

# IFT 6113

## MESH PARAMETERIZATION

[tiny.cc/6113](http://tiny.cc/6113)

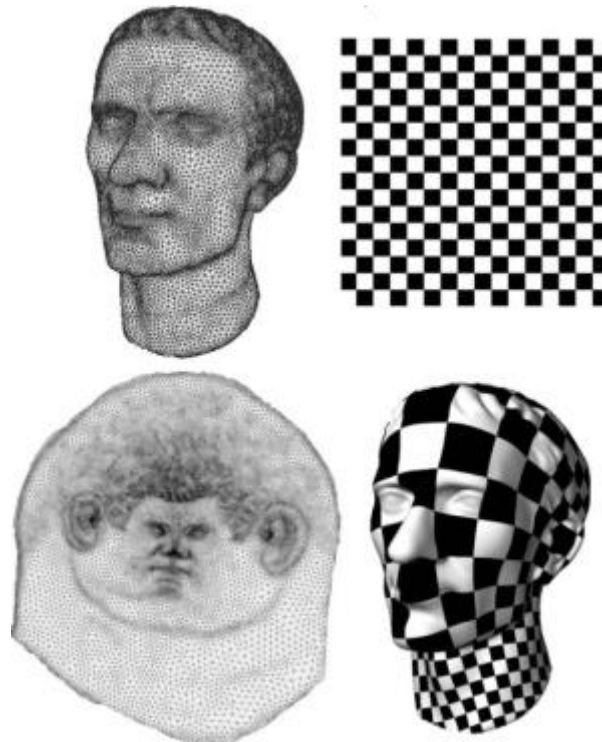
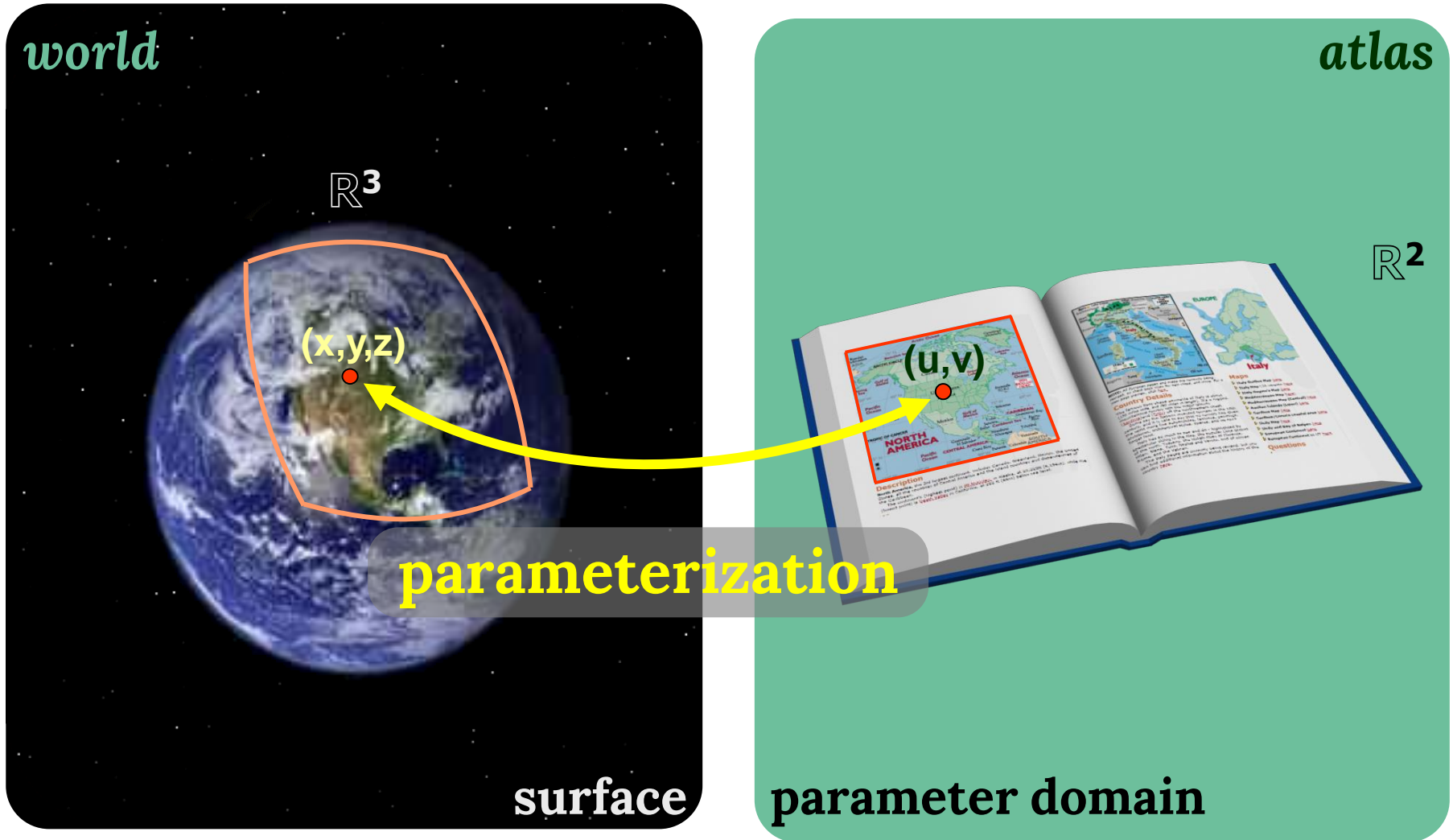


Image from [https://doc.cgal.org/latest/Surface\\_mesh\\_parameterization/index.html](https://doc.cgal.org/latest/Surface_mesh_parameterization/index.html)

# Mikhail Bessmeltsev

# What is Parameterization?

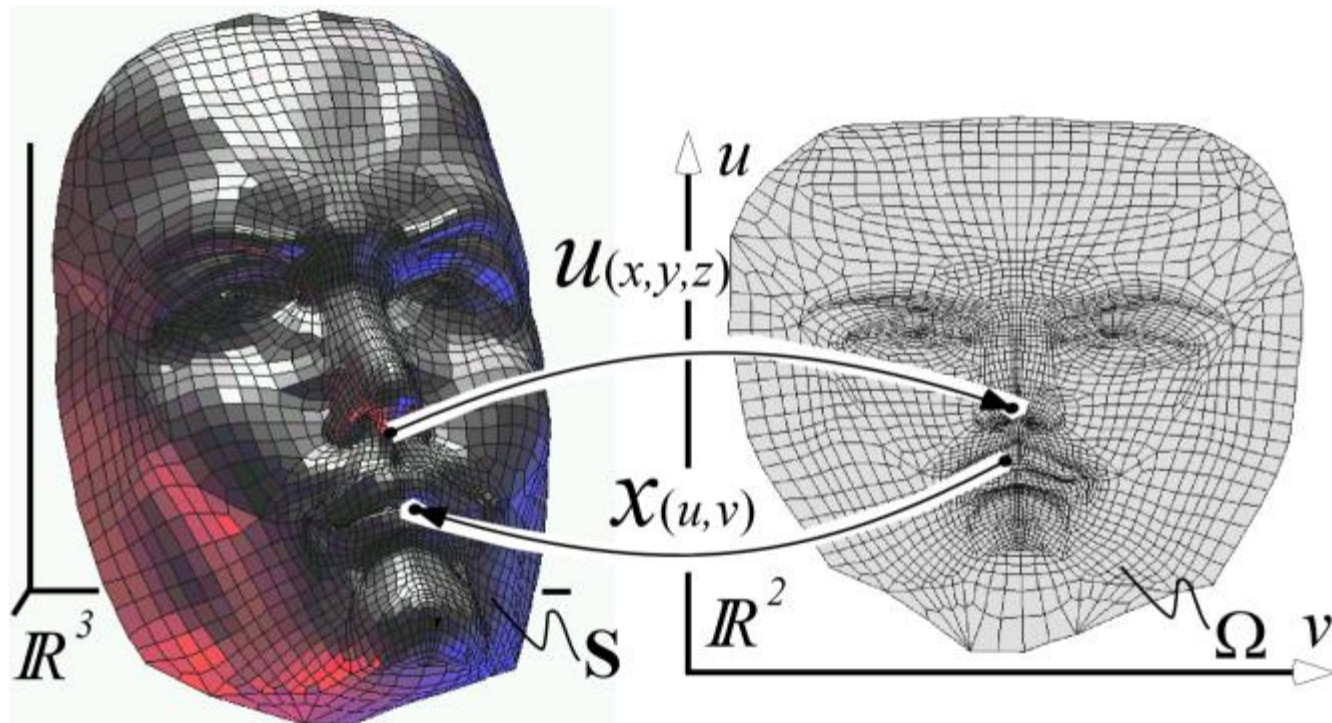


# Problem Definition

**Given:** surface  $S \subseteq \mathbb{R}^3$

domain  $D \subset \mathbb{R}^2$

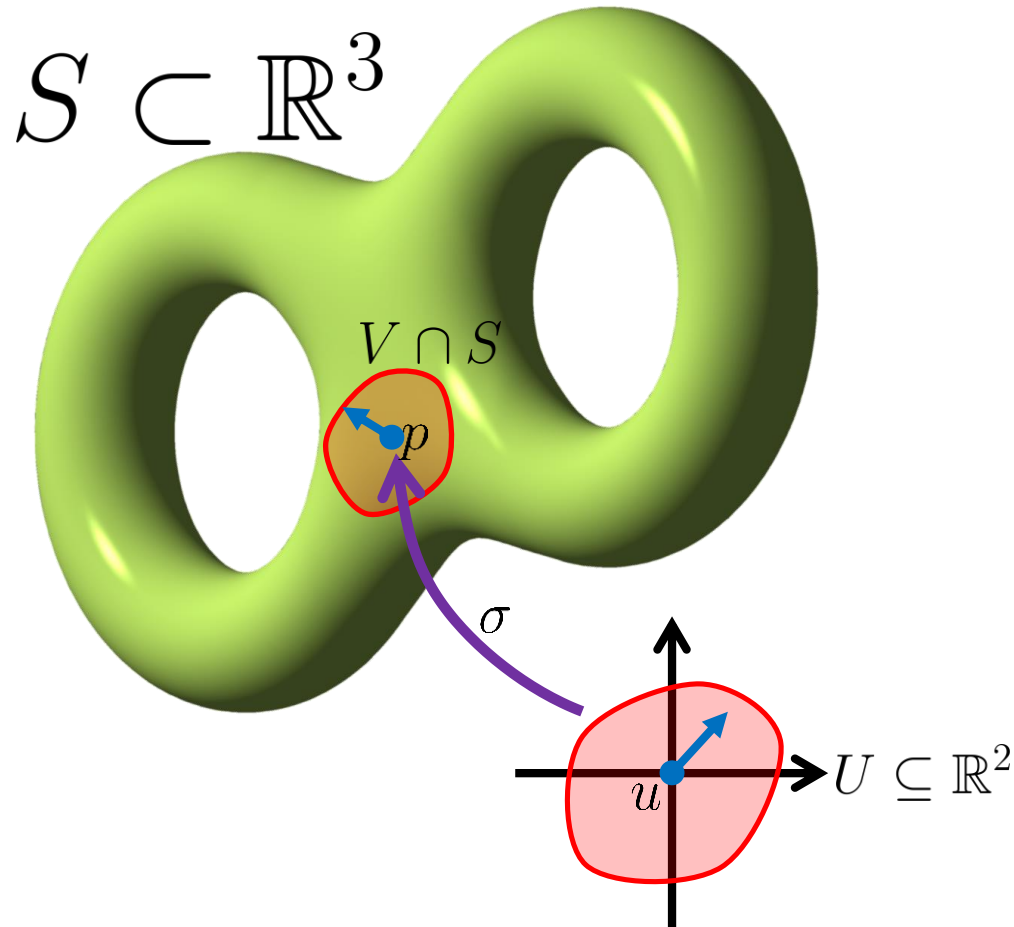
**Find** a bijective  $f: S \rightarrow D$



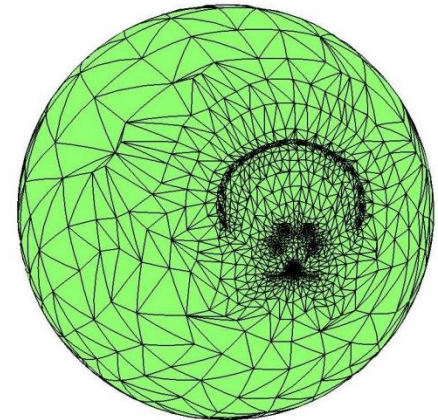
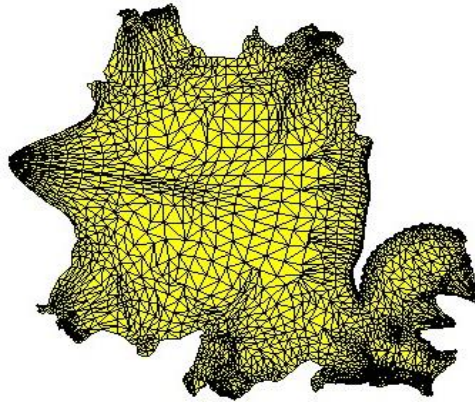
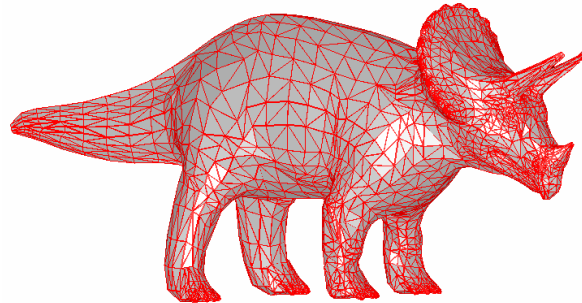
*Recall:*

# Tangent Space

$$T_p S := \text{Image}(D\sigma_u)$$



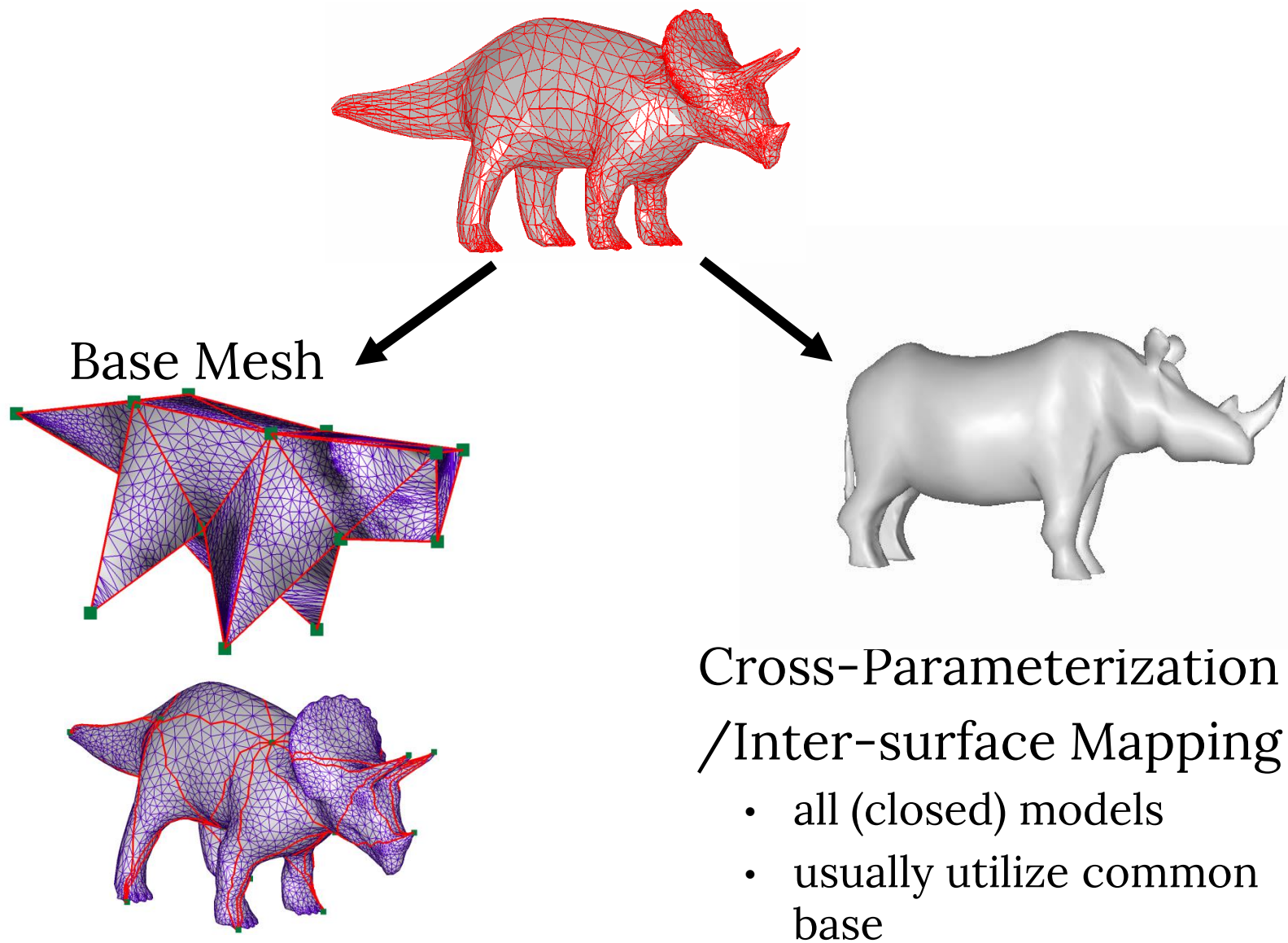
# Typical domains:



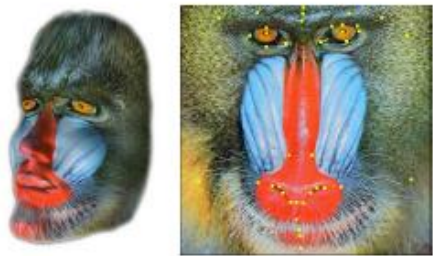
$\subseteq R^2$   
boundary

Sphere (no boundary)

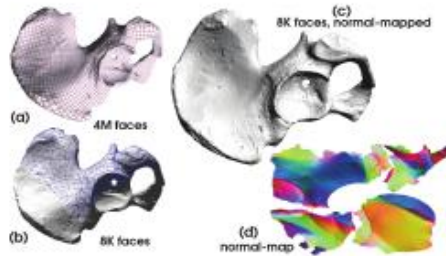
# Typical domains:



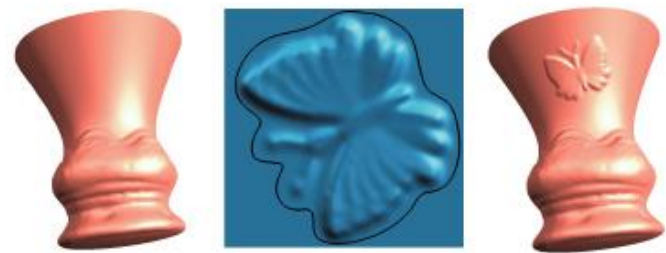
# Why Do We Need It?



