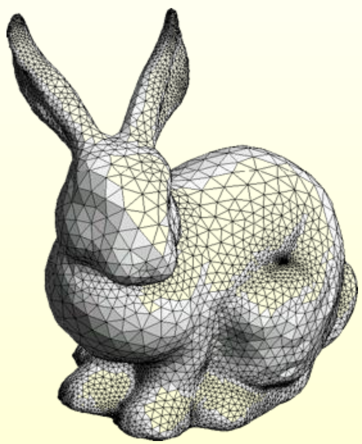
CS348a: Computer Graphics -- Geometric Modeling and Processing



Surface Parameterization

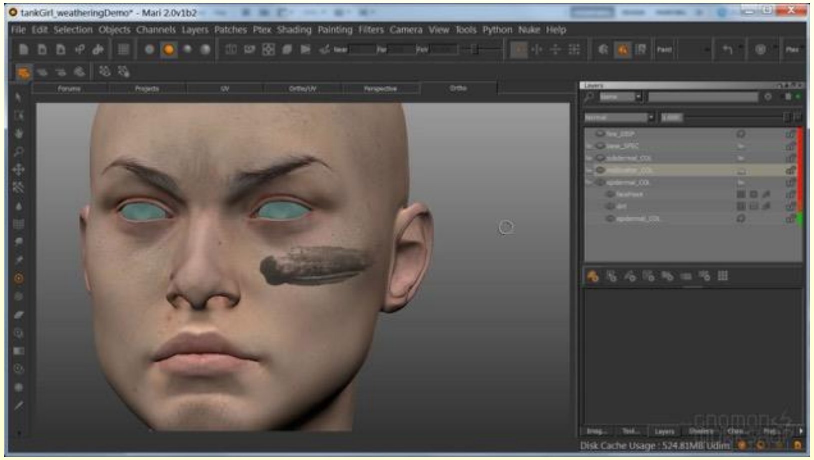
Chengcheng Tang
Computer Science Dept.
Stanford University

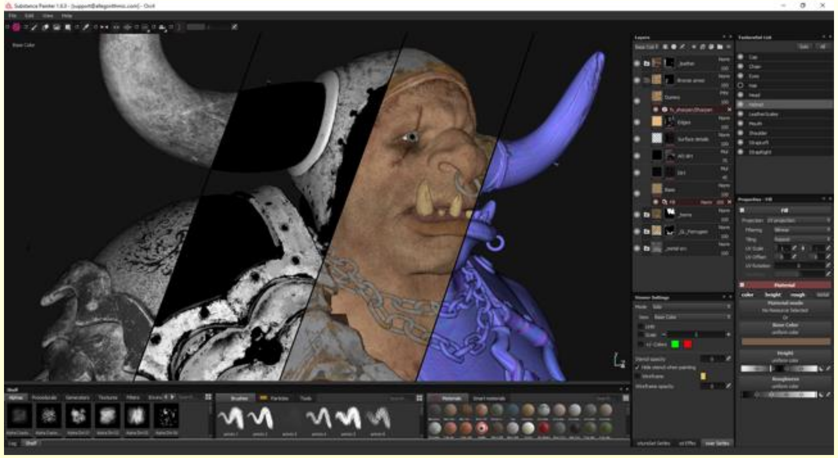
15 March 2017

Maks Ovsjanikov
Raif Rustamov
Justin Solomon
Mirela Ben-Chen
Julien Tierny
SIGGRAPH 2008 Course
Others…

