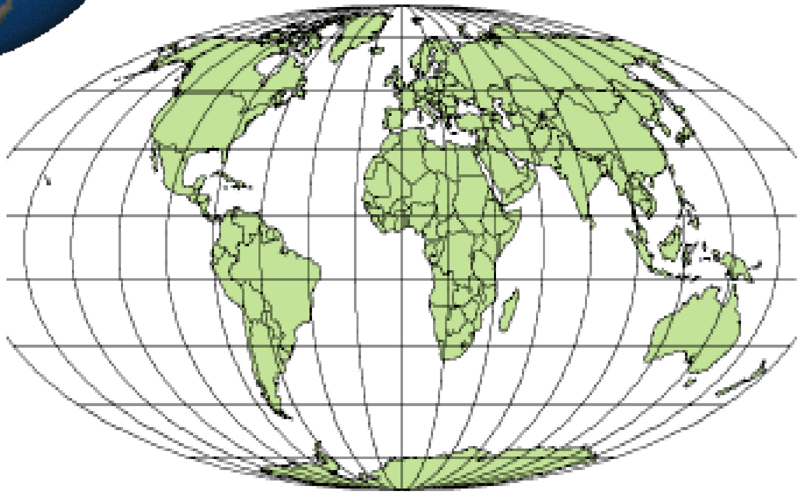
Parametrization

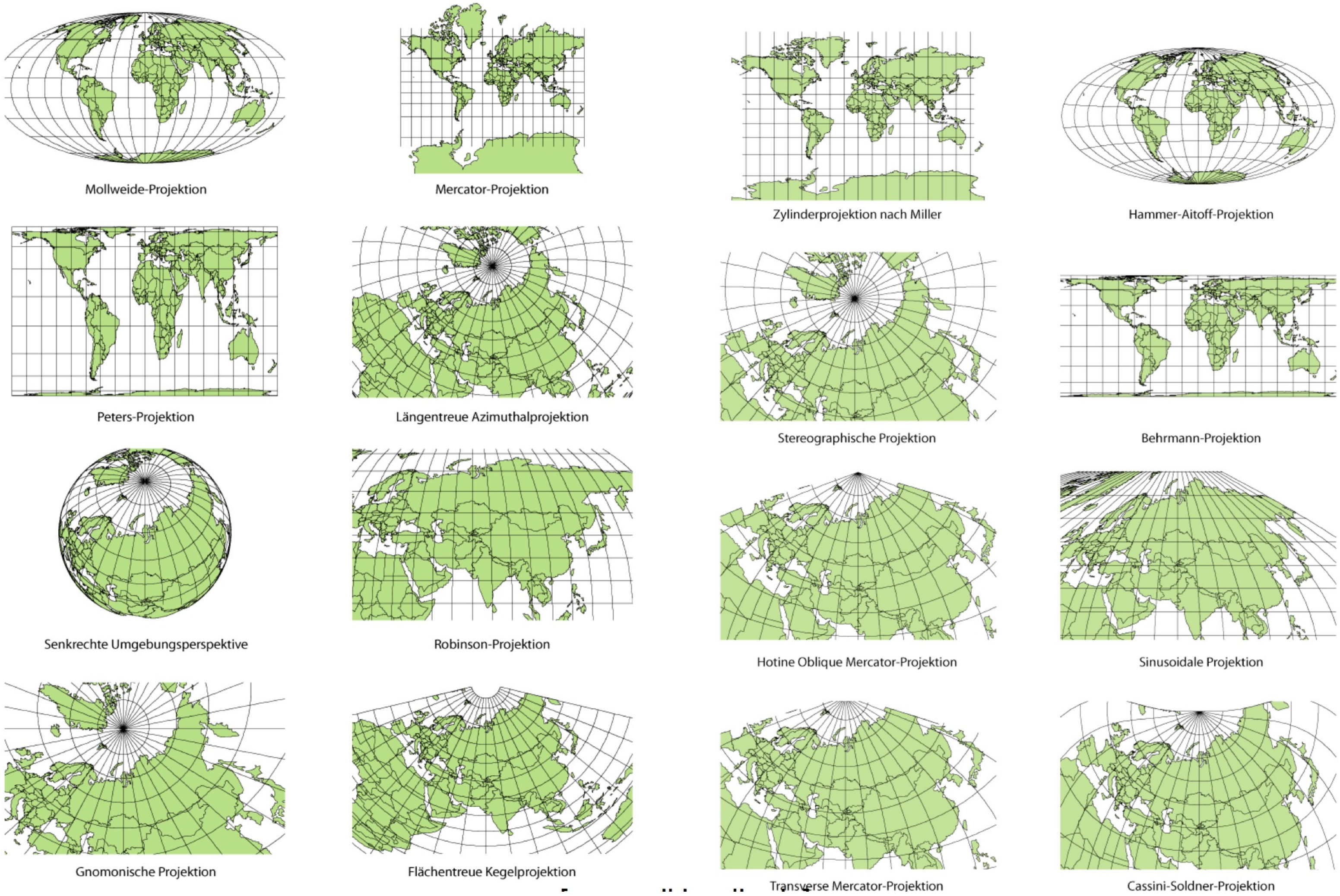
Paolo Cignoni

3D GEOMETRIC MODELING & PROCESSING

What is a parametrization?



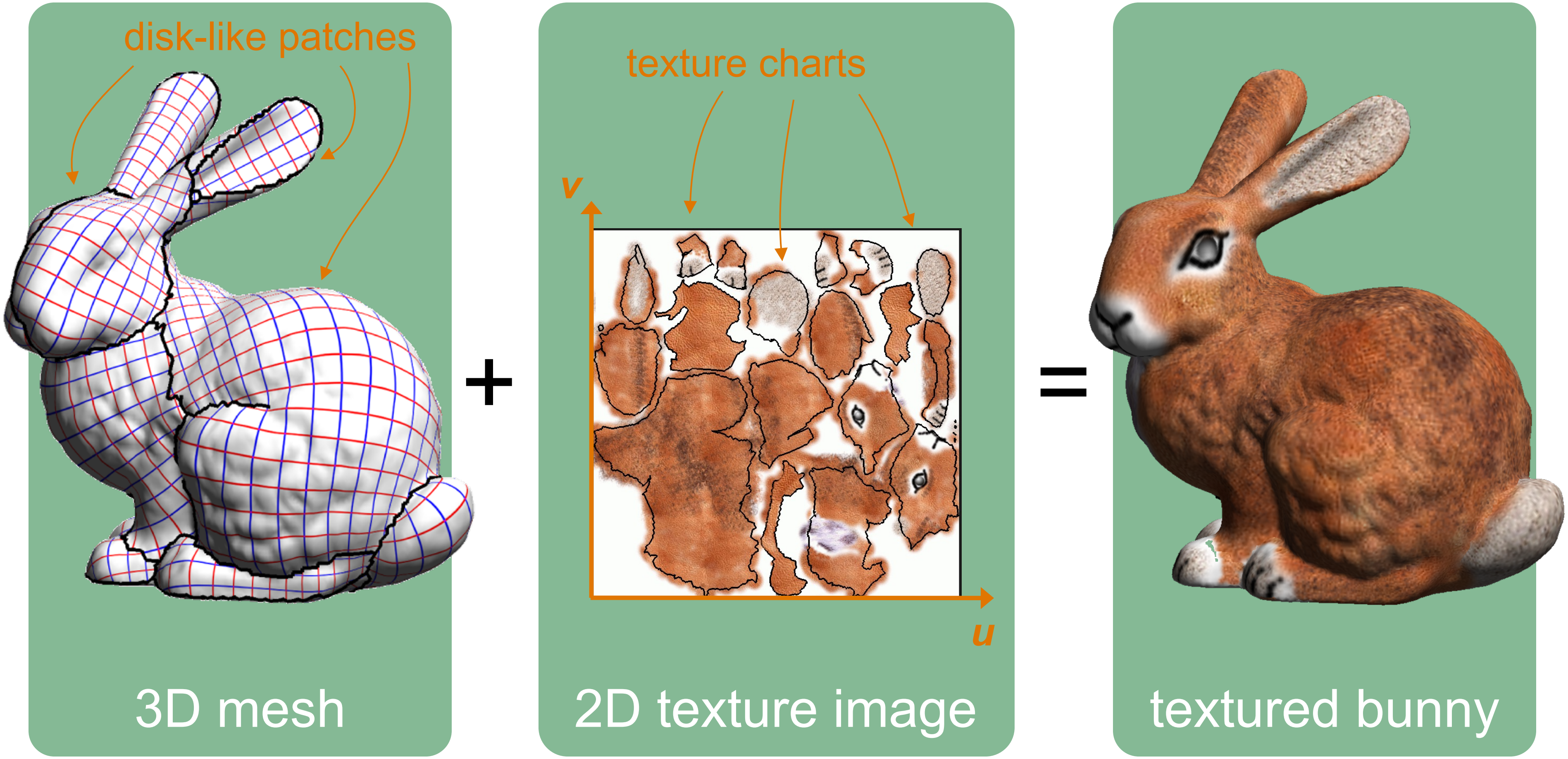
What is a parametrization?



http://vcg.isti.cnr.it/~tarini/spinnableworldmaps/

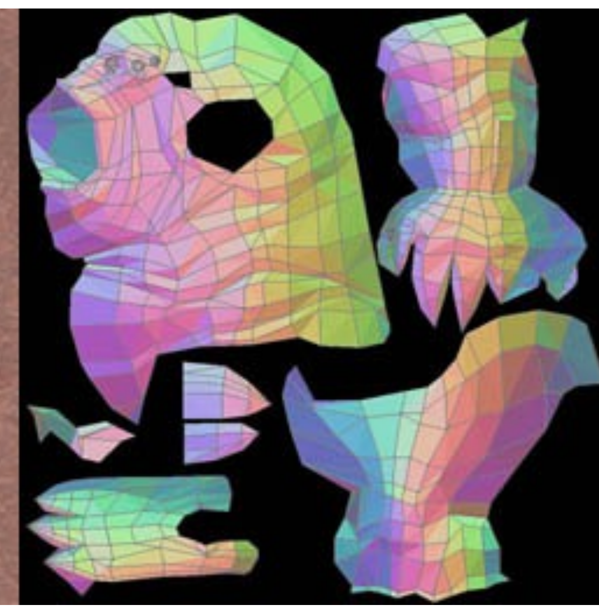
Why Parametrization?

- Texture Mapping



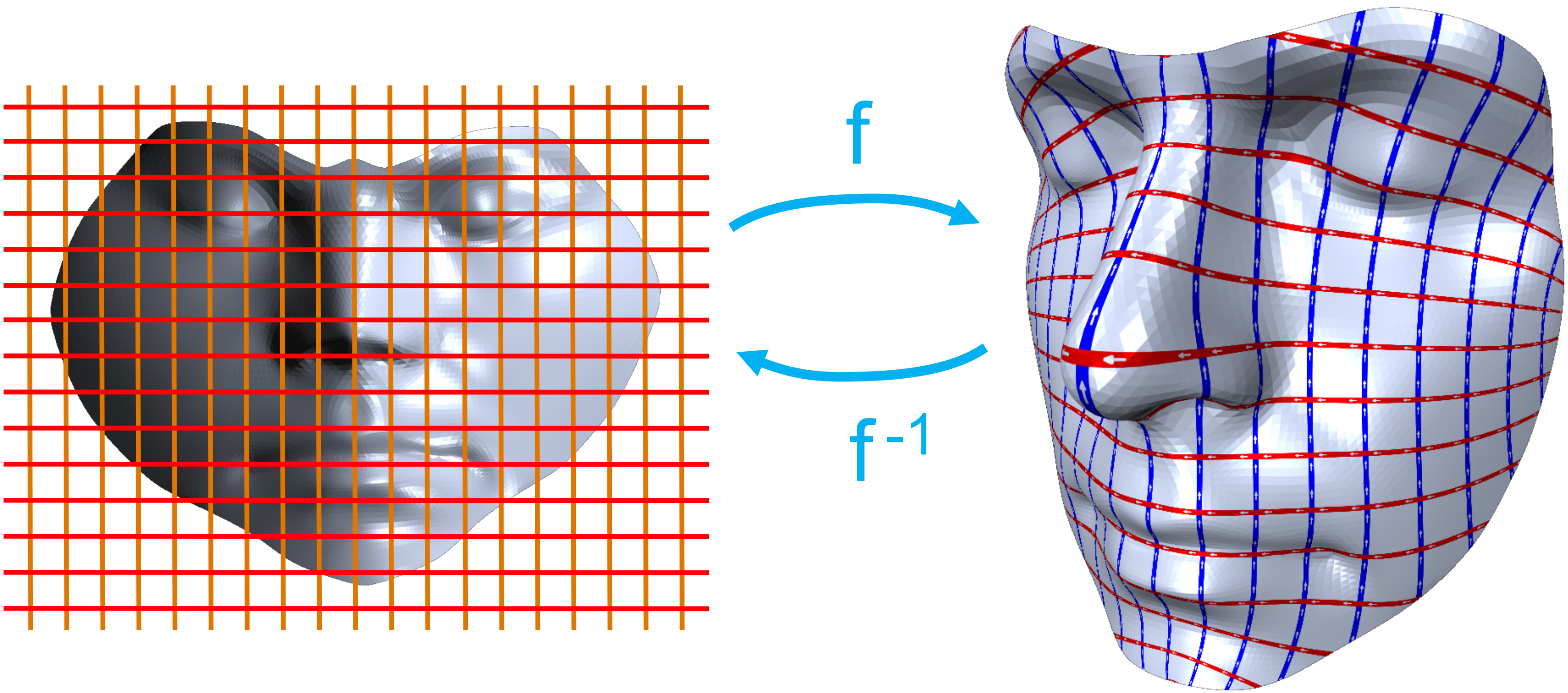
Why Parametrization?

- Manual UV mapping
- An advanced artistic skill



Why Parametrization?

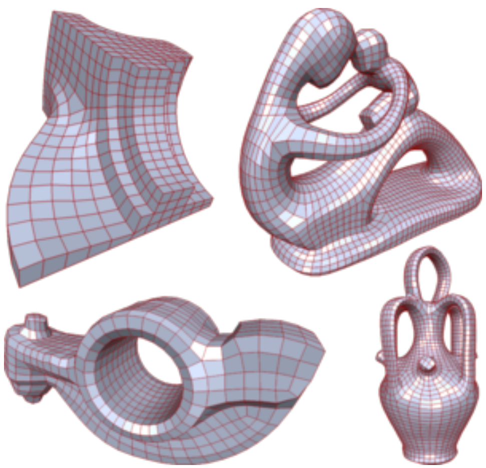
- Remeshing



Why Parametrization?