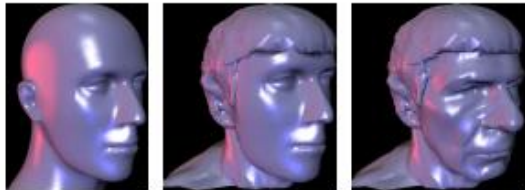
Texture Mapping



Normal Mapping



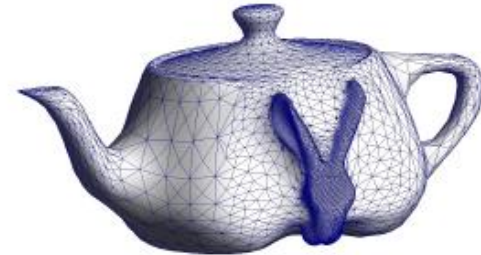
Detail Transfer



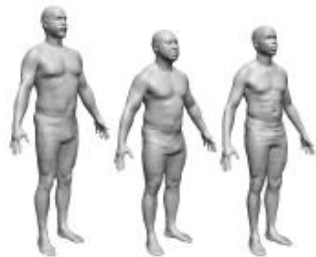
Morphing



Mesh Completion



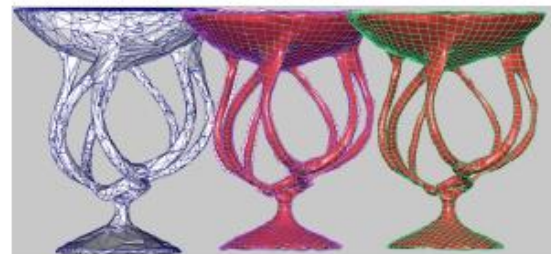
Editing



Databases



Remeshing



Surface Fitting

# Why Do We Need It?

## AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation

Thibault Groueix<sup>1\*</sup>, Matthew Fisher<sup>2</sup>, Vladimir G. Kim<sup>2</sup>, Bryan C. Russell<sup>2</sup>, Mathieu Aubry<sup>1</sup>

<sup>1</sup>LIGM (UMR 8049), École des Ponts, UPE, <sup>2</sup>Adobe Research

<http://imagine.enpc.fr/~groueixt/atlasnet/>

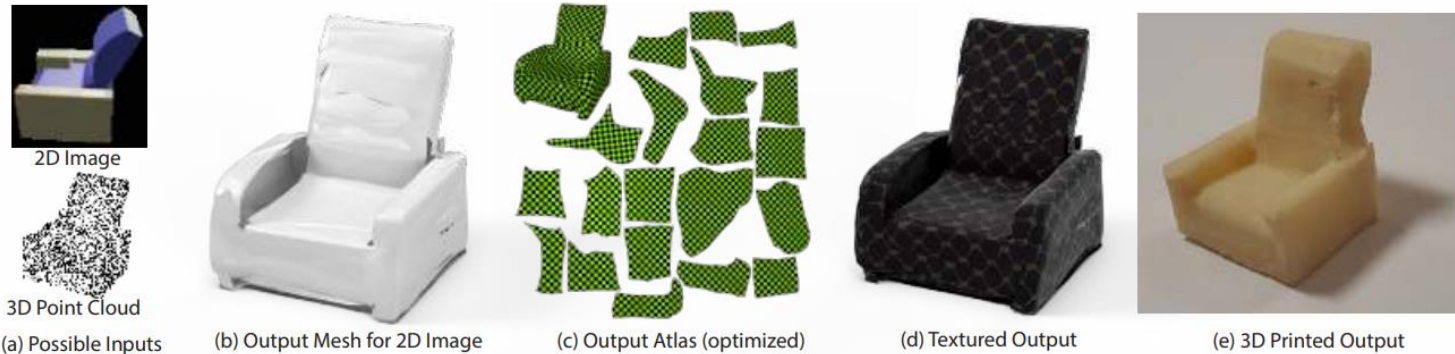


Figure 1. Given input as either a 2D image or a 3D point cloud (a), we automatically generate a corresponding 3D mesh (b) and its atlas parameterization (c). We can use the recovered mesh and atlas to apply texture to the output shape (d) as well as 3D print the results (e).

.CVJ 15 Feb 2018



Databases

Abstract



Remeshing

on surface details [9]. Efficiently leveraging capacity of the

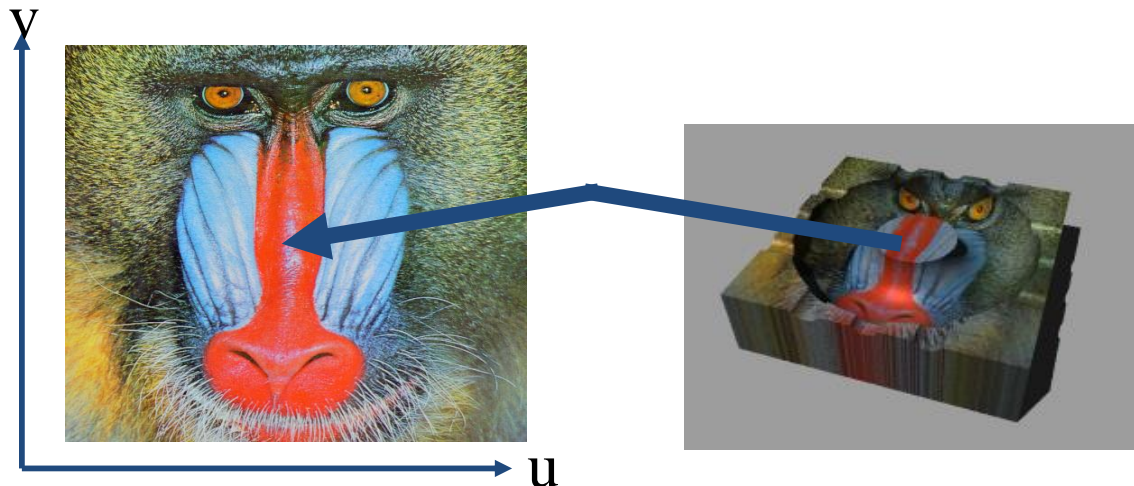


Surface Fitting

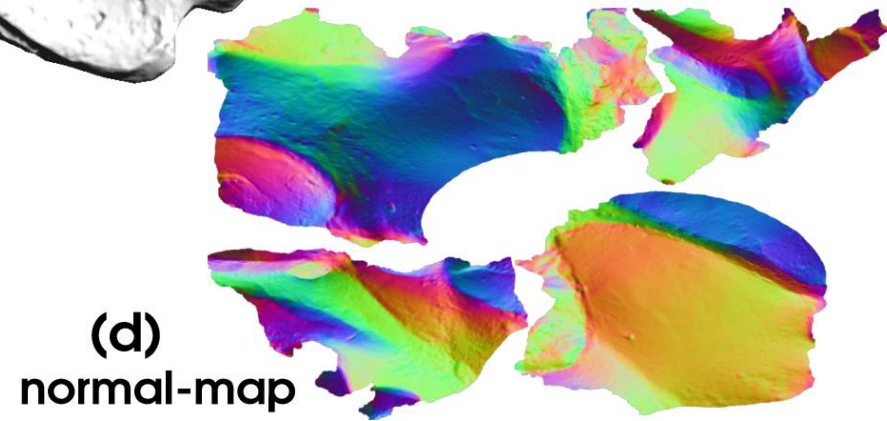
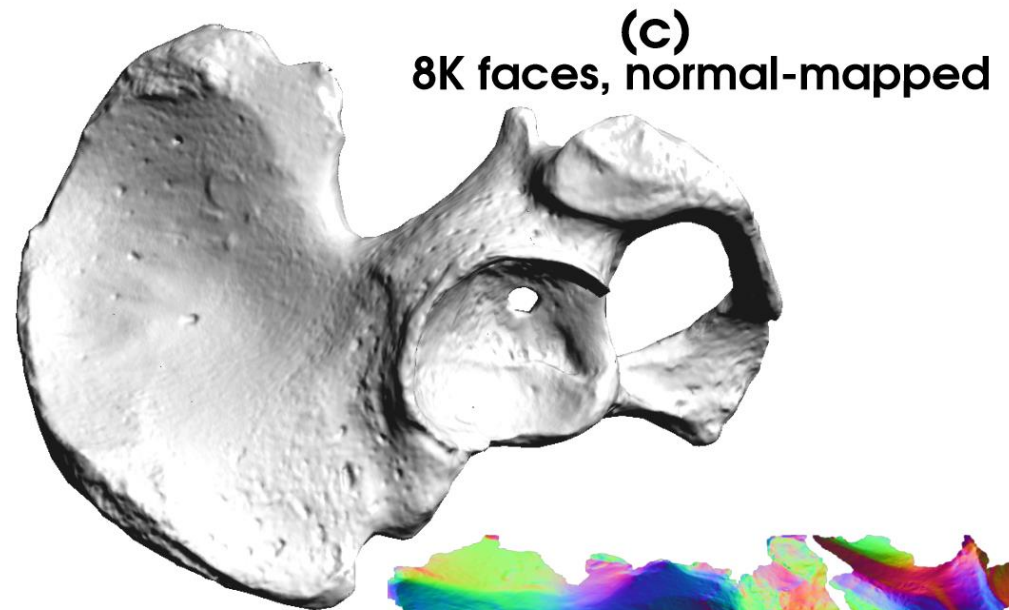
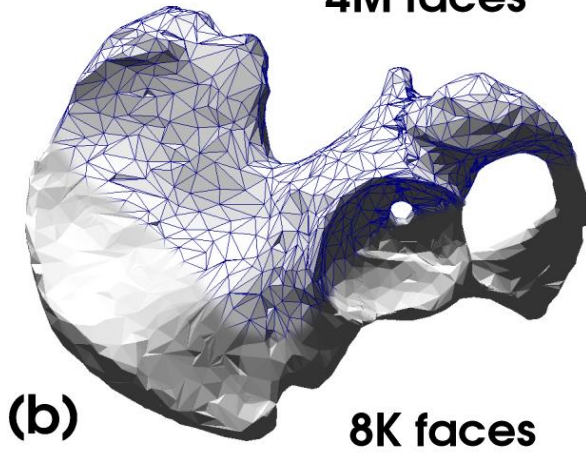
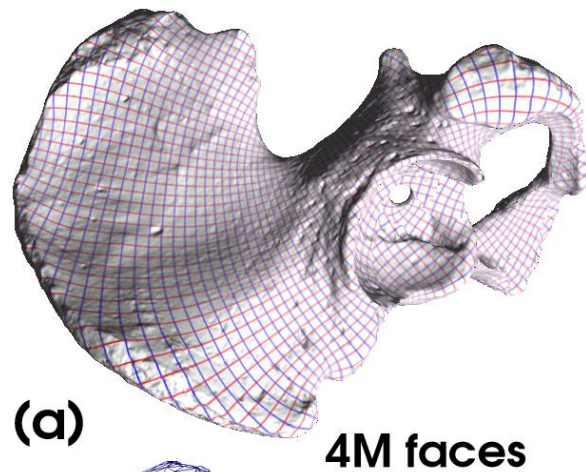


# Texture Mapping

- Define color for each point on object surface
- Map 2D texture to model surface:
  - Texture pattern defined over 2D domain  $(u, v)$
  - Assign  $(u, v)$  coordinates to each point on surface

