Introduction to Mesh Parameterization

Richard Liu

September 23, 2021

1 Introduction

Surface parameterization involves the problem of constructing a (bijective) map between two surfaces with similar topology. This problem has its roots in map-making from over 2000 years ago – attempting to make a map of the Earth, or in essence taking a 3D sphere and flattening it into the 2D plane. An unfortunate fact is that it is in fact impossible to get a "perfect" map of the Earth, in that the resulting map perfectly preserves both lengths and angles (an **isometric** parameterization). We will investigate this further later in the notes, but in the context of a subproblem that is fundamental to computer graphics: mesh parameterization.

Mesh parameterization is the problem of computing a mapping between a surface represented by a triangular mesh and another surface. The surface that the mesh is mapped to is referred to as the **parameter domain**. Mesh parameterization is a fundamental problem in computer graphics that has only become more ubiquitous in the past two decades due to its growing use as a tool in mesh processing applications. Some examples of these applications include:

- Detail mapping and transfer: by storing the per-triangle details (e.g. color, bump maps, animation data, etc.) and mapping them to a common parameter domain (in 2D), one can transfer this data between surfaces
- Mesh completion: meshes generated from scans often contain holes are disconnected components, which can be repaired if a template already exists that can be mapped to the problematic scan in question
- Correspondence between objects: by mapping to a common domain, one can also analyze the common factors between objects to establish a correspondence between common features/construct a taxonomy for a database
- Remeshing: certain triangulations are more desirable than others of the same surface (e.g. for numerical simulations). A common technique is to parameterize a surface,

triangulate the parameterization in a desired way, then map the triangulation back.

2 Basic Definitions and Framework

Here we review some basic definitions and properties from differential geometry and analysis.

Given a function $f : X \to Y$,

Definition 2.1. f is **injective** or **one-to-one**, if $\forall x, x' \in X, f(x) = f(x') \Rightarrow x = x'$.

Definition 2.2. f is **surjective** or **onto**, if $\forall y \in Y, \exists x \in X s.t. y = f(x)$.

Definition 2.3. f is **bijective** if f is both injective and surjective. Equivalently, f is bijective iff it is **invertible**.

Note that if the codomain (Y) of f is equal to its image ($f(X)$), then f is trivially surjective. In most cases for mesh parameterization, we assume the image to be the codomain, so **injectivity is the property we mostly care about**.

Now we establish the framework for parameterization. Let $\Omega \subset \mathbb{R}^2$ be a **simply connected** (without holes) region. Let $f : \Omega \to \mathbb{R}^3$ be continuous and injective. Then we call the image of f a **surface**

$$S = f(\Omega) = \{f(u, v) : (u, v) \in \Omega\}$$

We say that f is a **parameterization** of S over the **parameter domain** Ω. Observe that f is a bijection between Ω and S by definition. Below is a basic example for the parameterization of a unit cylinder.

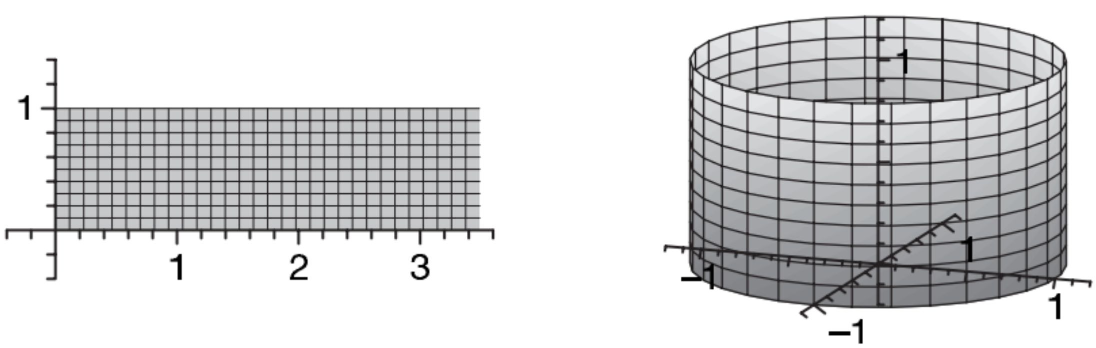


Figure 1: Parameterization of Unit Cylinder

- *Parameter domain:* $\Omega = \{(u, v) \in \mathbb{R}^2 : u \in [0, 2\pi), v \in [0, 1]\}$

- *Surface:* $S = \{(x, y, z) \in \mathbb{R}^{:} x^2 + y^2 = 1, z \in [0, 1]\}$
- *Parameterization:* $f(u, v) = (\cos u, \sin u, v)$
- *Inverse:* $f^{-1}(x, y, z) = (\arccos x, z)$

Remark. A parameterization $f : \Omega \to S$ is never unique. Given any bijection $\gamma : \Omega \to \Omega$, $g = f \circ \gamma$ is a parameterization of S over Ω.

We just need a few more assumptions on f to be able to use it for deriving some important **intrinsic surface properties**, or properties that are independent of how the surface sits in space (extrinsic geometry). Another useful way of understanding this is that intrinsic properties are those knowable to a tiny observer living **on** the surface (like how humans can observe the curvature of the Earth without ever knowing that the Earth is actually hurtling 30 km per second around the Sun).

First observe that the partial derivatives of f, $f_u = \frac{\partial f}{\partial u}$ and $_v = \frac{\partial f}{\partial v}$ (note that these are functions from $\mathbb{R}^2$ to $\mathbb{R}^3$) span the **local tangent plane**. Thus we can derive the **surface normal**

$$n_f = \frac{f_u \times f_v}{||f_u \times f_v||}$$

Here we require that our parameterization f is **regular**.