- Remeshing

QUADRILATERAL

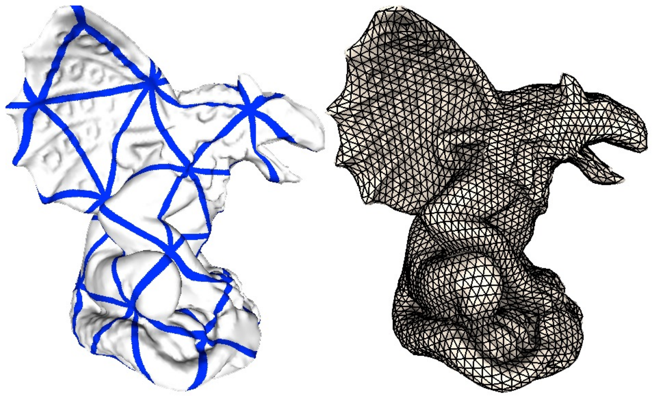


Bommes, et AL.: *Mixed Integer Quadrangulation*

HEXAGONAL

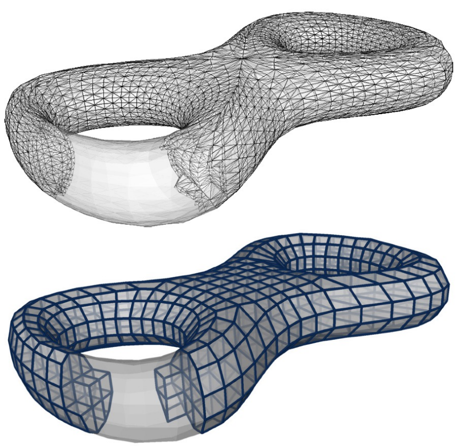
Nieser et al.: *Hexagonal Global Parameterization of Arbitrary Surfaces*

TRIANGULAR



Pietorni, et AL. :Almost isometric mesh parameterization through abstract domains

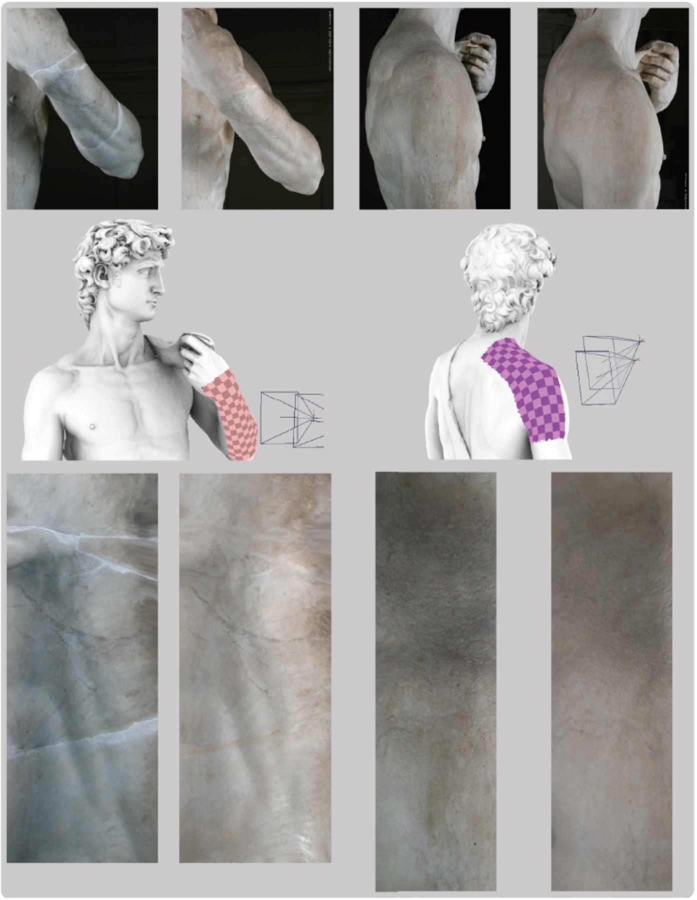
HEXAHEDRAL



Nieser, et AL. : CUBECOVER – Parameterization of 3D Volumes

Why Parametrization?

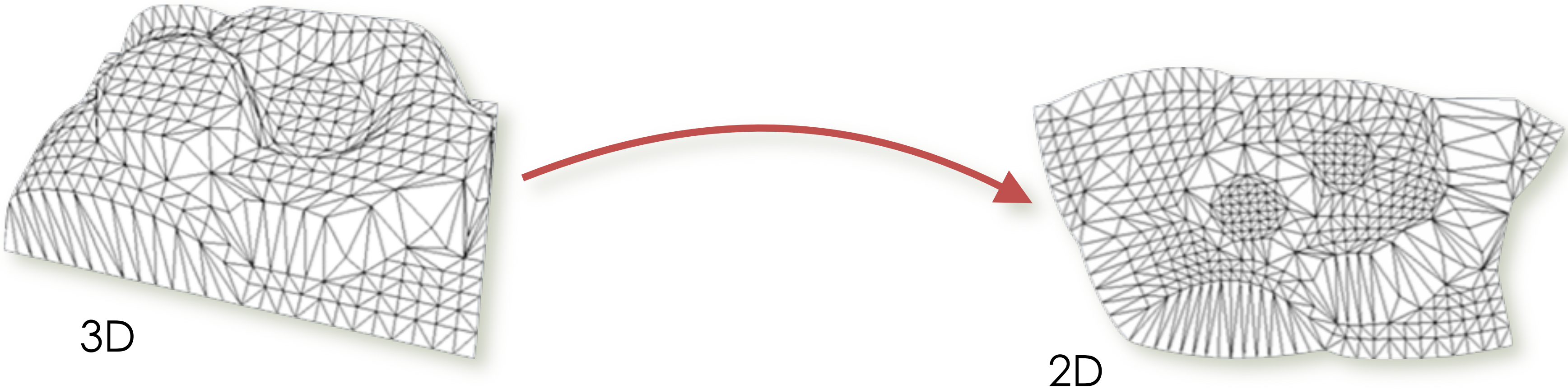
- Analysis…. 2D is easier than 3D



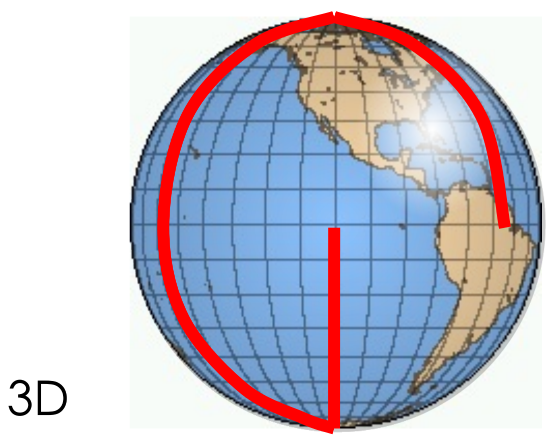
Pietroni, et AL.: *An Interactive Local Flattening Operator to Support Digital Investigations on Artwork Surfaces*

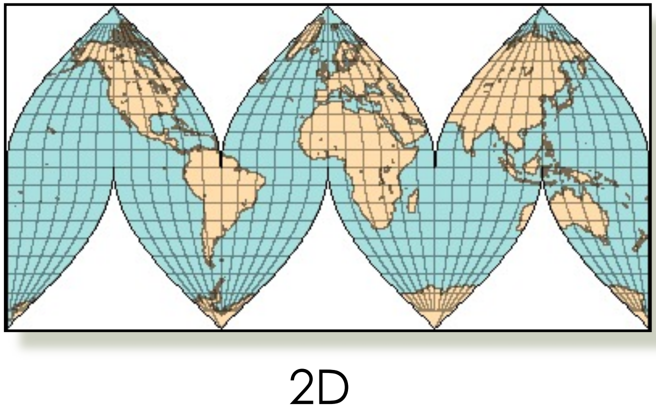
Parametrization: what we need?

- A strategy to flatten a 3D surface on 2D domain
  - Introducing as few distortion as possible



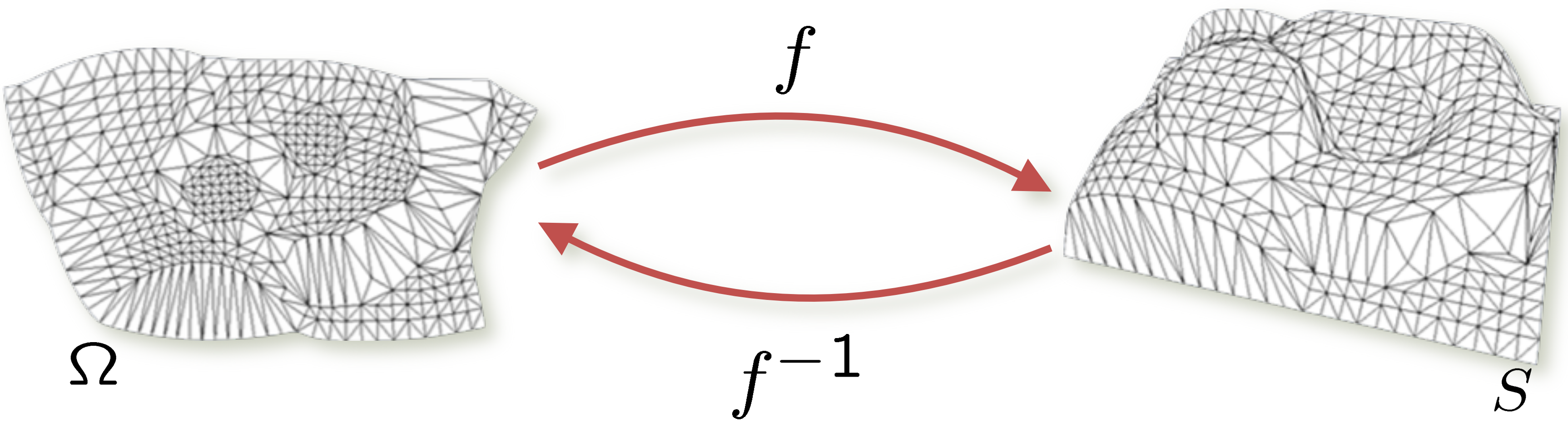
- A strategy to introduce cuts



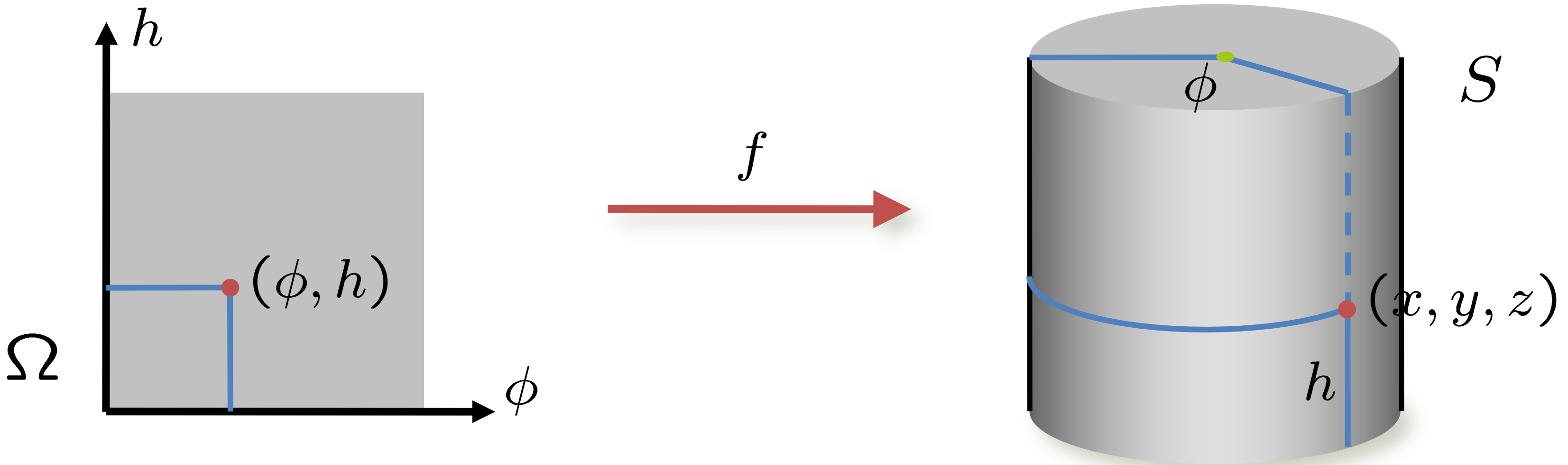


Flattening a surface

- surface $S \subset \mathbb{R}^3$
- parameter domain $\Omega \subset \mathbb{R}^2$
- mapping $f : \Omega \rightarrow S$ and $f^{-1} : S \rightarrow \Omega$



Parametrization: Cylindrical coords



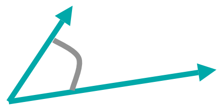
$$S = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 = 1, z \in [0, 1]\}$$

$$\Omega = \{(\phi, h) \in \mathbb{R}^2 : \phi \in [0, 2\pi), h \in [0, 1]\}$$

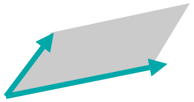
$$f(\phi, h) = (\sin\phi, \cos\phi, h)$$

Minimize Distortion

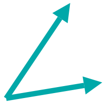
- Angle preservation: conformal



- Area preservation: equiareal



- Area and Angle: Isometric



Distortion

What happens to the surface point f(u,v) as we move a tiny little bit away from (u,v) in the parameter domain?

- Approximate with first order Taylor expansion

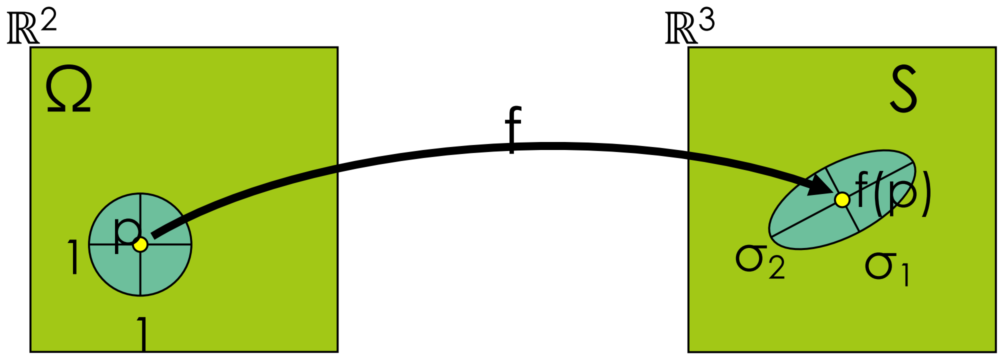
$$\tilde{f}(u+\Delta u, v+\Delta v) = f(u,v) + f_u(u,v)\Delta u + f_v(u,v)\Delta v.$$

$$f_u = \frac{\partial f}{\partial u} \quad \text{and} \quad f_v = \frac{\partial f}{\partial v}$$

$$\tilde{f}(u+\Delta u, v+\Delta v) = \boldsymbol{p} + J_f(\boldsymbol{u})\binom{\Delta u}{\Delta v},$$

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

  - J_f Jacobian of f, i.e. the 3×2 matrix with partial derivatives of f as column vectors



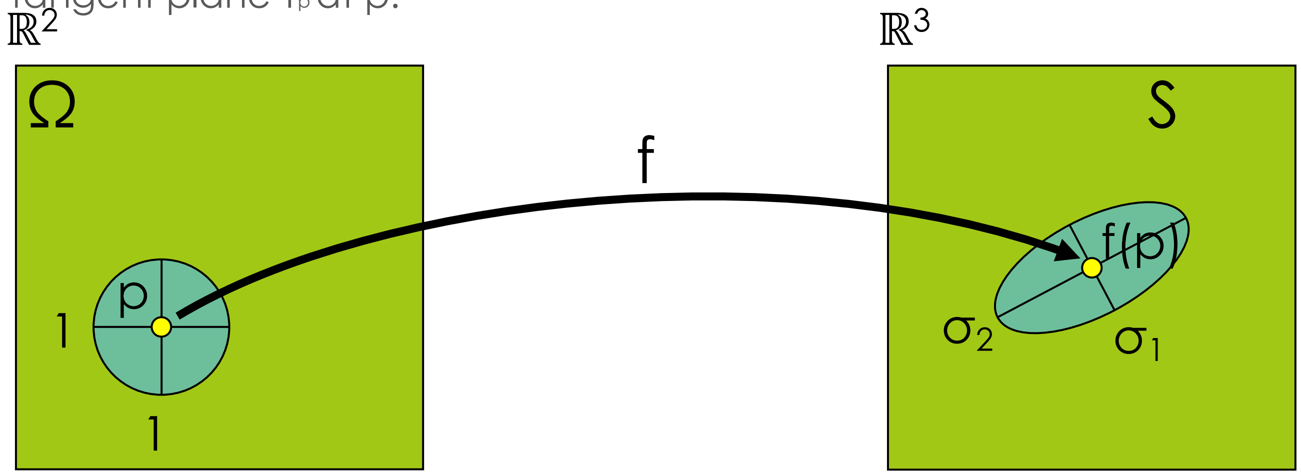
Distortion

$$\tilde{f}(u+\Delta u, v+\Delta v) = \boldsymbol{p} + J_f(\boldsymbol{u})\binom{\Delta u}{\Delta v}, \qquad J_f = U\Sigma V^T = U\begin{pmatrix}\sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0\end{pmatrix}V^T,$$

- Consider singular value decomposition of the Jacobian

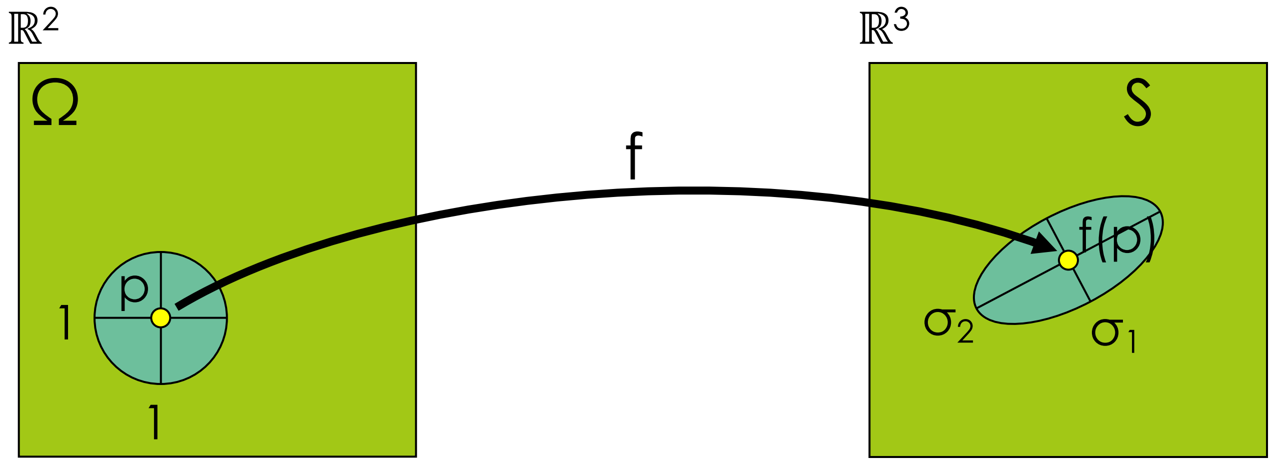
  singular values $\sigma_1 \geq \sigma_2 > 0$ and *orthonormal* matrices $U \in \mathbb{R}^{3\times3}$ and $V \in \mathbb{R}^{2\times2}$

  - The transformation V^T first rotates all points around u such that the vectors V_1 and V_2 are in alignment with the u- and the v-axes afterwards.
  - The transformation Σ then **stretches** by the factor σ_1 in the u- and by σ_2 in the v-direction.
  - The transformation U finally maps the unit vectors (1, 0) and (0, 1) to the vectors U_1 and U_2 in the tangent plane T_p at p.