Today

- Painting on Surfaces



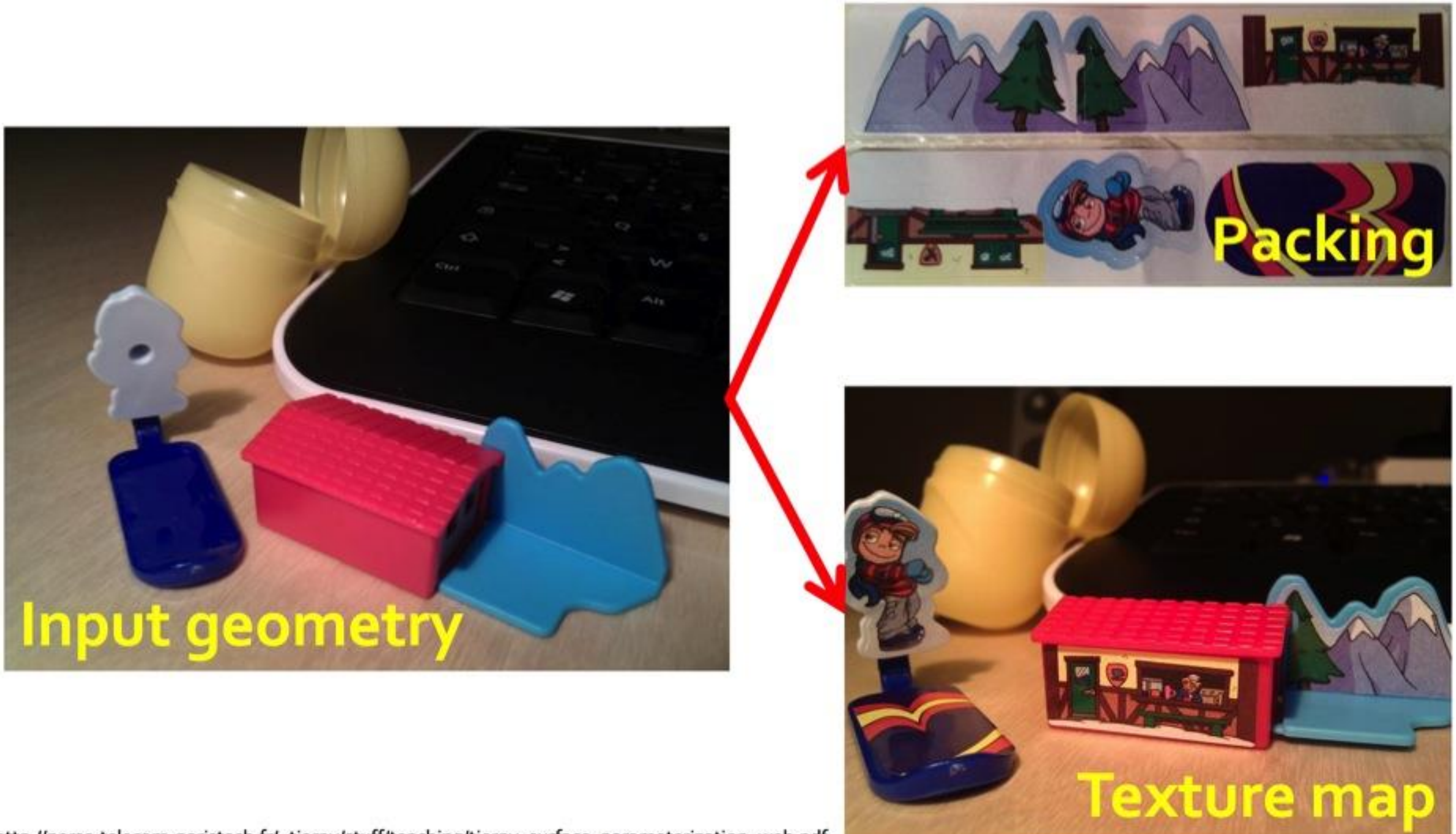


Painting directly on the 3d object

