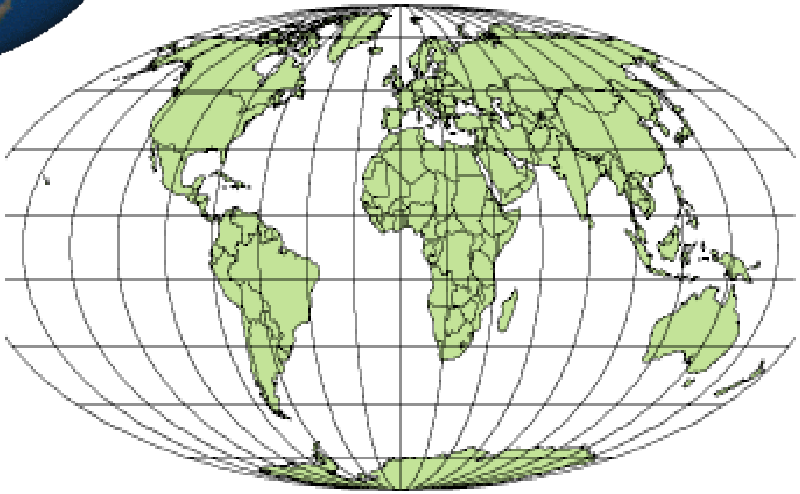
Parametrization & Remeshing

Nico Pietroni (nico.pietroni@isti.cnr.it)

ISTI – CNR

What is a parametrization?



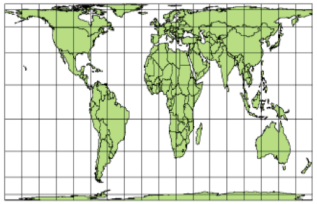
What is a parametrization?

Mollweide-Projektion

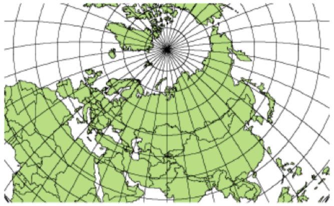
Mercator-Projektion

Zylinderprojektion nach Miller

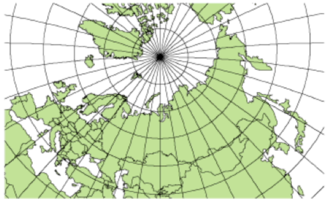
Hammer-Aitoff-Projektion



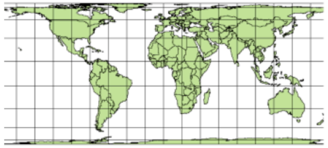
Peters-Projektion



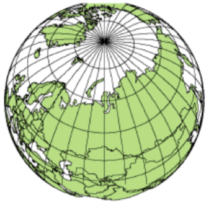
Längentreue Azimuthalprojektion



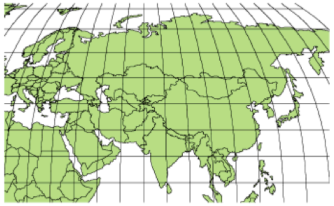
Stereographische Projektion



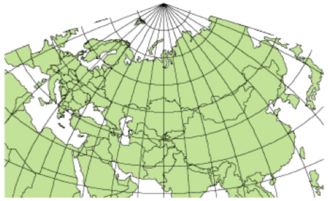
Behrmann-Projektion



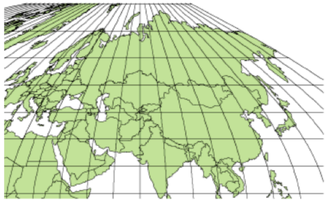
Senkrechte Umgebungsperspektive



Robinson-Projektion

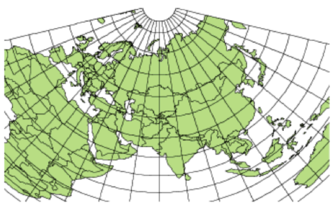


Hotine Oblique Mercator-Projektion

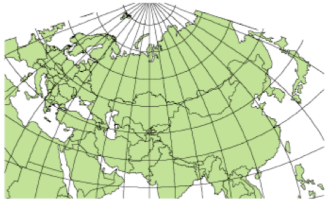


Sinusoidale Projektion

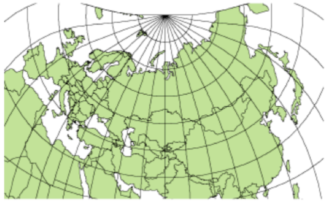
Gnomonische Projektion



Flächentreue Kegelprojektion



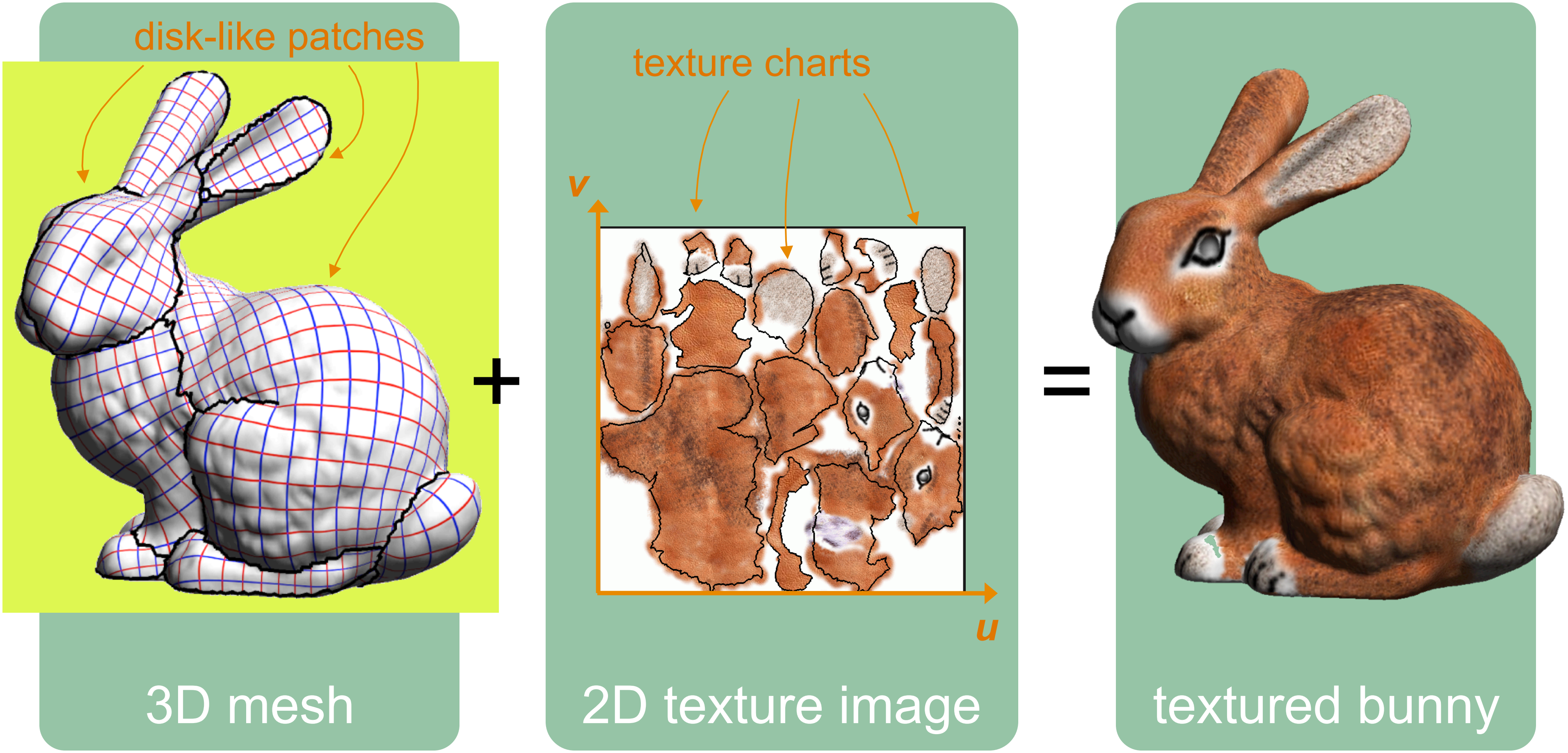
Transverse Mercator-Projektion



Cassini-Soldner-Projektion

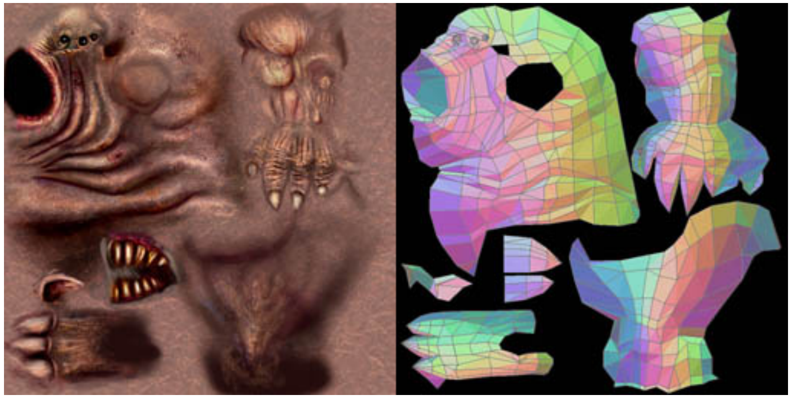
Why Parametrization?