Definition 2.4. A parameterization $f : \Omega \subset \mathbb{R}^2 \to S \subset \mathbb{R}^3$ is **regular** if the tangent vectors f_u and f_v are always linearly independent.

This ensures that the surface normal n_f is non-zero.

Remark. The surface normal is always **independent** of the parameterization, regardless of the surface, making it an **intrinsic** property.

The parameterization also relates directly to the **first fundamental form** and **second fundamental form**. They are fundamental precisely because they determine the key metric properties of a surface, such as the **gaussian curvature**, **mean curvature**, and **surface area**.

Definition 2.5. The first fundamental form is defined through the parameterization f as

$$\mathbf{I}_f = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_v \cdot f_u & f_v \cdot f_v \end{pmatrix} = \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

We can then define the area of our surface

$$A(S) = \int_\Omega \sqrt{\det(\mathbf{I}_f)} du dv$$

In order to derive the second fundamental form, we require that the parameterization is **twice differentiable**.

Definition 2.6. The second fundamental form is defined as

$$\mathbf{II}_f = \begin{pmatrix} f_{uu} \cdot n_f & f_{uv} \cdot n_f \\ f_{uv} \cdot n_f & f_{vv} \cdot n_f \end{pmatrix} = \begin{pmatrix} L & M \\ M & N \end{pmatrix}$$

Definition 2.7. The Gaussian curvature is

$$K = \det(\mathbf{I}_f^{-1}\mathbf{II}_f) = \frac{\det \mathbf{II}_f}{\det \mathbf{I}_f} = \frac{LN - M^2}{EG - F^2}$$

Definition 2.8. The mean curvature is

$$H = \frac{1}{2}\text{trace}(\mathbf{I}_f^{-1}\mathbf{II}_f) = \frac{LG - 2MF + NE}{2(EG - F^2)}$$

Applying these definitions to the parameterization of the cylinder above, we get that $K = 0$ and $H = \frac{1}{2}$. The cylinder is often the canonical example of a surface with 0 gaussian curvature.

Definition 2.9. A surface S is **developable** if $\forall p \in S$, $K(p) = 0$, i.e. the Gaussian curvature is 0 everywhere on S.

The above definition will be key in our discussion of parameterization properties.

Now we turn to the Jacobian of f

Definition 2.10. The **Jacobian** of parameterization f is the 3 x 2 matrix of partial derivatives of f.

$$J_f = (f_u, f_v)$$

Observe that the first fundamental form can be rewritten as

$$J_f^T J_f = \begin{pmatrix} f_u^T \\ f_v^T \end{pmatrix} (f_u f_v) = \mathbf{I}_f$$

which should make it clear that $\mathbf{I}$ is a symmetric matrix. Thus the eigenvalues λ_1 and λ_2 are given by

$$\lambda_{1,2} = \frac{1}{2}((E + G) \pm \sqrt{4F^2 + (E - G)^2}$$

.

We can also make use of the following fact from linear algebra

Remark. For a matrix A, the singular values are the square roots of the eigenvalues of $A^T A$.

where the singular values are defined as the diagonal entries σ_1, σ_2 in the singular value decomposition.

Definition 2.11. SVD For any $m \times n$ matrix J, the **singular value decomposition** (SVD) is given by

$$J = U\Sigma V^T$$

where Σ is an $m \times n$ diagonal matrix, and U and V are $m \times m$ and $n \times n$ orthonormal matrices, respectively.

By the above, the SVD of the Jacobian is thus

$$J_f = U \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix} V^T$$

where $\sigma_1 = \sqrt{\lambda_1}$, $\sigma_2 = \sqrt{\lambda_2}$.

With the above in place, we can now start talking about what makes a "good" parameterization.

3 Properties of Parameterizations

As mentioned in the introduction, most of the time there does **not** exist an **isometric** map from a surface to the plane, i.e. a map which perfectly preserves both angles and areas. Maps which preserve each individually are called **conformal** and **equiareal**, respectively.

Definition 3.1. A parameterization is **conformal**, or **angle-preserving**, when the singular values of the Jacobian are equal, i.e. $\sigma_1 = \sigma_2$.

Definition 3.2. A parameterization is **equiareal/authalic**, or **area-preserving**, when the singular values of the Jacobian multiply to 1, i.e. $\sigma_1\sigma_2 = 1$.

Definition 3.3. A parameterization is **isometric**, or **length-preserving** iff it is conformal and equiareal, i.e. $\sigma_1 = \sigma_2 = 1$.

As a matter of fact, Gauss showed the following in 1827.

Theorem 1. *Globally isometric parameterizations (from 3D to 2D) only exist for* ***developable*** *surfaces (i.e. $K = 0$ everywhere).*

Examples of developable surfaces include planes, cones, and cylinders, but most surfaces are not developable. Due to the above theoretical constraint, surface parameterization approaches will focus on finding a mapping that is either conformal, equiareal, or some combination of the two. From the above definitions, we know that the singular values of the Jacobian completely determine the distortion from parameterization. Thus a common form of a function to minimize to obtain the "best" parameterization is taking the average distortion over the whole domain,

$$E(f) = \int_{\Omega} E(\sigma_1(u,v), \sigma_2(u,v))dudv/A(\Omega)$$

where E is some function with minimal value(s) defined depending on the desired property (e.g. for conformal parameterization you want E to take on a minimum value along the entirety of (x, x) for $x \in \mathbb{R}_+$).