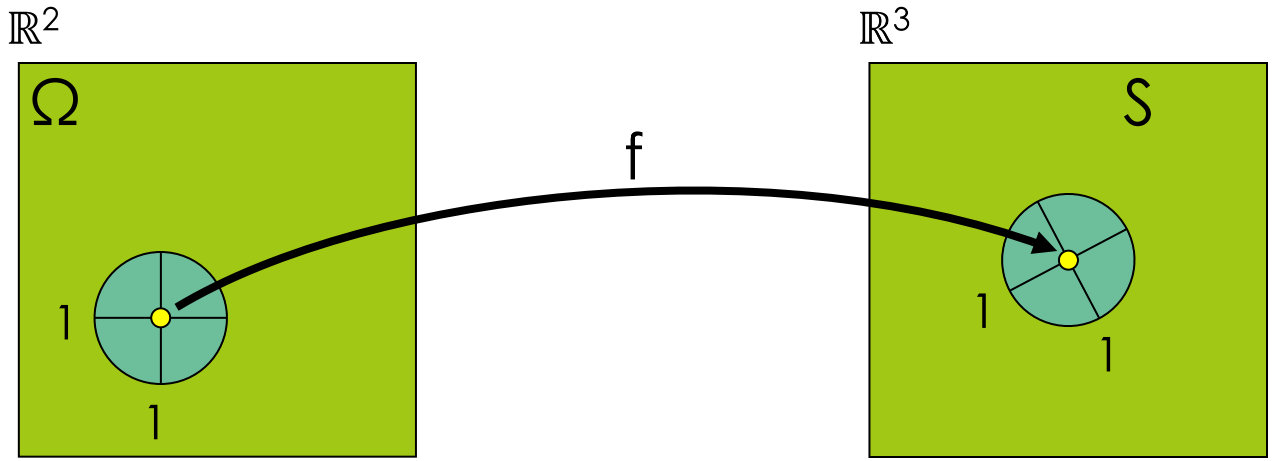
Distortion

- In practice σ_1 and σ_2 describe local deformations



Isometric Mapping

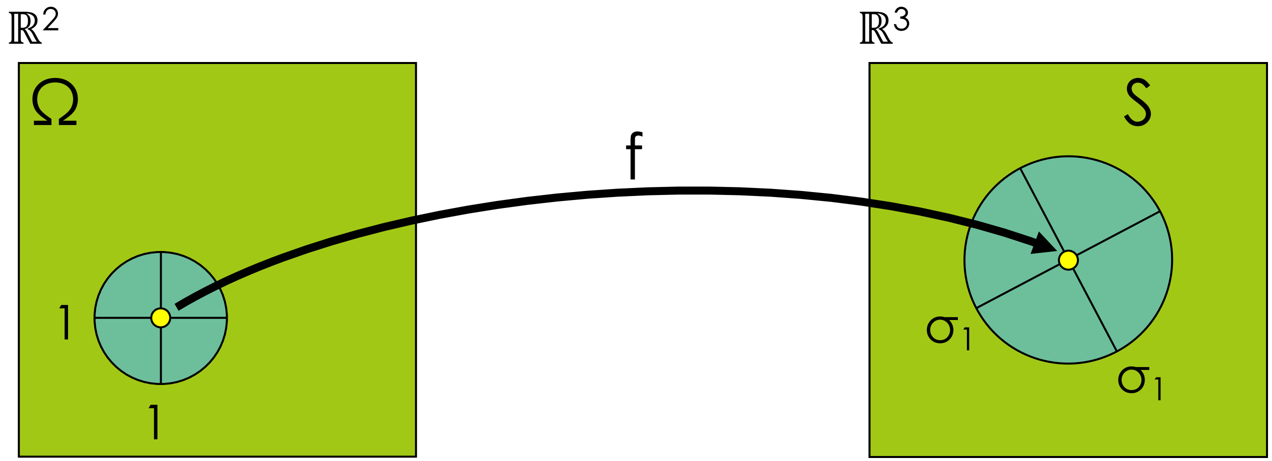
- $\sigma_1 = \sigma_2 = 1$



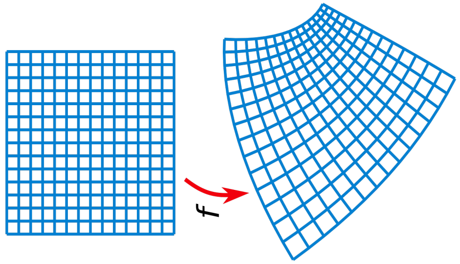
- preserves **areas**, **angles** and **lengths**

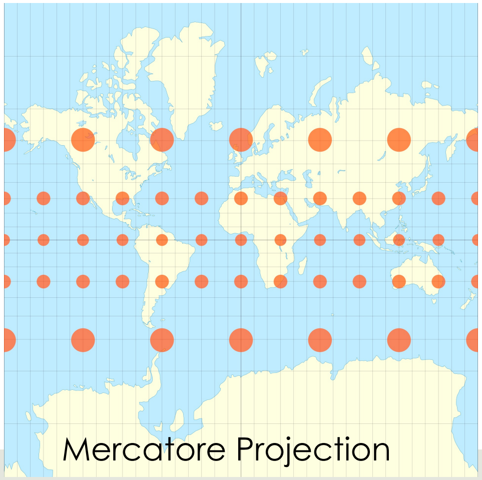
Conformal Mapping

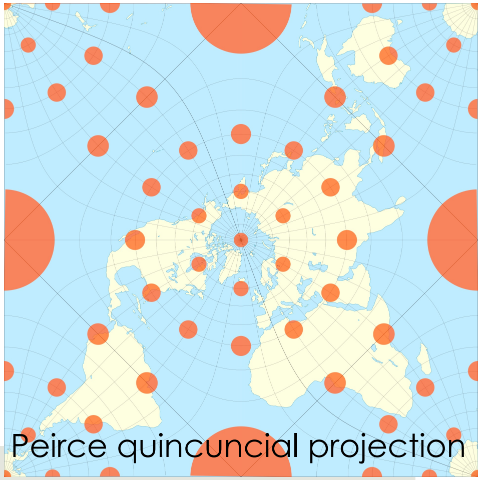
- $\sigma_1 / \sigma_2 = 1$



- preserves **angles**

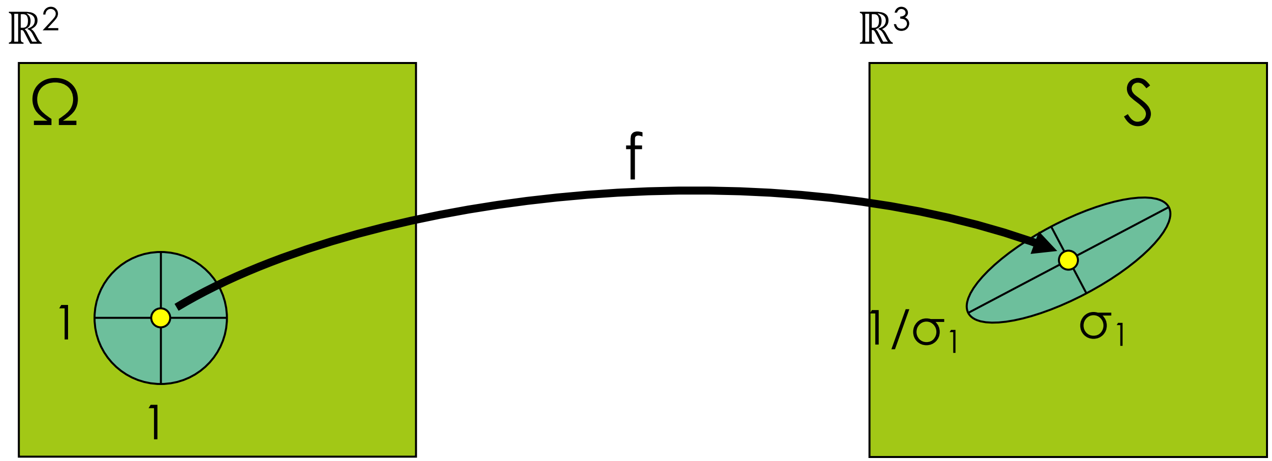




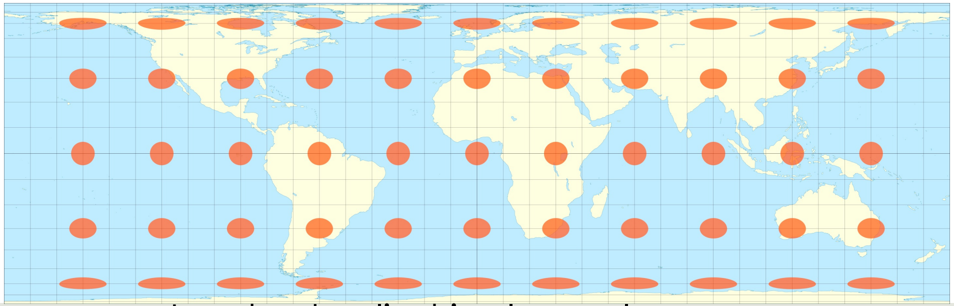


Equiareal Mapping

- $\sigma_1 \cdot \sigma_2 = 1$



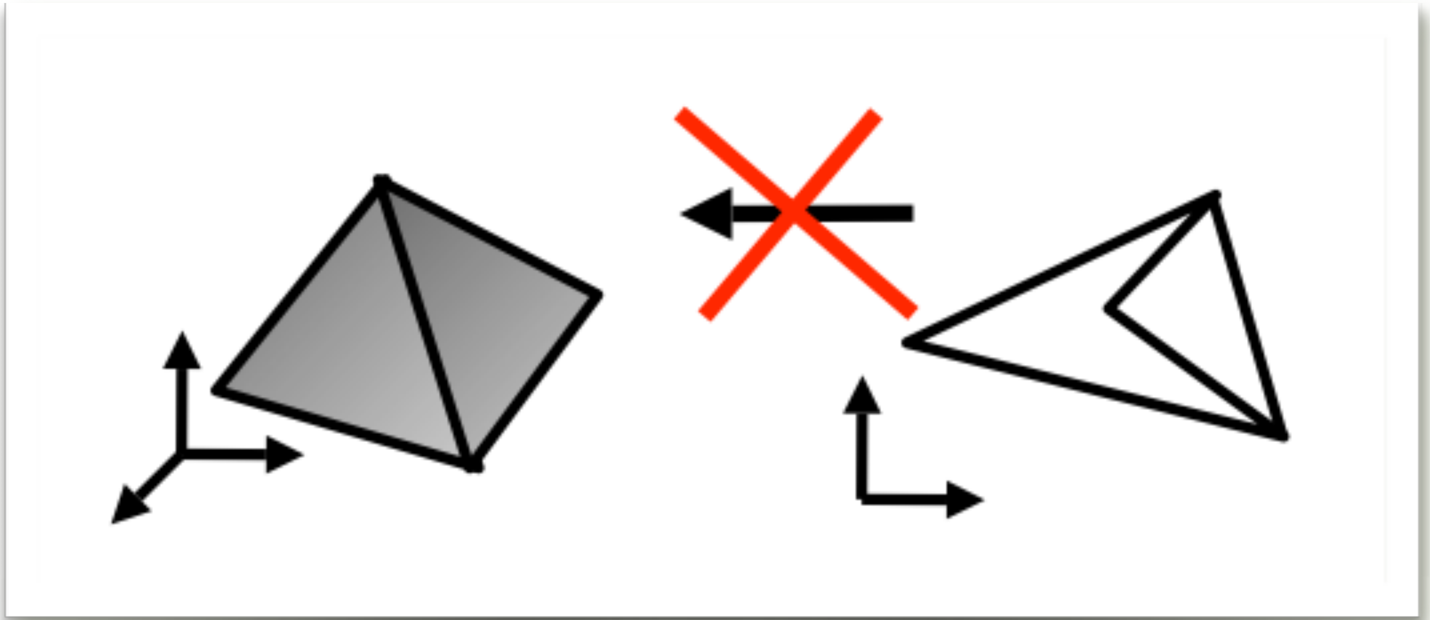
- preserves **areas**



Lambert cylindrical equal-area

Bijectivity

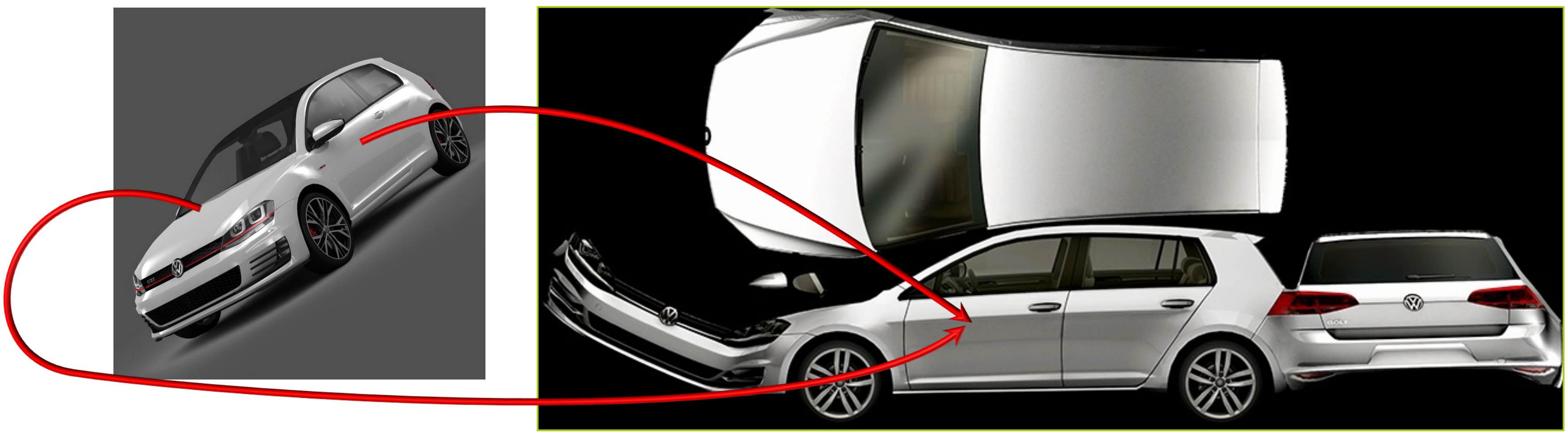
- Parametrization map must be bijective ⇔ triangles in parametric domain do not overlap (no triangle flips)