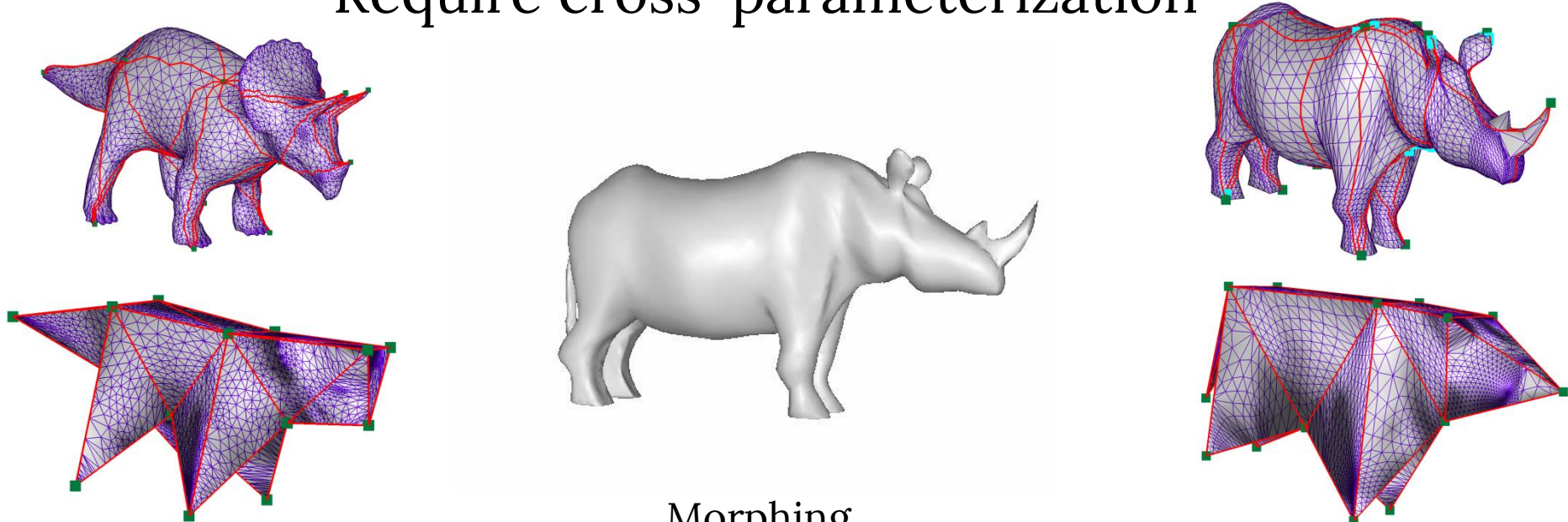


# Normal/Bump mapping

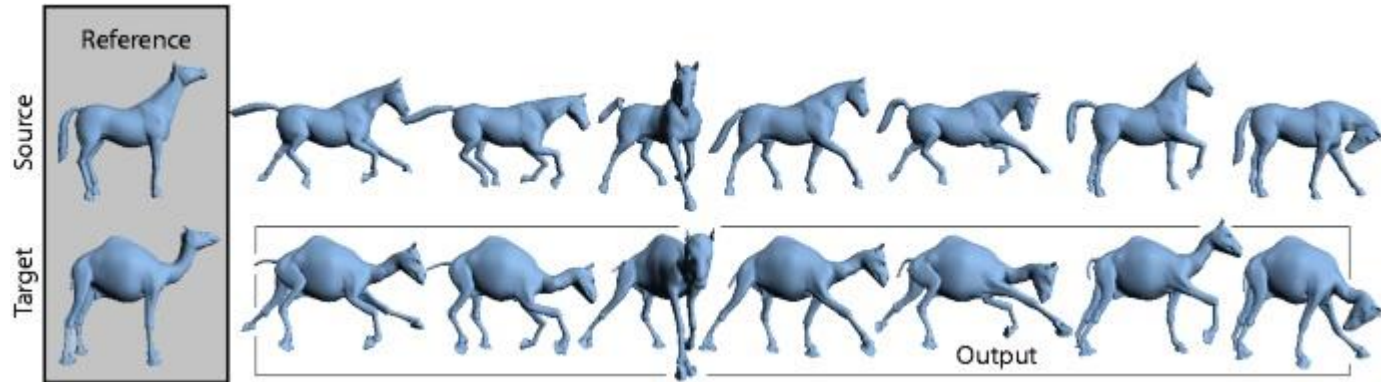


# Morphing/Properties Transfer

Require cross-parameterization

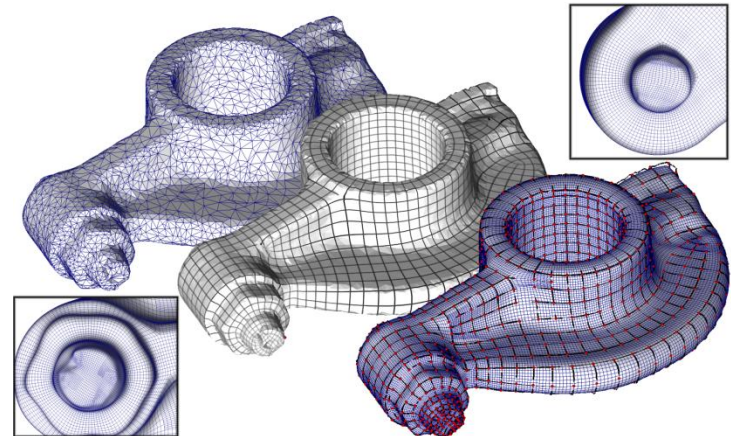
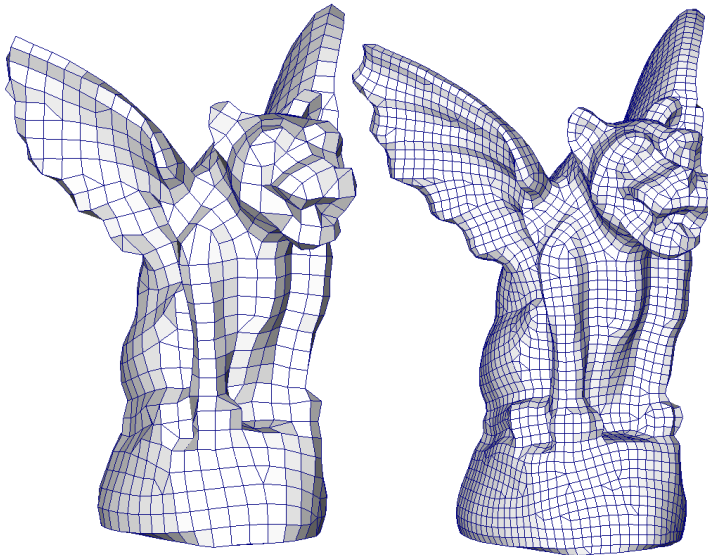
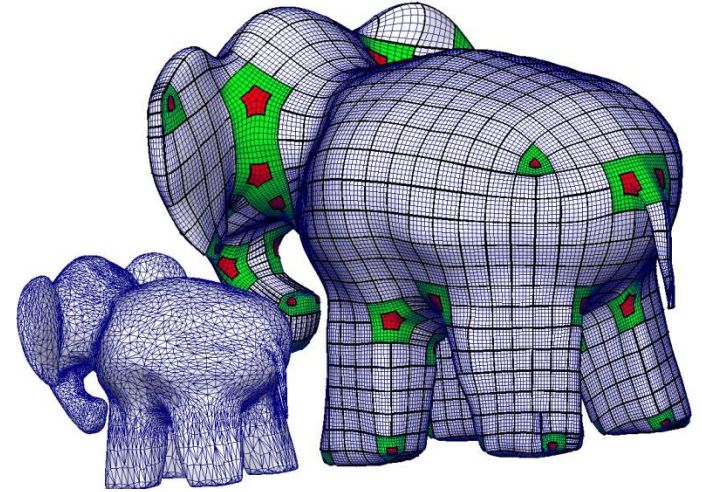
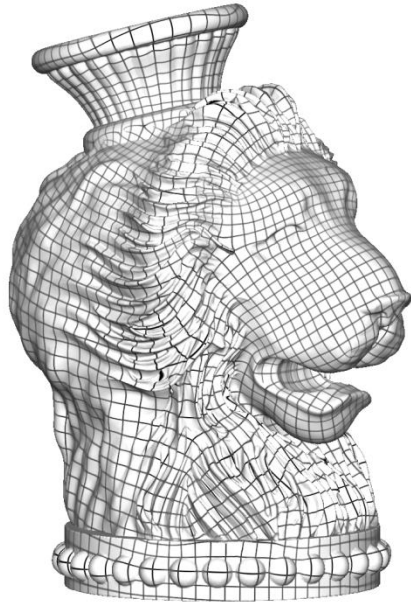
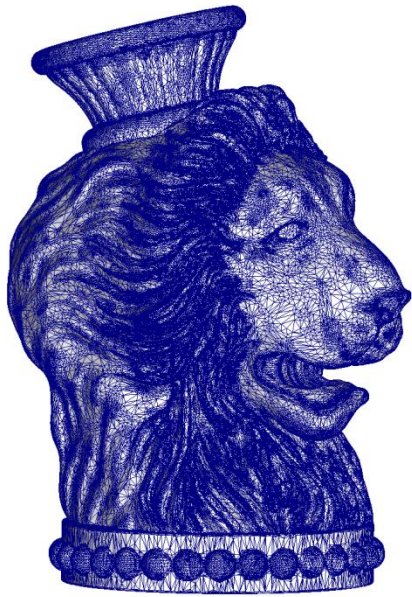


Morphing



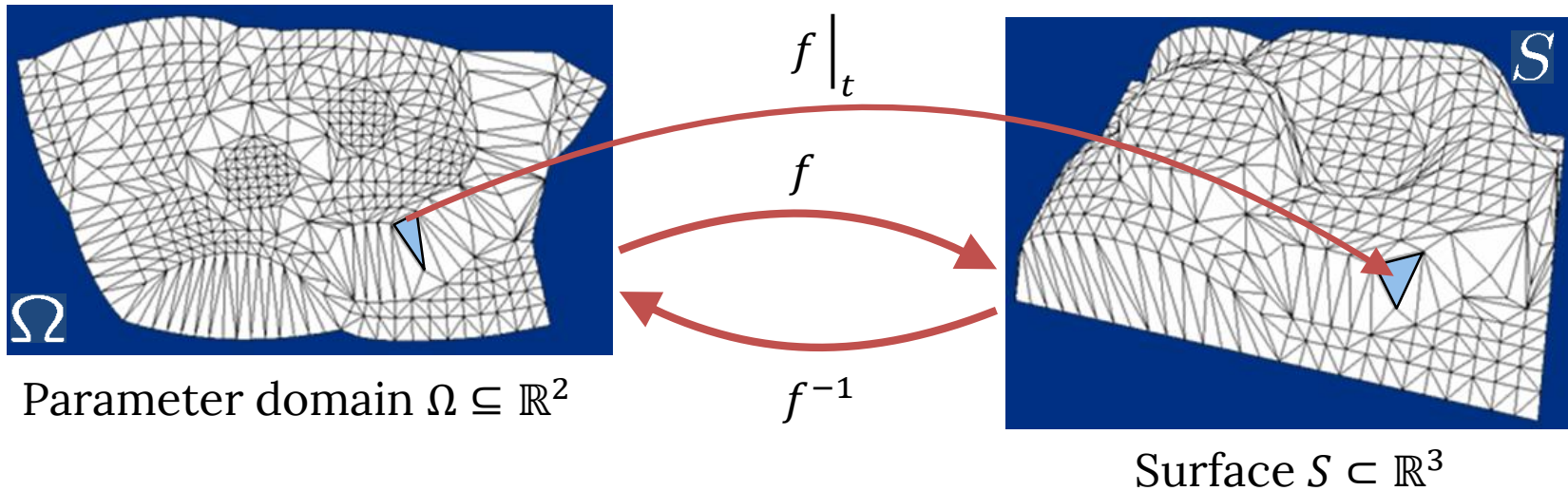
Deformation Transfer

# Remeshing & Surface Fitting



THEORY/BACKGROUND

# Mesh Parameterization



$f: \Omega \rightarrow S$ , s. t.

- $f$  is piecewise linear
  - $f|_t$  is linear
- $f$  is **bijective**
  - at least locally

# Earth maps!

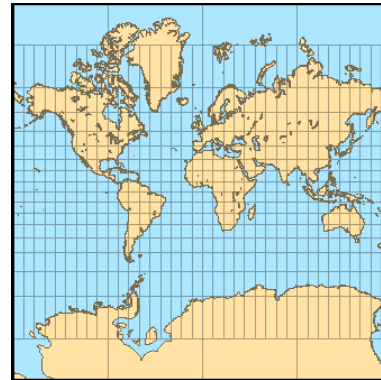
- Distortion is inevitable
- But we can preserve some properties exactly



orthographic  
~ 500 B.C.



stereographic  
~ 150 B.C.



Mercator  
1569



Lambert  
1772

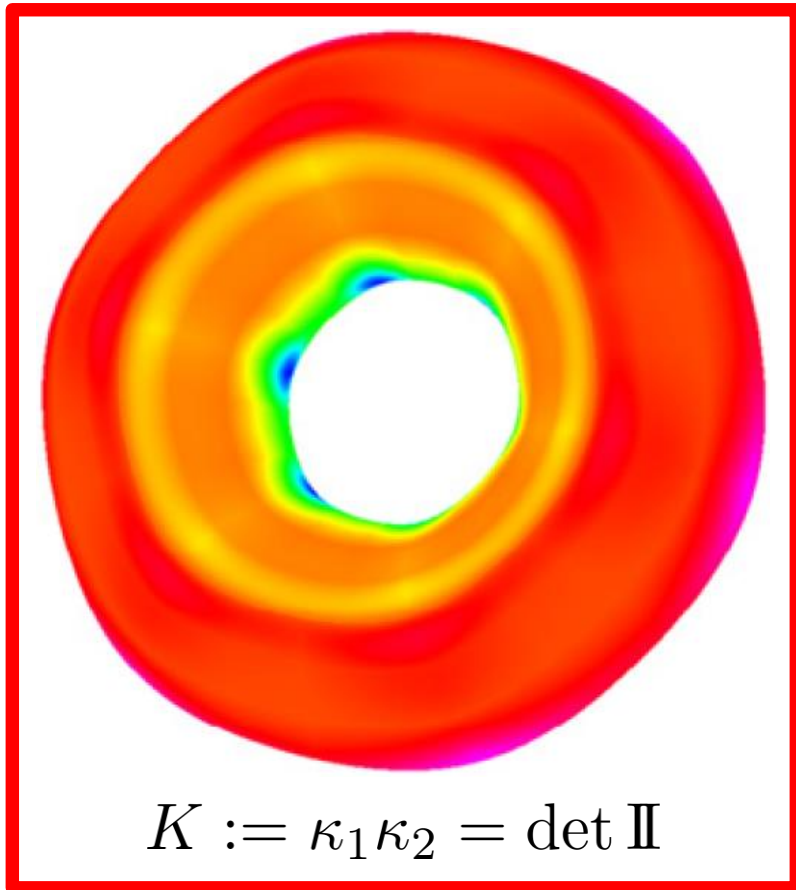


**conformal**  
(angle-preserving)

**equiareal**  
(area-preserving)

*Recall:*

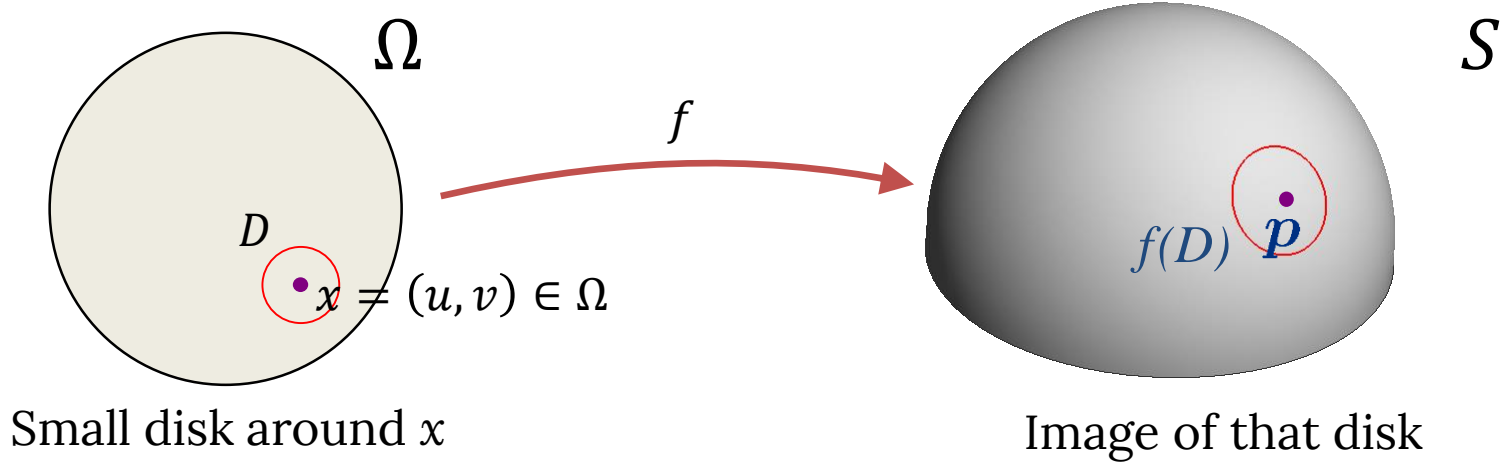
# Intrinsic Descriptor



**Theorema Egregium**  
 (“Totally Awesome  
Theorem”):  
Gaussian curvature is  
**intrinsic.**



# What is Distortion?

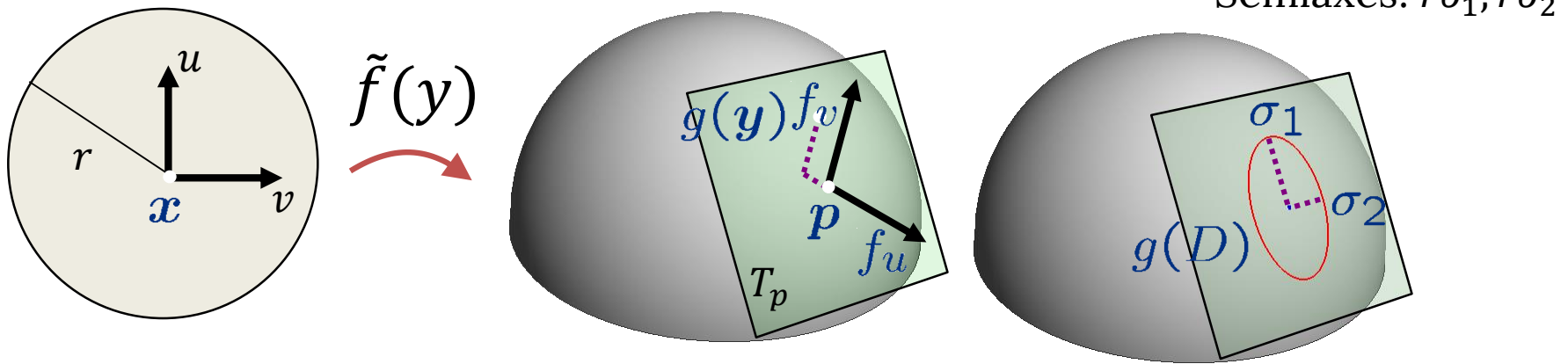


- Distortion (at  $x$ ): How different is  $f(D)$  from  $D$ 
  - How to measure?

How to quantify distortion?  
 Study **differential**/**Jacobian**.

$$f(y) = f(x) + \underbrace{\left( \frac{\partial f}{\partial x_i} \right) (y - x)}_{\text{linearization } \tilde{f}(y)} + O(\|y - x\|^2)$$

linearization  $\tilde{f}(y)$

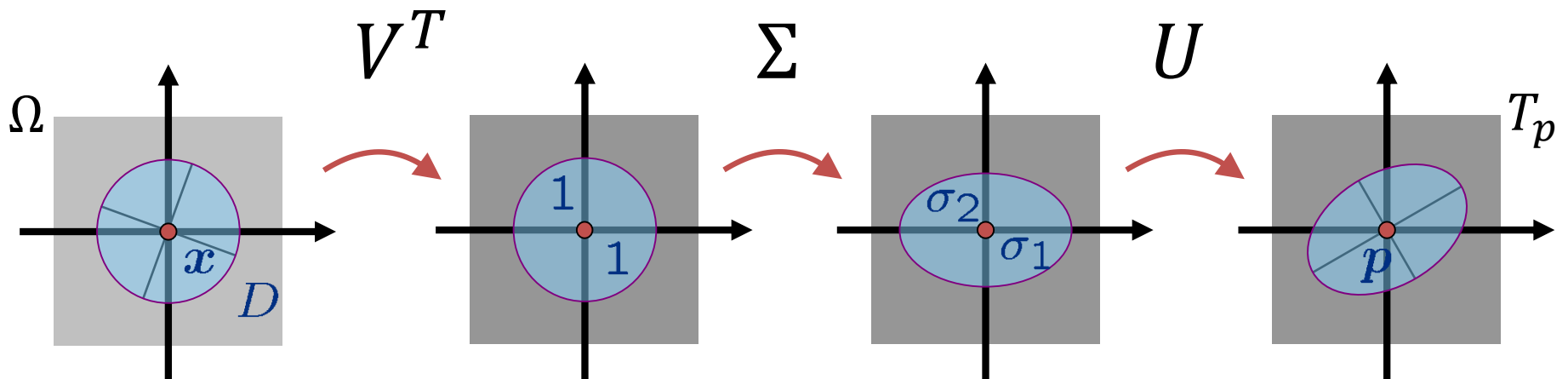


# Linear Map Surgery

- **Singular Value Decomposition** (SVD) of  $J_f$

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

with rotations  $U \in \mathbb{R}^{3 \times 3}$ ,  $V \in \mathbb{R}^{2 \times 2}$   
and scale factors (singular values)  $\sigma_1, \sigma_2 \geq 0$



How to quantify distortion?  
Study **Metric Tensor = 1<sup>st</sup> fundamental form.**

$$I = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_u \cdot f_v & f_v \cdot f_v \end{pmatrix}$$

How to quantify distortion?  
Study **Metric Tensor = 1<sup>st</sup> fundamental form.**

$$I = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_u \cdot f_v & f_v \cdot f_v \end{pmatrix} \stackrel{\text{"Gram Matrix"}}{=} J_f J_f^T$$

How to quantify distortion?  
Study **Metric Tensor = 1<sup>st</sup> fundamental form.**

$$I = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_u \cdot f_v & f_v \cdot f_v \end{pmatrix} = J_f^T J_f$$

$$= (U \Sigma V^T)^T (U \Sigma V^T) = V \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} V^T$$

Eigenvalues of  $I$

# Pointwise distortion

**Isometric**

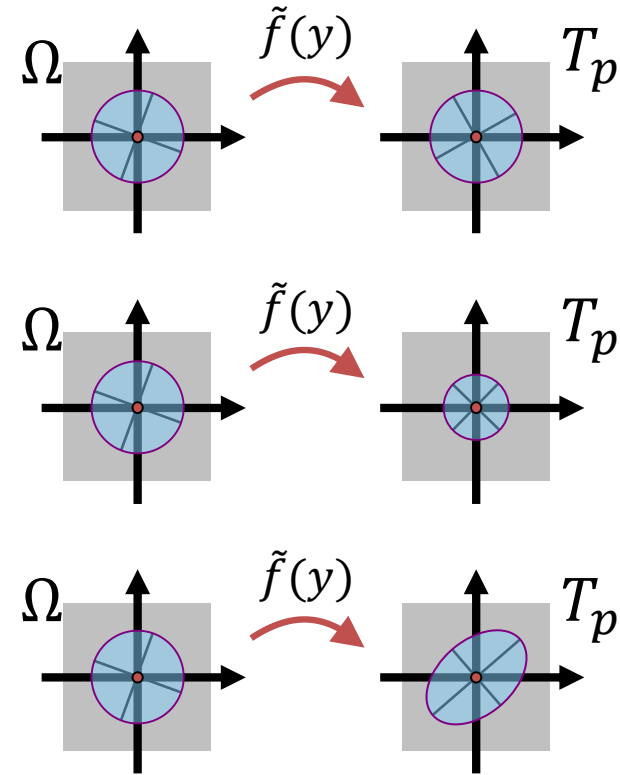
$$\sigma_1 = \sigma_2 = 1$$

**Conformal** or **angle-preserving**

$$\sigma_1 = \sigma_2$$

**Equiareal** or **area-preserving**

$$\sigma_1 \cdot \sigma_2 = 1$$



Defined pointwise on  $\Omega$

# Measuring Distortion

- **Local** distortion measure function of  $\sigma_1$  and  $\sigma_2$

$$E: (\mathbb{R}_+ \times \mathbb{R}_+) \rightarrow \mathbb{R}, \quad (\sigma_1, \sigma_2) \mapsto E(\sigma_1, \sigma_2)$$