4 Discrete/Mesh Setting

The above framework establishes the **general** framework for thinking about surface parameterizations. Now let's focus on the issue of triangle meshes specifically, which can be considered as piecewise linear surfaces.

Definition 4.1. A mesh is a triangulation $M = (V, E, F)$, where $V = \{v_i\} \subset \mathbb{R}^3$, $E = \{e_{ij}\}$, and $F = \{f_{ijk}\}$ are the vertex, edge, and face sets, respectively. More formally, edge e_{ij} represents the convex hull between vertices v_i and v_j (i.e. line segment), and face f_{ijk} is the convex hull of non-collinear points v_i, v_j, v_k.

Not confusingly at all, we now care about the **inverse** of the parameterization f of M, $g = f^{-1}$. g is uniquely determined by the following:

- **Parameter points** $u_i = g(v_i) \forall v_i \in V$
- g is **continuous** and **linear** within each triangle F
- For each triangle $f_{ijk} = \{v_i, v_j, v_k\}$, g is a linear map from f_{ijk} to **parameter triangle** $\tilde{f}_{ijk} = \{u_i, u_j, u_k\}$
- The **parameter domain** is defined as the union of all parameter triangle $\Omega = \bigcup \tilde{f}_{ijk}$

4.1 Desirable Properties

We already discussed two key properties of parameterizations in general: **conformality** and **equiareality**. An important goal in the case of applications of mesh parameterization is **bijectivity**. Bijectivity is a requirement for most, if not all, mesh processing applications. An example for texture mapping is that when the map is not bijective, a single point in the texture could map to multiple, disconnected regions of the surface, which makes annotating different regions of the surface impossible.

However, certain applications are satisfied with just **local injectivity/bijectivity** (recall from above that we assume the image and codomain are the same).

Definition 4.2. A mesh parameterization is **locally injective** if no triangles change orientation ("flip" or "fold over") during the parameterization. Concretely, this results in an inverted normal of the flipped triangle.

Definition 4.3. A mesh parameterization is **globally bijective** if it is locally injective and the boundary of the parameterization does not intersect itself.

An example of a parameterization that is locally injective, but not globally bijective is shown in figure 5. The wavy cone is isometrically flattened without error, but the boundary ends up self-intersecting.

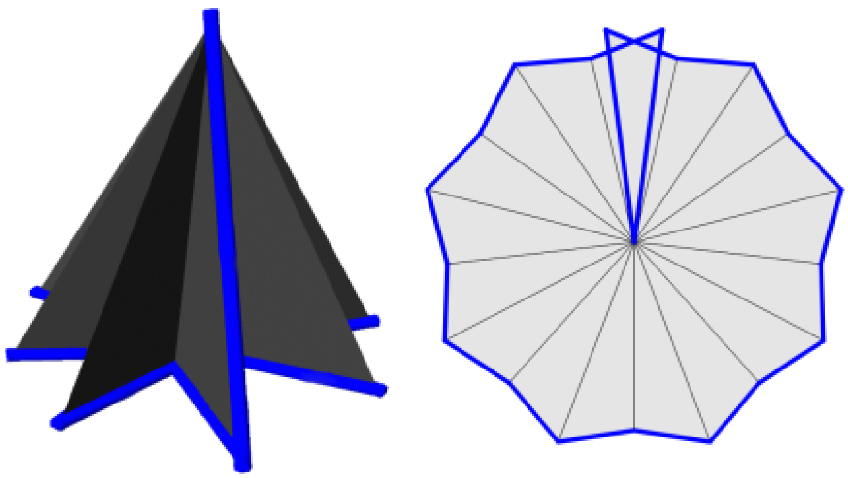


Figure 2: Locally injective but not globally bijective parameterization

Early methods of mesh parameterization relied on fixing the boundary of the surface to the boundary of a convex polygon that forms the parameter domain. In certain instances, this guarantees bijectivity, at the expense of potentially high distortion because the surface boundary is complex/non-convex. As a result, **boundary-free** methods were introduced. In the following section, I will categorize the existing methods starting with this boundary vs boundary-free distinction, due to the somewhat chronological order they are introduced.

In general, mesh parameterization methods can be characterized by the following set of properties:

- Distortion minimized: {angle (conformal), area (equiareal), distance (isometric)}
- Boundary: {fixed, free}

- Bijectivity: {global, local}

5 Methods

5.1 Boundary-Based Maps

Boundary-based, or **barycentric** mappings all follow the same general procedure.

1. Choose the **shape of the boundary** of the parameter domain and the **distribution** of the parameter points around the boundary.
2. Compute **barycentric coordinates** for the interior vertices
3. Solve a **linear system** based around minimizing the spring energy of the mesh represented as a spring network.

Barycenteric coordinates are simply a way of representing an interior point in a polygon (typically triangle) as a linear combination of its vertices. These coordinates have found their way into many essential applications in rendering, geometric modeling, etc.

Definition 5.1. For a point x in the interior of a triangle $f_{ijk} = \{v_i, v_j, v_k\}$, values λ_i, λ_j, λ_k are **barycentric coordinates** of x with respect to the vertices of f_{ijk} if:

1. $x = \lambda_i v_i + \lambda_j v_j + \lambda_k v_k$
2. $\lambda_i + \lambda_j + \lambda_k = 1$

Note that the above definition is applied to triangle meshes specifically, but can easily be generalized to any n-gon meshes.