- Texture Mapping



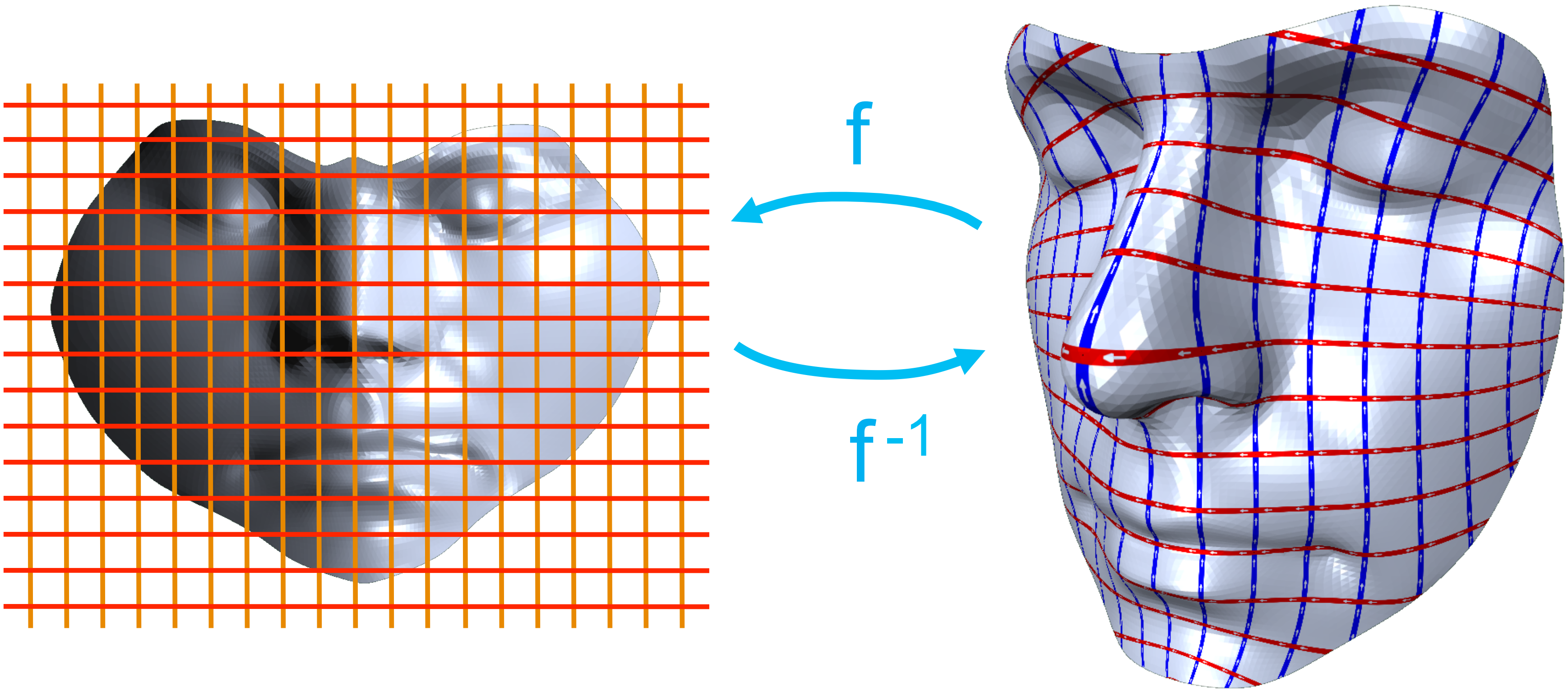
Why Parametrization?

- Manual UV mapping
- An advanced artistic skill



Why Parametrization?

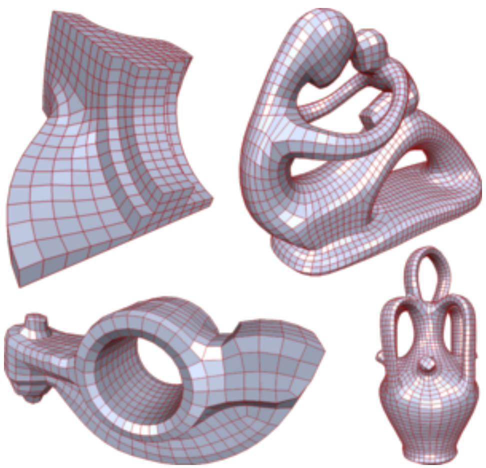
- Remeshing



Why Parametrization?

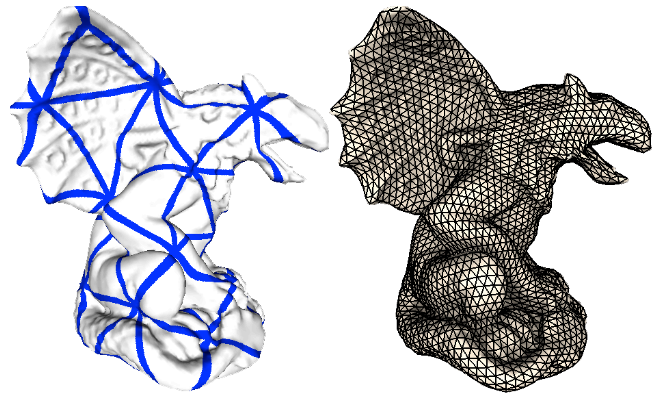
- Remeshing

QUADRILATERAL



Bommes, et AL.: *Mixed Integer Quadrangulation*

TRIANGULAR

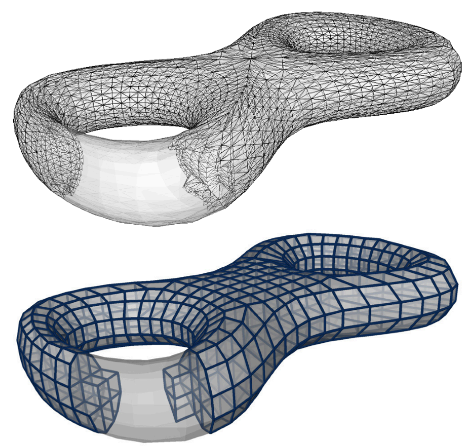


Pietorni, et AL. :Almost isometric mesh parameterization through abstract domains

HEXAGONAL

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

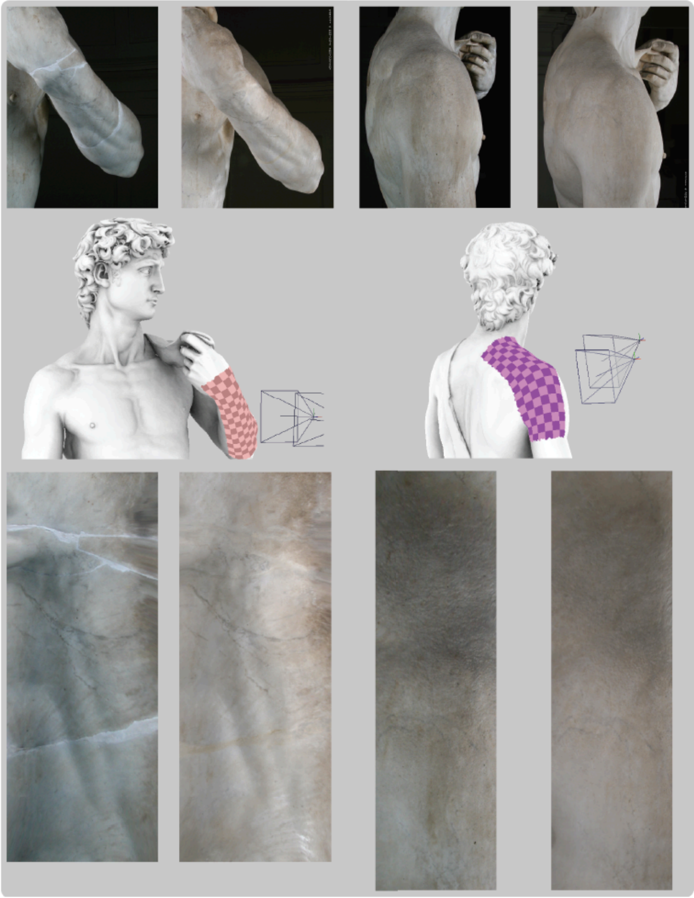
HEXAHEDRAL



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

Why Parametrization?

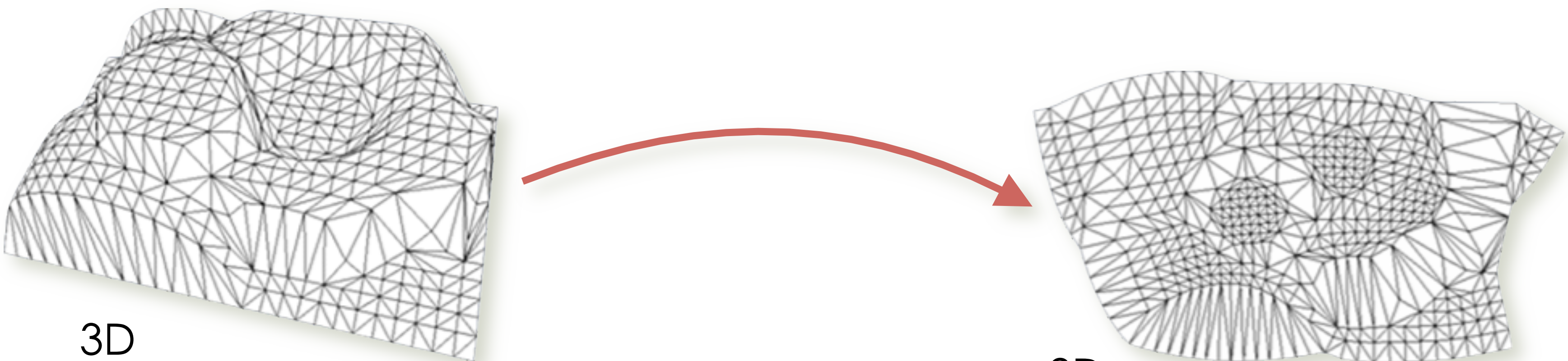
- Analysys…. 2D is easier than 3D



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

Parametrization: what we need?

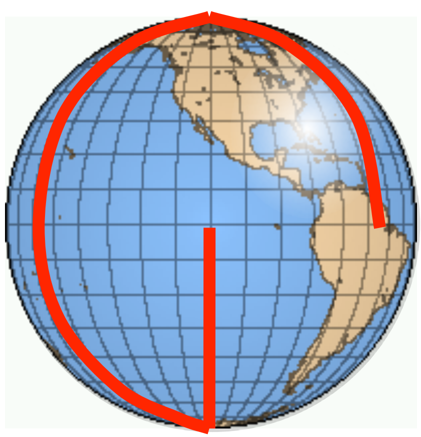
- A strategy to flatten a 3D surface on 2D domain
  - Introducing as few distortion as possible