Mari software

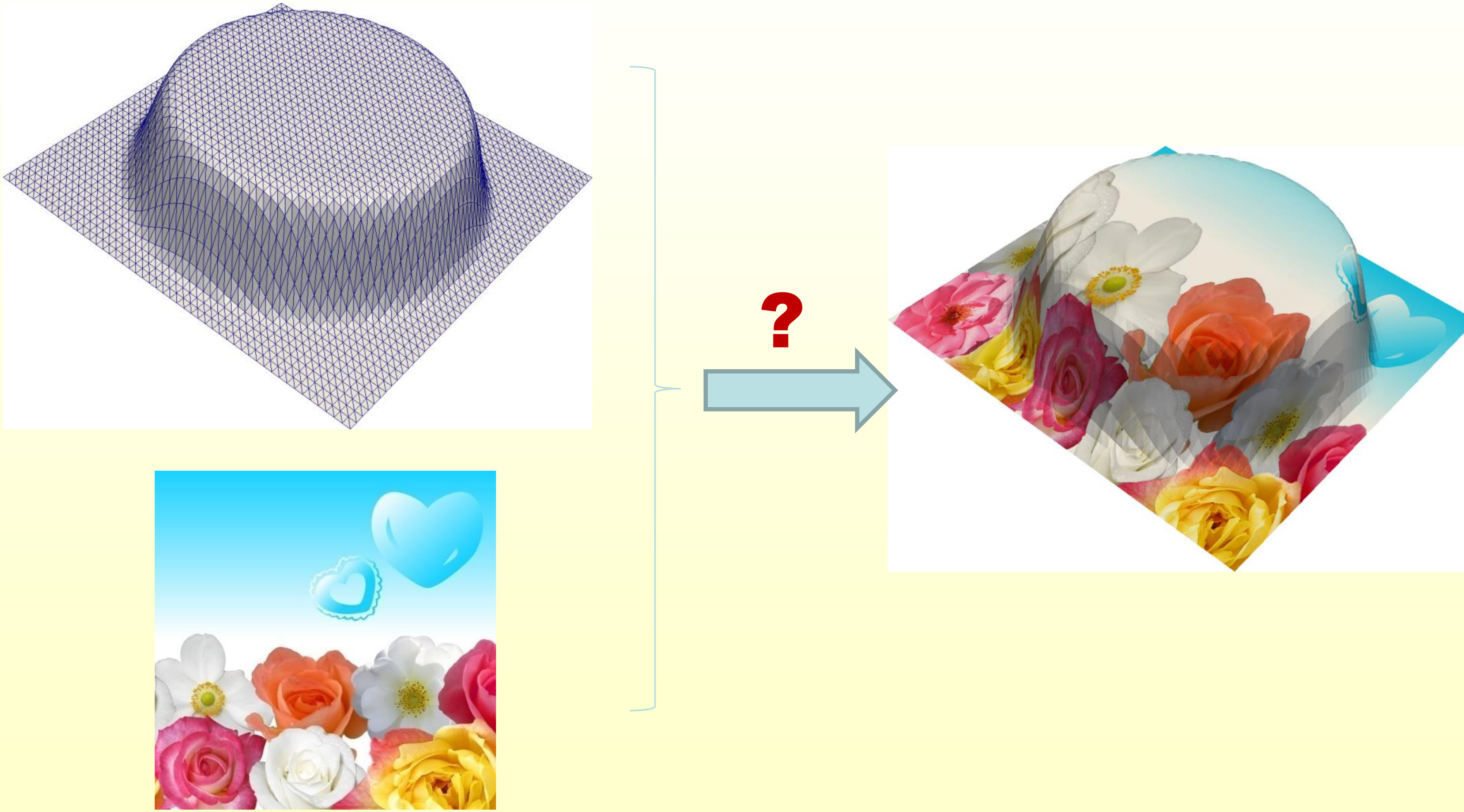
Substance 3D Painter

What we would like to do

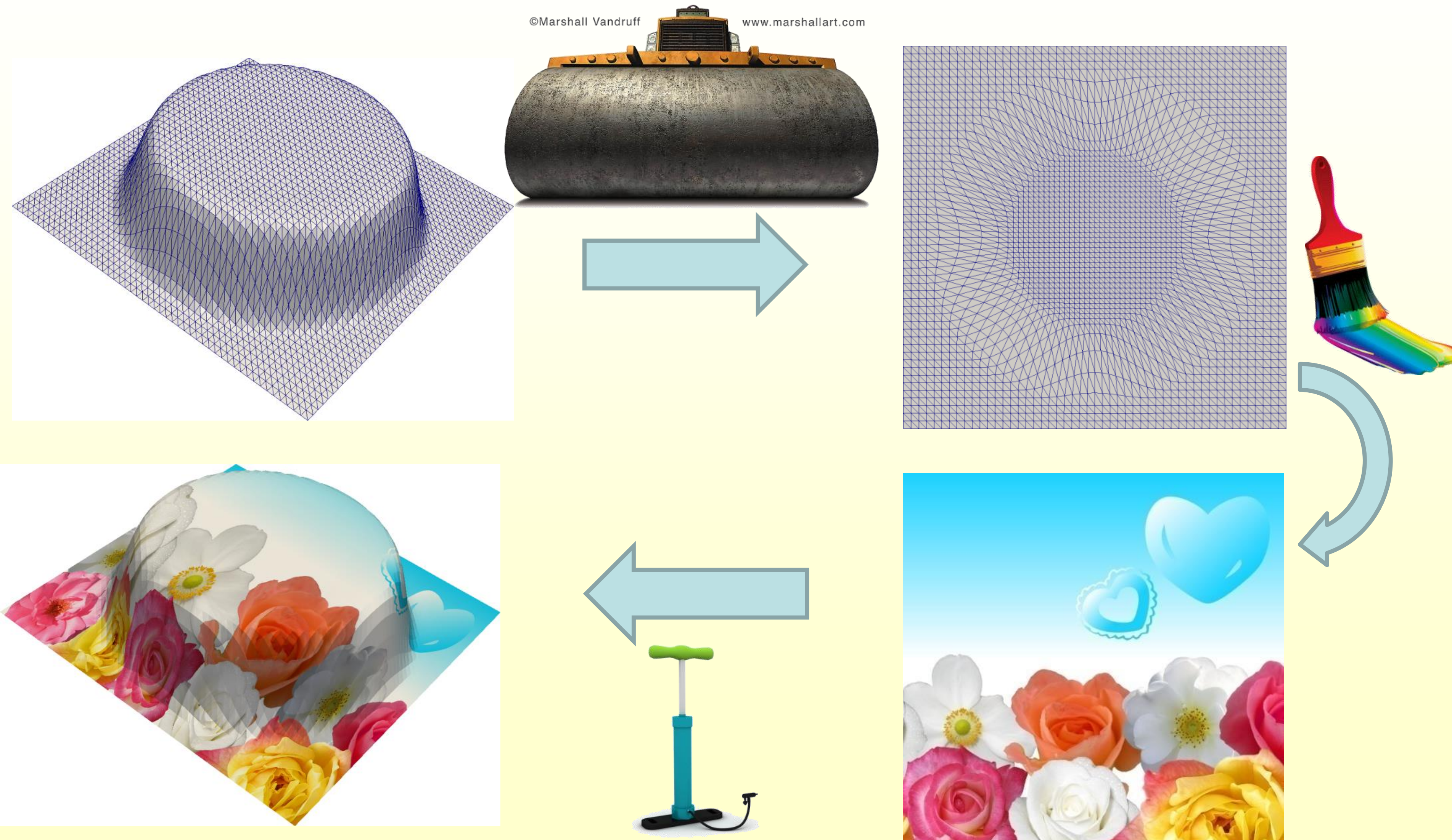


http://perso.telecom-paristech.fr/~tierny/stuff/teaching/tierny_surface_parameterization_web.pdf