Remark. For triangle meshes, the barycentric coordinates of an interior point are **unique**. This is not the case for any polygons with more than 3 vertices.

Suppose we have a mesh M with n interior vertices and b boundary vertices. For a vertex x_i, let N_i represent the set of its neighboring vertices. Finally, for $x_j \in N_i$, let λ_{ij} be the barycentric coordinate at x_j for source vertex x_i. The linear system is then represented as

$$AU = \overline{U}$$
$$AV = \overline{V}$$

where $U = (u_1, \ldots, u_n)$ and $V = (v_1, , v_n)$ are the parameter coordinates of the interior vertices of the mesh. $\overline{U}$ and $\overline{V}$ are column vectors defined by

$$\overline{u}_i = \sum_{u_j \in N_i, j>n} \lambda_{ij} u_j$$
$$\overline{v}_i = \sum_{v_j \in N_i, j>n} \lambda_{ij} v_j$$

$A = (a_{ij})$ is an $n \times n$ matrix defined by

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\lambda_{ij} & \text{if } j \in N_i \\ 0 & \text{otherwise} \end{cases}$$

Below is a brief synopsis of the most popular boundary fixed methods, along with their properties and drawbacks.

1. (Tutte, 1963) was the first to introduce the above-described framework into the mesh parameterization context with his seminal work on straight-line embeddings of planar graphs, where the coefficients λ_{ij} are defined uniformly, so that $\lambda_{ij} = 1/|N_i|$, which are notably **not** barycentric coordinates. Tutte's embeddings **do not** offer any guarantee of distortion minimization, but do guarantee **bijectivity**.

2. (Eck et al., 1995) method makes use of **harmonic coordinates**, or **cotangent weights**, which are another type of barycentric coordinates, defined by normalizing $w_{ij} = \cot\gamma_{ij} + \cot\gamma_{ji}$ across the vertex neighbors of x_i, with γ representing the adjacent edge angles between x_i and x_j. The harmonic parameterization aims to minimize **angle distortion**. One major drawback is that if the mesh contains **obtuse angles, then the weights can be negative**, which results in a non-bijective parameterization.

3. (Floater, 2003) discretized the mean value theorem in order to present another set of barycentric coordinates called **mean value coordinates**. Like (Eck et al., 1995)'s method, mean value coordinates do not require the 1-ring of neighboring faces to be coplanar. These coordinates are also applied in minimizing angular distortion in parameterization.

4. The above methods all perform poorly when the 3D meshes have non-convex boundaries or boundaries that differ significantly from the boundary of the parameter domain. Lee et al. 2007 introduce the idea of a "virtual" boundary, which augments the 3D boundary with extra triangles to make the boundary nicer and ultimately decrease the large distortions generated by problematic boundaries. Lee et al. then apply mean value coordinates to the actual parameterization.

5. (Jiang et al., 2017) offer a more recent look at the use of **scaffolding**, which similarly involves generating a virtual boundary with a particular triangulation, which is then

iteratively updated based on some metric, an isometric distortion energy in this case. Here the boundary isn't completely fixed, but is not totally free either. This method also suffers from high computational cost, requiring a re-triangulation of the scaffold for every iteration.

As discussed above, fixing the boundary comes with the convenience of limited guarantees of bijectivity and fast, linear-time solutions. However, it comes at the cost of high distortion in cases where the surface has highly non-convex boundaries, or there is no "natural" way to fix the 3D border on a convex polygon. In the next section we discuss state-of-the-art methods that "free" the boundary, dragging the choice of boundary vertices into the optimization process.

5.2 Boundary-Free Maps

Boundary-free methods are far more varied, but can be divided into three broad camps on the basis of their distortion-minimizing objective: **conformal, authalic/equiareal, and isometric**. Due to the sheer volume of methods that have been developed for each of these objectives, I will focus primarily on the conformal methods, with one exception of ARAP (an isometric method), and provide a brief primer on deformation analysis in order to motivate mathematical intuition behind the techniques.

5.2.1 Deformation analysis for meshes

Recall in the context of mesh parameterization, the parameterization function is piecewise linear for each triangle on the mesh. We can thus supply a **local basis** to each triangle, setting one of the vertices to the origin, as demonstrated in Figure 3

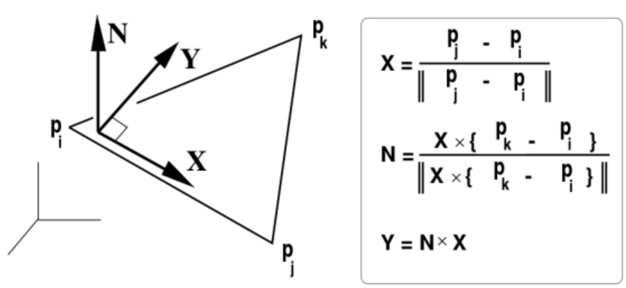


Figure 3: Local Basis for 3D Triangle

In this local basis, we can define the coordinates of any point p in the triangle as (p_X, p_Y), or (X, Y) for convenience (just remember that this is **different** from the orthonormal bases

X and Y). So now our **inverse parameterization function** becomes

$$\begin{cases} u(X,Y) = \lambda_i u_i + \lambda_j u_j + \lambda_k u_k \\ v(X,Y) = \lambda_i v_i + \lambda_j v_j + \lambda_k v_k \end{cases} \tag{1}$$

where λ is computed as the barycentric coordinates of our point p in **3D space**.

$$\begin{pmatrix} \lambda_i \\ \lambda_j \\ \lambda_k \end{pmatrix} = \frac{1}{2|T|_{X,Y}} \begin{pmatrix} Y_j - Y_k & X_k - X_j & X_j Y_k - X_k Y_j \\ Y_k - Y_i & X_i - X_k & X_k Y_i - X_i Y_k \\ Y_i - Y_j & X_j - X_i & X_i Y_j - X_j Y_i \end{pmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{2}$$

where $2|T|_{X,Y}$ is double the area of the triangle in **3D space**.