- **Overall** distortion

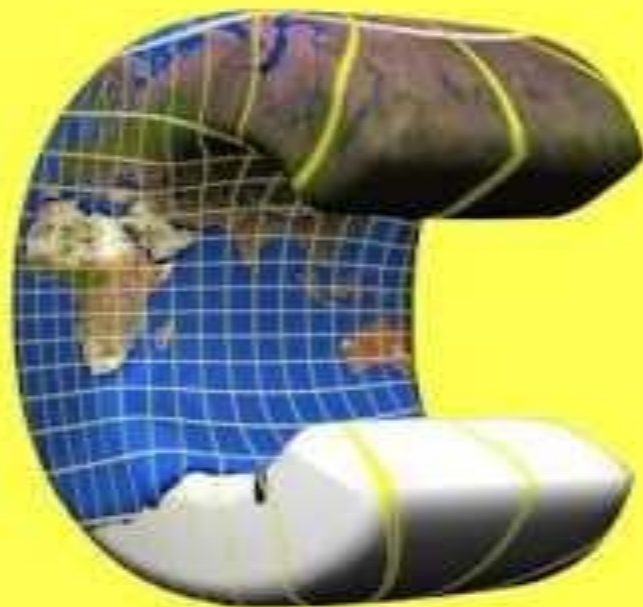
$$E(f) = \int_{\Omega} E(\sigma_1(u, v), \sigma_2(u, v)) \, du \, dv / \text{Area}(\Omega)$$

- On mesh constant per triangle

$$E(f) = \sum_{t \in \Omega} E(t) A(t) / \sum_{t \in \Omega} A(t)$$



# CONFORMAL MAPS



# Stereographic Projection

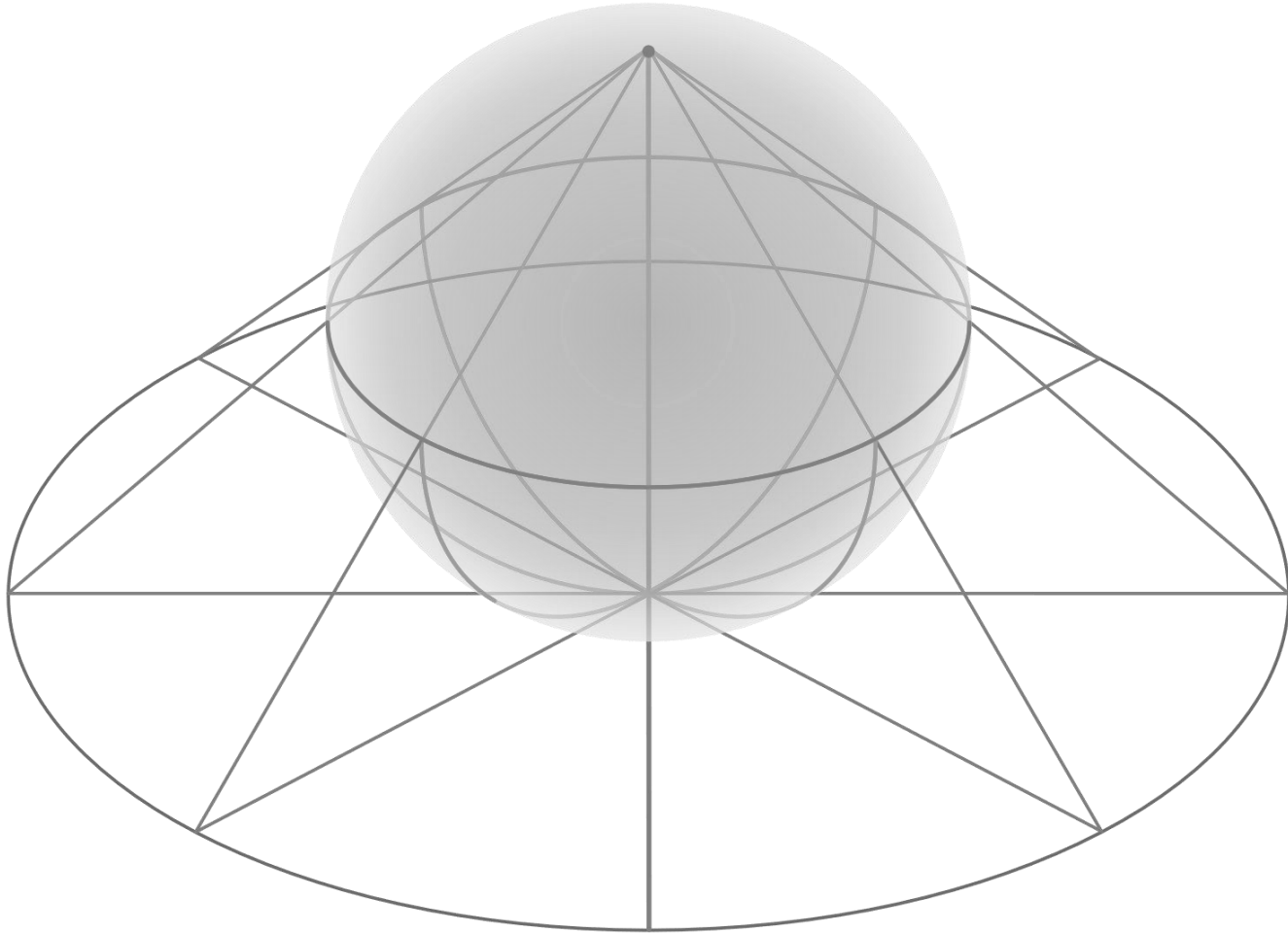
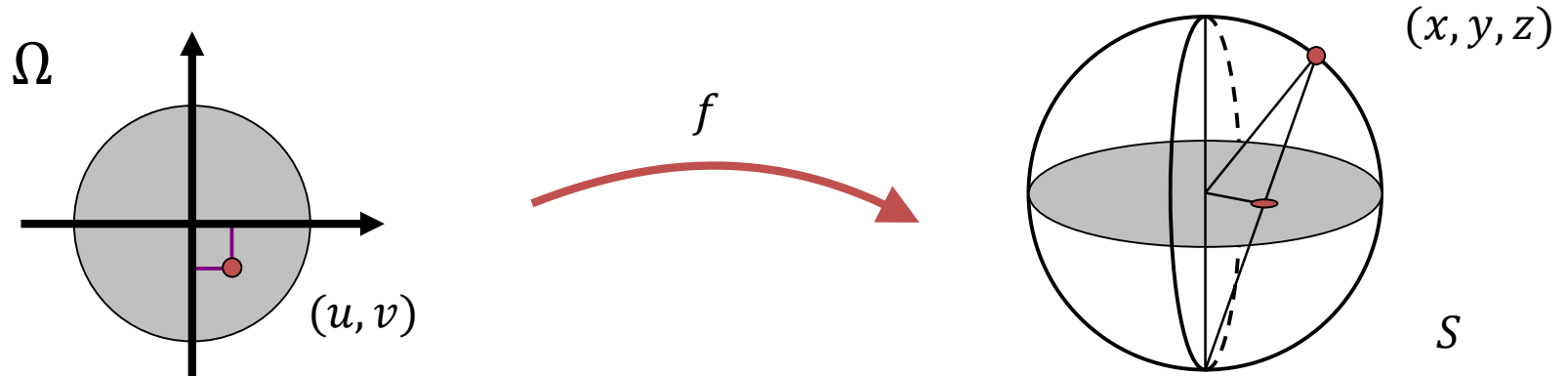


Image from Wikipedia

# Stereographic Projection



parameterization:  $f(u, v) = (2ud, 2vd, (1 - u^2 - v^2)d)$  with  $d = \frac{1}{1+u^2+v^2}$

Jacobian:  $J_f = \begin{pmatrix} 2d-4u^2d^2 & -4uvd^2 \\ -4uvd^2 & 2d-4v^2d^2 \\ -4ud^2 & -4vd^2 \end{pmatrix}$

first fundamental form:  $\mathbf{I}_f = \begin{pmatrix} 4d^2 & 0 \\ 0 & 4d^2 \end{pmatrix}$

eigenvalues:  $\lambda_1 = 4d^2, \quad \lambda_2 = 4d^2$

$\sigma_1 = \sigma_2 = 2d \quad \Rightarrow \quad \text{conformal}$

# Stereographic Projection

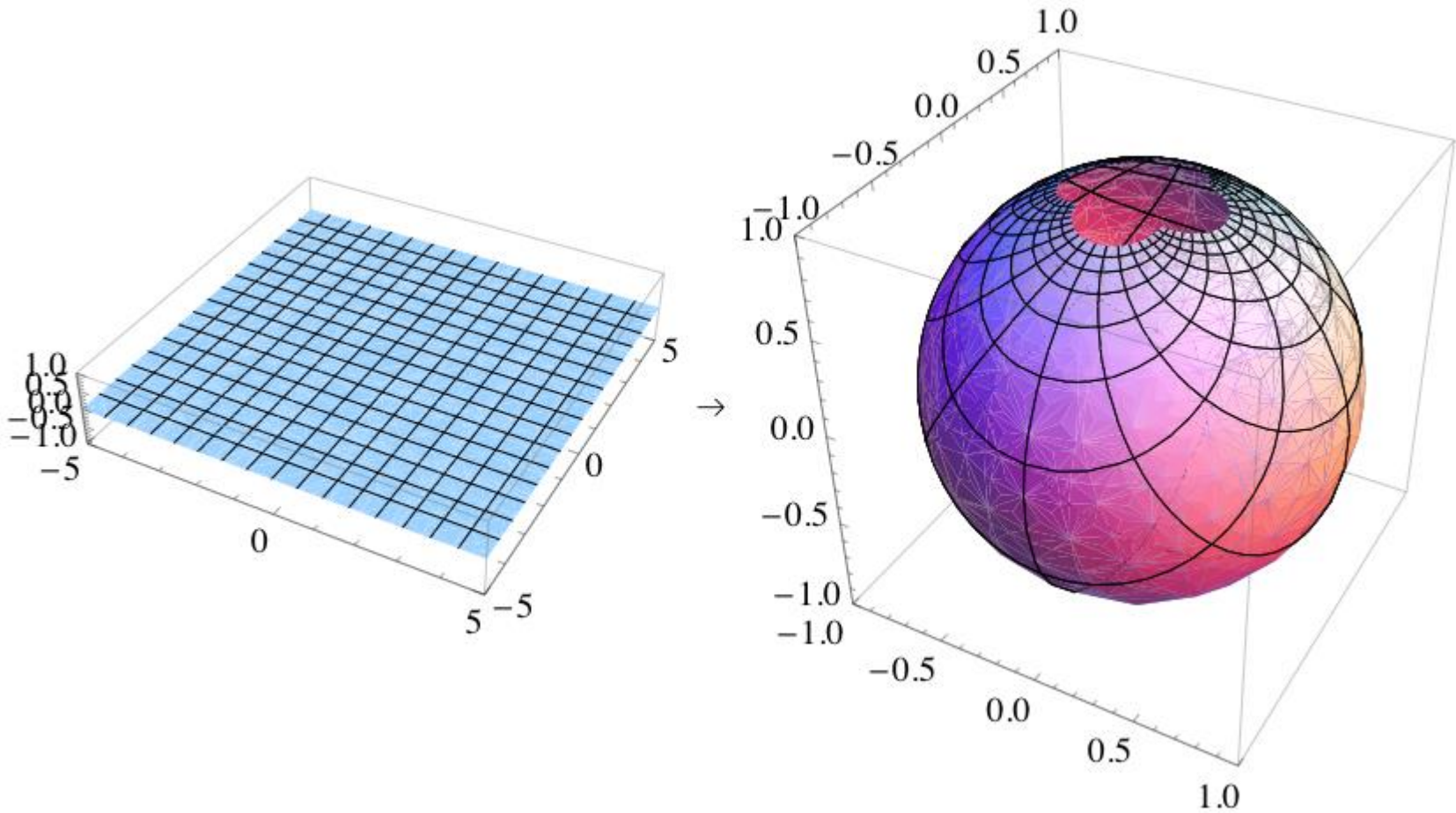


Image from Wikipedia

# Stereographic Projection



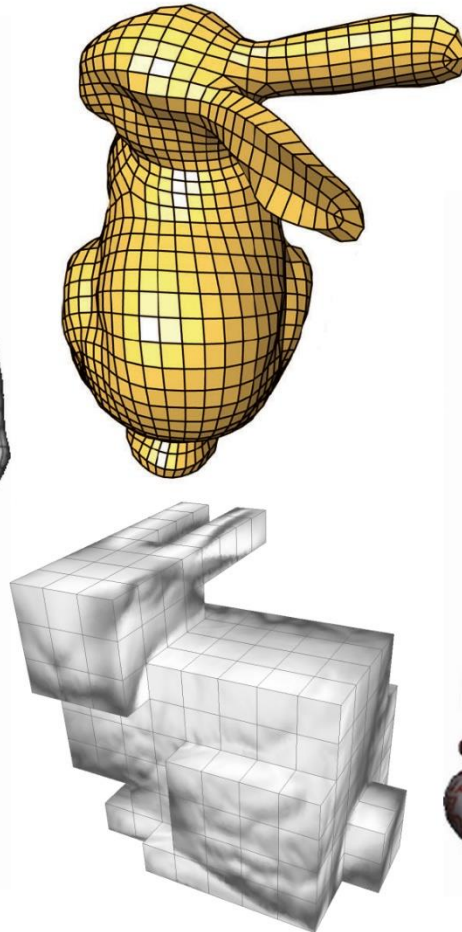
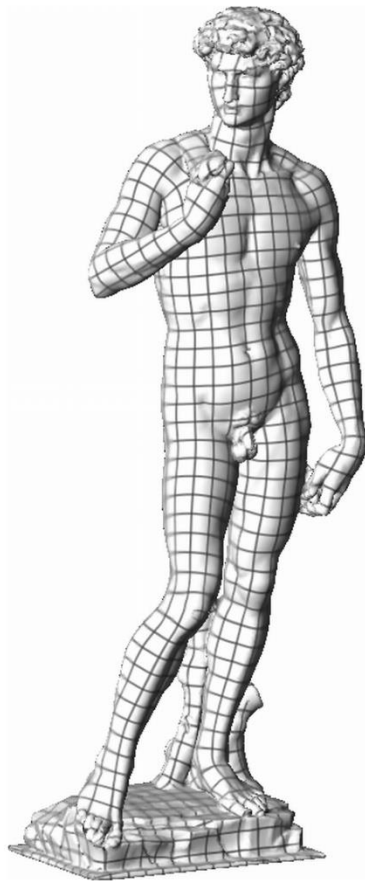
# Does conformal map exist?

## **Uniformization theorem**

“Every simply connected Riemann surface is conformally equivalent to one of three Riemann surfaces:

1. open unit disk
2. complex plane
3. Riemann sphere”

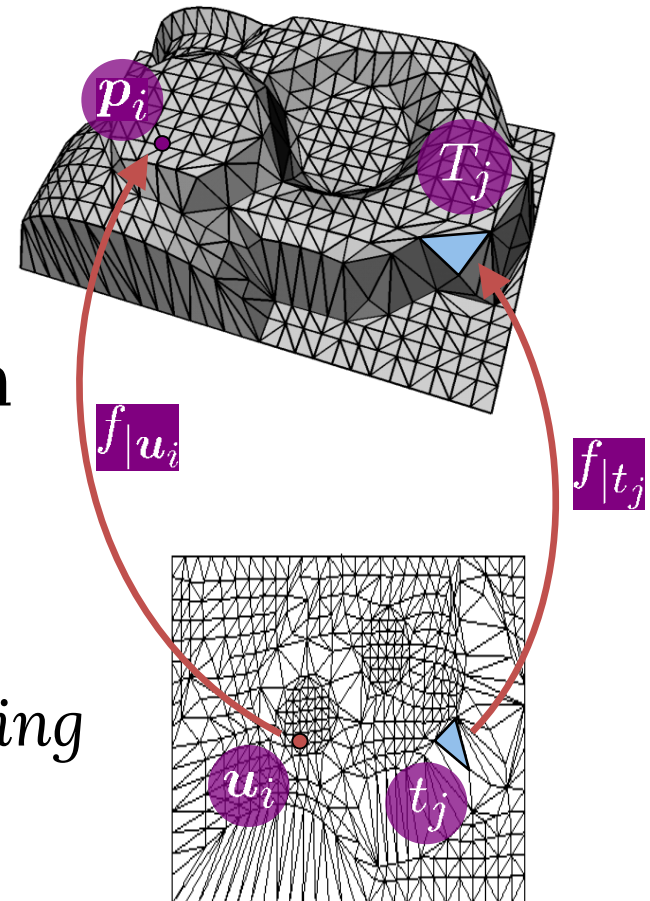
# Parameterization: Practice





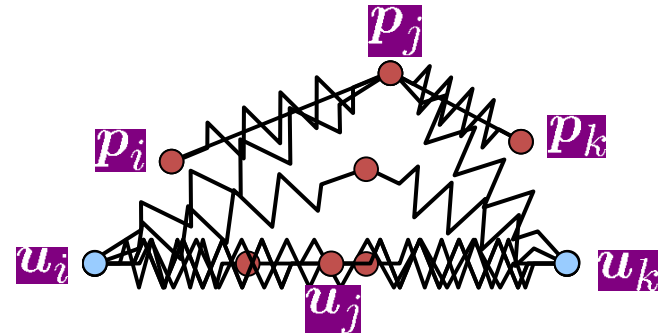
# Mesh Parameterization Methods

- Fixed Boundary
  - 2D “convex” embeddings
    - Variety of weights
- Free boundary –
  - Direct energy minimization
    - Example: Least Squares Conformal Map (LSCM)....
  - Indirect
    - Example: Angle Based Flattening (ABF)....



# Spring Model

- Edges  $\rightarrow$  springs



- Fix boundary vertices on a *convex* polygon
- Let go of the springs
  - “Relaxation”

# Spring Model

Hooke's Law: Energy of spring between  $p_i$   
and  $p_j$ :  $0.5 D_{ij} \|u_i - u_j\|_2^2$

**Total energy**

$$E = \sum_{(i,j) \in E} \frac{1}{2} D_{ij} \|u_i - u_j\|_2^2$$

# Spring Model

Hooke's Law: Energy of spring between  $p_i$   
and  $p_j$ :  $0.5 D_{ij} \|u_i - u_j\|_2^2$

Will this provide a  
**bijection**  
parameterization?