Bijectivity

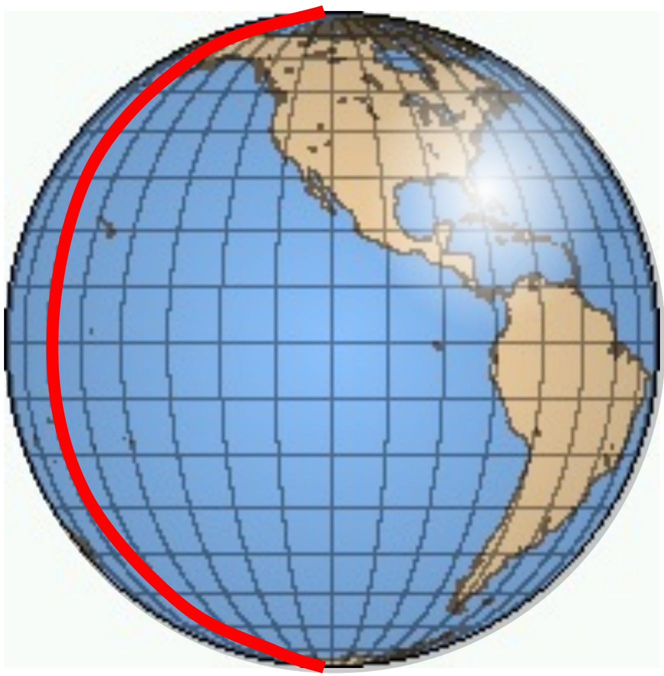
should

- Parametrization map ~~must~~ be bijective ⇔ triangles in parametric domain do not overlap (no triangle flips)



Cuts 1

- Clearly needed for closed surfaces

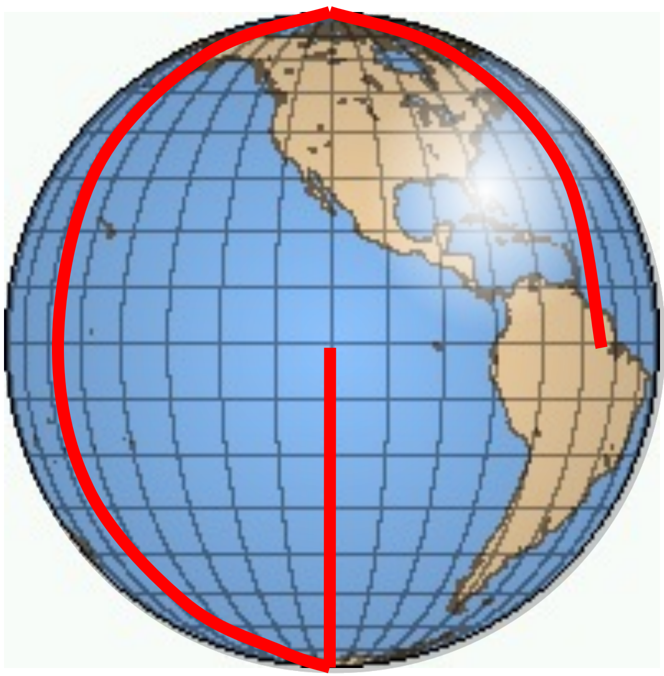


sphere in 3D

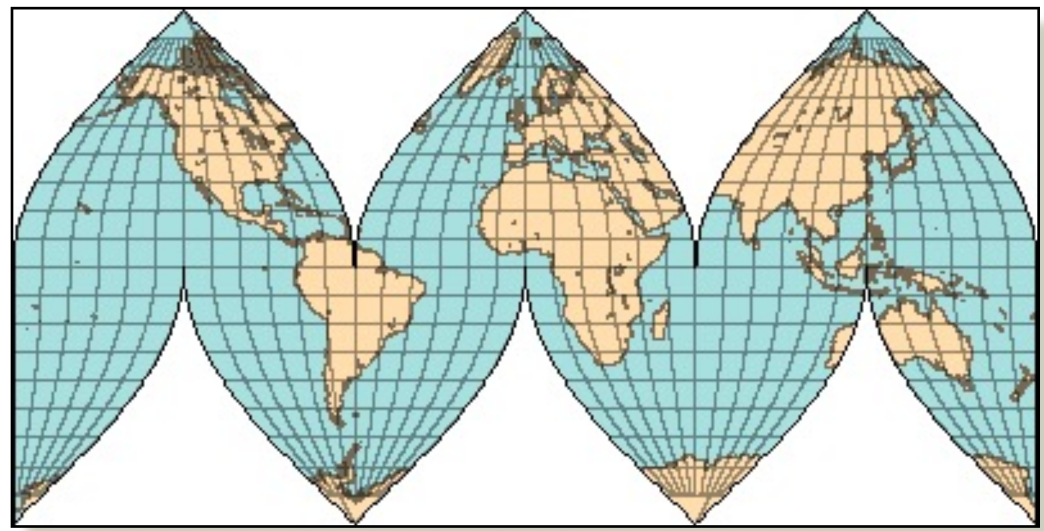
2D surface disk

Cuts 2

- Usually more cuts -> less distortion



sphere in 3D



2D surface

Cuts 3: closed surfaces

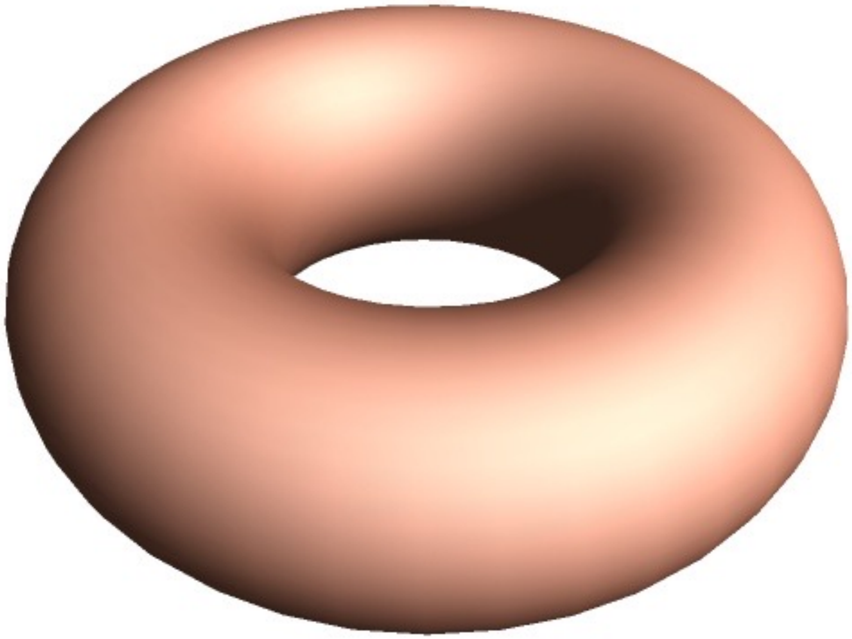
- How many cuts?

for a genus 0 surface ?

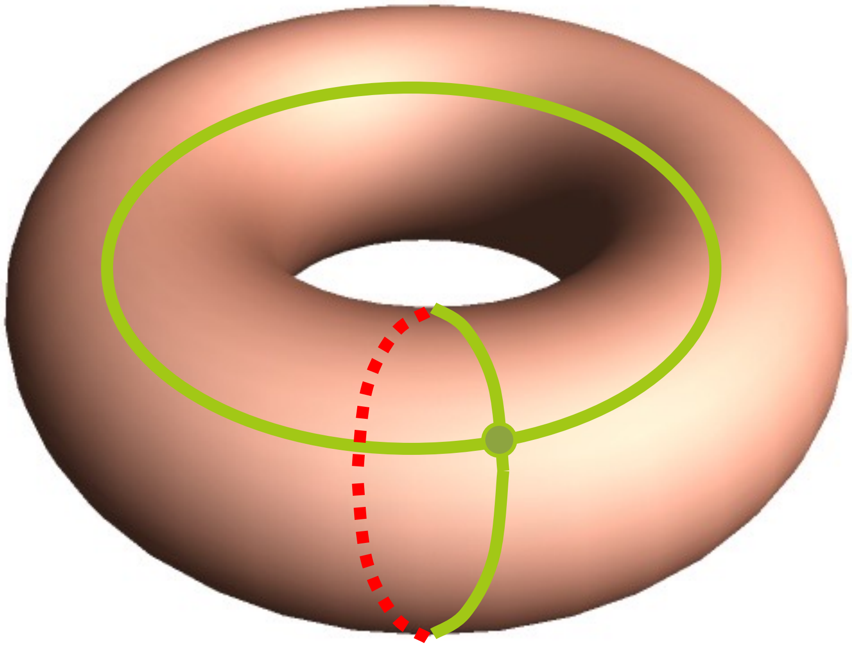
any tree of cuts

Cuts 3: closed surfaces

- How many cuts?



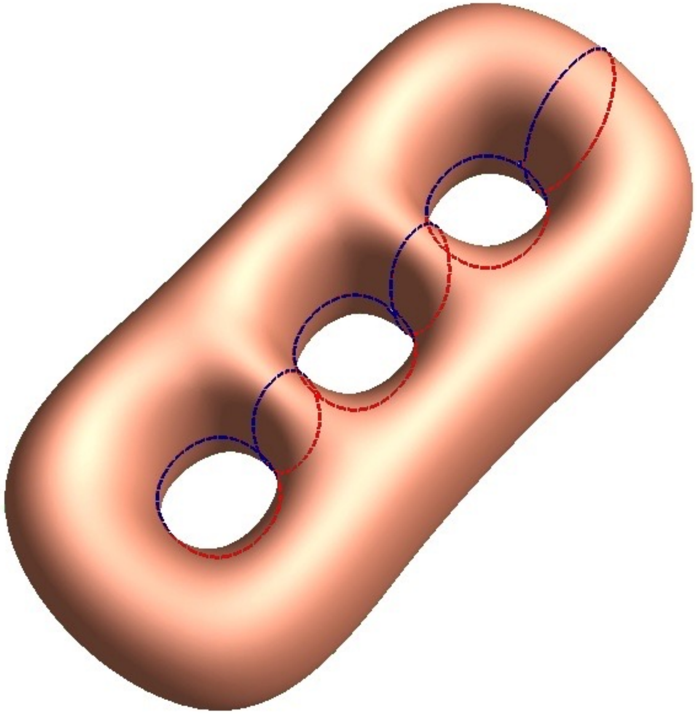
for a genus 1 surface ?



two looped cuts

Cuts 3: closed surfaces

- How many cuts?

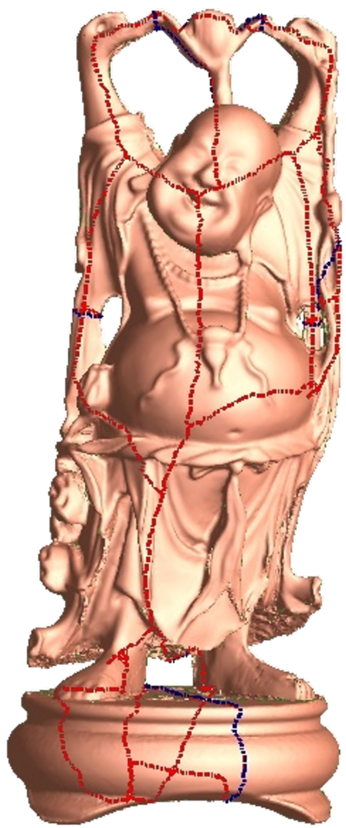


for a genus 3 surface ?

6 looped cuts

Cuts 3: closed surfaces

- How many cuts?



genus 6

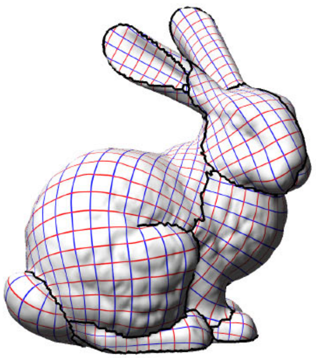
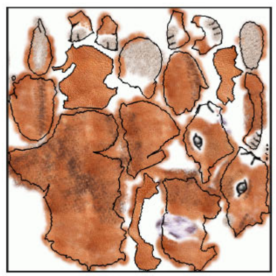
for a genus n surface ?

2n looped cuts

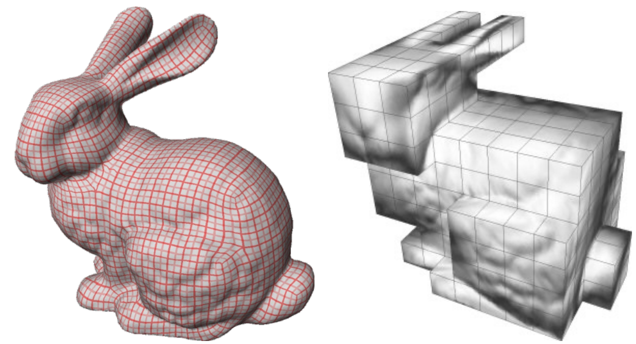
Generic Cut Strategies

- Texture Mapping

UNSTRUCTURED CUTS

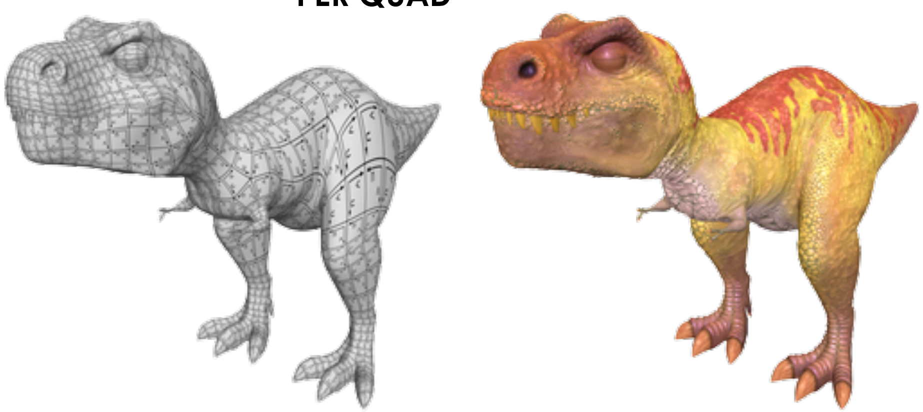

IMPLICIT



Tarini, et AL.: *PolyCube Maps*

Lévy, et AL.: *Least squares conformal maps for automatic texture atlas generation*

PER QUAD



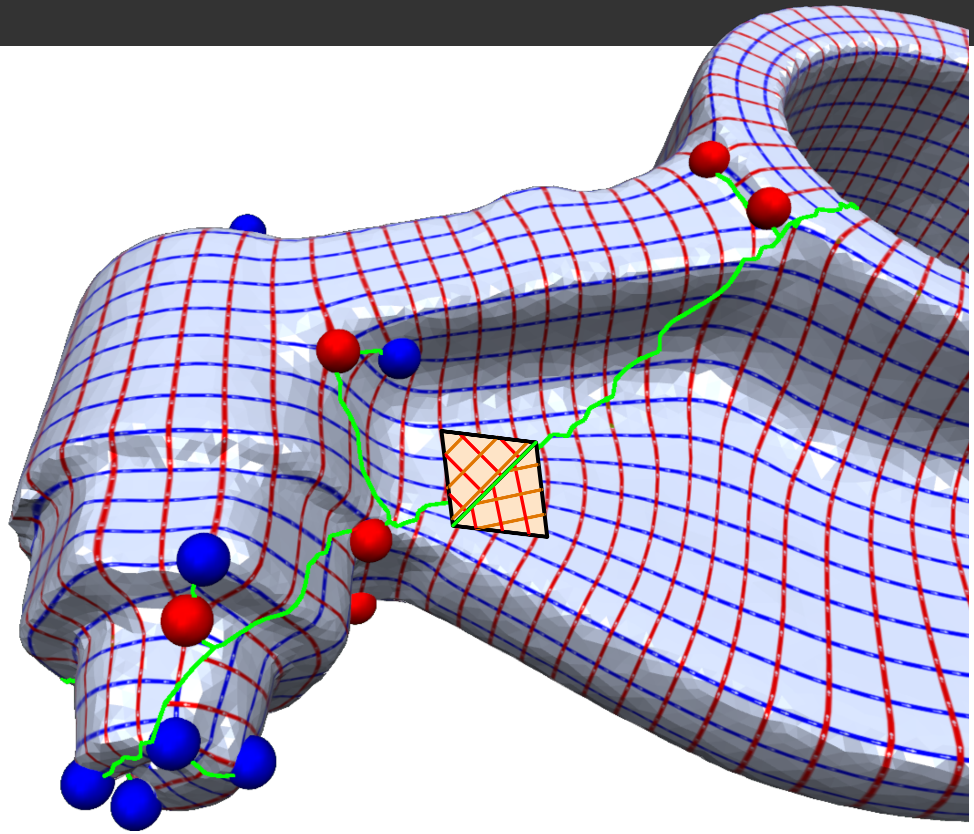
Brent Burley et al : Ptex: Per-Face Texture Mapping for Production Rendering