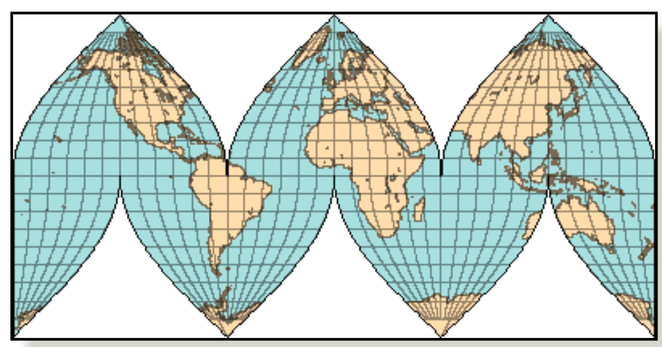
3D

2D

- A strategy to introcuce cuts



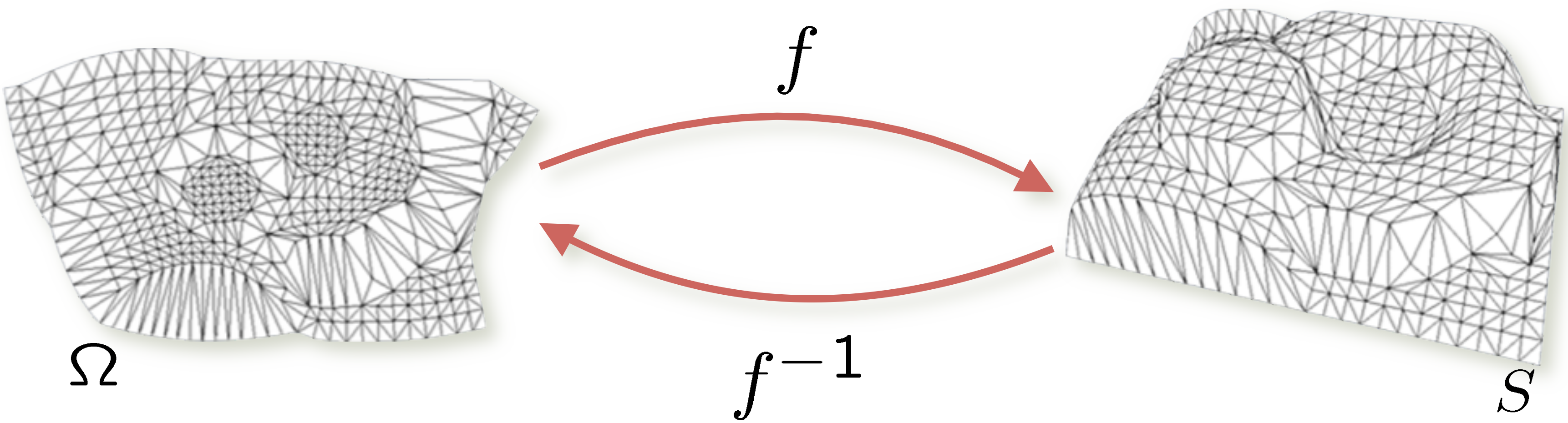
3D



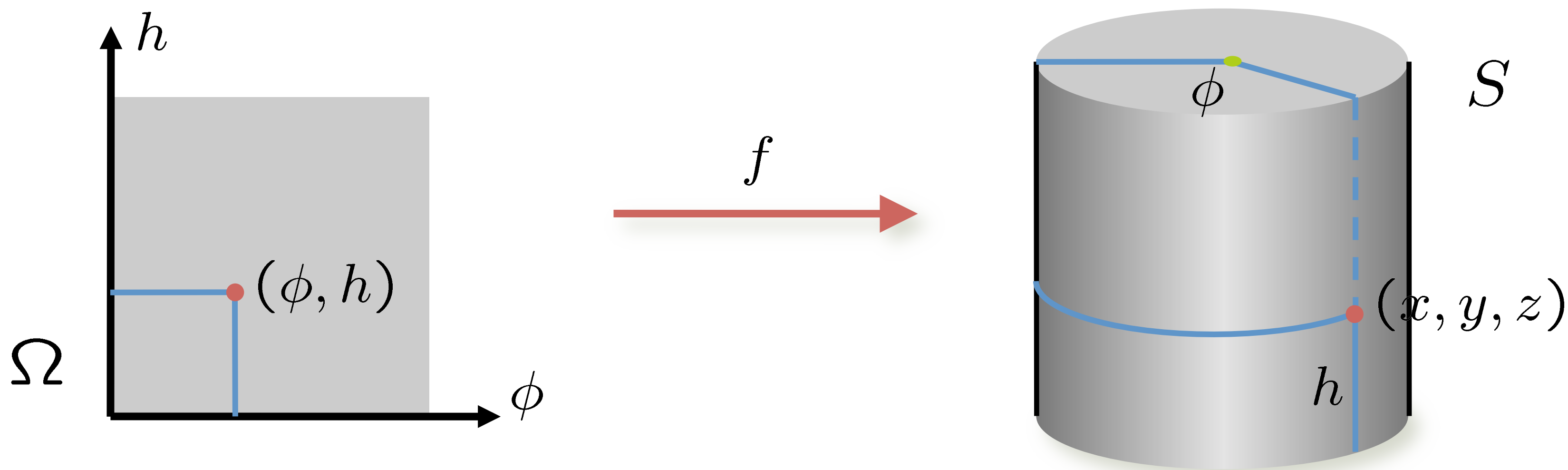
2D

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

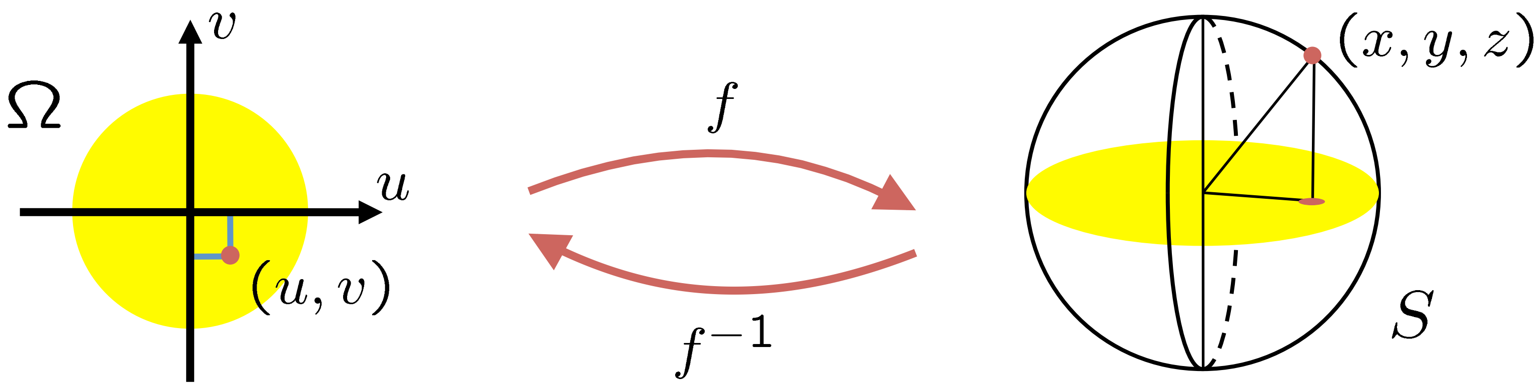


$$S = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 = 1, z \in [0, 1]\}$$

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

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

Parametrization: Ortho Projection



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

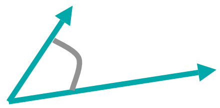
$$\Omega = \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}$$

$$f^{-1}(x, y, z) = (x, y)$$

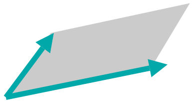
$$f(u, v) = (u, v, \sqrt{1 - u^2 - v^2})$$