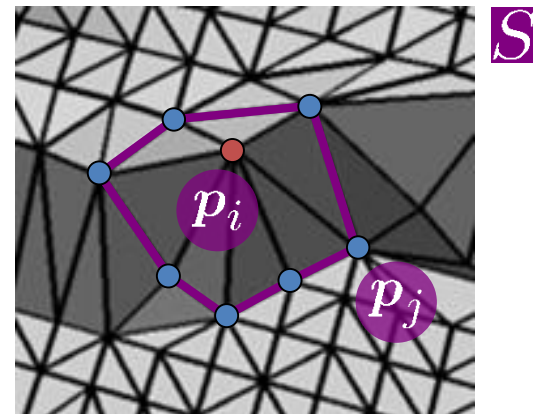
$$E = \sum_{(i,j) \in E} \frac{1}{2} D_{ij} \|u_i - u_j\|_2^2$$

- Rewrite

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i} \frac{1}{2} D_{ij} \|u_i - u_j\|^2$$

- Gradient

$$\frac{\partial E}{\partial u_i} = \sum_{j \in N_i} D_{ij} (u_i - u_j)$$



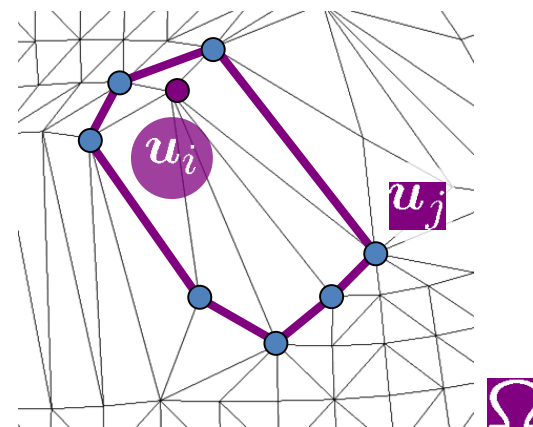
- Gradient = 0, when

Convex Combination

$$u_i = \sum_{j \in N_i} \lambda_{ij} u_j$$

with weights

$$\lambda_{ij} = D_{ij} / \sum_{k \in N_i} D_{ik}$$

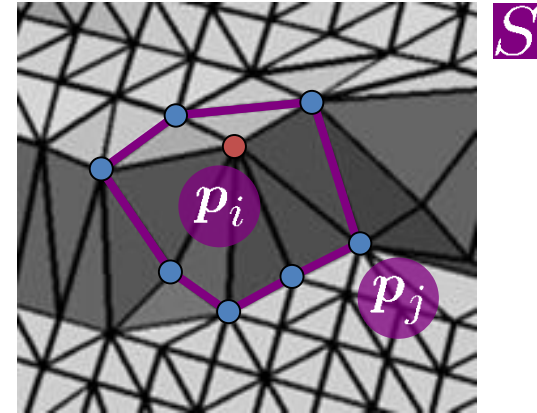


- Rewrite

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{j \in N_i} \frac{1}{2} D_{ij} \|u_i - u_j\|^2$$

- Gradient

$$\frac{\partial E}{\partial u_i} = \sum_{j \in N_i} D_{ij} (u_i - u_j)$$



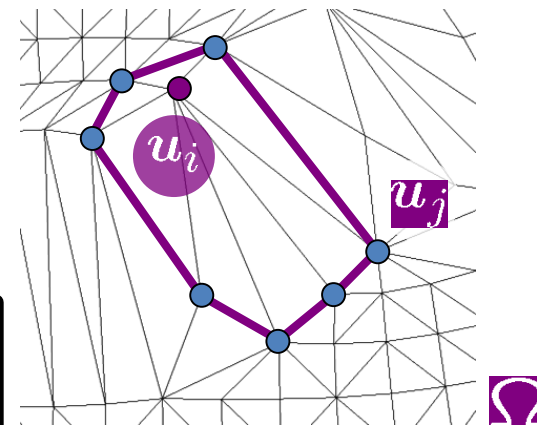
- Gradient = 0, when

Convex Combination

$$u_i = \sum_{j \in N_i} \lambda_{ij} u_j$$

with weights

A generalization of barycentric coordinates



# Linear System

- Separation of variables

$$u_i - \underbrace{\sum_{j \in N_i, j \leq n} \lambda_{ij} u_j}_{\text{unknown parameter points}} = \underbrace{\sum_{j \in N_i, j > n} \lambda_{ij} u_j}_{\text{fixed}} = \bar{u}_i$$

- Linear system

$$AU = \bar{U},$$
$$A = \begin{cases} 1, & i = j \\ -\lambda_{ij}, & j \in N_i \\ 0, & \text{else} \end{cases}$$

$$\lambda_{ij} = D_{ij} / \sum_{k \in N_i} D_{ik}$$

- Solve separately for  $u$  and  $v$

**Theorem** [Tutte'63,Floater'01,Maxwel'1864]:

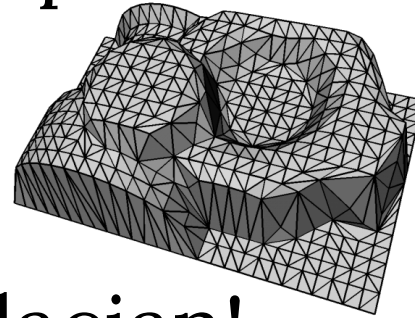
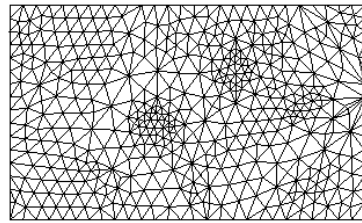
If  $G$  is a 3-connected planar graph  
(*triangular mesh*) then any convex  
combination embedding ( $\lambda_{ij} > 0$ ) provides  
bijective parameterization



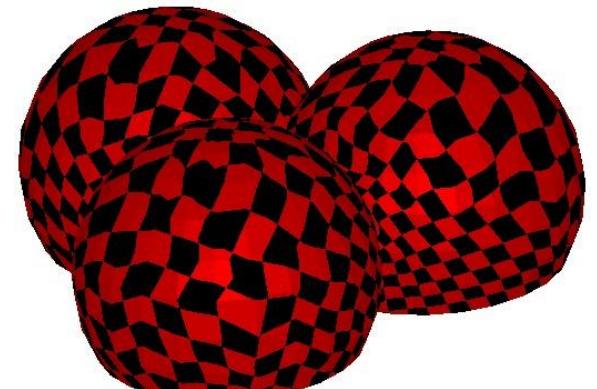
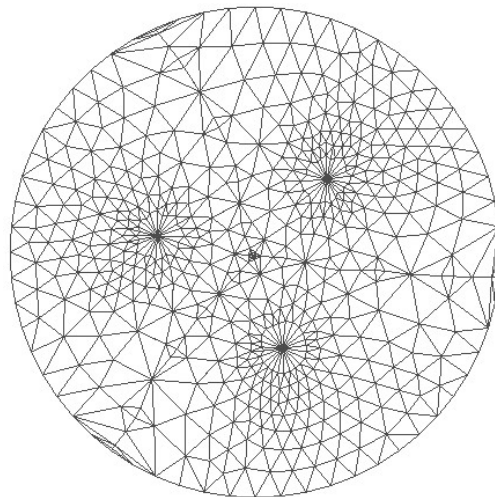
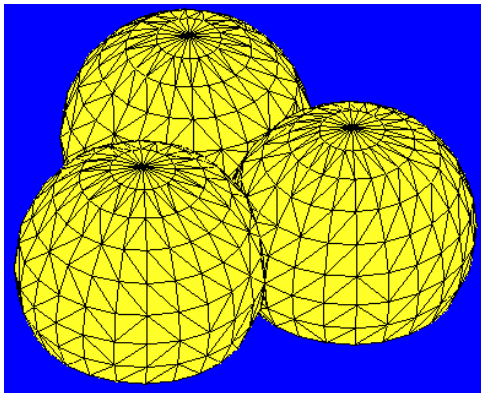
# Choice of Weights: Uniform (Tutte)

$$D_{ij} = 1 \qquad \lambda_{ij} = \frac{1}{\#N_i}$$

No shape preservation – equilateral triangles



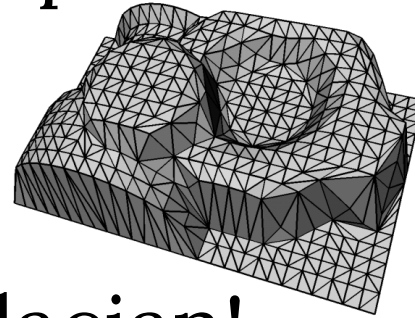
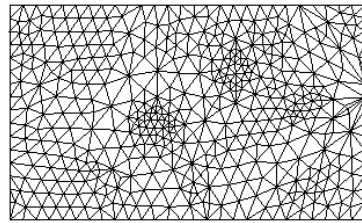
Graph Laplacian!



# Choice of Weights: Uniform (Tutte)

$$D_{ij} = 1 \quad \lambda_{ij} = \frac{1}{\#N_i}$$

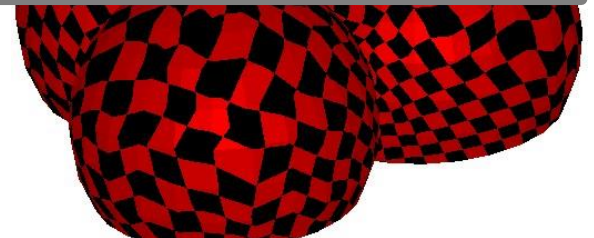
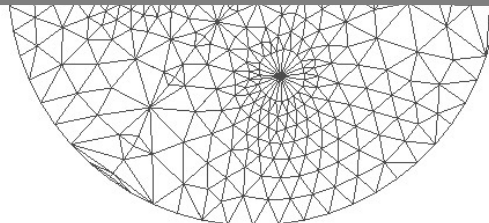
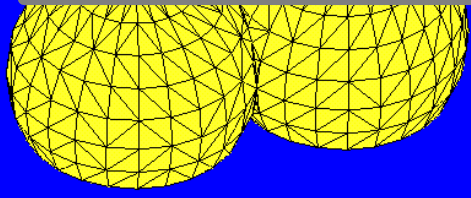
No shape preservation - equilateral triangles



Graph Laplacian!

$$E = \sum_{(i,j) \in E} \frac{1}{2} D_{ij} \|u_i - u_j\|^2 \text{ is}$$

Dirichlet energy discretized on a graph!



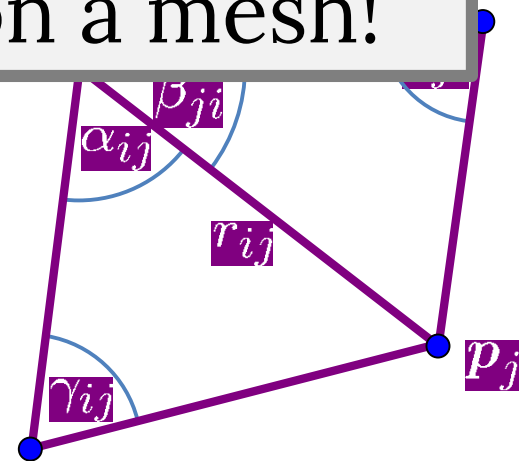
# Choice of Weights: Barycentric Harmonic/Conformal/FEM Laplacian

$$\omega_{ij} = \cot\gamma_{ij} + \cot\gamma_{ji}$$

$$\lambda_{ij} = \omega_{ij} / \sum_k \omega_{ik}$$

$$E = \sum_{(i,j) \in E} \frac{1}{2} D_{ij} \|u_i - u_j\|^2 \text{ is}$$

Dirichlet energy discretized on a mesh!



# Issue

Point is inside, but the coordinate can be  $<0$

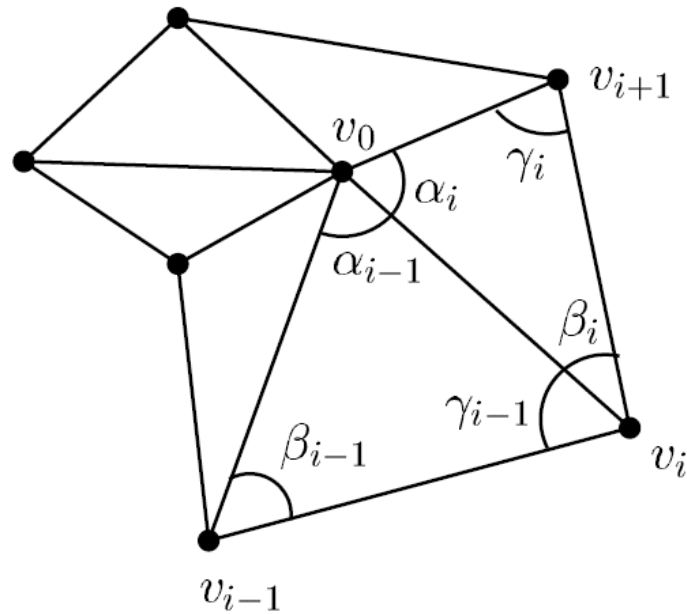


Figure 1. Star-shaped polygon.

# Issue

Point is inside, but the coordinate can be  $< 0$   
 $\Rightarrow \exists \lambda_{ij} < 0$

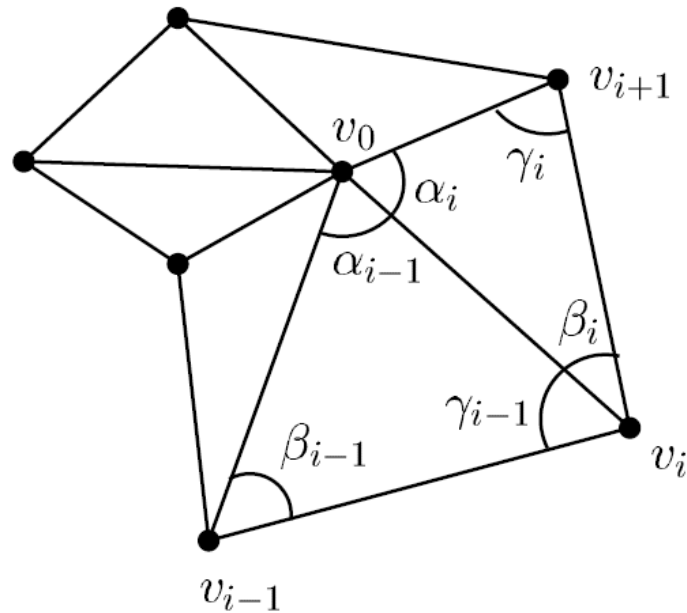


Figure 1. Star-shaped polygon.

# Issue

Point is inside, but the coordinate can be  $< 0$

$$\Rightarrow \exists \lambda_{ij} < 0$$

Local non-bijection

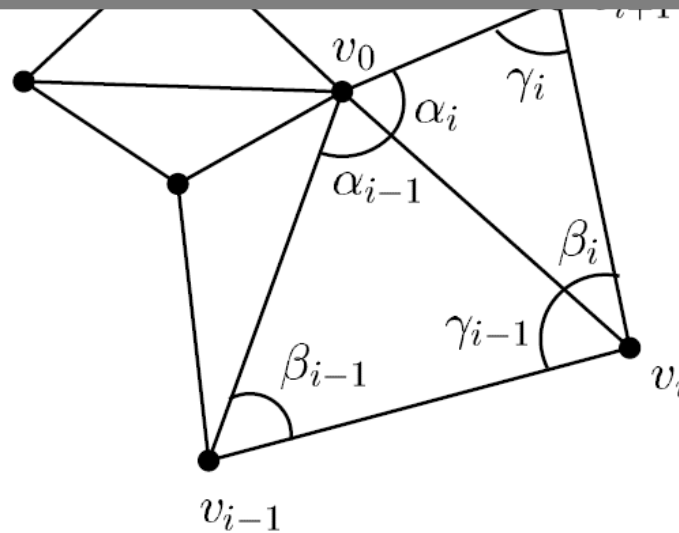
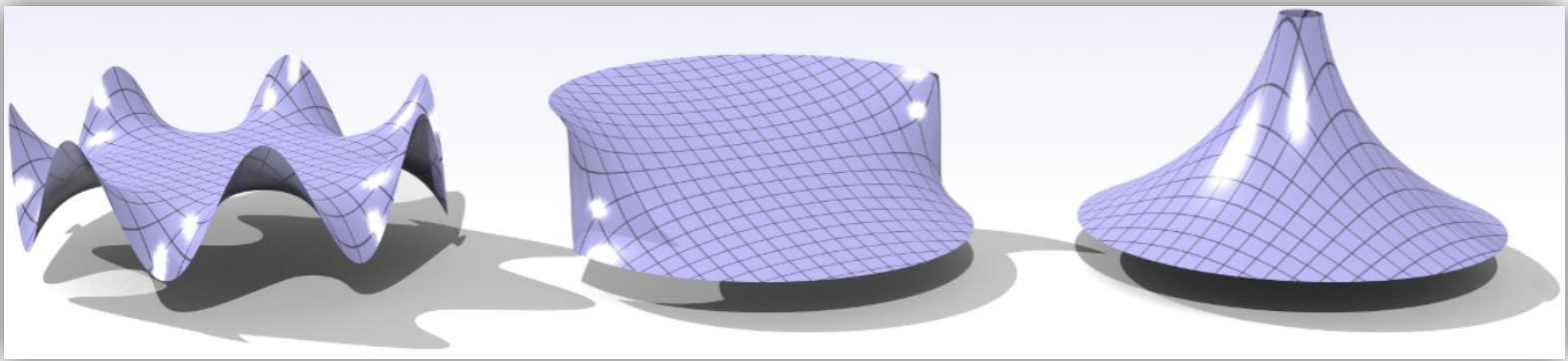


Figure 1. Star-shaped polygon.