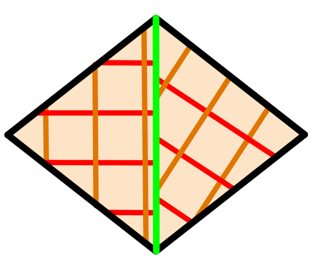
REGULAR CUTS


Pietroni, et AL.: *Almost isometric mesh parameterization through abstract domains*

Globally Smoothess

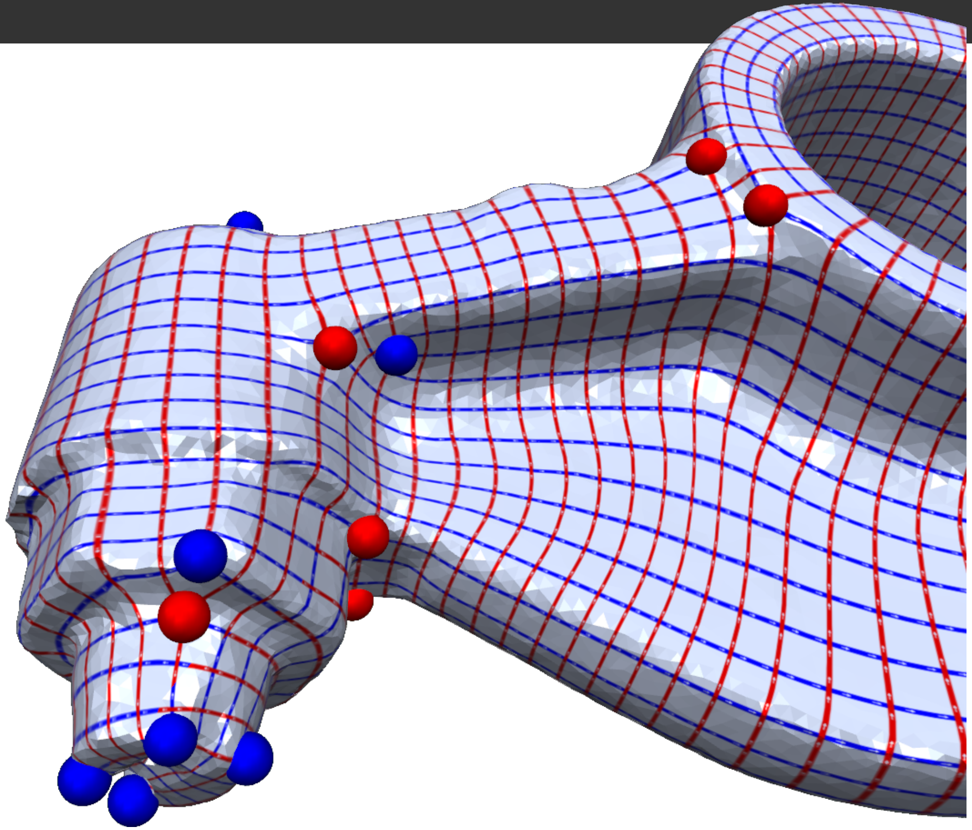
- Tangent directions varyes smoothly across seams

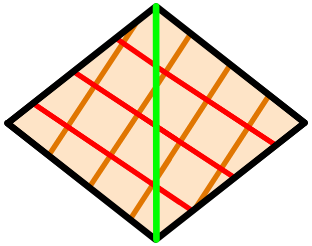




Globally Smoothness

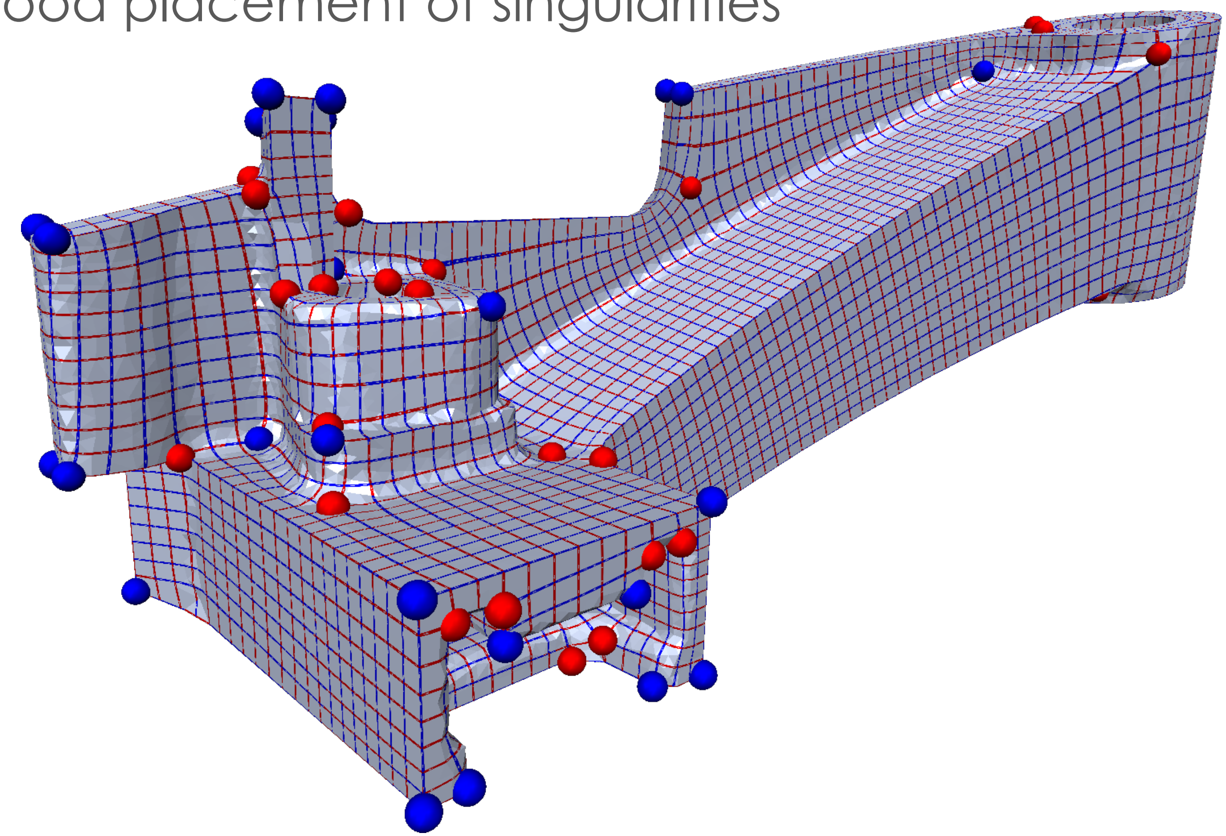
- Tangent directions vary smoothly across seams





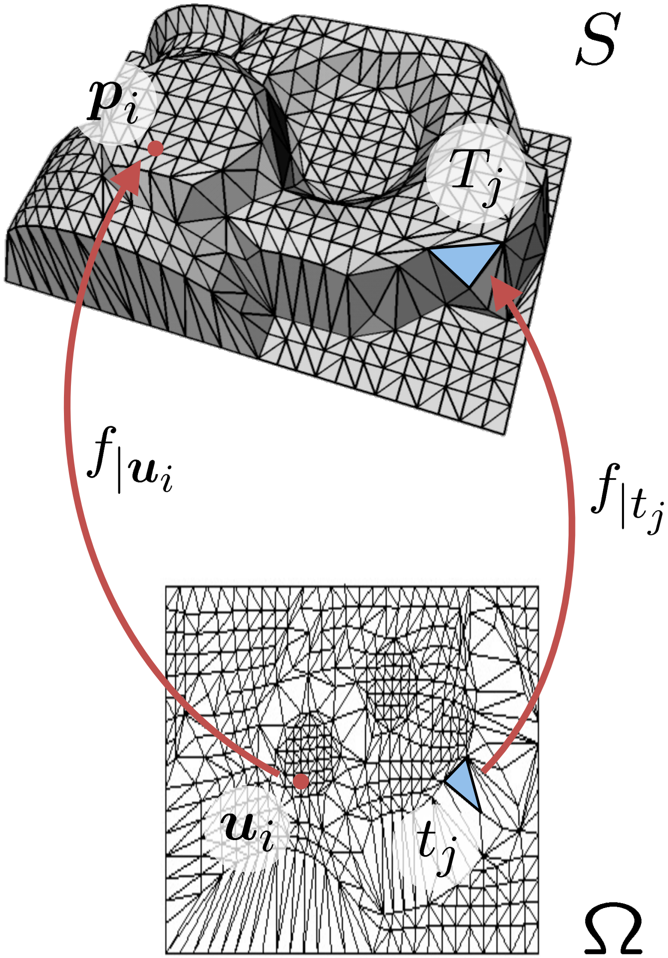
Feature Alignment

- Useful for quadrangulation
- Need good placement of singularities



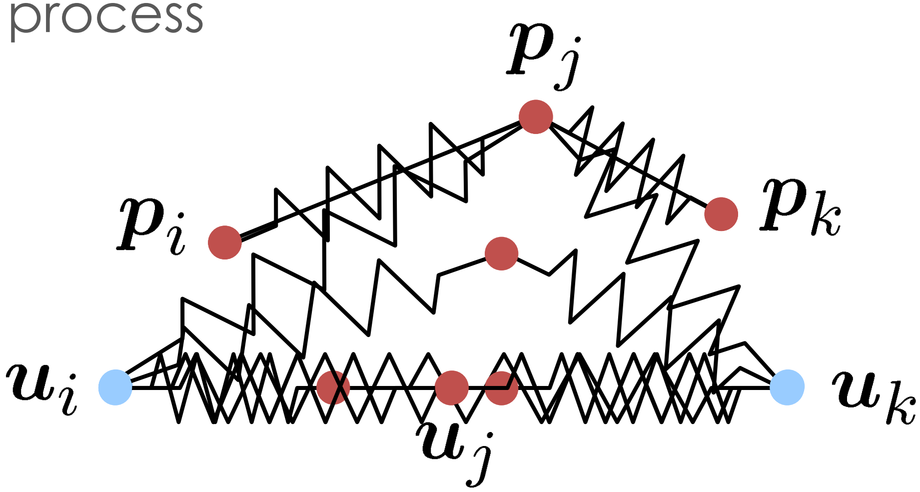
Details: Parametrization

- triangle mesh $S \subset \mathbb{R}^3$
  - vertices $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n+b}$
  - Triangles $T_1, \ldots, T_m$
- parameter mesh $\Omega \subset \mathbb{R}^2$
  - parameter points $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n+b}$
  - parameter triangles $t_1, \ldots, t_m$
- parameterization $f : \Omega \rightarrow S$
  - piecewise linear map $f(t_j) = T_j$



Parametrization: Mass-Spring

- replace **edges** by **springs**
- Position of vertices $p_0..p_n$
- UV Position of vertices $u_0..u_n$
- **relaxation** process



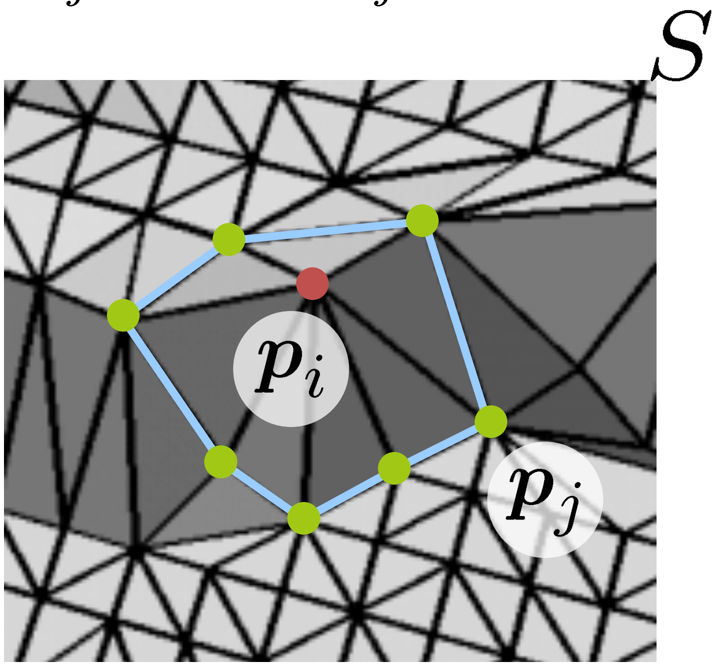
Energy Minimization

- energy of spring between p_i and p_j : $\frac{1}{2}D_{ij}{s_{ij}}^2$
- spring constant (stiffness) $D_{ij} > 0$
- spring length (in parametric space) $s_{ij} = \|\boldsymbol{u}_i - \boldsymbol{u}_j\|$
- total energy

$$E = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2}D_{ij}\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2$$

- partial derivative

$$\frac{\partial E}{\partial \boldsymbol{u}_i} = \sum_{j\in N_i} D_{ij}(\boldsymbol{u}_i - \boldsymbol{u}_j)$$



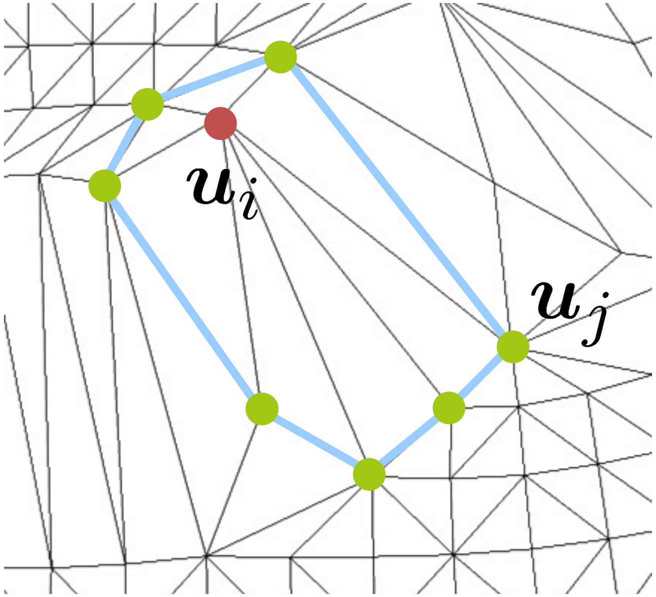
Linear System

- u_i is expressed as a **convex combination** of its neighbours u_j

$$\boldsymbol{u}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{u}_j$$

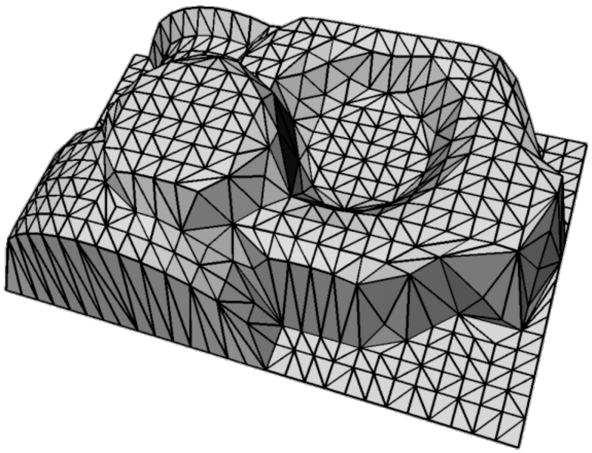
- With weights

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

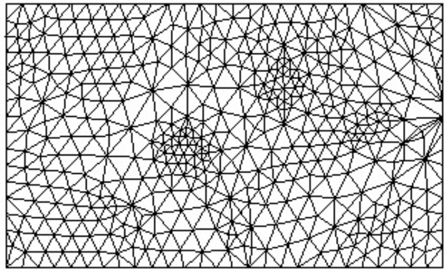


- LEAD to Linear System!