Through a simple substitution of the values of λ defined above into our inverse parameterization equation, we get the following equation for the gradient of u (equivalently for v)

$$\begin{pmatrix} \partial u/\partial X \\ \partial u/\partial Y \end{pmatrix} = M_T \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} = \frac{1}{2|T|_{X,Y}} \begin{pmatrix} Y_j - Y_k & Y_k - Y_i & Y_i - Y_j \\ X_k - X_j & X_i - X_k & X_j - X_i \end{pmatrix} \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} \tag{3}$$

Note that M_T is solely dependent on the geometry of triangle T. These gradients of the **inverse function** will prove to be useful in developing intuition for the conformal methods described below.

We now tie the concept of conformality back to our original framework. Recall from our differential geometry review that conformality implies that the singular values of the Jacobian (of the **forward** parameterization function f) are equal ($\sigma_1 = \sigma_2$). This implies that the Jacobian matrix is composed of a rotation and a scaling, aka a **similarity**.

Remark. Conformal maps **locally** correspond to similarities.

A visual way of describing this is that a conformal parameterization transforms elementary circles to elementary circles, and that the gradient vectors f_u and f_v are orthogonal, as show in Figure 4. Again this **does not** guarantee area/length preservation, which requires that $\sigma_1 = \sigma_2 = 1$ as well.

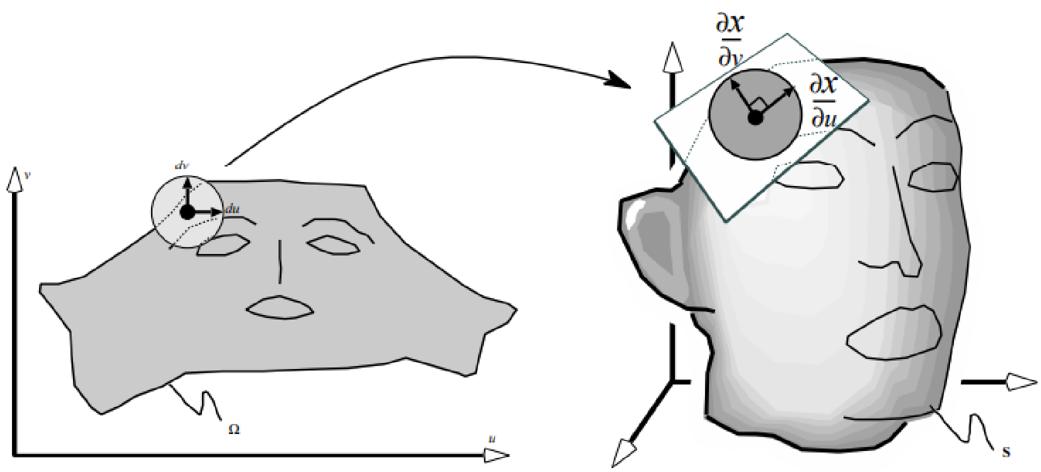


Figure 4.9: A conformal parameterization transforms an elementary circle into an elementary circle.

Figure 4

5.2.2 LSCM/DCP

The Least squares conformal maps (LSCM) method from (Lévy et al., n.d.) expresses the conformality condition as a quadratic optimization problem, specifically linear least squares. It has been well-established that a method introduced in the same year, Discrete Conformal Parameterization (DCP) from (Desbrun et al., 2002) are different derivations of the **same measure of conformal energy**, and thus equivalent optimization problems. We can understand the LSCM method in terms of a simple geometric relationship between the gradients of the **inverse parameterization function**. Take one triangle on the mesh, with orthonormal basis (X, Y). The conformality condition can be written as

$$\nabla v = \mathrm{rot}_{90}(\nabla u) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla u$$

where rot_{90} is a counterclockwise 90 degree rotation. Substituting in the gradient formula derived in 3, we get the following conformality condition

$$M_t \begin{pmatrix} v_i \\ v_j \\ v_k \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} M_t \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4}$$

Though we know from the Riemann mapping theorem that every continuous surface admits a conformal parameterization, this is restricted to **developable surfaces** for our discrete, piecewise linear setting. Thus, 4 does not have a solution for most meshes. Levy et al formulate a **conformal energy** metric based on minimizing the deviation of a mapping from 4.

$$E_{LSCM} = \sum_{T_{i,j,k}} |T| \left\| M_t \begin{pmatrix} v_i \\ v_j \\ v_k \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} M_t \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix} \right\|^2 \tag{5}$$

Note that 5 will look different than the term in Levy et al, as they take a complex analysis approach. I avoid the complex analysis discussion here for simplicity.

To avoid the trivial solution (sending all parameter coordinates to (0,0)), Levy et al. propose to **pin** two arbitrary vertices to the parameter domain at coordinates (0,0) and (1,1) (common endpoints for texture coordinates). A heuristic they use is to pin two diameter vertices, but any choice of pinned vertices would suffice.