*Recall:*

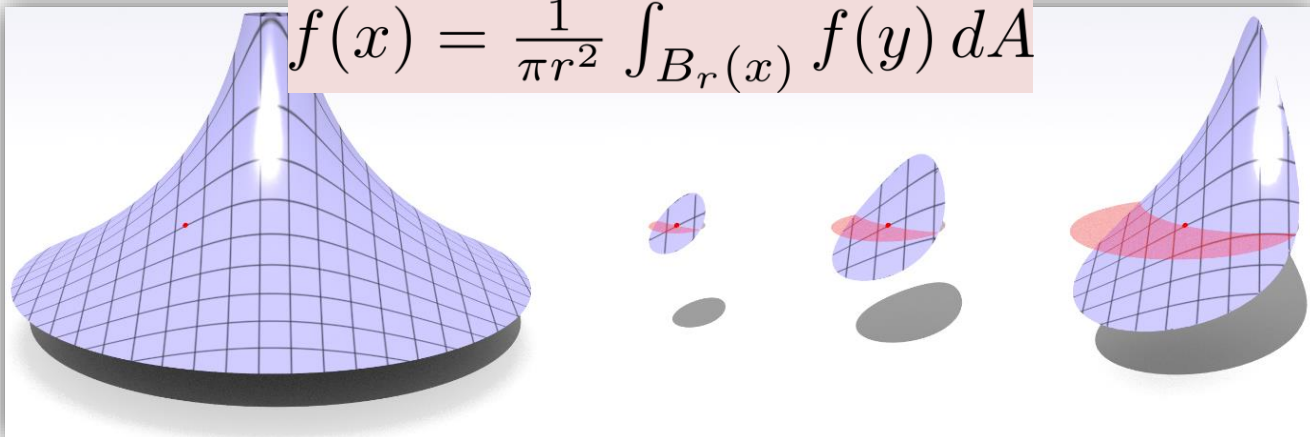
# Harmonic Functions

$$\Delta f \equiv 0$$



Mean value property:

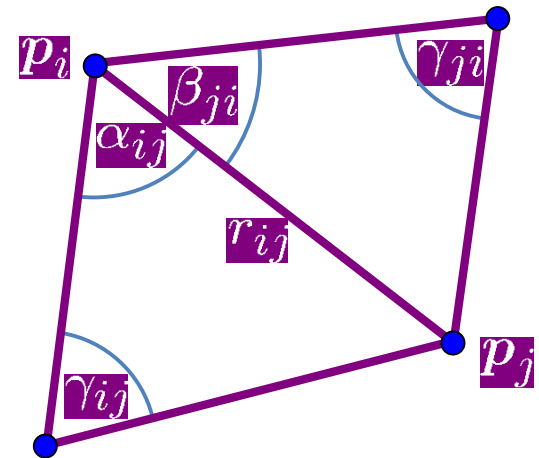
$$f(x) = \frac{1}{\pi r^2} \int_{B_r(x)} f(y) dA$$



# Choice of Weights: Mean Value

$$\omega_{ij} = \frac{\tan \frac{\alpha_{ij}}{2} + \tan \frac{\beta_{ji}}{2}}{r_{ij}}$$

$$\lambda_{ij} = \frac{\omega_{ij}}{\sum_{k \in N_i} \omega_{ik}}$$

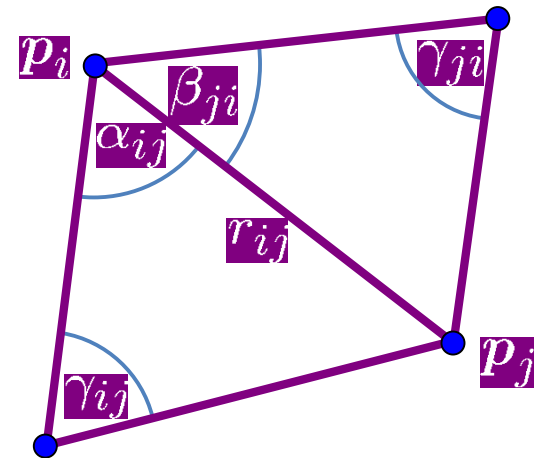




# Choice of Weights: Mean Value

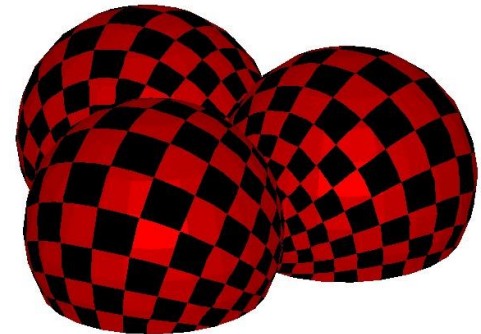
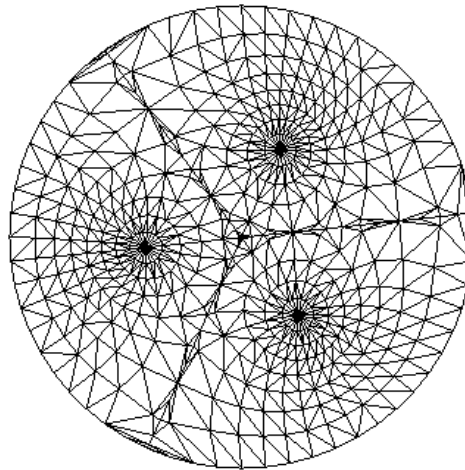
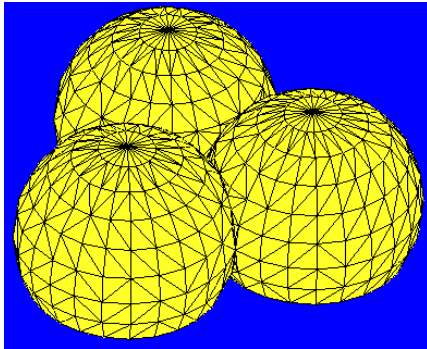
$$\omega_{ij} = \frac{\tan \frac{\alpha_{ij}}{2} + \tan \frac{\beta_{ji}}{2}}{r_{ij}}$$

Always non-negative



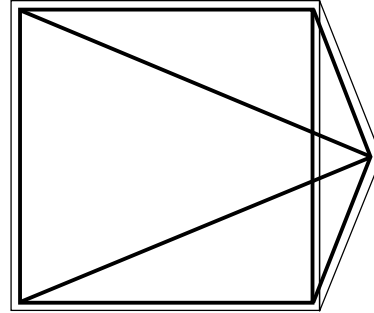
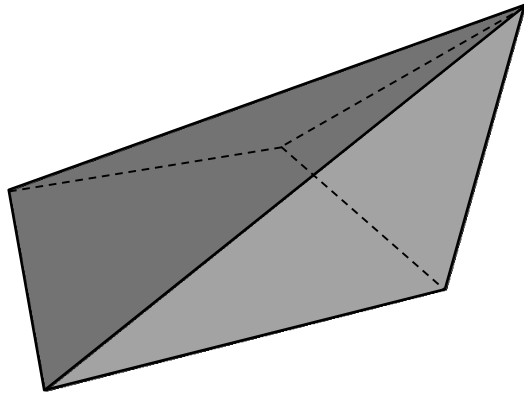
# Harmonic/Mean-Value Mappings

- Quasi-Conformal

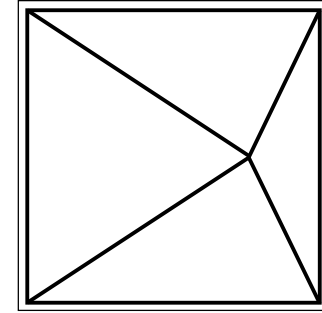


- Linear precision
  - Reproduce planar inputs (same boundary)

# Bijection (fold-overs)



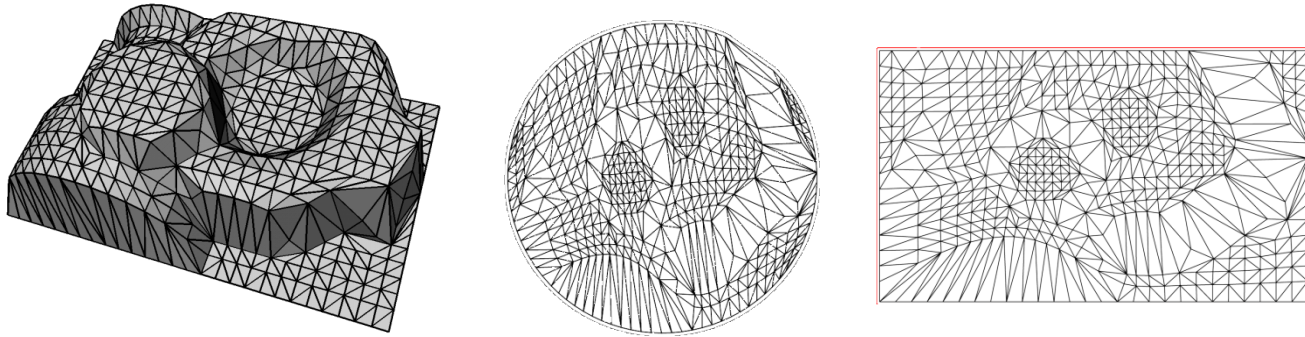
**harmonic**



**mean value**

- Can have fold-overs for negative coordinates
- Mean-value coordinates guaranteed to be positive

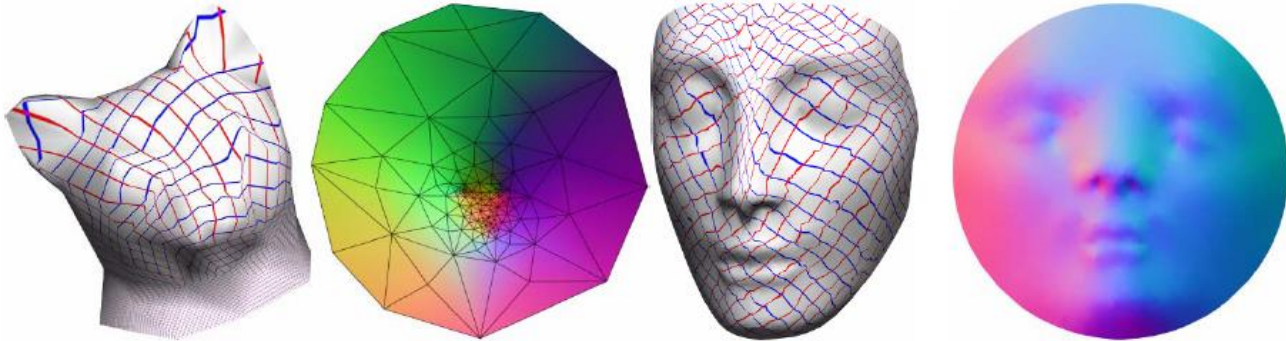
# Boundary Mapping



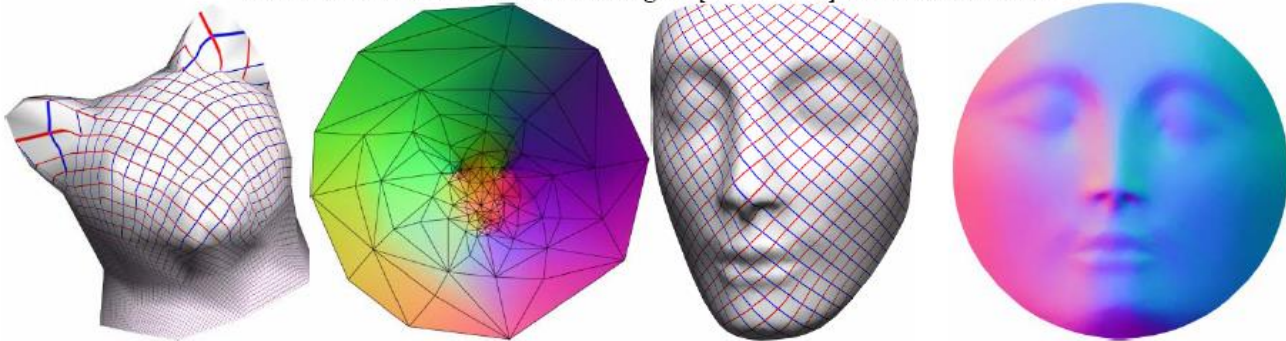
*Chordal* parameterization around *convex* shape

- circle
- rectangle
- triangle
- Choice often application specific
  - Reconstruction – rectangle
  - Mapping to base mesh – triangle

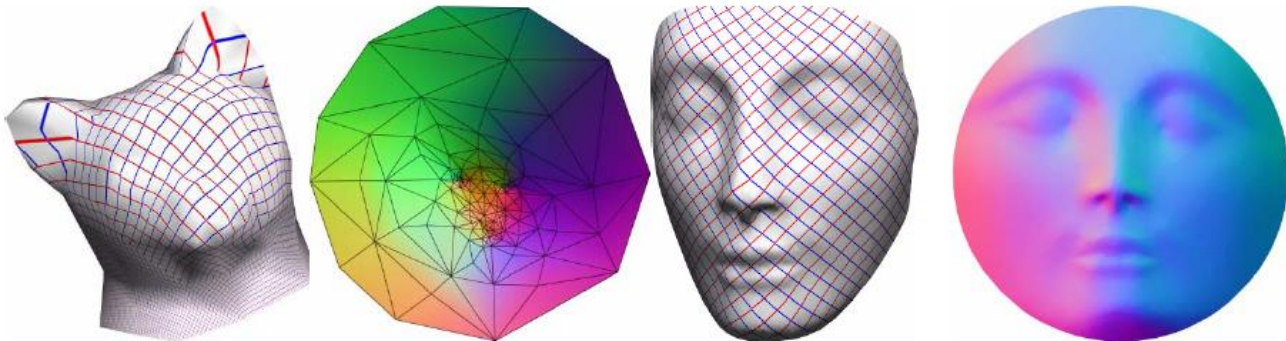
# Examples



Parameterization with uniform weights [Tutte 1963] on a circular domain.

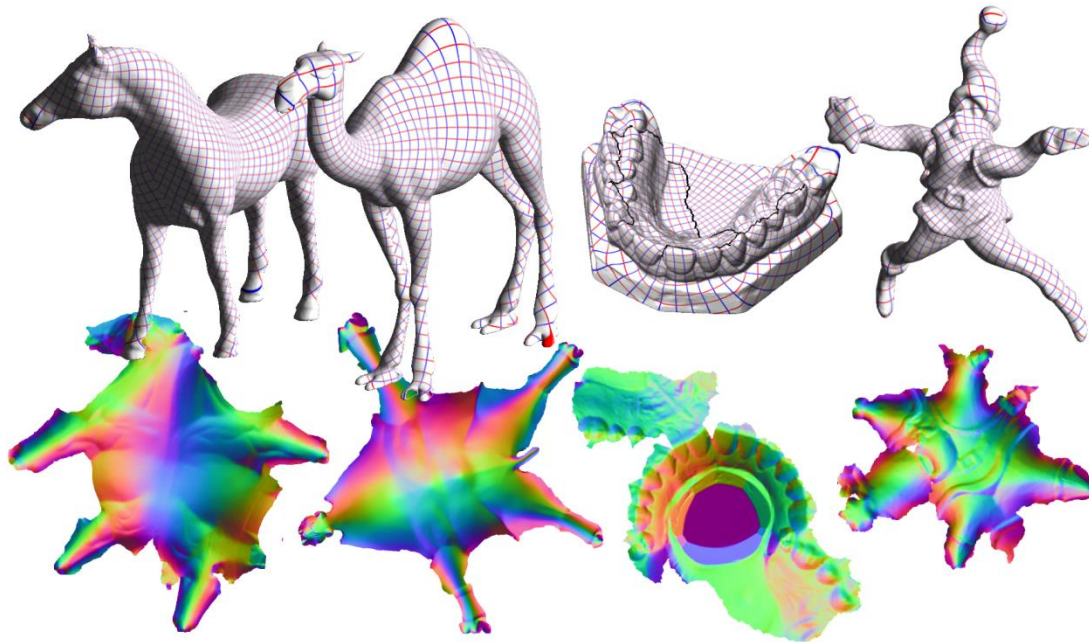


Parameterization with harmonic weights [Eck et al. 1995] on a circular domain.



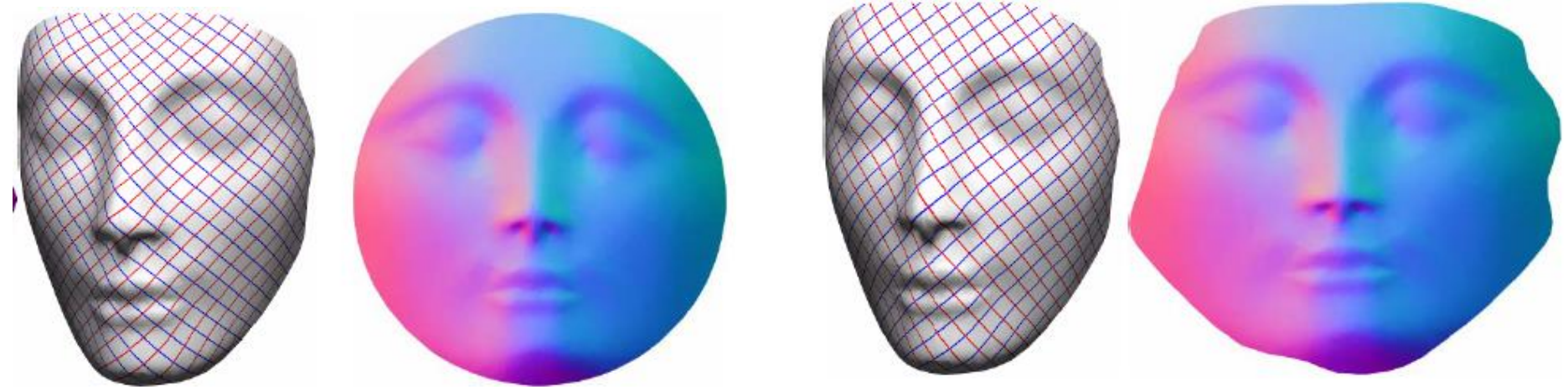
Parameterization with mean value weights [Floater 2003] on a circular domain.

# Parameterization: Free Boundary



# Free Boundary Methods

- Direct energy minimization
  - Example: Least Squares Conformal Map (LSCM)....
- Indirect
  - Example: Angle Based Flattening (ABF)....



rights [Floater 2003] on a circular domain.

LSCM [Lévy et al. 2002].