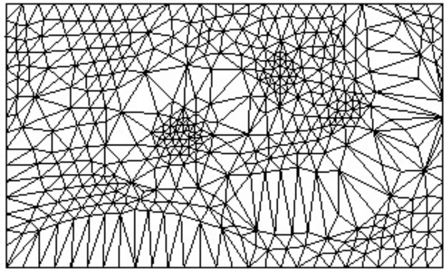
Which Weights?



- uniform spring constants

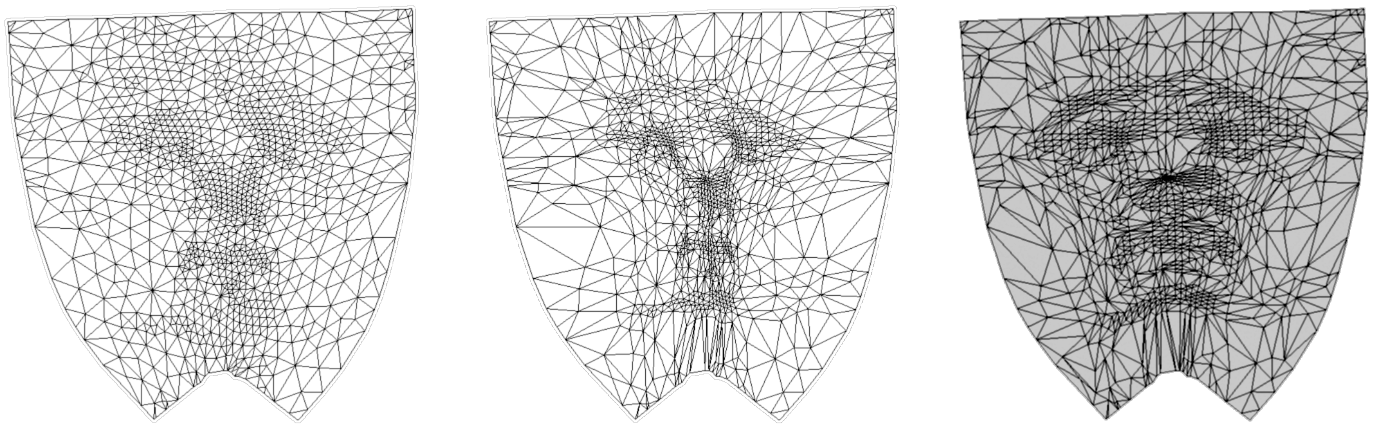


- Proportional to 3D distance



Which Weights?

- **NO** linear reproduction
- Planar mesh are distorted



Which Weights?

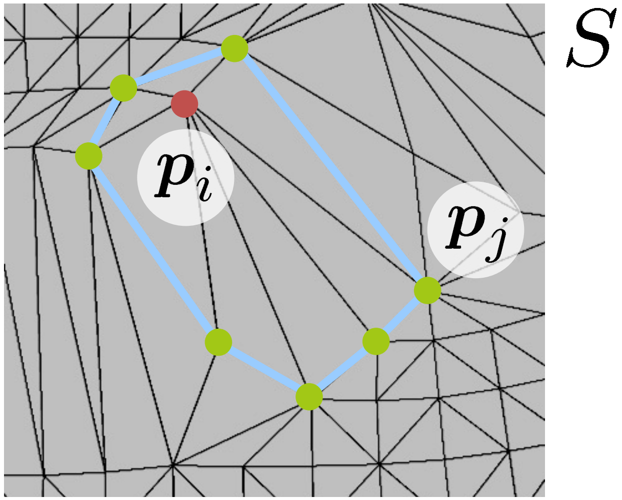
- suppose **S** to be is planar
- specify weights λ_{ij} such that

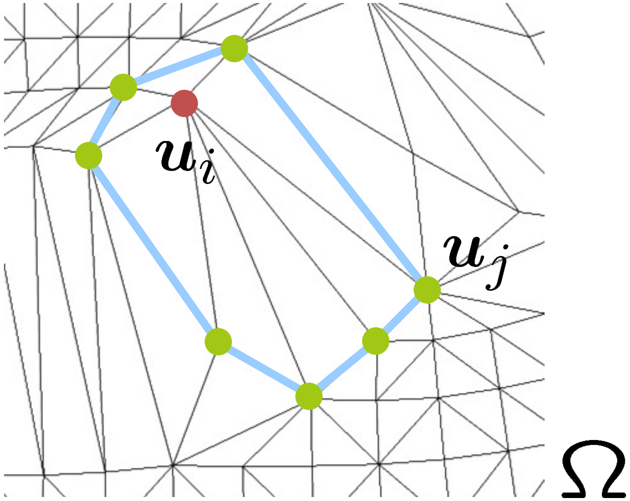
$$\boldsymbol{p}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{p}_j$$

- Then solving

$$\boldsymbol{u}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{u}_j$$

- Reproduces **S**





Which Weights?

- Wachspress coordinates

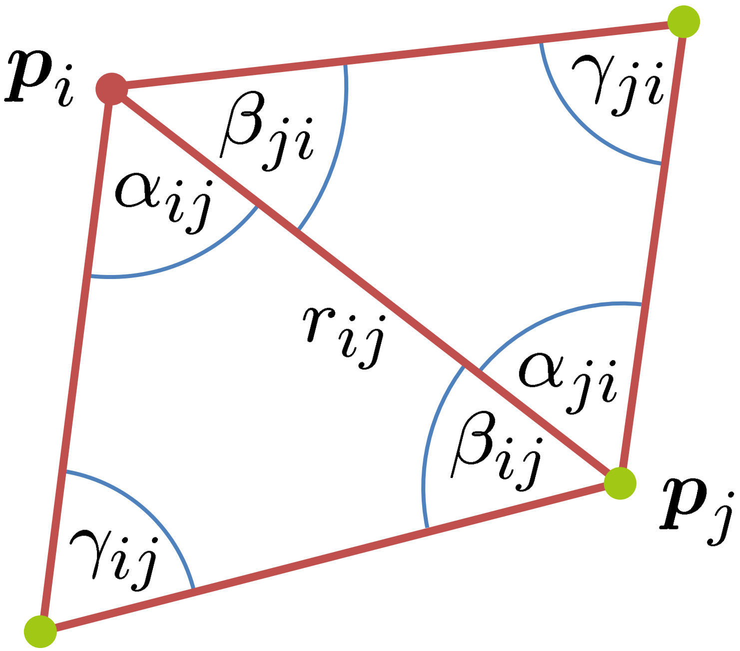
$$w_{ij} = \frac{\cot\alpha_{ji} + \cot\beta_{ij}}{{r_{ij}}^2}$$

- discrete harmonic coordinates

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

- mean value coordinates

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

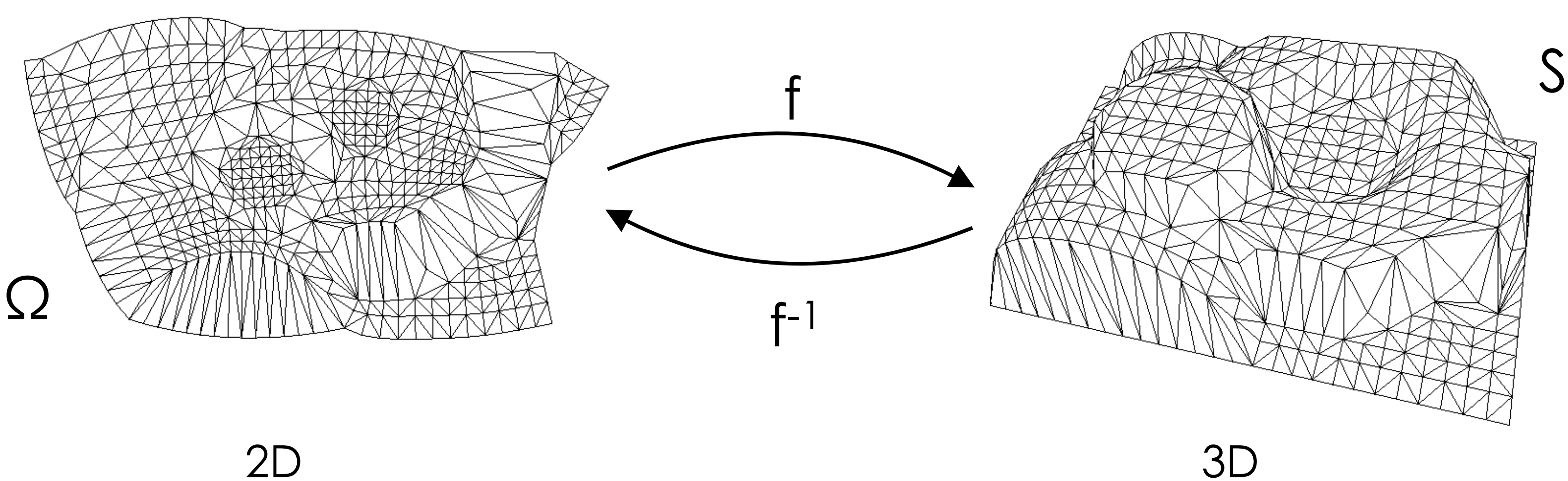


normalization

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

Recap

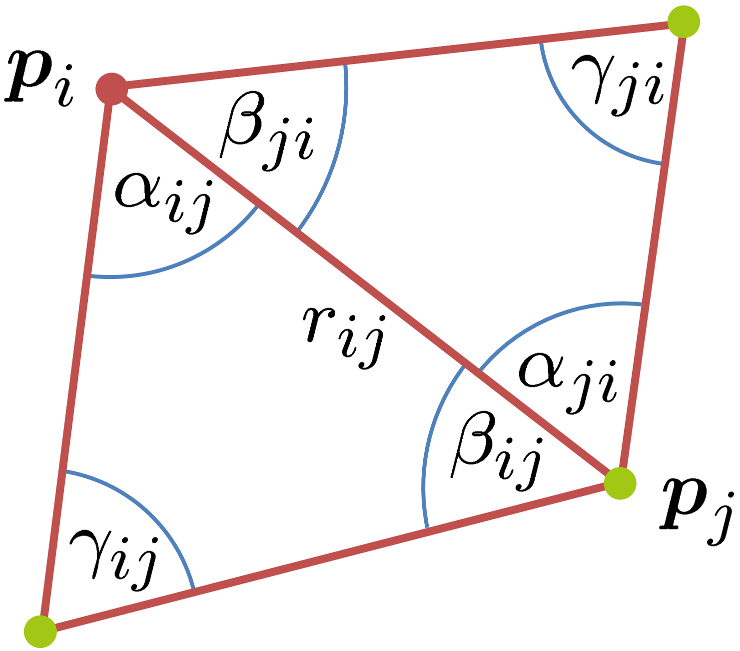
- Parametrization



Weight ed average

- discrete harmonic coordinates

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



normalization

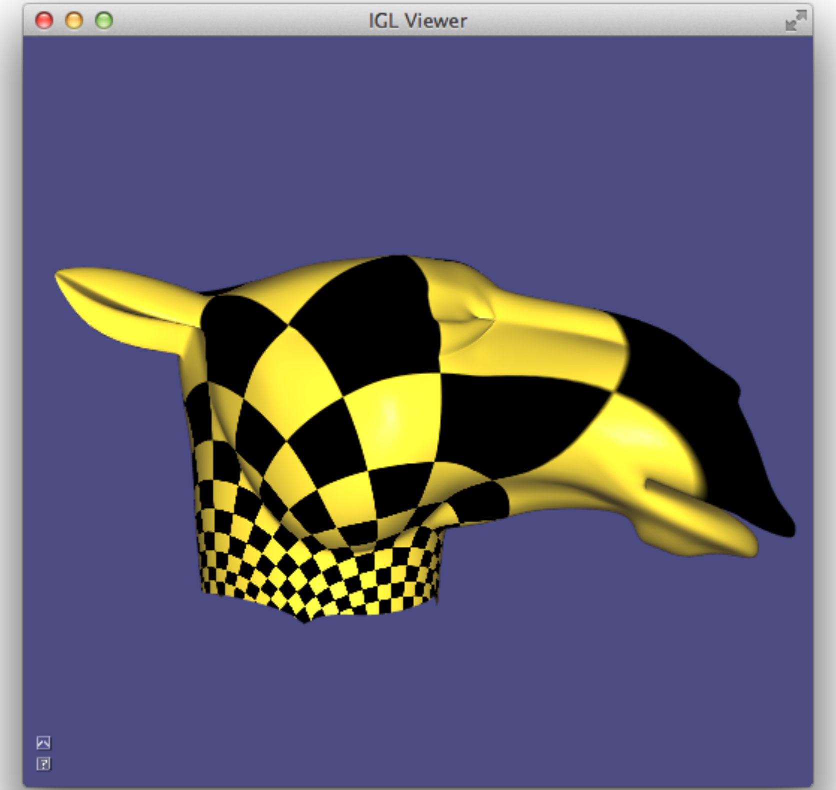
$$\lambda_{ij} = \frac{w_{ij}}{\sum_{k \in N_i} w_{ik}}$$

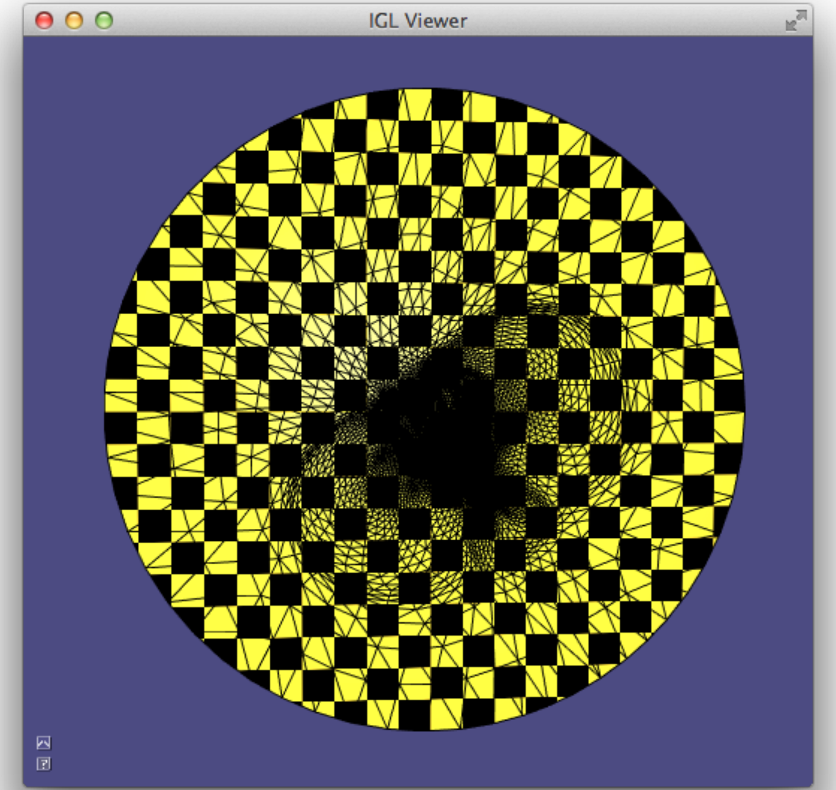
Harmonic parametrization

- Linear sistem
- Sparse matrix (2n x 2n), where n is number of vertices of the mesh
- Express each point as weighted sum of its neighbors
- Find x such that Ax=0
- x are the final UV coordinates!

Harmonic parametrization

- Fix the boundary of the mesh to UV
- Express each UV position as linear combination of neighbors
- Use cotangent weights!