Minimize Distortion

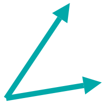
- Angle preservation: conformal



- Area preservation: equiareal

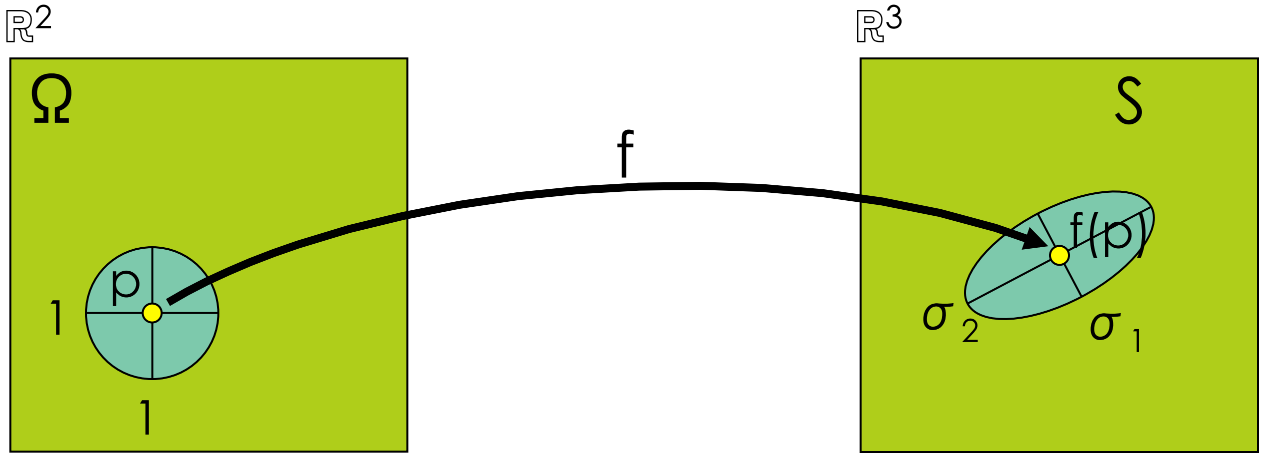


- Area and Angle: Isometric



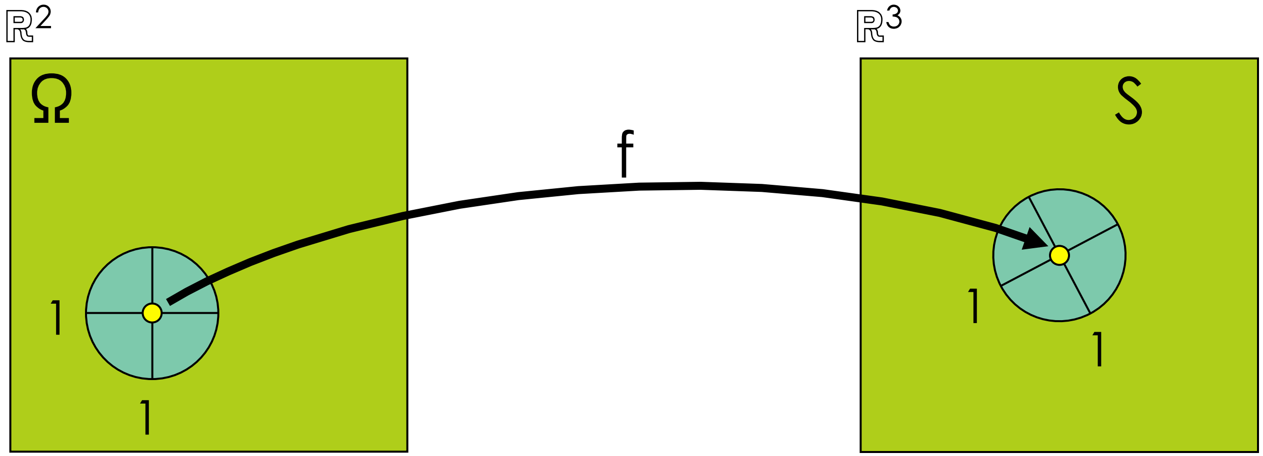
Distortion

- σ_1 and σ_2 describe local deformations



Isometric Mapping

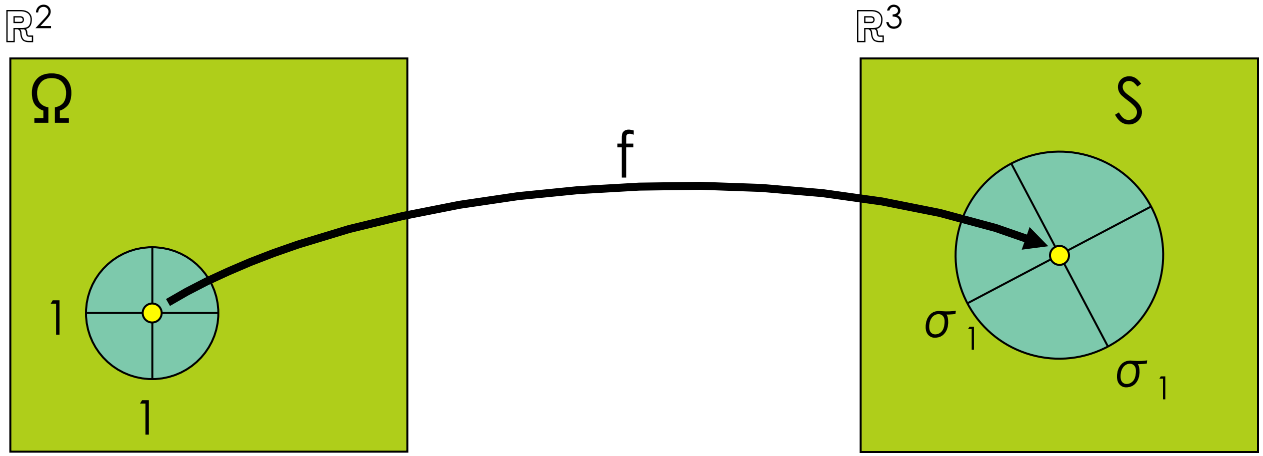
- $\sigma_1 = \sigma_2 = 1$



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

Conformal Mapping

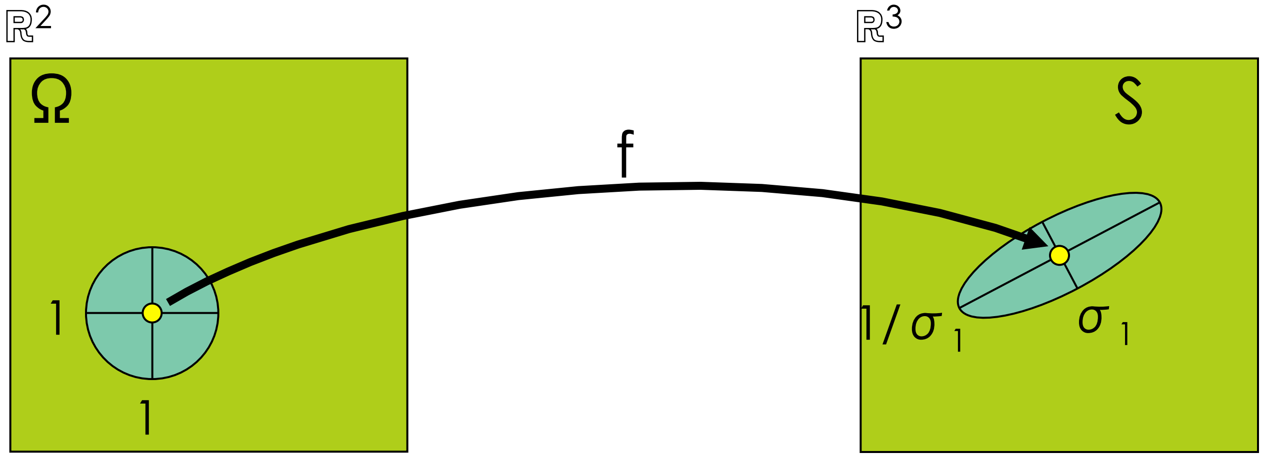
- $\sigma_1 / \sigma_2 = 1$



- preserves **angles**

Equiareal Mapping

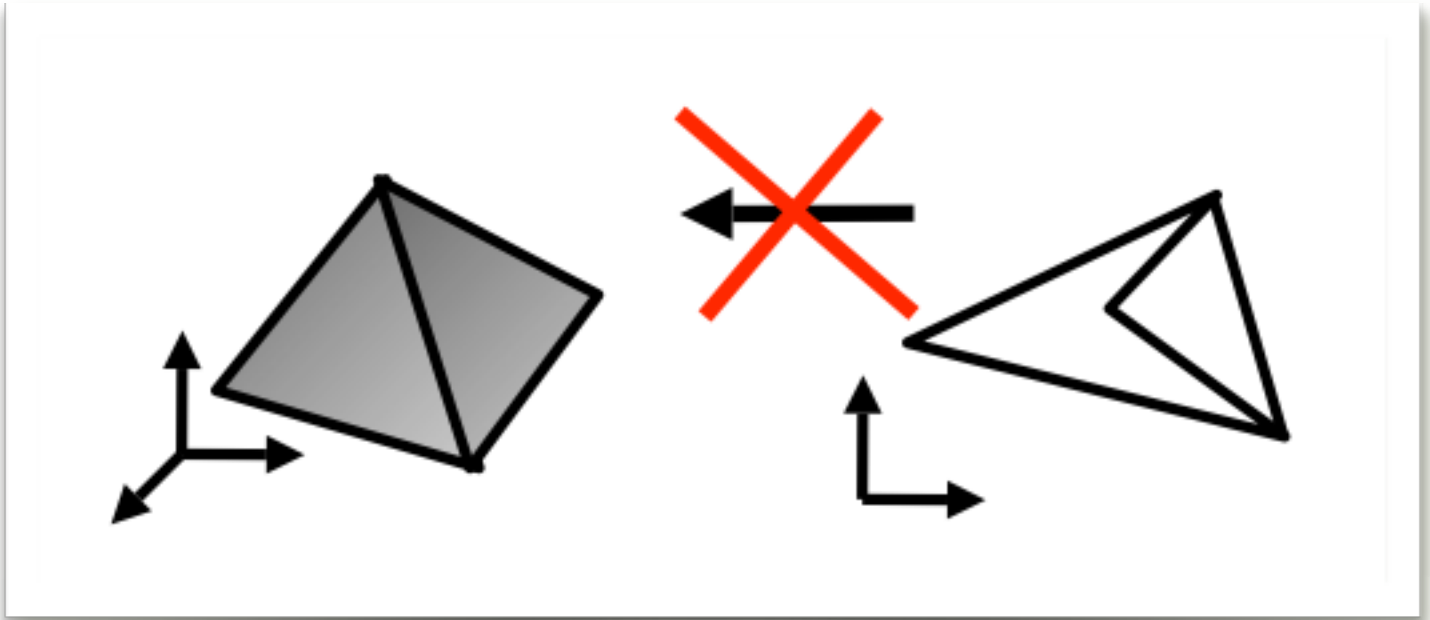
- $\sigma_1 \cdot \sigma_2 = 1$



- preserves **areas**

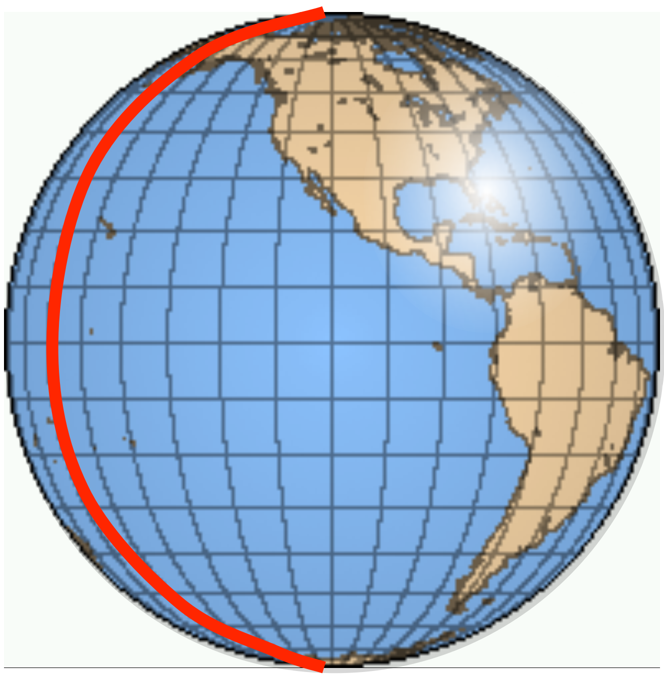
Bijectivity

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

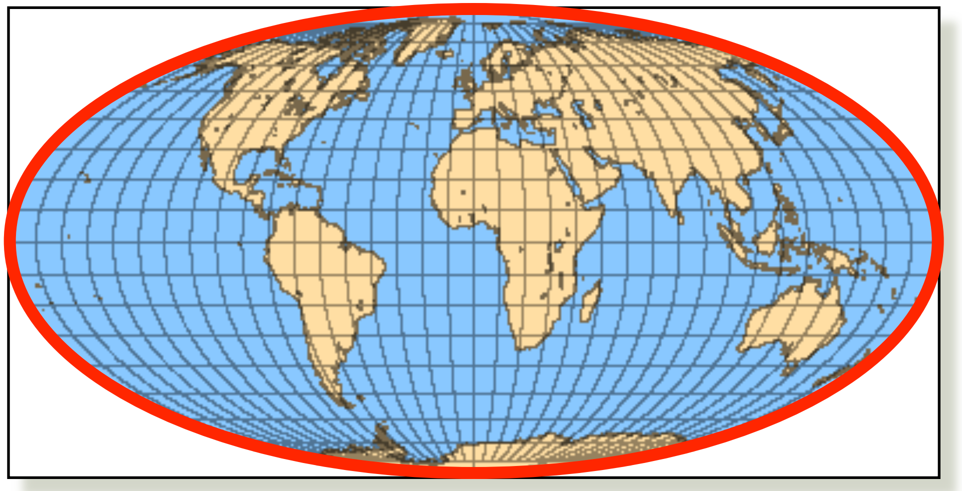


Cuts 1

- Clearly needed for closed surfaces



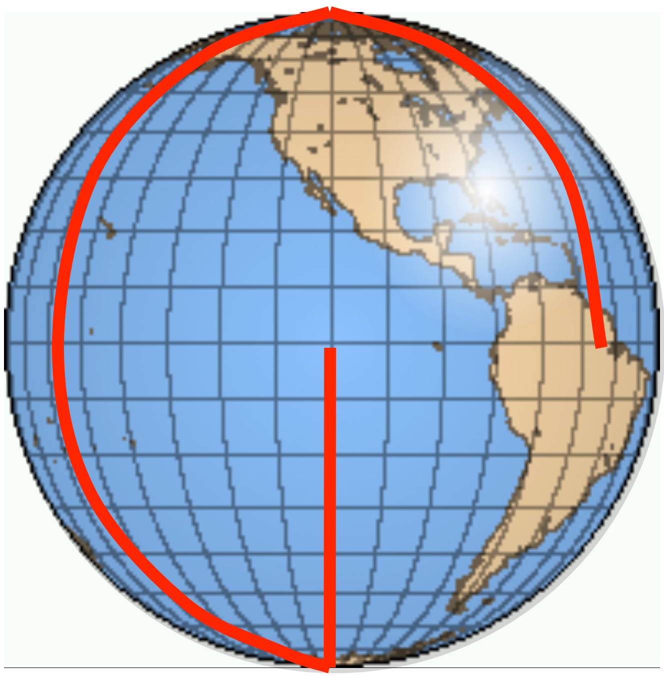
sphere in 3D