Desbrun et al. end up deriving the same conformal energy problem in DCP, albeit through the minimization of the **dirichlet energy**.

Definition 5.2. Given a parameterization $f : \Omega \subset \mathbb{R}^2 \to S \subset R^3$, the **Dirichlet energy** measures the integral of the squared norm of the gradients.

$$E_D = \frac{1}{2} \int_S ||f_u||^2 + ||f_v||^2 dp$$

The Dirichlet energy can also be expressed in terms of the singular values σ_1, σ_2 of the Jacobian

$$E_D = \frac{\sigma_1^2 + \sigma_2^2}{2}$$

The equivalence of the minimization can easily be seen if we also write our conformal energy formula in terms of the singular values.

$$E_{LSCM} = E_C = \frac{1}{2} \int_S ||f_v - \text{rot}_{90}(f_u X)||^2 dp = \frac{(\sigma_1 - \sigma_2)^2}{2}$$

So now we have

$$E_D - E_C = \sigma_1 \sigma_2$$

But note that this is the product of the singular values of the Jacobian, which **is equivalent to the determinant**! The determinant of the Jacobian also has an incredibly useful geometric interpretation, which is simply the **area of the 3D surface**. Thus, we have the final relationship between conformal energy and Dirichlet energy

$$E_D - E_C = \det(J) = \text{Area}(S)$$

Since the 3D surface area is constant, the DCP and LSCM methods are equivalent!

Unfortunately, one striking weakness of LSCM/DCP is that they **do not** guarantee local injectivity nor prevent global overlaps.

Other properties of LSCM/DCP include

- The LSCM energy is a flawed metric: the energy is scaled by the area of the parameter domain, which is dependent on the choice of pinned vertices
- Linear complexity (just as fast as the fixed-boundary methods)

A noteworthy extension to LSCM/DCP is spectral conformal parameterization. (Mullen et al., 2008) are able to find a solution that minimizes conformal energy without having to pin 2 boundary vertices. Their approach makes use of generalized eigenvalue problem of the form

$$L_c u = \lambda B u$$

where u is the target parameterization and B is a degenerate diagonal binary matrix whose nonzero entries correspond to the boundary vertices.

5.2.3 MIPS

The mostly isometric parameterization of surfaces method ((Hormann et al., n.d.)) is the first known method to compute natural boundaries. This approach minimizes the **Dirichlet energy per parameter-space area**.

$$K_F(J_T) = ||J_T||_F ||J_T^{-1}||_F = \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} = \frac{\text{trace}(\mathbf{I}_T)}{\det J_T}$$

(Hormann et al., n.d.) are able to guarantee bijectivity through an iterative approach, moving vertices to minimize the energy, while simultaneously checking for flips and boundary overlaps. This iterative approach comes at the cost of being extremely slow.

5.2.4 ABF

The angle-based flattening approach by Sheffer and de Sturler 2001 is based on the following key observation:

Remark. A planar triangulation is defined by the corner angles of triangles (up to similarity).

They thus formulate the objective in **angle space** rather than over vertices like the previous methods. Specifically they minimize the objective

$$D(\alpha_i) = \sum_{i=1}^{3T} (\alpha_i - \beta_i)^2$$

where β_i are the known 3D angles and α_i are the unknown 2D angles.

Sheffer and Sturler impose the following constraints on the 2D parameter angles to ensure a valid triangulation

- All angles are positive
- All angles in each triangle sum to π
- The sum of angles around each vertex is 2π
- Edges shared by adjacent triangles have same length

This approach has the following properties

- Locally bijective (but not global)
- Non-linear (slow) and unstable for large meshes

Some noteworthy extensions of ABF include Zayer et al 2003, who enforce **convex boundaries** on the parameter domain to guarantee global bijectivity, and (Kharevych et al., n.d.), who introduce cone singularities to generate a **global parameterization**. This parameterization is continuous up to translation and rotation, other than at the singularities. They also need to compute edge paths between the cone singularity vertices to parameterize to the plane. Figure 5 demonstrates an example of a global parameterization with cone singularities.