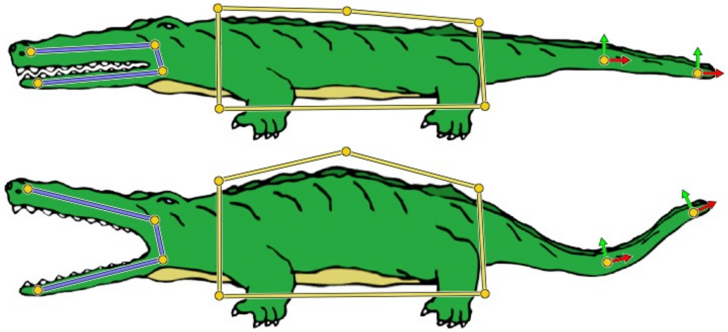




Harmonic Weights

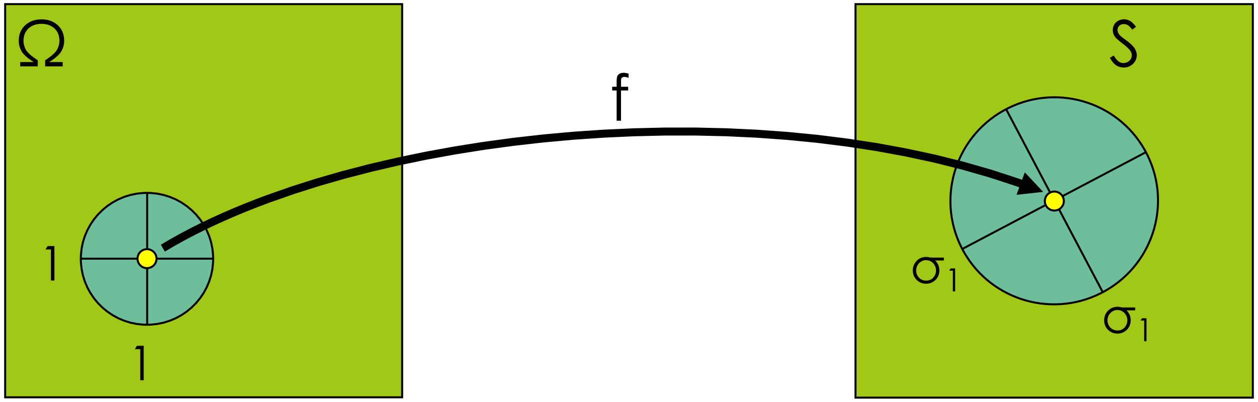
- Used to smoothly interpolate scalar values over a mesh given some sparse constraint

- Useful to interpolate deformations



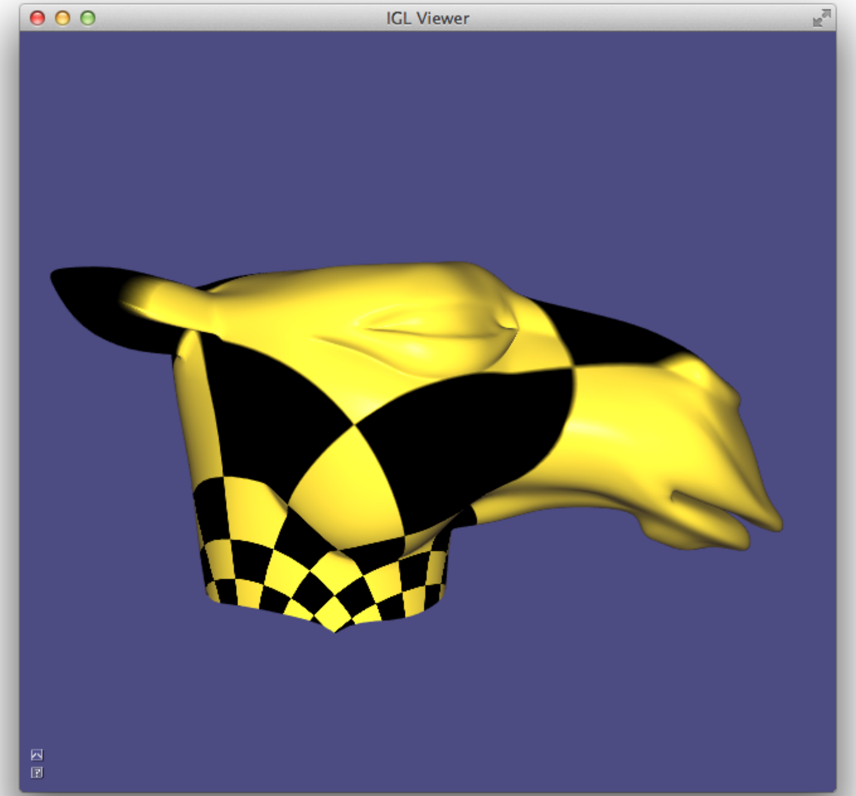
Least Squares Conformal maps

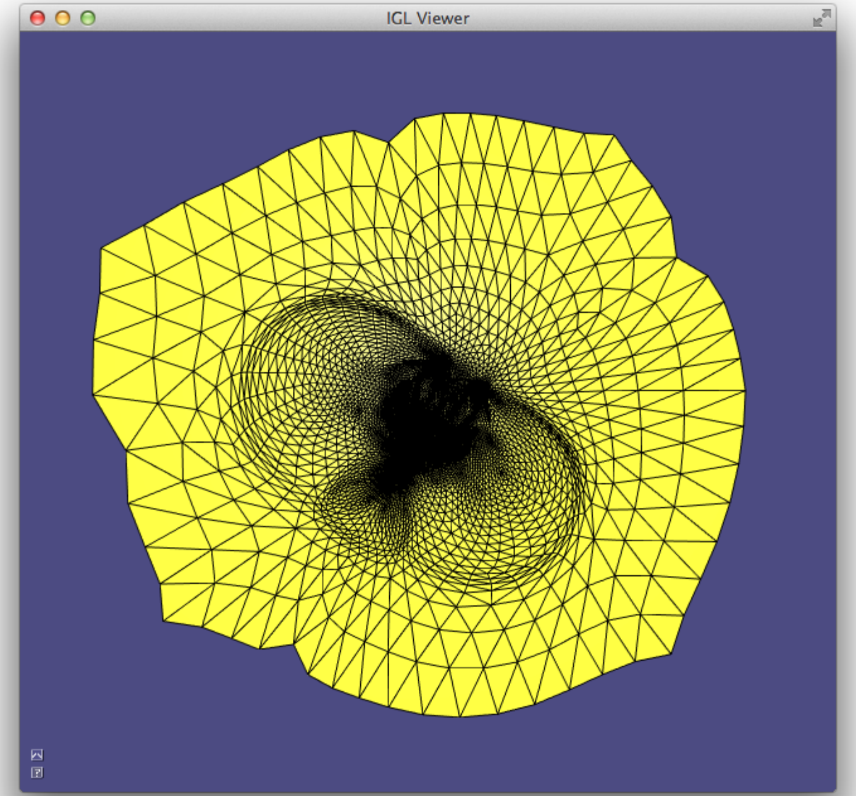
- Doesn't need the entire boundary to be fixed
- Imposing that two vectors on UV maps to 2 orthogonal, same length vectors in 3D.



Least Squares Conformal maps

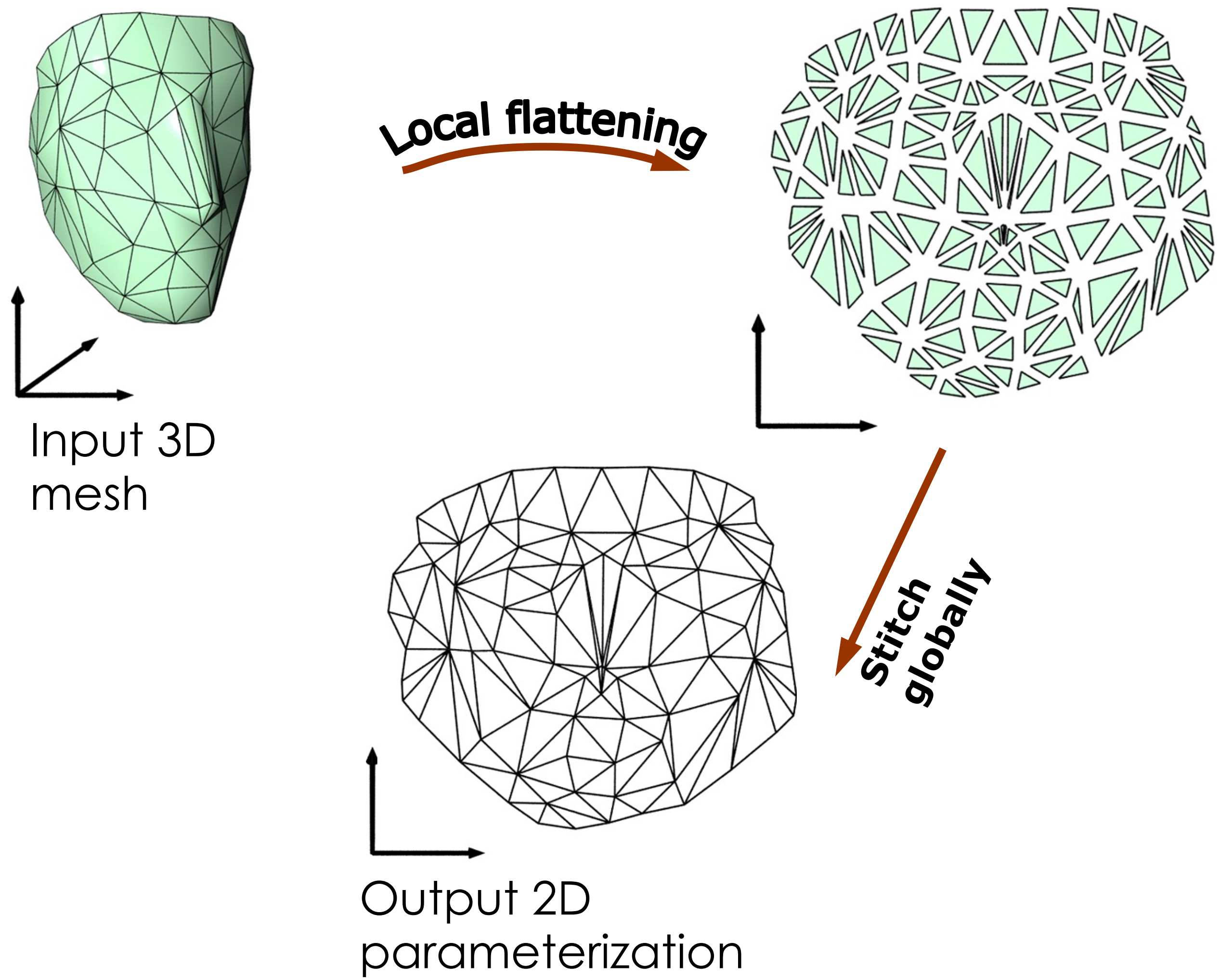
- Need to fix only 2 vertices to disambiguate
- Why?





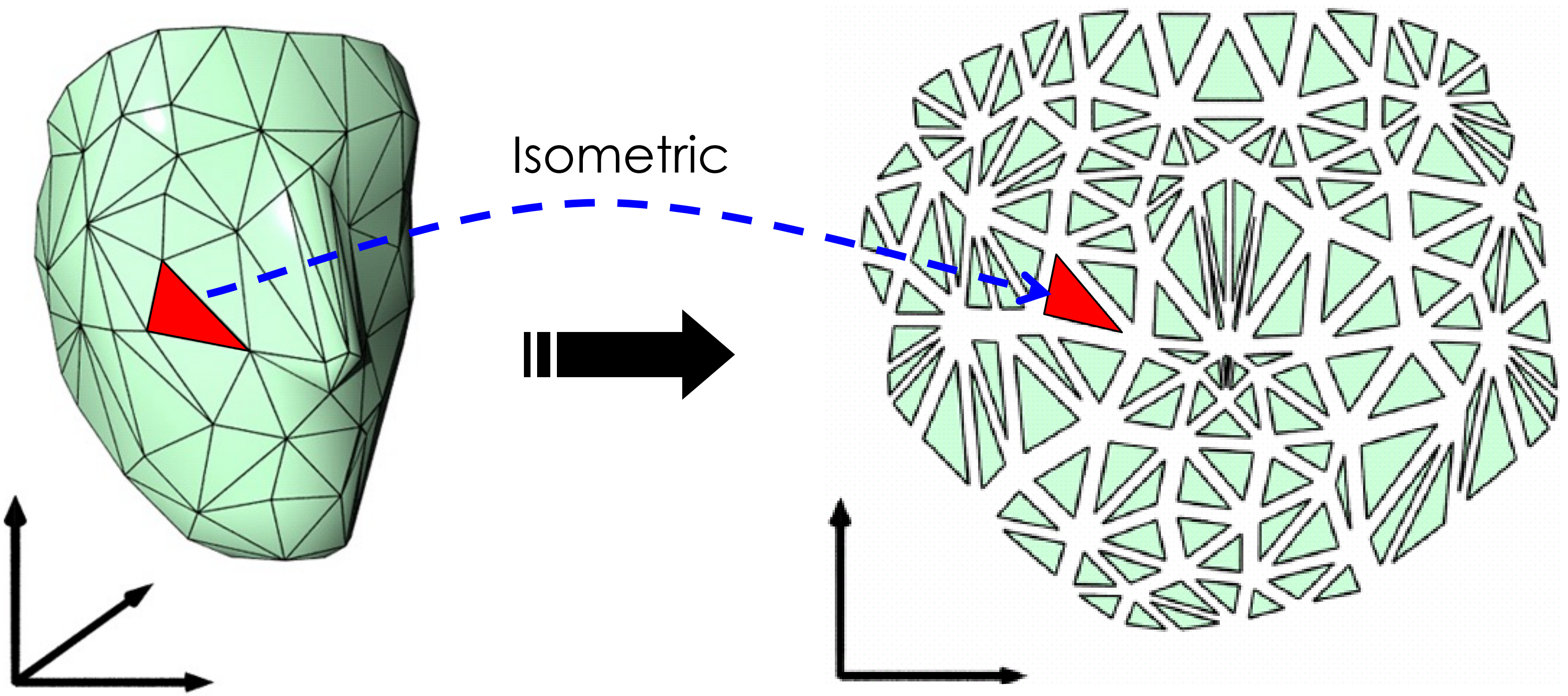
As-rigid-as-possible parametrization (0)

Local-Global Approach



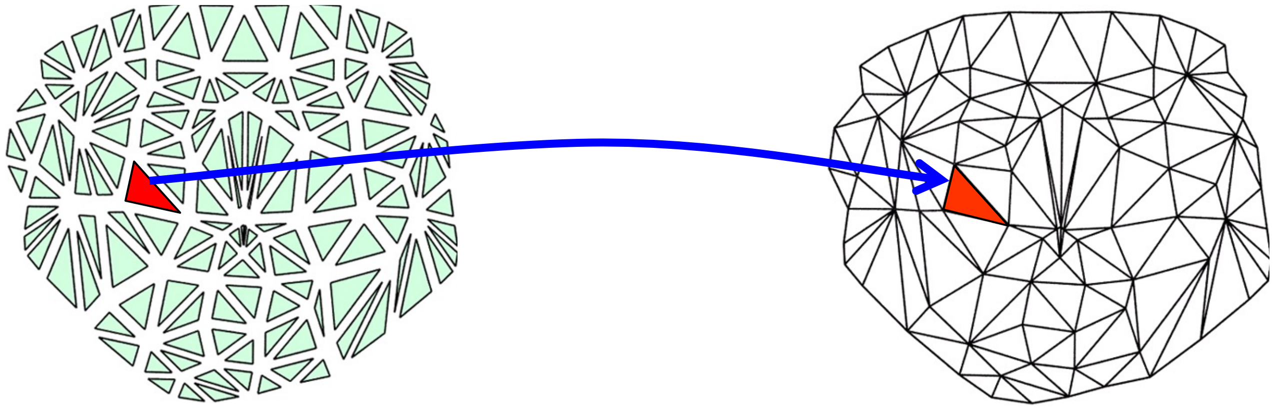
As-rigid-as-possible parametrization (1)

- Each individual triangle is independently flattened into plane without any distortion



As-rigid-as-possible parametrization (1)

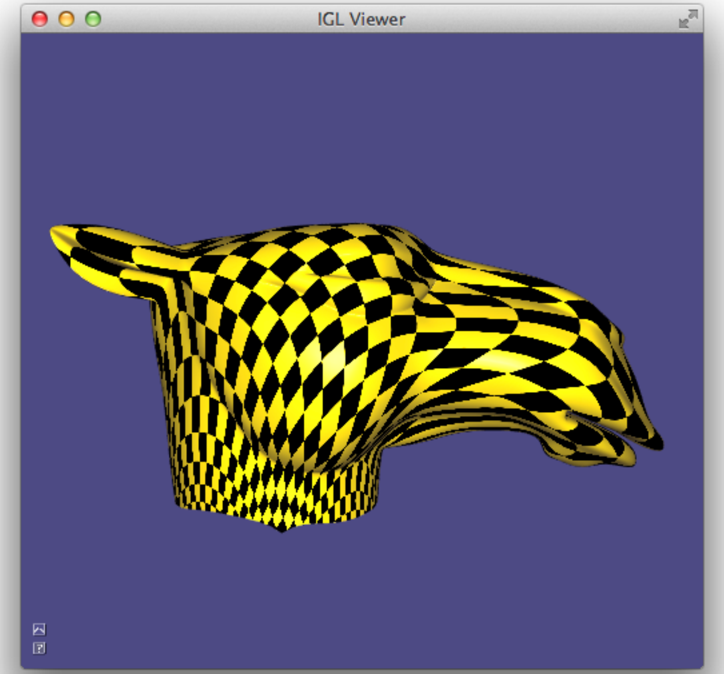
- Merge in UV space (averaging or more sophisticated strategied)