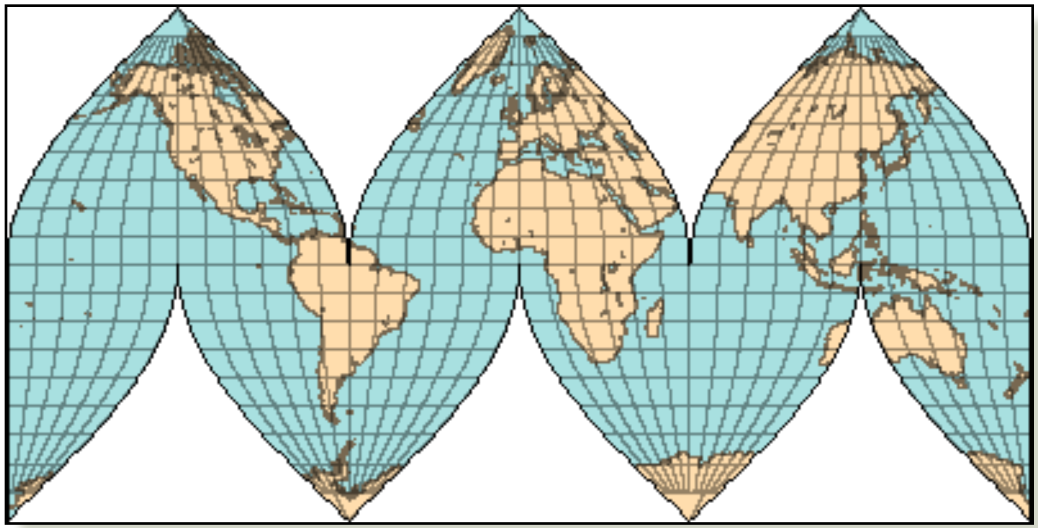
2D surface disk

Cuts 2

- Usually more cuts -> less distortion



sphere in 3D



2D surface

Cuts 3: closed surfaces

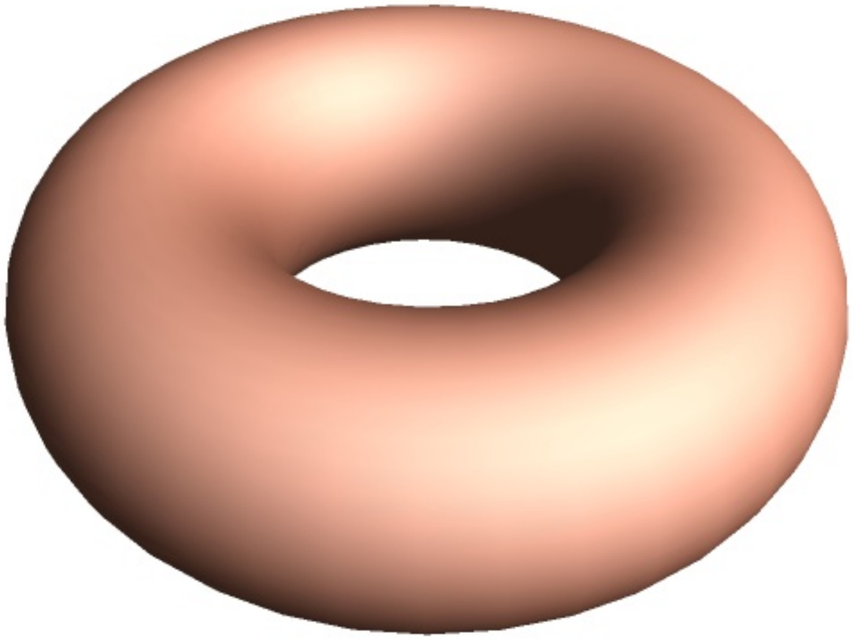
- How many cuts?

for a genus 0 surface ?

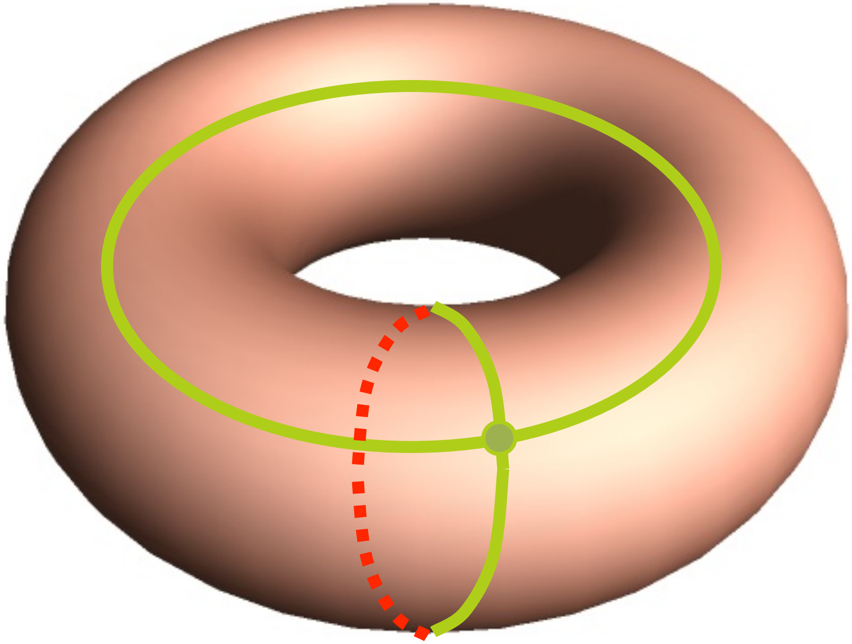
any tree of cuts

Cuts 3: closed surfaces

- How many cuts?



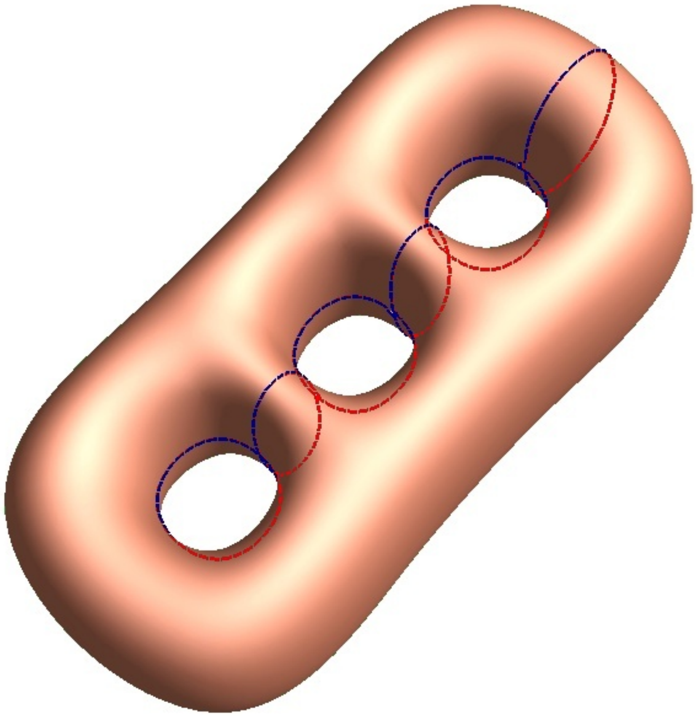
for a genus 1 surface ?



two looped cuts

Cuts 3: closed surfaces

- How many cuts?

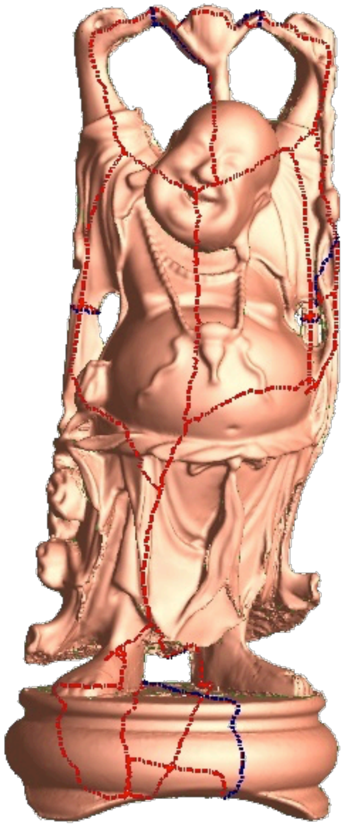


for a genus 3 surface ?

6 looped cuts

Cuts 3: closed surfaces

- How many cuts?



genus 6

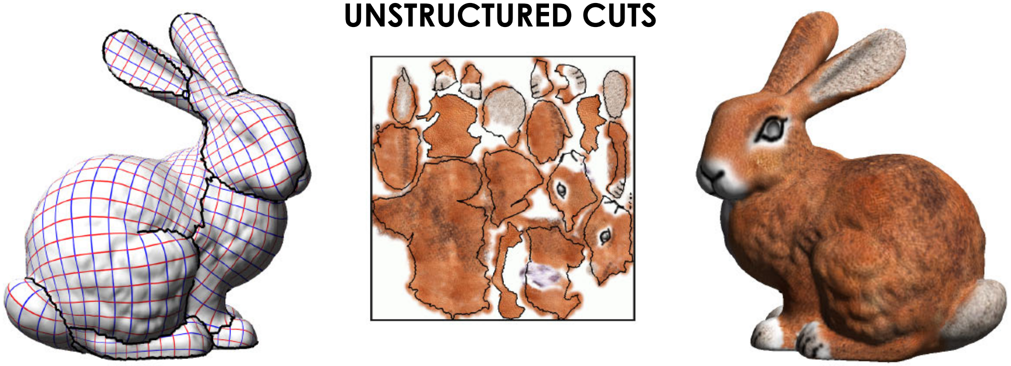
for a genus n surface ?

2n looped cuts

Generic Cut Strategies

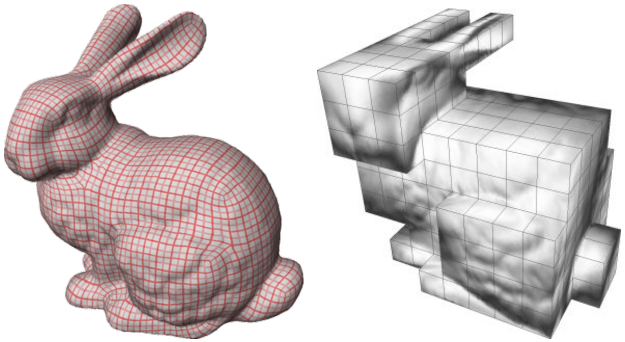
- Texture Mapping

UNSTRUCTURED CUTS



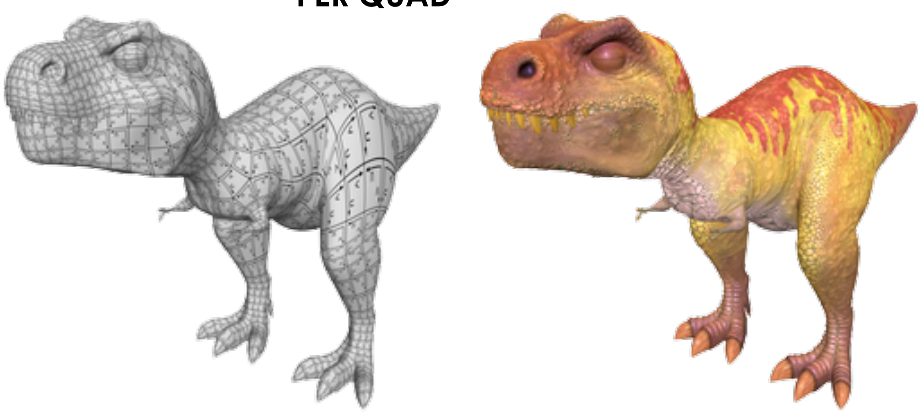
Lévy, et AL.: *Least squares conformal maps for automatic texture atlas generation*