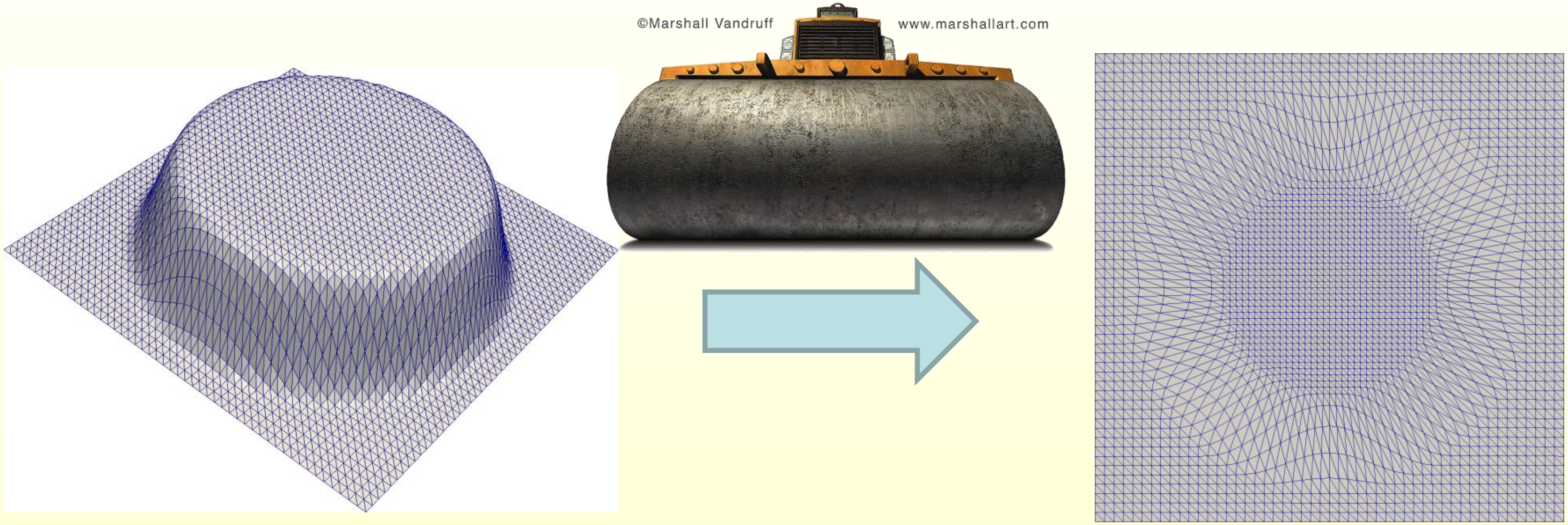
The Basic Problem



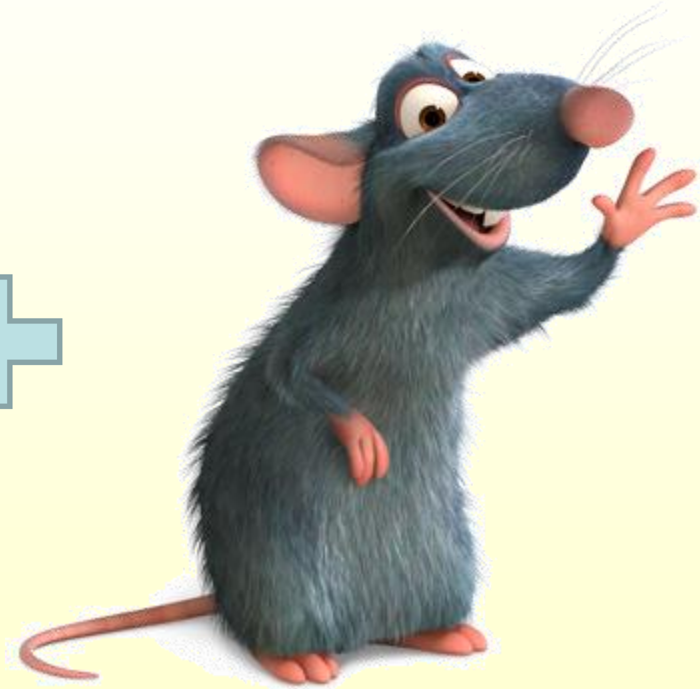
Solution



Parameterization is …