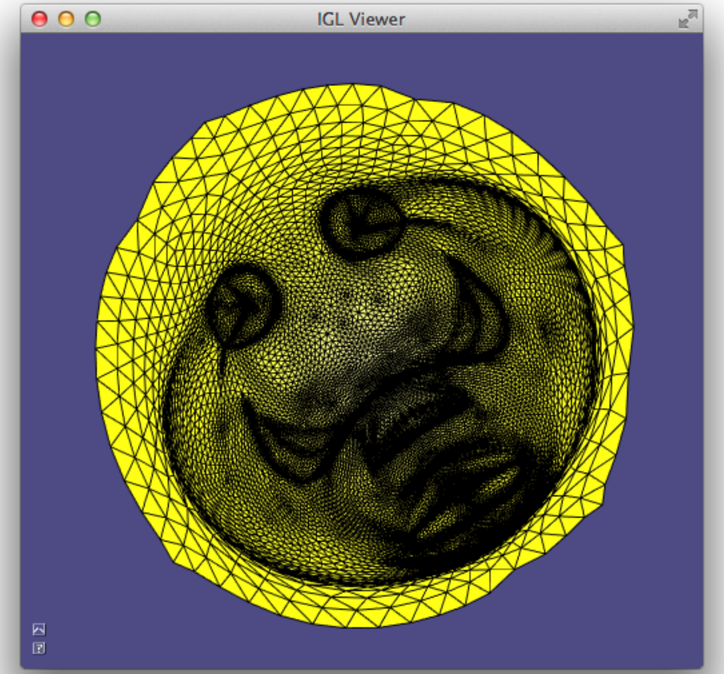


Reference triangles x　　　　　　　　Parameterization *u*

As-rigid-as-possible parametrization (1)

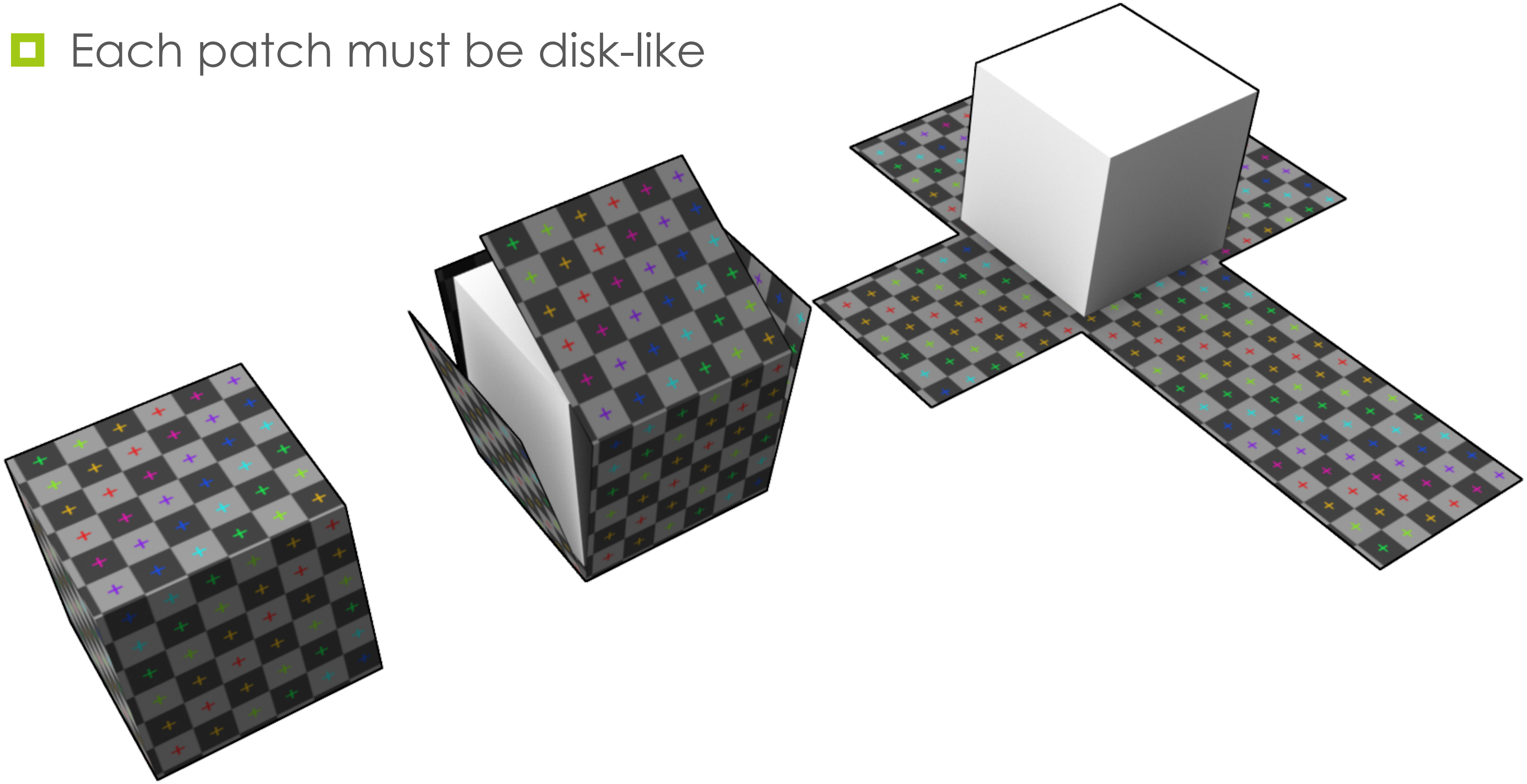
- Warning: it does not guarantee injectivity…





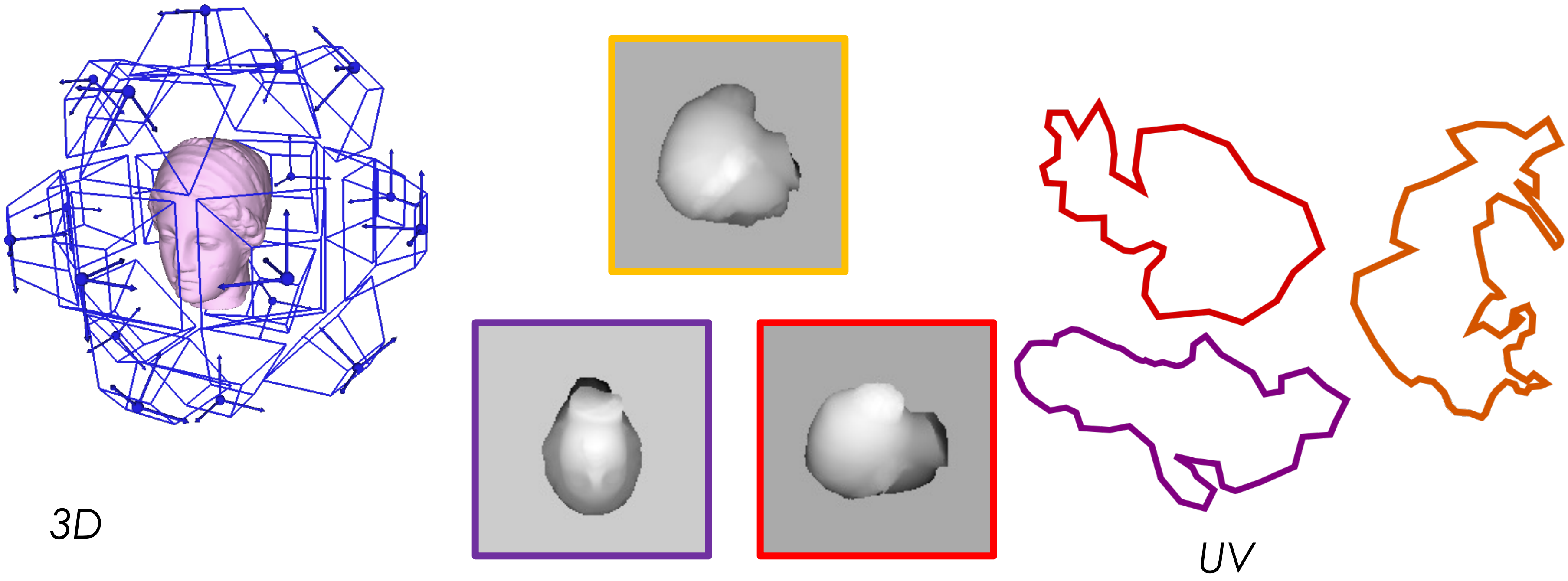
Deriving Cuts

- Splitting the mesh in sub-partitions
- Each patch must be disk-like



Orthoprojection (0)

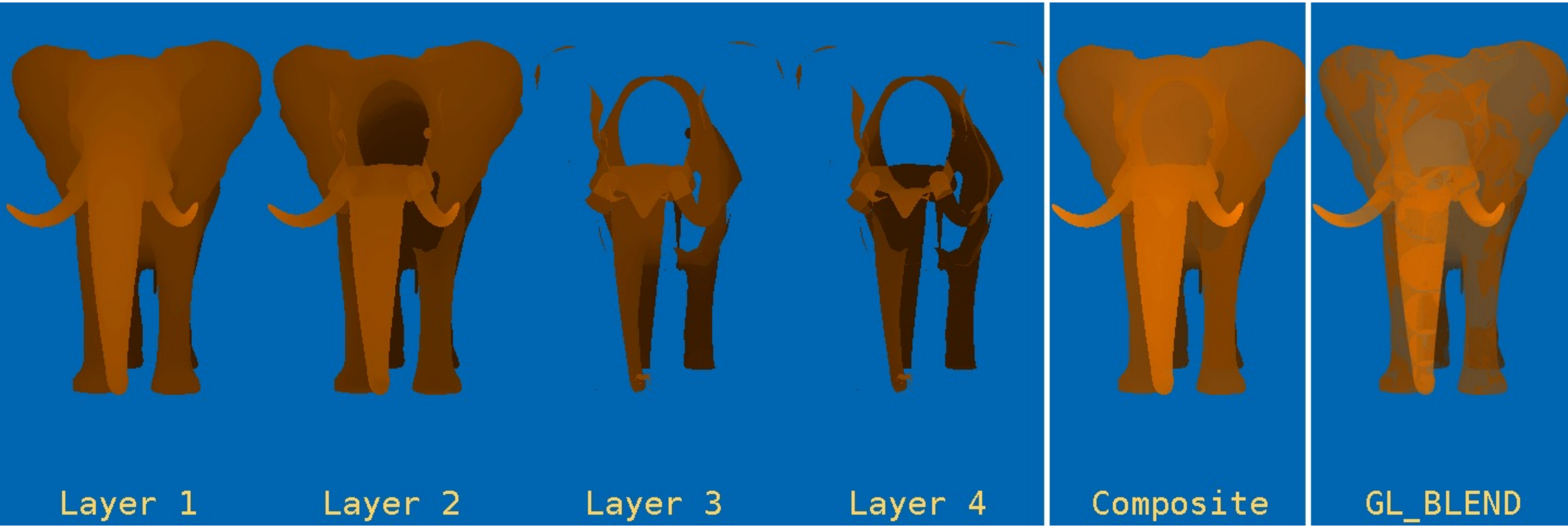
- Use orthographics Projection from multiple directions
- Map each triangle in the "best projection"
- Use depth peeling for handling overlapping parts



Depth peeling

- Depth peeling is **a multipass technique to render translucent polygonal geometry without sorting polygons**.
  (zbuffer and transparency do not work well together)

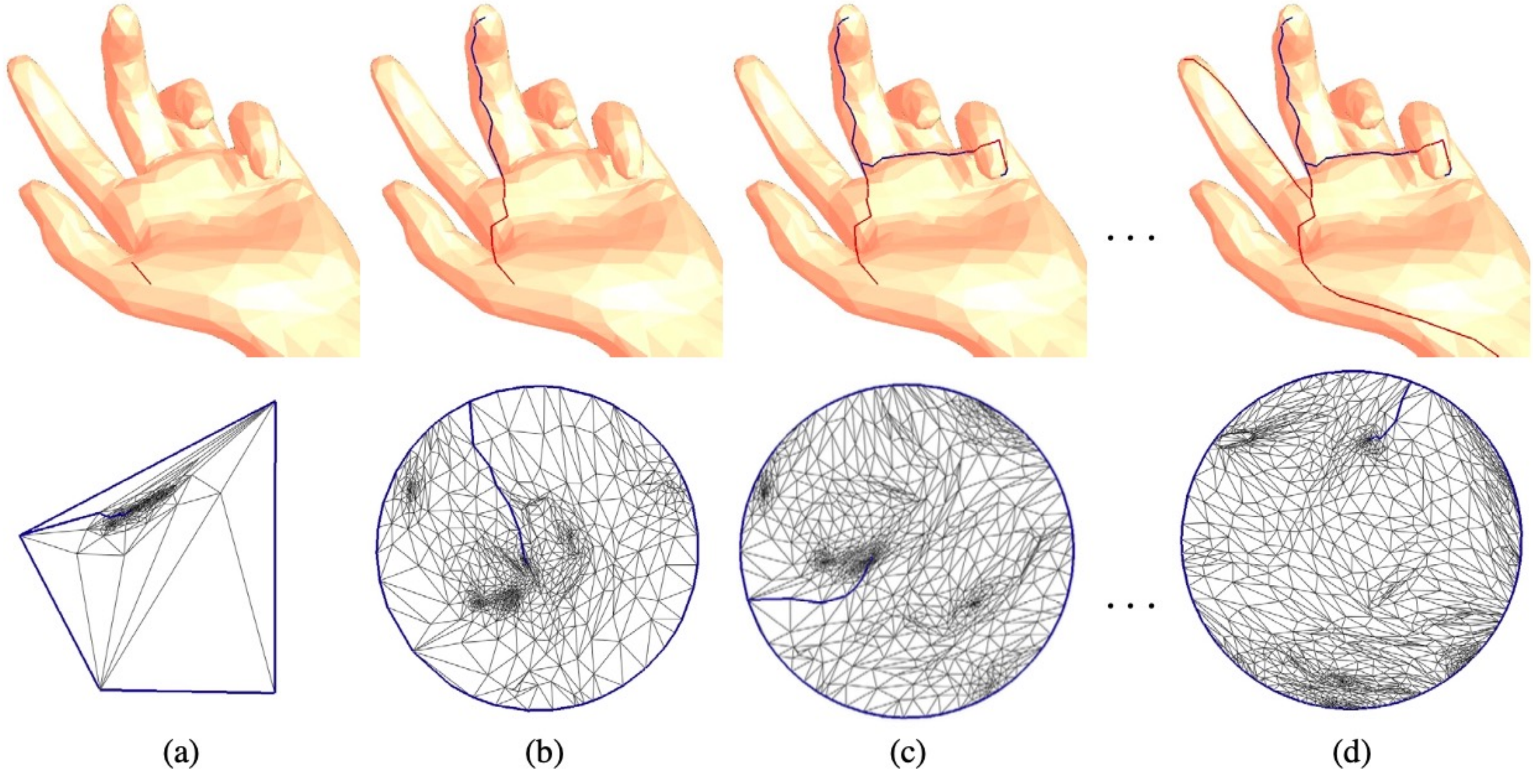
- The idea is to to peel geometry from front to back until there is no more geometry to render.



Orthoprojection (1)

- Small isolated pieces are removed and merged with bigger areas, to avoid fragmentation
- Useful for Color-to-Geometry mapping
- If you have a set of photos aligned over a 3D object they induce a direct parametrization by simply assigning each triangle to the best photo

Growing Cuts



Find the shortest path from the point with the highest distortion to the boundary.
Iterate.