IMPLICIT



Tarini, et AL.: *PolyCube Maps*

PER QUAD



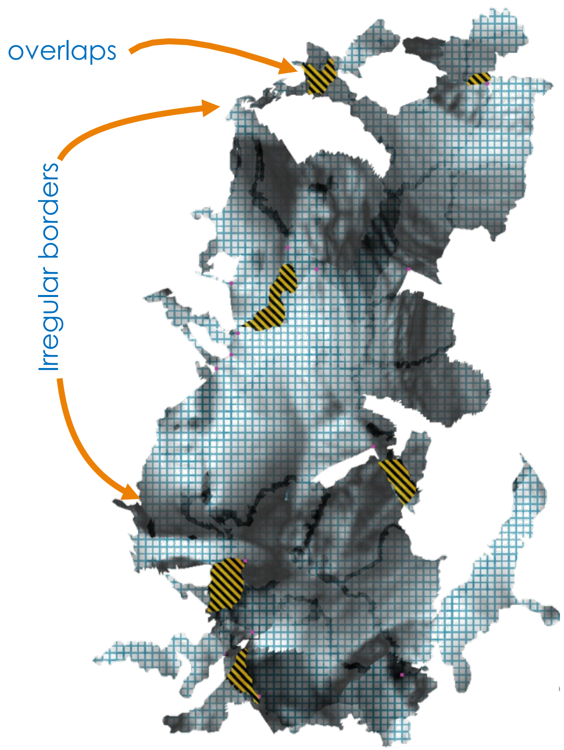
Brent Burley et al : Ptex: Per-Face Texture Mapping for Production Rendering

REGULAR CUTS

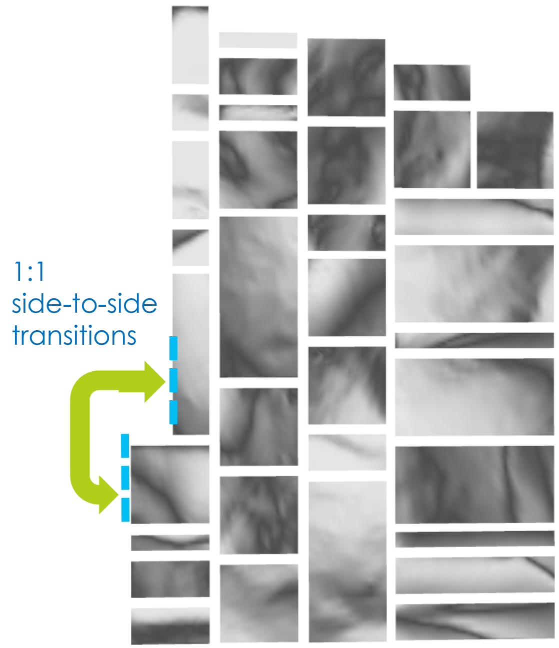
Pietroni, et AL.: *Almost isometric mesh parameterization through abstract domains*

What is a good Cut?

- **Simplicity of parametric domain**



Pietroni, et AL.: *Global Parametrization of Range Image Sets*



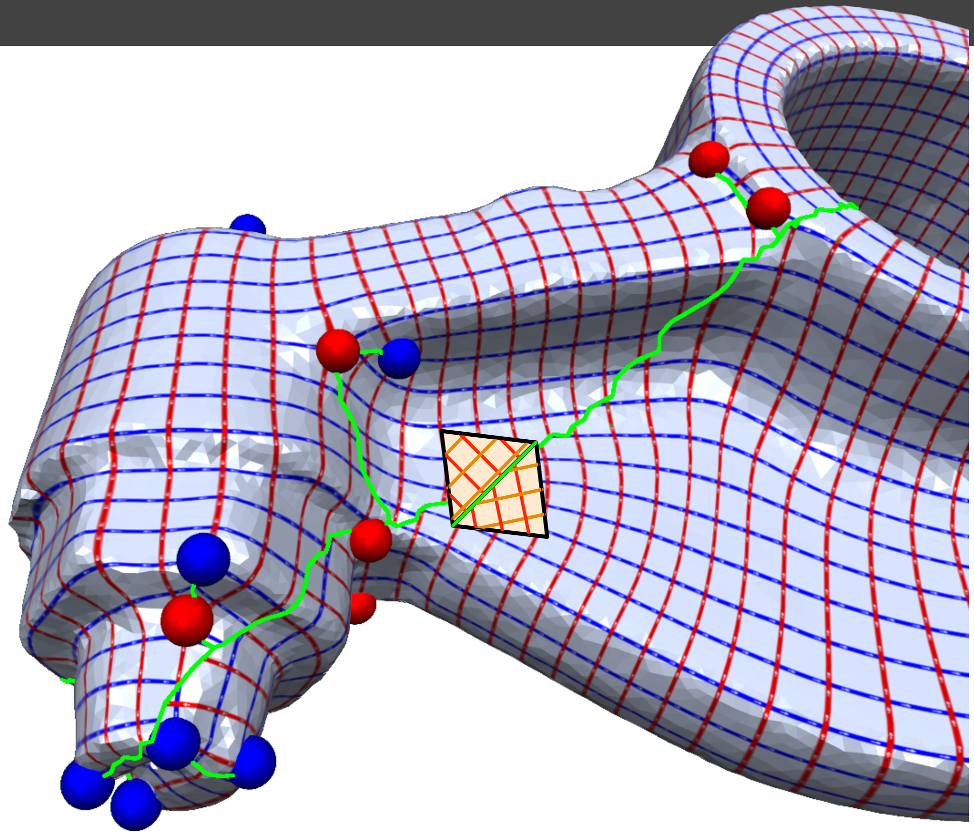
Tarini, et AL.: *Simple Quad Domains for Field Aligned Mesh Parametrization*

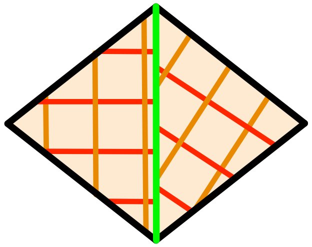
Gu, et AL.: *Goemetry Images*

Versatility ⟷ Simplicity

Globally Smoothess

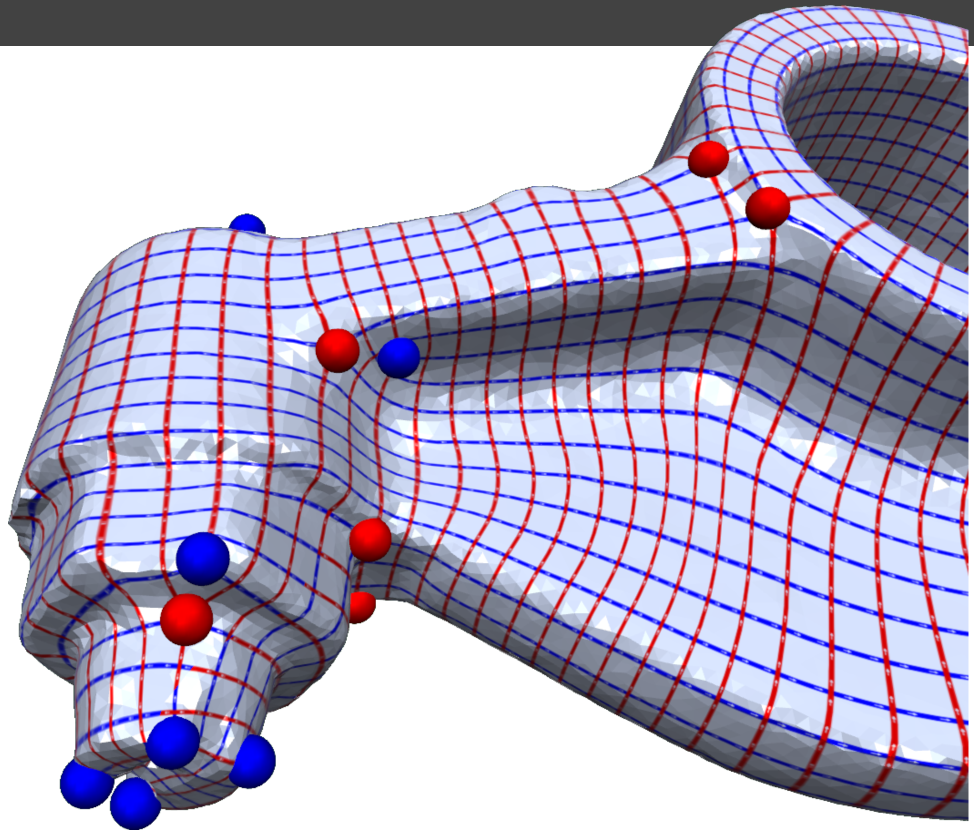
- Tangent directions varyes smoothly across seams