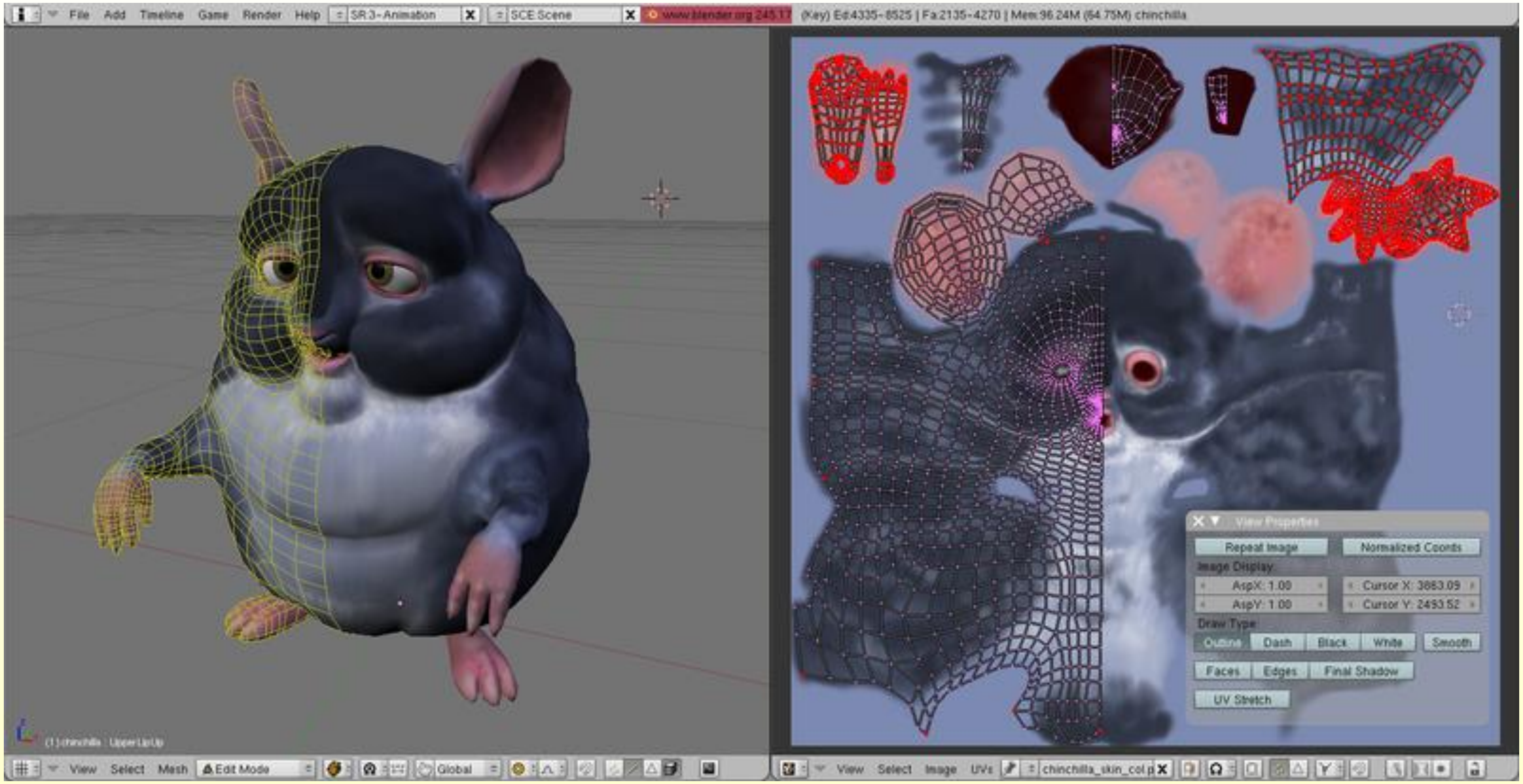


Why Parameterize?



R.I.P.
Really
Interested in
Parameterization

Why Parameterize?



