- [ZLL\*17] ZHAO H., LEI N., LI X., ZENG P., XU K., GU X.: Robust Edge-Preserved Surface Mesh Polycube Deformation. In *Pacific Graphics Short Papers* (2017), Barbic J., Lin W.-C., Sorkine-Hornung O., (Eds.), The Eurographics Association. doi:10.2312/pg.20171319. 3, 5
- [ZMT05] ZHANG E., MISCHAIKOW K., TURK G.: Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics (TOG)* 24, 1 (2005), 1–27. 7
- [ZRKS05] ZAYER R., RÖSSL C., KARNI Z., SEIDEL H.-P.: Harmonic guidance for surface deformation. In *Computer Graphics Forum* (2005), vol. 24, Wiley Online Library, pp. 601–609. 3