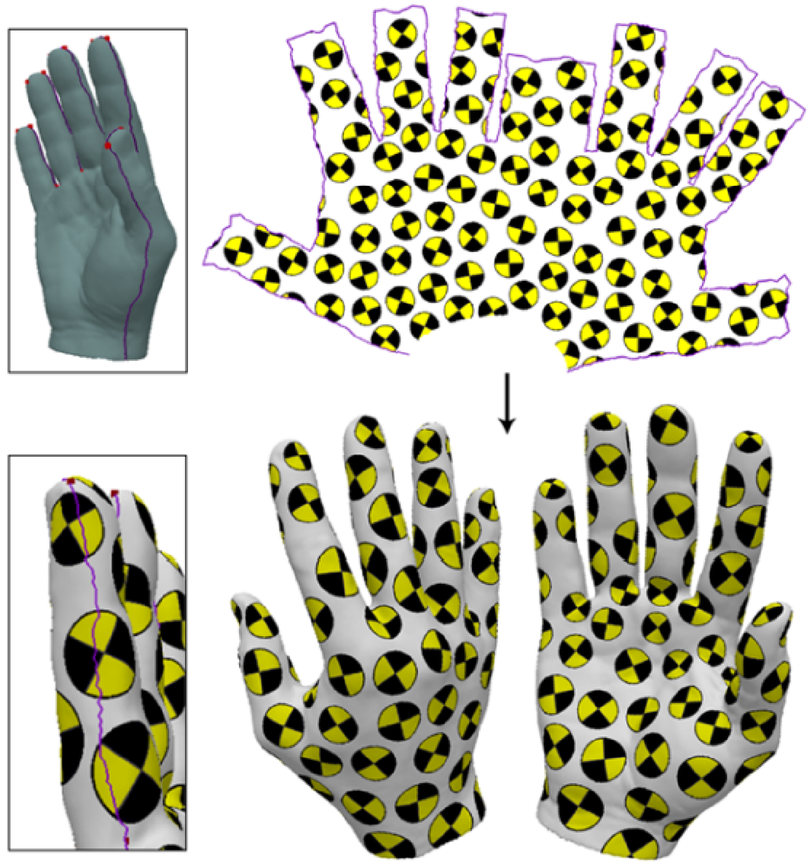


Figure 5: Global parameterization with cone singularities

5.2.5 Other Methods

As-rigid-as-possible parameterization was introduced by (Liu et al., 2008). to optimize a measure of isometric distortion. The authors apply a local-global optimization approach, where they iterate over each triangle, find the best rigid transformation (using the ARAP energy in Sorkine & Alexa), and stitch the triangles together through global linear system of equations. Similar to LSCM, however, ARAP does not offer any theoretical guarantees of local/global bijectivity. ARAP has been demonstrated to improve the area distortion when initialized with a conformal parameterization, such as LSCM, in Figure 6.

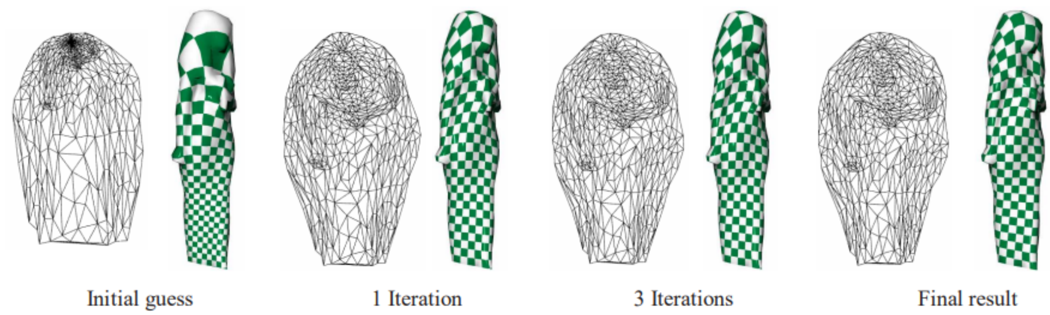
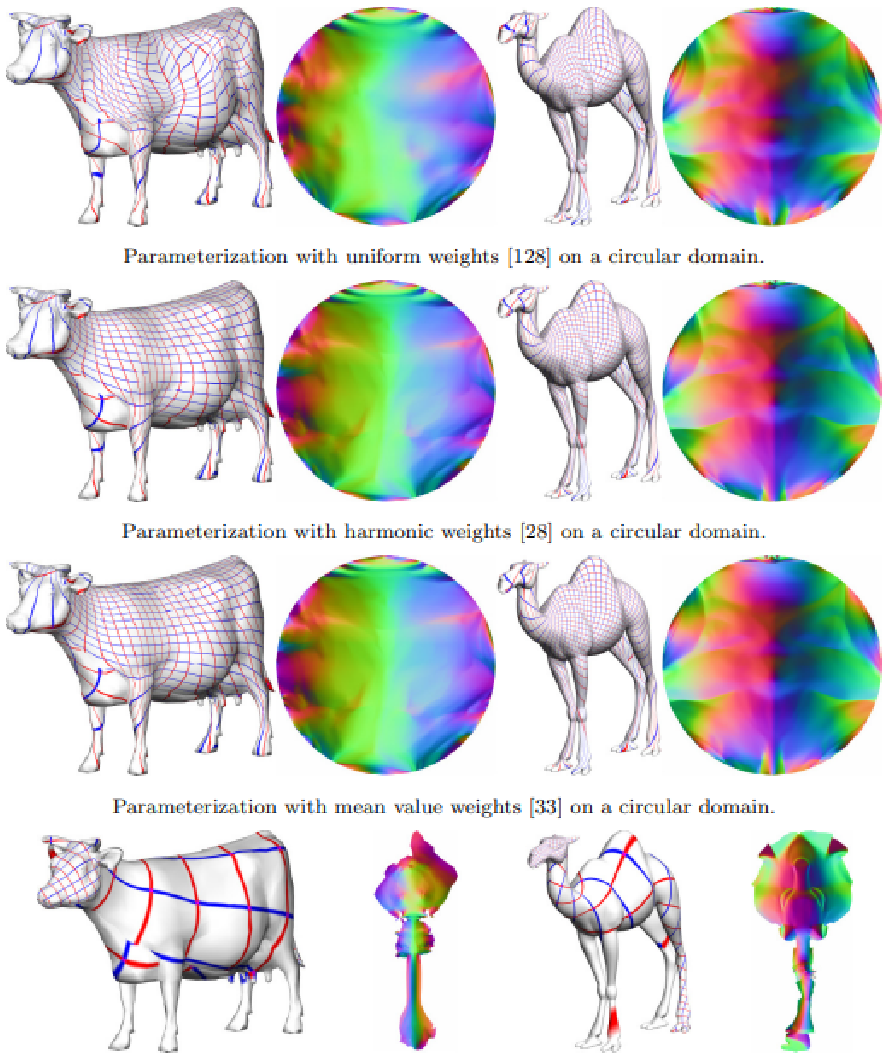


Figure 6: ARAP Progression when Initialized with LSCM

5.3 Discussion

Mesh parameterizations methods in general can be categorized by whether they fix the boundary, which distortion they are minimizing, and their complexity (linear vs nonlinear). In general the barycentric mappings, along with LSCM/DCP, are linear and fast to compute. However, they do not guarantee bijectivity and may induce substantial area distortion in the presence of high curvature, as shown in Figure 7. In contrast, non-linear methods such as MIPS and ABF are slow, but make some guarantees as to bijectivity and may outperform the linear methods, as shown in Figure 8.