Free vs Fixed

# LSCM – Geometric Interpretation

We're minimizing conformal energy

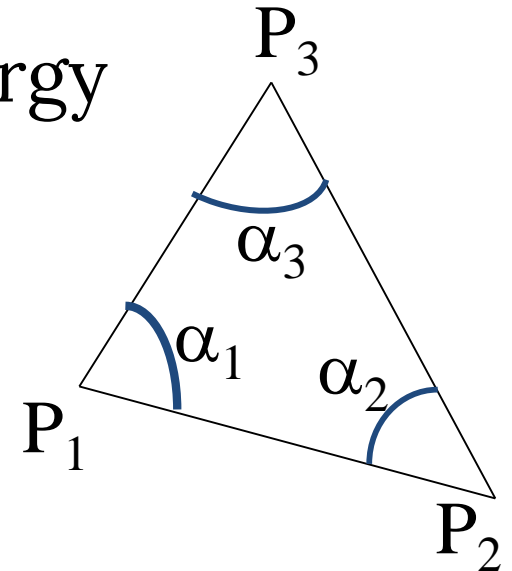
$$E_C = (\sigma_1 - \sigma_2)^2 / 2$$

Geometric Interpretation:

- Use triangle similarity
- Given angles  $\alpha_1, \alpha_2, \alpha_3$  of a triangle  $P_1P_2P_3$  in 2D we have

$$P_3 - P_1 = \frac{\sin \alpha_2}{\sin \alpha_3} R_{\alpha_1} (P_2 - P_1),$$

$$R_\alpha = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix}$$





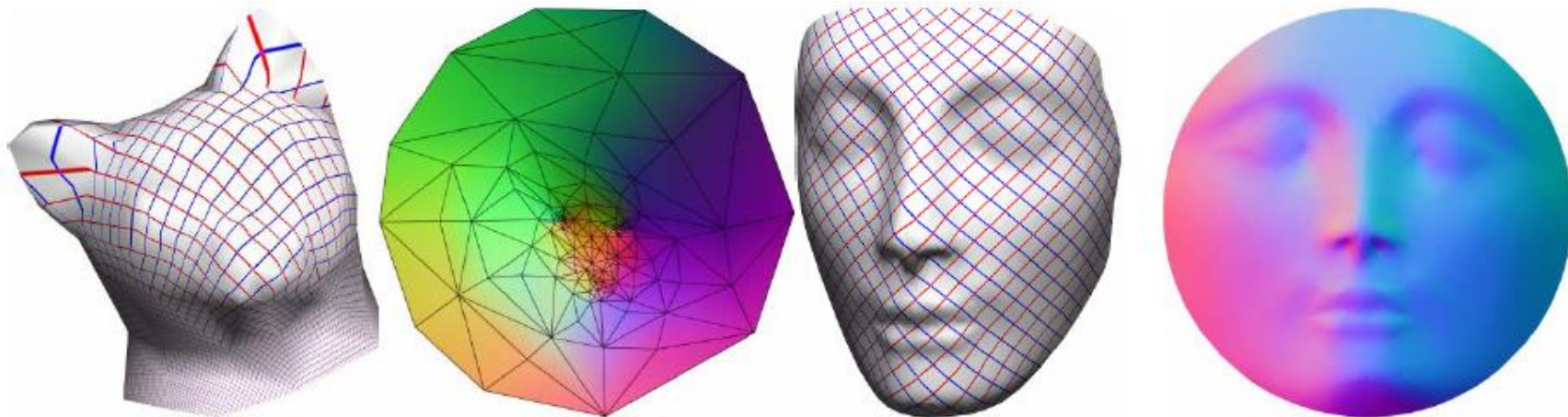
# LSCM

- In map from 3D to 2D might be impossible to keep angles exactly
  - Use least-squares

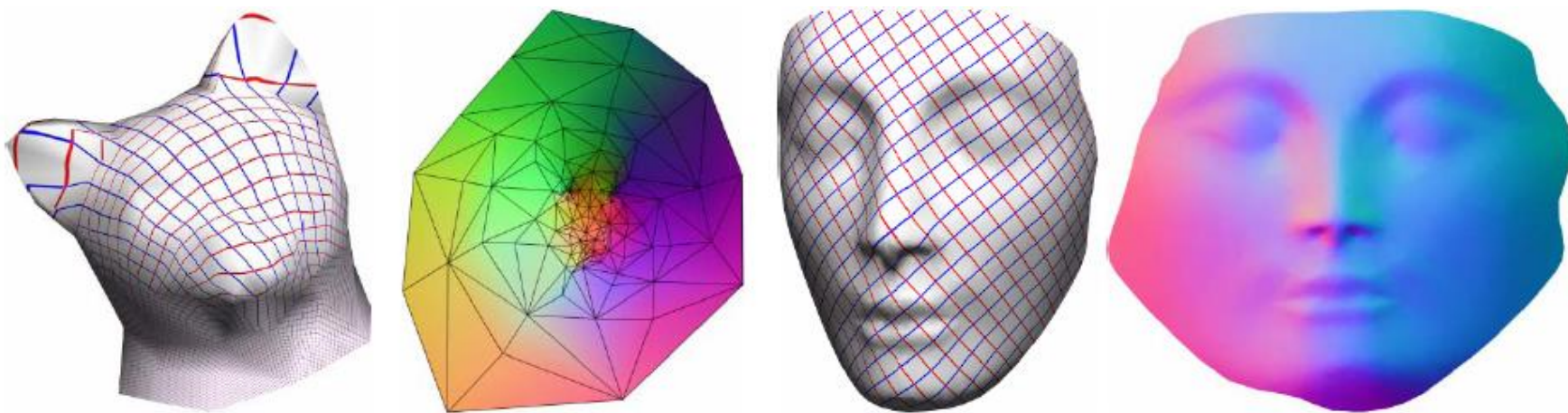
$$\min \sum_i (P_3^i - P_1^i - \frac{\sin \alpha_2^i}{\sin \alpha_3^i} R_{\alpha_1^i} (P_2^i - P_1^i))^2$$

- To solve need to fix two vertices
  - Obtain linear system
  - Choice of vertices affects solution
- Can have flips

# Examples



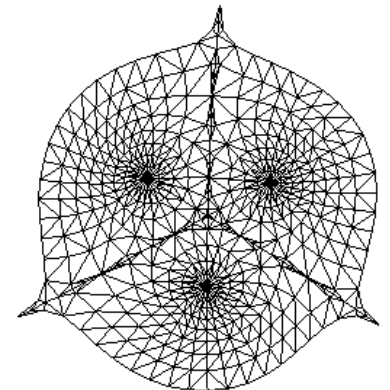
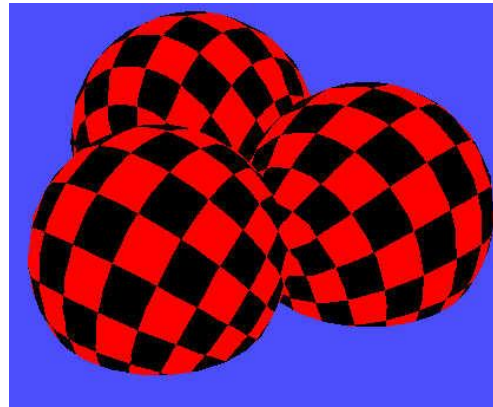
**Parameterization with mean value weights [Floater 2003] on a circular domain.**



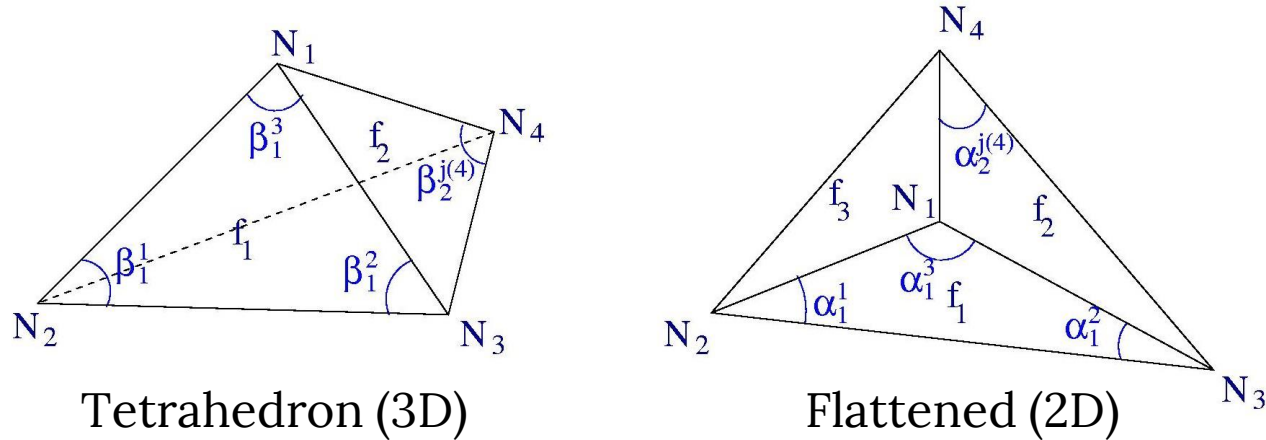
**Parameterization with LSCM [Lévy et al. 2002].**

# ABF: Angle Based Flattening

- Triangular 2D mesh is defined by its angles
- Formulate parameterization as problem in angle space
- Angle based formulation:
  - Distortion as function of angles (conformality)
  - Validity: set of angle constraints
  - Convert solution to UV



# ABF Formulation



- Distortion:
  - 2D/3D angle difference

$$\sum_{t \in T, j=1 \dots 3} w_j^t (\alpha_j^t - \beta_j^t)^2, w_j^t = 1/\beta_j^{t^2}$$

# ABF Formulation

- **Constraints:**

- Triangle validity:

$$\forall t \in T, \quad \alpha_1^t + \alpha_2^t + \alpha_3^t - \pi = 0;$$

- Planarity:

$$\forall v \in V_{int}, \quad \sum_{(t,k) \in v^*} \alpha_k^t - 2\pi = 0$$

- Reconstruction

$$\forall v \in V_{int}, \quad \prod_{(t,k) \in v^*} \sin \alpha_{k \oplus 1}^t - \prod_{(t,k) \in v^*} \sin \alpha_{k \ominus 1}^t = 0$$

- Positivity

$$\alpha_j^t > 0$$

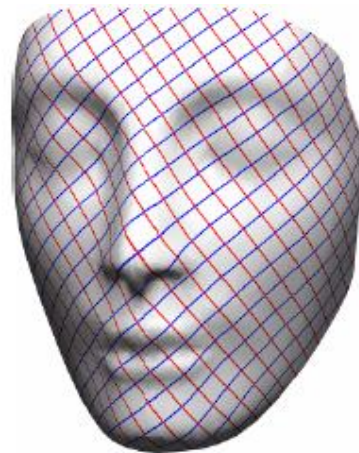
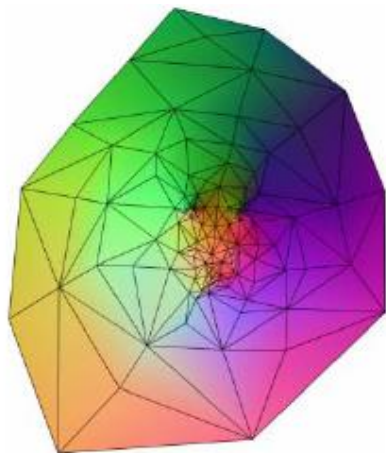
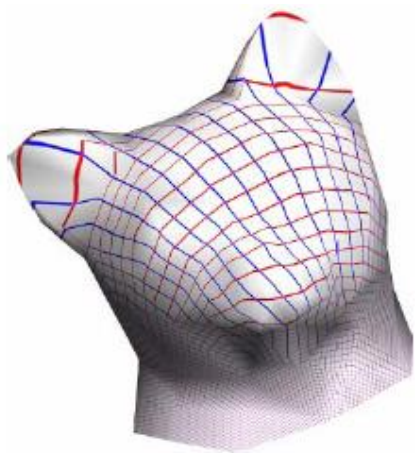
- Distortion:

$$\sum_{t \in T, j=1 \dots 3} w_j^t (\alpha_j^t - \beta_j^t)^2, w_j^t = 1/\beta_j^{t^2}$$

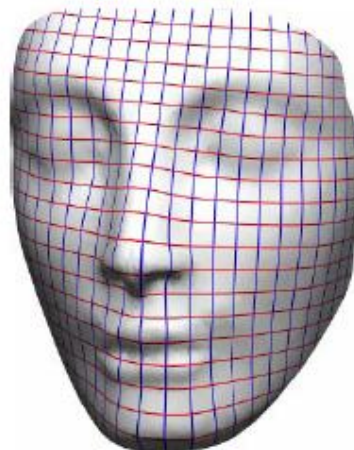
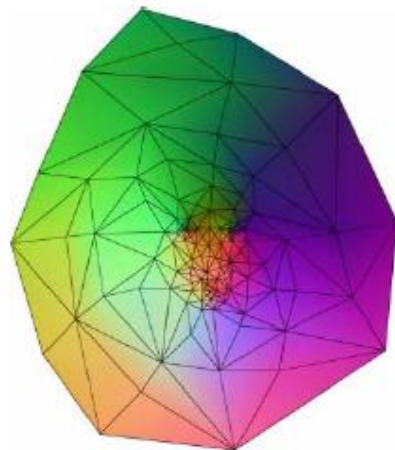
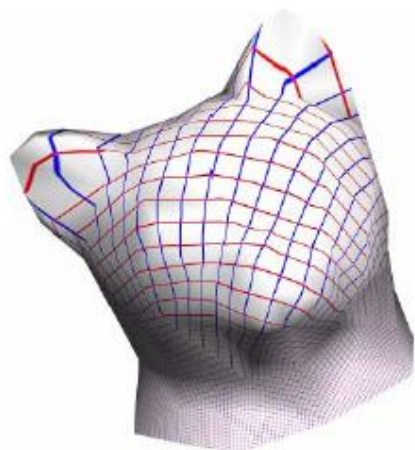
# Angle to UV Conversion

- Use computed angles as input to LSCM (it is a reproducing method..)

# Examples

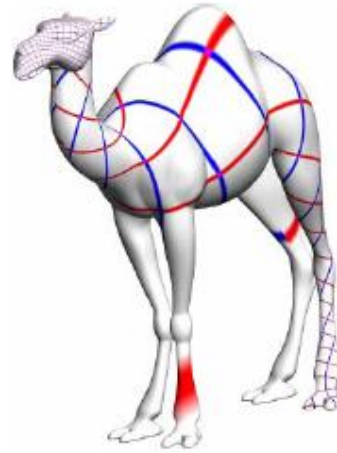
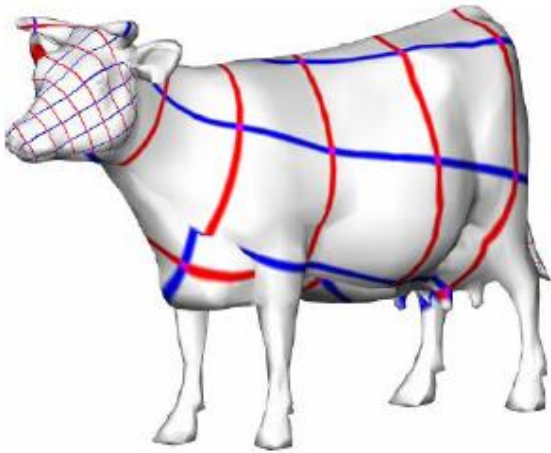


**Parameterization with LSCM [Lévy et al. 2002].**

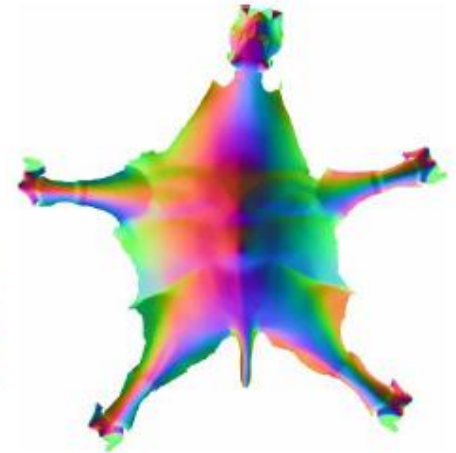
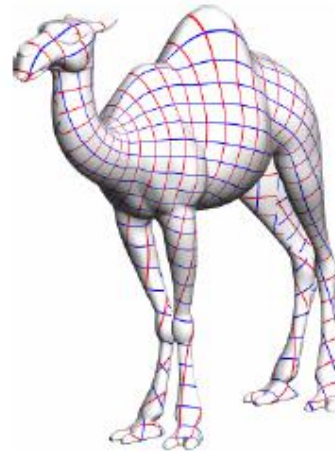
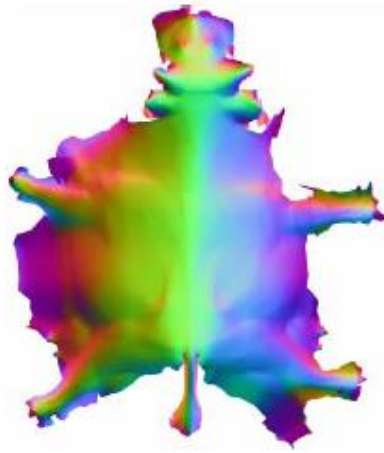
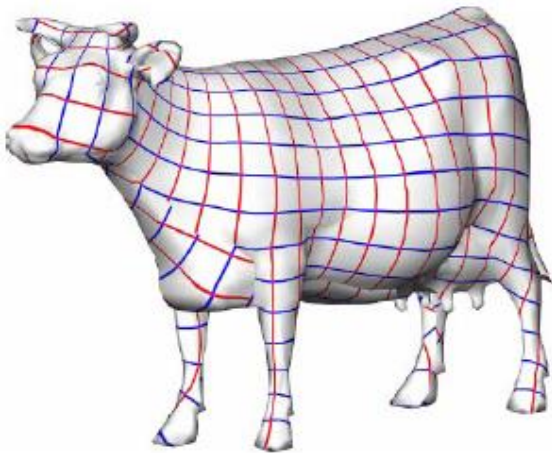


**Parameterization with ABF++ [Sheffer et al. 2005].**

# Examples



Parameterization with LSCM [Lévy et al. 2002].



Parameterization with ABF++ [Sheffer et al. 2005].