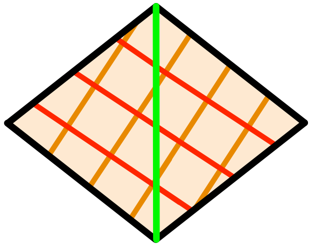




Globally Smoothess

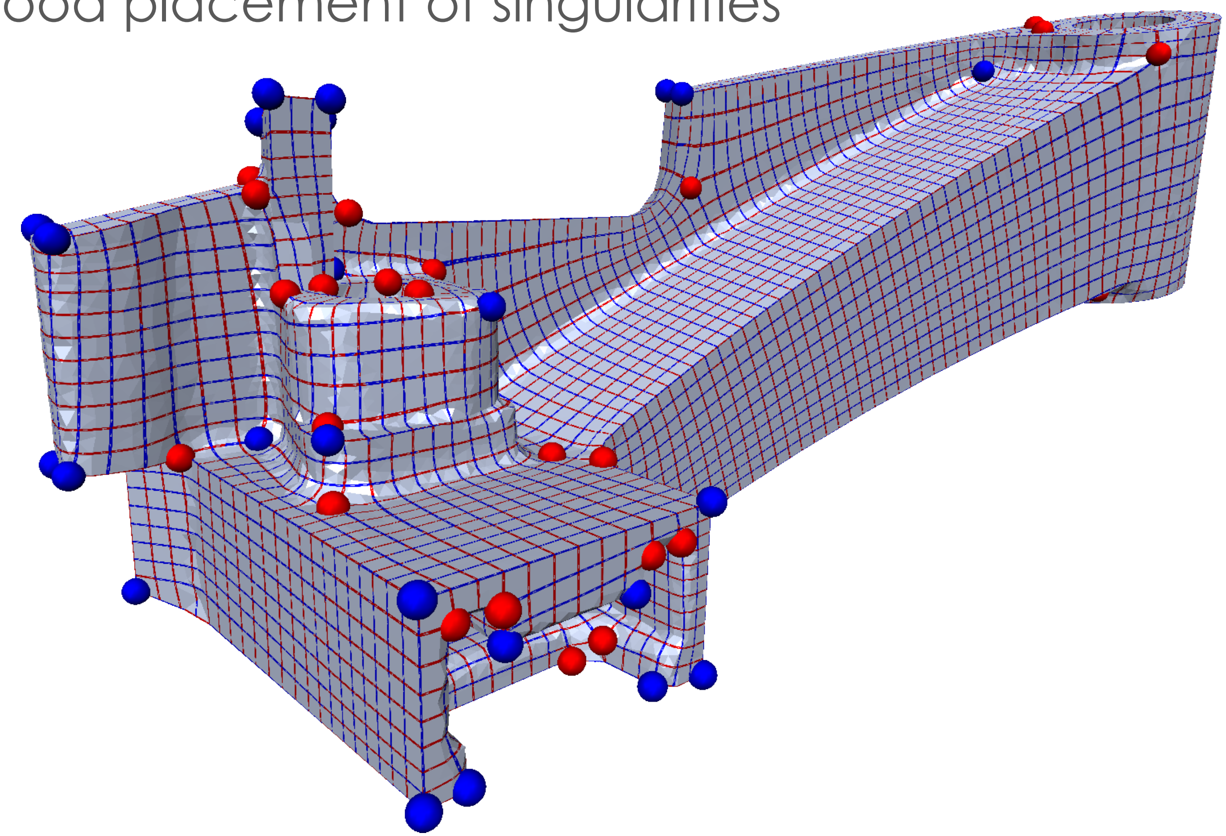
- Tangent directions varyes smoothly across seams





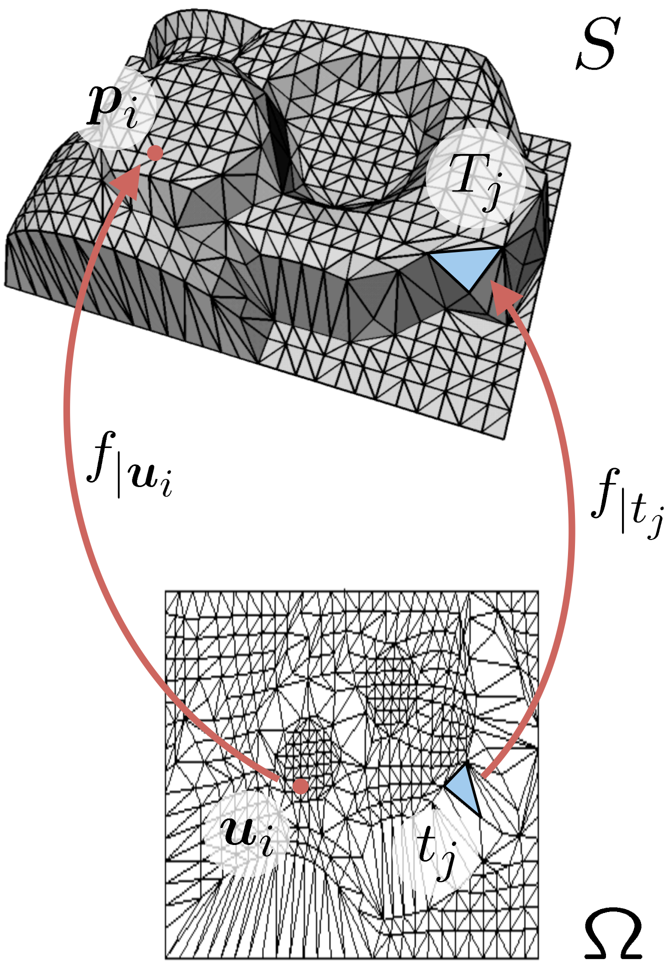
Feature Alignment

- 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 (VCG)

- triangle mesh
  - vertices
  - Triangles

```
MyTrimesh m;
MyTrimesh::VertexType *v=&m.vert[0];
MyTrimesh::CoordType pos=v->P();
```

```
MyTrimesh m;
MyTrimesh::FaceType *f=&m.face[0];
MyTrimesh::VertexType *v=f->V(0);
```

- parameter mesh

```
class MyTriFace;
class MyTriVertex;
struct TriUsedTypes: public vcg::UsedTypes<. . .>;

class MyTriVertex:public vcg::Vertex<TriUsedTypes,…vcg::vertex::TexCoord2d>{};
class MyTriFace:public vcg::Face<TriUsedTypes,…vcg::face::WedgeTexCoord2d>{};

class MyTriMesh: public vcg::tri::TriMesh< std::vector<MyTriVertex>,std::vector<MyTriFace > >
```

  - Per vertex UV

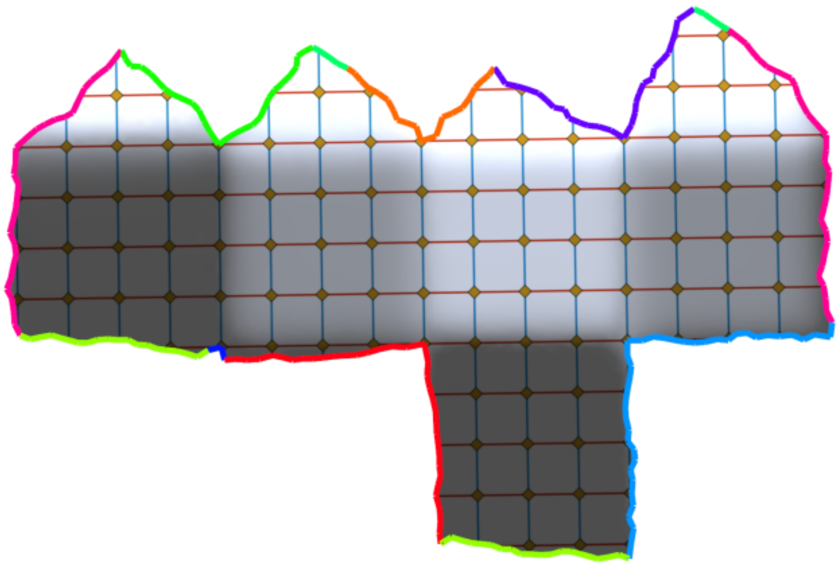
```
MyTrimesh m;
MyTrimesh::VertexType *v=&m.vert[0];
vcg::Point2d uv=v->T().P();
```

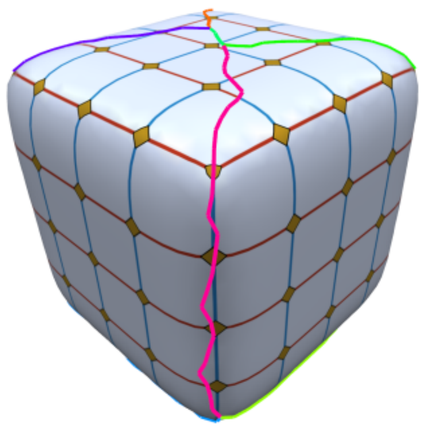
  - Per Wedge UV

```
MyTrimesh m;
MyTrimesh::FaceType *f=&m.face[0];
vcg::Point2d uv=f->T(0).P();
```

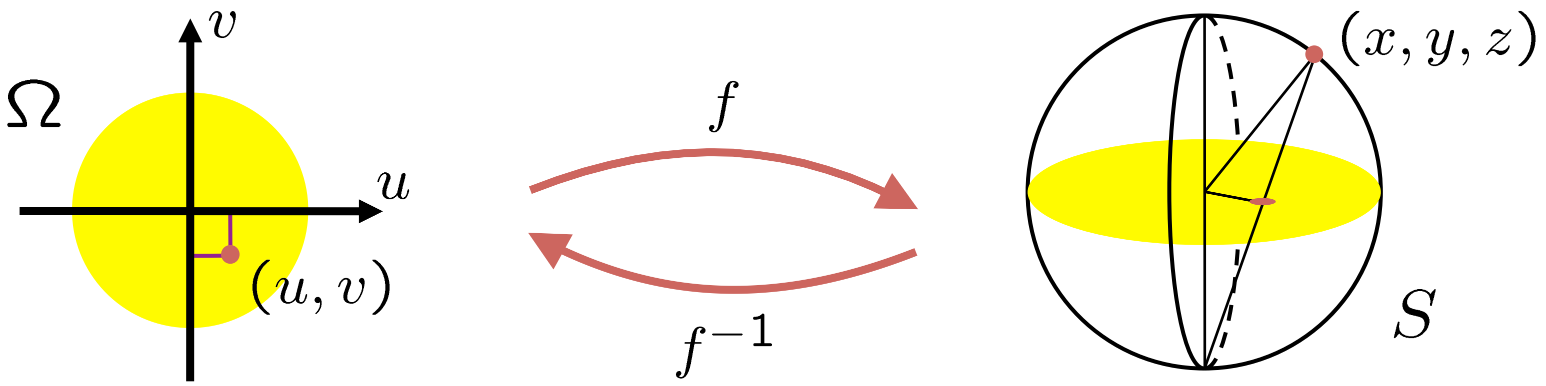
Parametrization (VCG)

- Per Wedge = per triangle per vertex
- Why ?
- Needed to handle seams





Parametrization: Stereo Projection



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

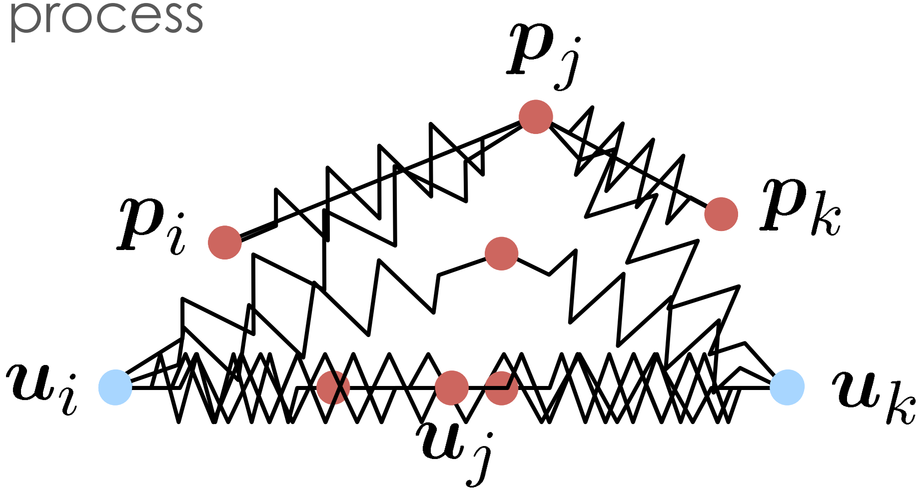
$$\Omega = \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 \leq 1\}$$

$$f^{-1}(x, y, z) = \left(\frac{x}{1+z}, \frac{y}{1+z}\right)$$

$$f(u, v) = \left(\frac{2u}{1+u^2+v^2}, \frac{2v}{1+u^2+v^2}, \frac{1-u^2-v^2}{1+u^2+v^2}\right)$$