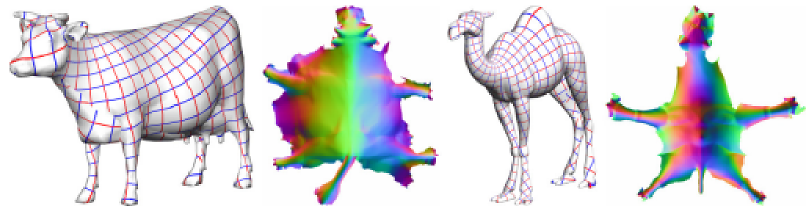
Parameterization with uniform weights [128] on a circular domain.

Parameterization with harmonic weights [28] on a circular domain.

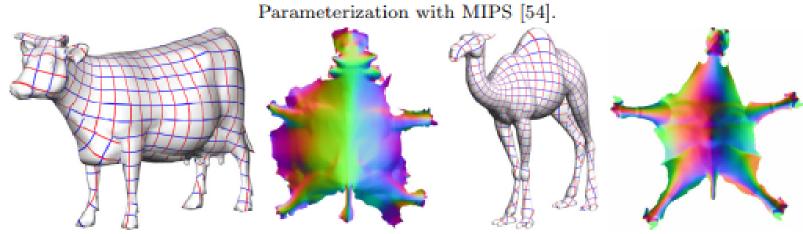
Parameterization with mean value weights [33] on a circular domain.

Parameterization with LSCM [79].

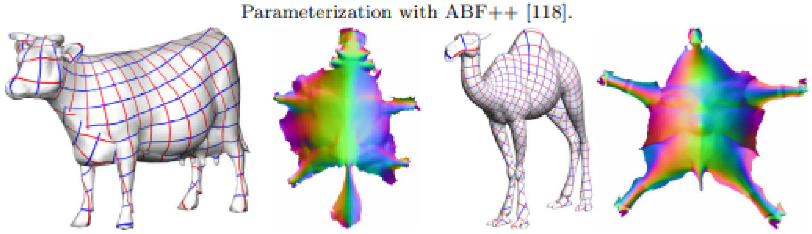
Figure 7: Linear methods



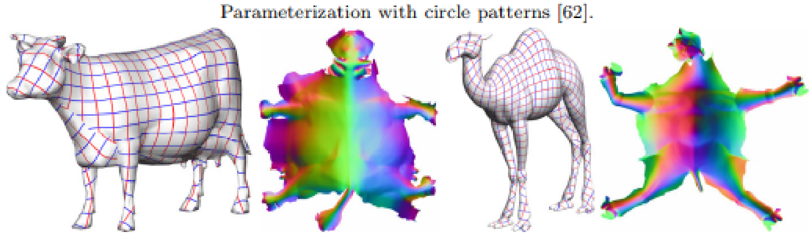
Parameterization with MIPS [54].



Parameterization with ABF++ [118].



Parameterization with circle patterns [62].



Stretch minimizing parameterization [107].

Figure 8: Non-linear methods

5.4 Acknowledgements

Much of the material presented in this document were derived from the excellent and thorough course notes on mesh parameterization created by (Hormann et al., n.d.).

References

Desbrun, Mathieu, Mark Meyer, and Pierre Alliez, "Intrinsic Parameterizations of Surface Meshes," *Computer Graphics Forum*, September 2002, *21* (3), 209–218.

Eck, Matthias, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle, "Multiresolution analysis of arbitrary meshes," in "Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95" ACM Press Not Known 1995, pp. 173–182.

Floater, Michael S., "Mean value coordinates," 2003.

Hormann, Kai, Bruno Lévy, and Alla She, "Mesh Parameterization: Theory and Practice," p. 122.

Jiang, Zhongshi, Scott Schaefer, and Daniele Panozzo, "Simplicial complex augmentation framework for bijective maps," *ACM Transactions on Graphics*, November 2017, *36* (6), 1–9.

Kharevych, Liliya, Boris Springborn, TU Berlin, and Peter Schroder, "Discrete Conformal Mappings via Circle Patterns," p. 8.

Liu, Ligang, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler, "A Local/Global Approach to Mesh Parameterization," *Computer Graphics Forum*, July 2008, *27* (5), 1495–1504.

Lévy, Bruno, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot, "Least Squares Conformal Maps for Automatic Texture Atlas Generation," p. 10.

Mullen, Patrick, Yiying Tong, Pierre Alliez, and Mathieu Desbrun, "Spectral Conformal Parameterization," *Computer Graphics Forum*, July 2008, *27* (5), 1487–1494.

Tutte, W. T., "How to Draw a Graph," *Proceedings of the London Mathematical Society*, 1963, *s3-13* (1), 743–767.