# Cone Singularities [Kharevych:06]

- What separates boundary from interior in angle space?
- Answer: Sum of angles at vertex
- Formulation specific

– ABF/ABF++

- Planarity & Reconstruction

$$\forall v \in V_{int}, \quad \sum_{(t,k) \in v^*} \alpha_k^t - 2\pi = 0$$

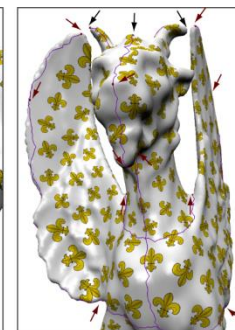
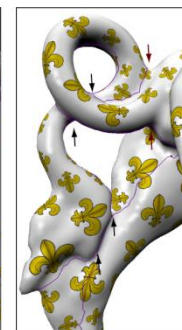
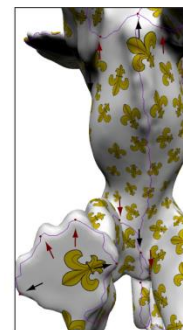
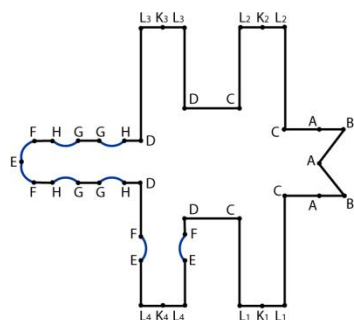
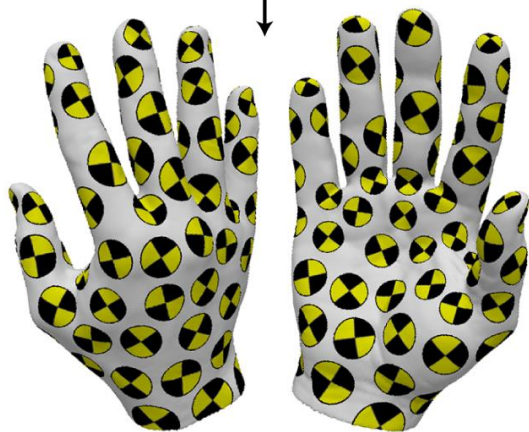
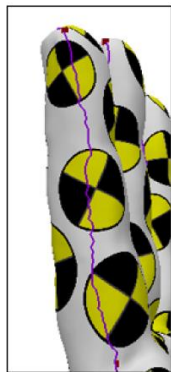
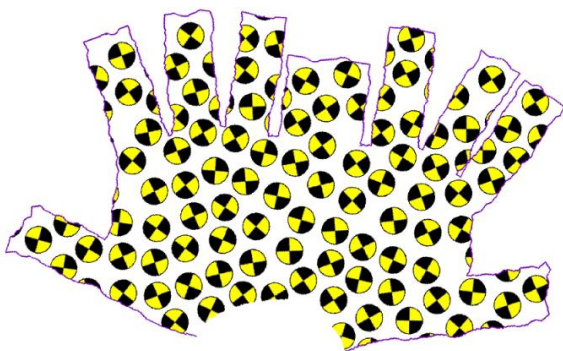
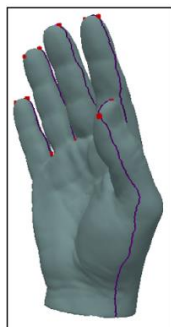
- But... reconstruction can be enforced on boundaries

$$\forall v \in V_{int}, \quad \prod_{(t,k) \in v^*} \sin \alpha_{k \oplus 1}^t - \prod_{(t,k) \in v^*} \sin \alpha_{k \ominus 1}^t = 0$$

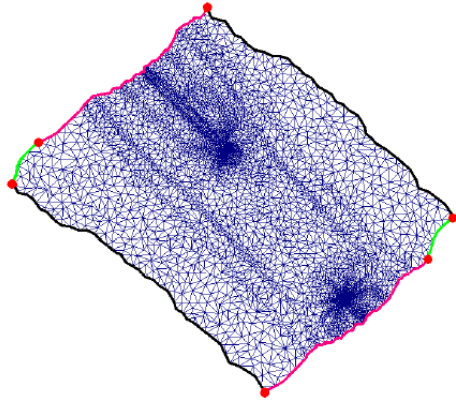
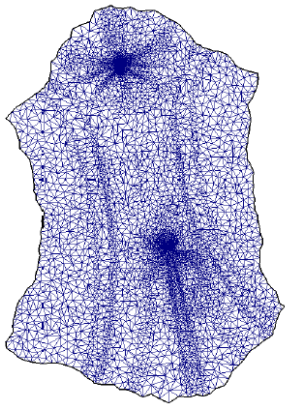
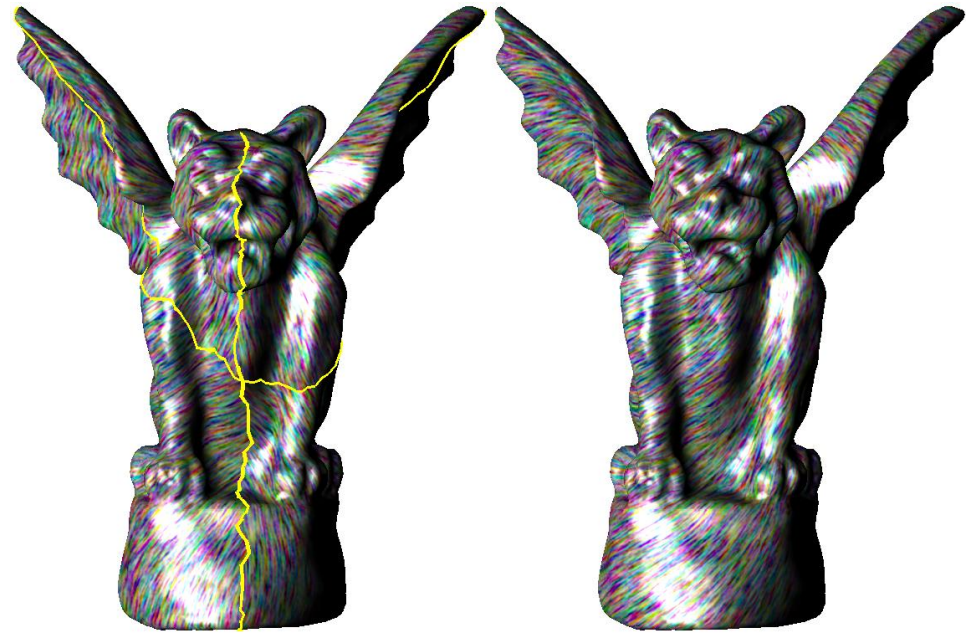
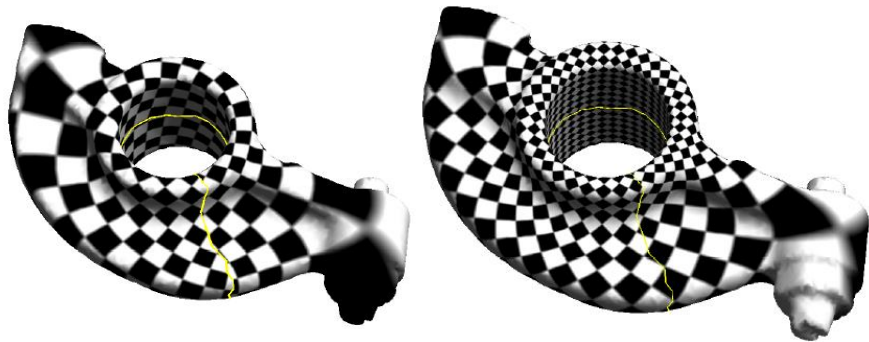
# Cone Singularities

- Idea: Reduce boundary to small set of vertices
- Implementation:
  - Enforce “interior” constraints at all other vertices
- To unfold choose any sequence of edges connecting “boundary” vertices

# Circle Patterns + Cone Singularities



# ABF + Cone Singularities



# General Framework

- Choose an energy

Name	$\mathcal{D}(\mathbf{J})$	$\mathcal{D}(\sigma)$
Symmetric Dirichlet	$\ \mathbf{J}\ _F^2 + \ \mathbf{J}^{-1}\ _F^2$	$\sum_{i=1}^n (\sigma_i^2 + \sigma_i^{-2})$
Exponential Symmetric Dirichlet	$\exp(s(\ \mathbf{J}\ _F^2 + \ \mathbf{J}^{-1}\ _F^2))$	$\exp(s \sum_{i=1}^n (\sigma_i^2 + \sigma_i^{-2}))$
Hencky strain	$\ \log \mathbf{J}^\top \mathbf{J}\ _F^2$	$\sum_{i=1}^n (\log^2 \sigma_i)$
AMIPS	$\exp(s \cdot \frac{1}{2} (\frac{\text{tr}(\mathbf{J}^\top \mathbf{J})}{\det(\mathbf{J})} + \frac{1}{2} (\det(\mathbf{J}) + \det(\mathbf{J}^{-1}))))$	$\exp(s (\frac{1}{2} (\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}) + \frac{1}{4} (\sigma_1 \sigma_2 + \frac{1}{\sigma_1 \sigma_2})))$
Conformal AMIPS 2D	$\frac{\text{tr}(\mathbf{J}^\top \mathbf{J})}{\det(\mathbf{J})}$	$\frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2}$
Conformal AMIPS 3D	$\frac{\text{tr}(\mathbf{J}^\top \mathbf{J})}{\det(\mathbf{J})^{\frac{2}{3}}}$	$\frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2}{(\sigma_1 \sigma_2 \sigma_3)^{\frac{2}{3}}}$

# Conformal Energies

- **Conformal** energy

$$E_C = (\sigma_1 - \sigma_2)^2 / 2$$

[Pinkall & Polthier 1993]

[Lévy et al. 2002]

[Desbrun et al. 2002]

- **MIPS** energy

$$E_M = \kappa_F(J_f) = \|J_f\|_F \|J_f^{-1}\|_F = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}$$

[Hormann & Greiner 2000]

- *Riemann theorem*: any  $C^1$  continuous surface in  $\mathbb{R}^3$  can be mapped conformally to fixed domain in  $\mathbb{R}^2$ 
  - Nearly true for meshes

# Detailed Example

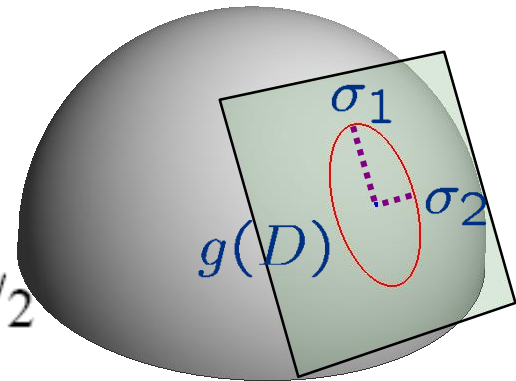
Given a triangle  $T$  with 2D texture coordinates  $p_1, p_2, p_3$ ,  $p_i = (s_i, t_i)$ , and corresponding 3D coordinates  $q_1, q_2, q_3$ , the unique affine mapping  $S(p) = S(s, t) = q$  is

$$S(p) = (\langle p, p_2, p_3 \rangle q_1 + \langle p, p_3, p_1 \rangle q_2 + \langle p, p_1, p_2 \rangle q_3) / \langle p_1, p_2, p_3 \rangle$$

$$S_s = \partial S / \partial s = (q_1(t_2 - t_3) + q_2(t_3 - t_1) + q_3(t_1 - t_2)) / (2A)$$

$$S_t = \partial S / \partial t = (q_1(s_3 - s_2) + q_2(s_1 - s_3) + q_3(s_2 - s_1)) / (2A)$$

$$A = \langle p_1, p_2, p_3 \rangle = ((s_2 - s_1)(t_3 - t_1) - (s_3 - s_1)(t_2 - t_1)) / 2$$



Singular values:  $\sqrt{0.5((a + c) \pm \sqrt{(a - c)^2 + 4b^2})}$

$$a = S_s \cdot S_s, \quad b = S_s \cdot S_t, \quad \text{and} \quad c = S_t \cdot S_t$$

# General Framework

- Choose an energy
- Start with an initial bijective parameterization
  - E.g. Tutte



# General Framework

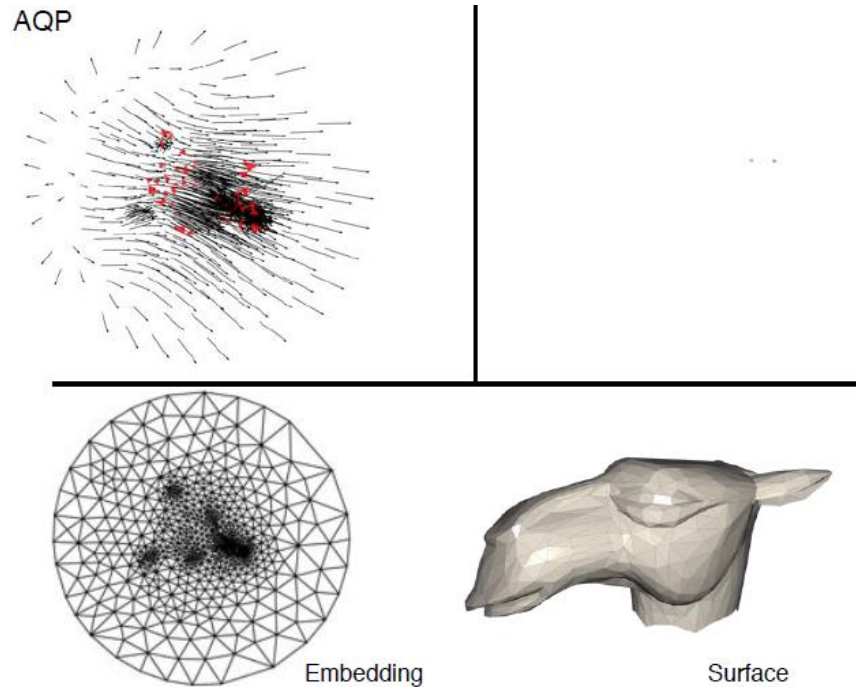
- Choose an energy
- Start with an initial bijective parameterization
  - E.g. Tutte
- Use nonlinear optimization tools to minimize
  - Gradient descent
  - Quasi-Newton methods
  - ...

# General Framework

- Choose an energy
- Start with an initial bijective parameterization
  - E.g. Tutte
- Use nonlinear optimization tools to minimize
  - Gradient descent
  - Quasi-Newton methods
  - ...
- How to preserve **bijectivity**?

# What is gradient of E w/r to positions?

## A vector field!



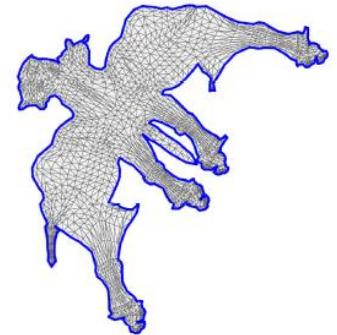
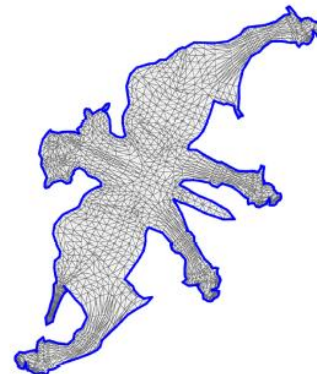
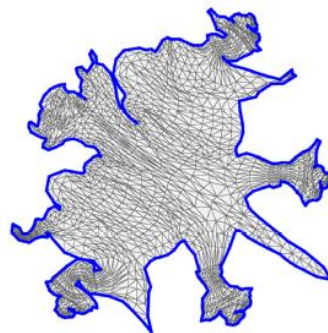
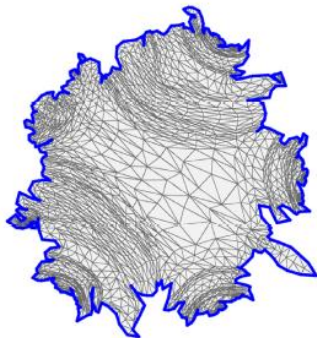
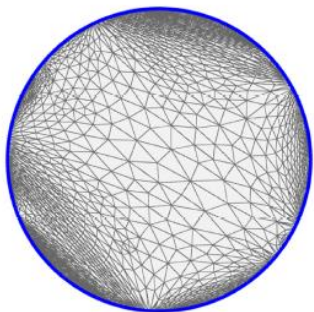
# Scaling vector field

- Find a scale parameter, s.t. nothing flips if we add the vector field
- Limit line search step to this value

## Bijjective Parameterization with Free Boundaries

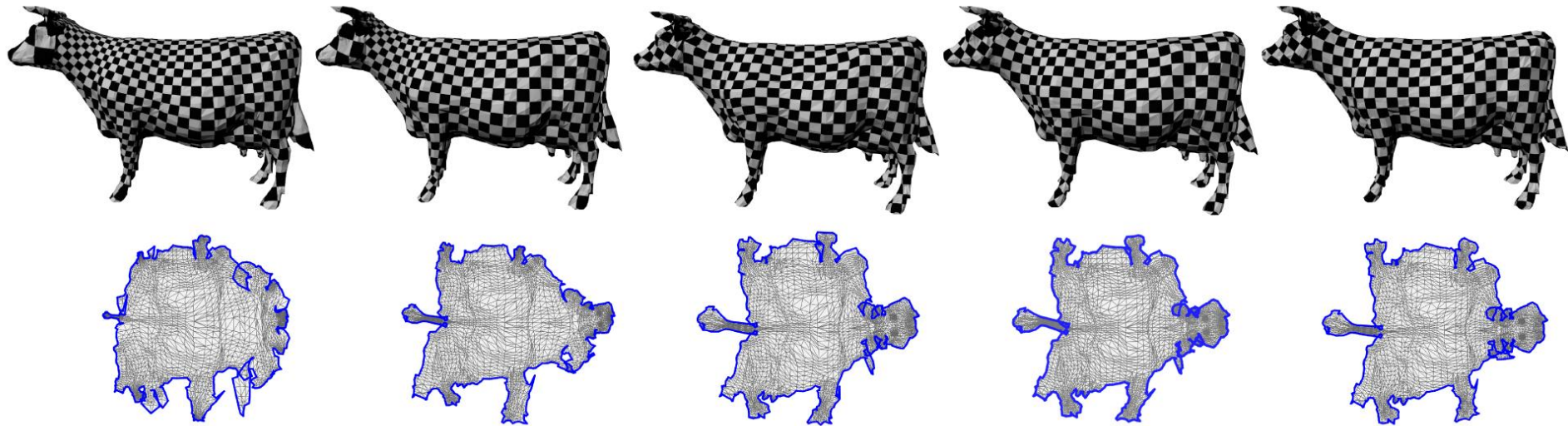
Jason Smith\*  
Texas A&M University

Scott Schaefer†  
Texas A&M University



# Issues

- Only local injectivity



# Issues

- Only local injectivity
- Sometimes the step size is too small
  - One almost inverted triangle is enough