Parametrization: Mass-Spring

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



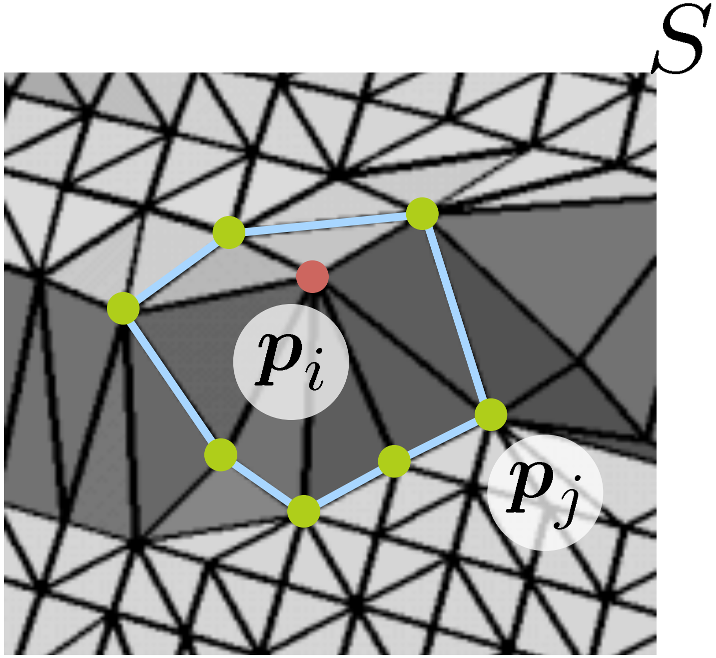
Energy Minimization

- 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

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



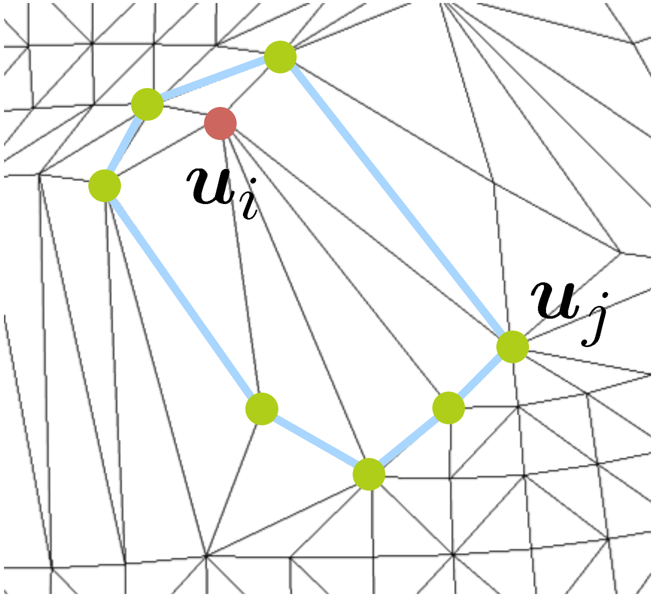
Linear System

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

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

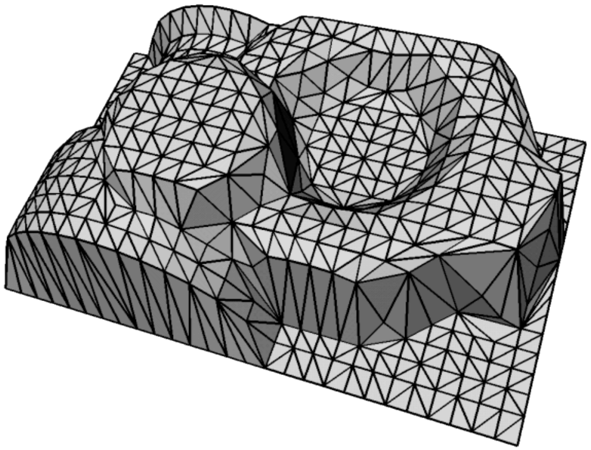
- With weights

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

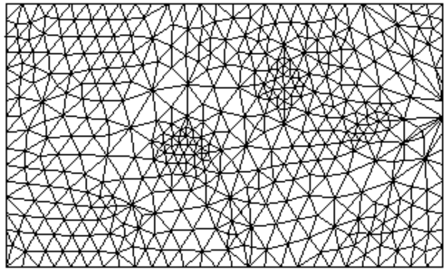


- LEAD to Linear System!

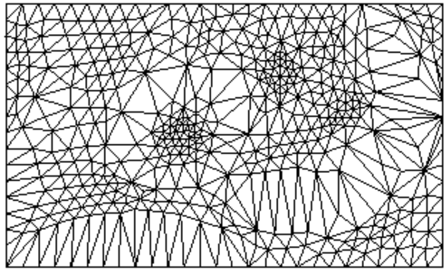
Which Weights?



- uniform spring constants

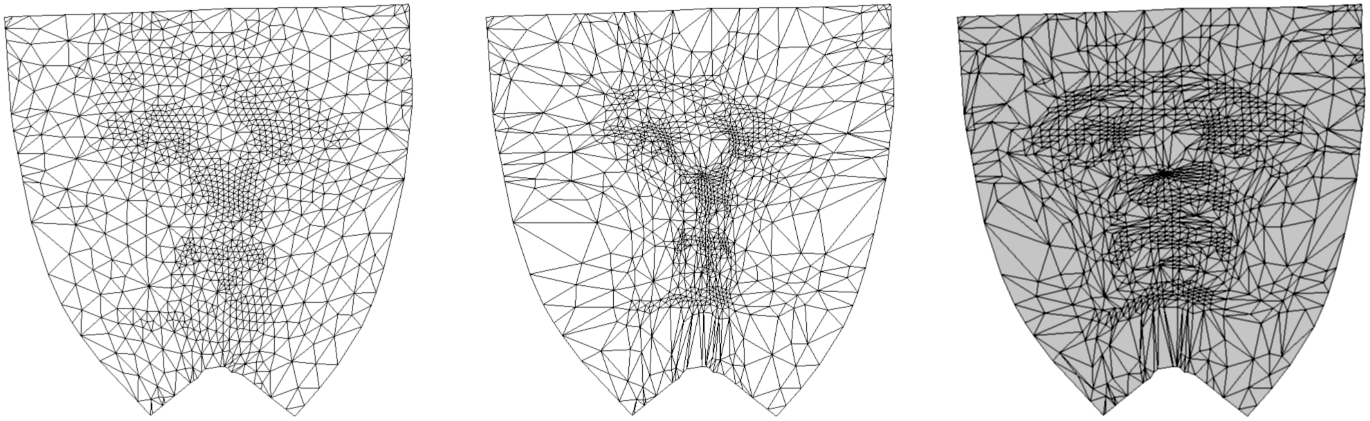


- Proportional to 3D distance



Which Weights?

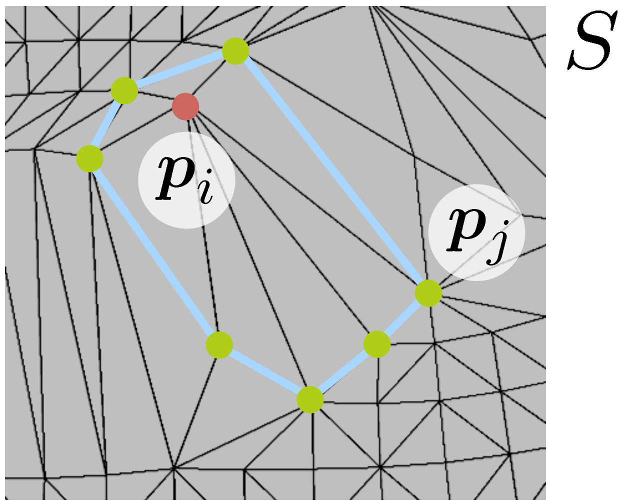
- **NO** linear reproduction
- Planar mesh are distorted



Which Weights?

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

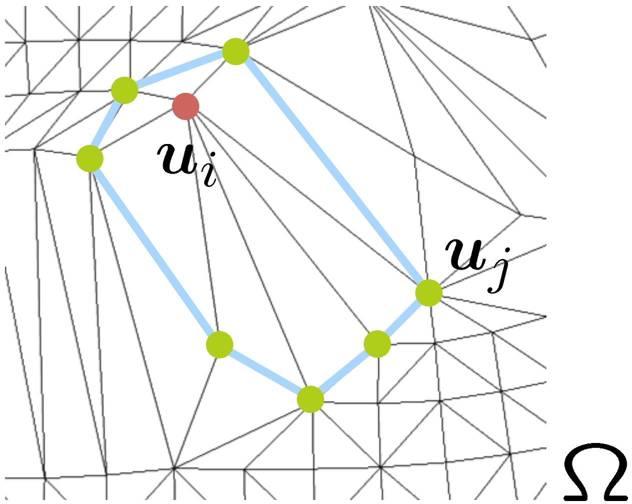
$$\boldsymbol{p}_i = \sum_{j \in N_i} \lambda_{ij} \boldsymbol{p}_j$$



- Then solving

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

- Reproduces **S**



Which Weights?

- Wachspress coordinates

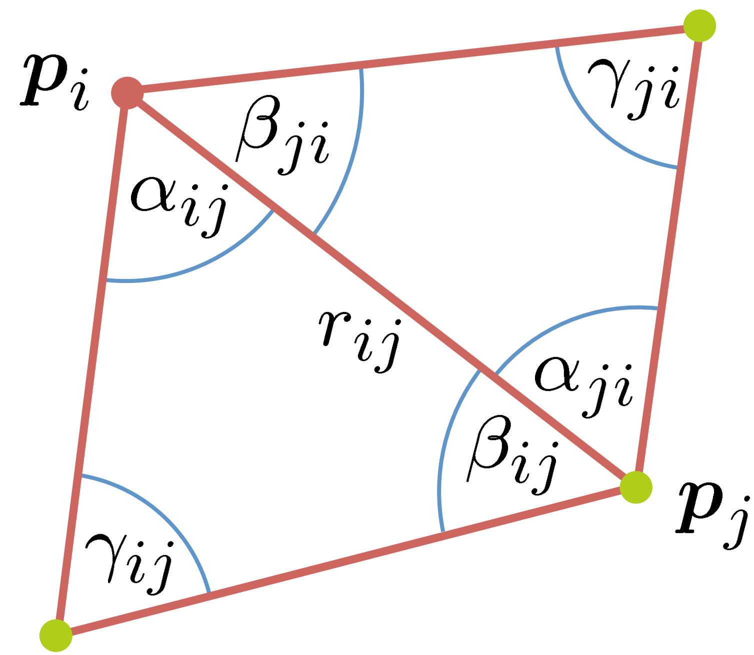
$$w_{ij} = \frac{\cot \alpha_{ji} + \cot \beta_{ij}}{{r_{ij}}^2}$$

- 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

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

Parametrization & Remeshing

Nico Pietroni (nico.pietroni@isti.cnr.it)

ISTI – CNR