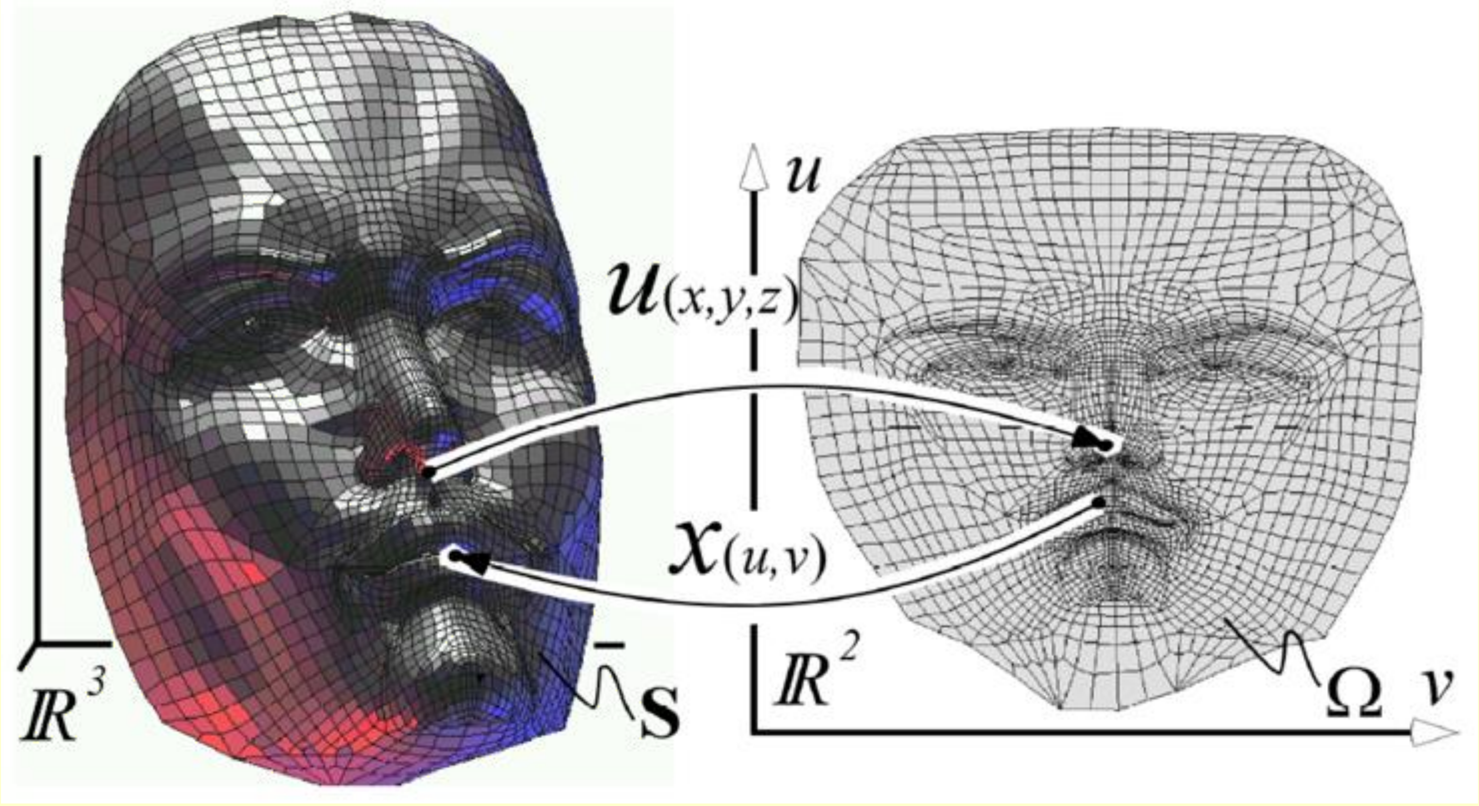
http://www.blender.org/development/release-logs/blender-246/uv-editing/

Texture Mapping

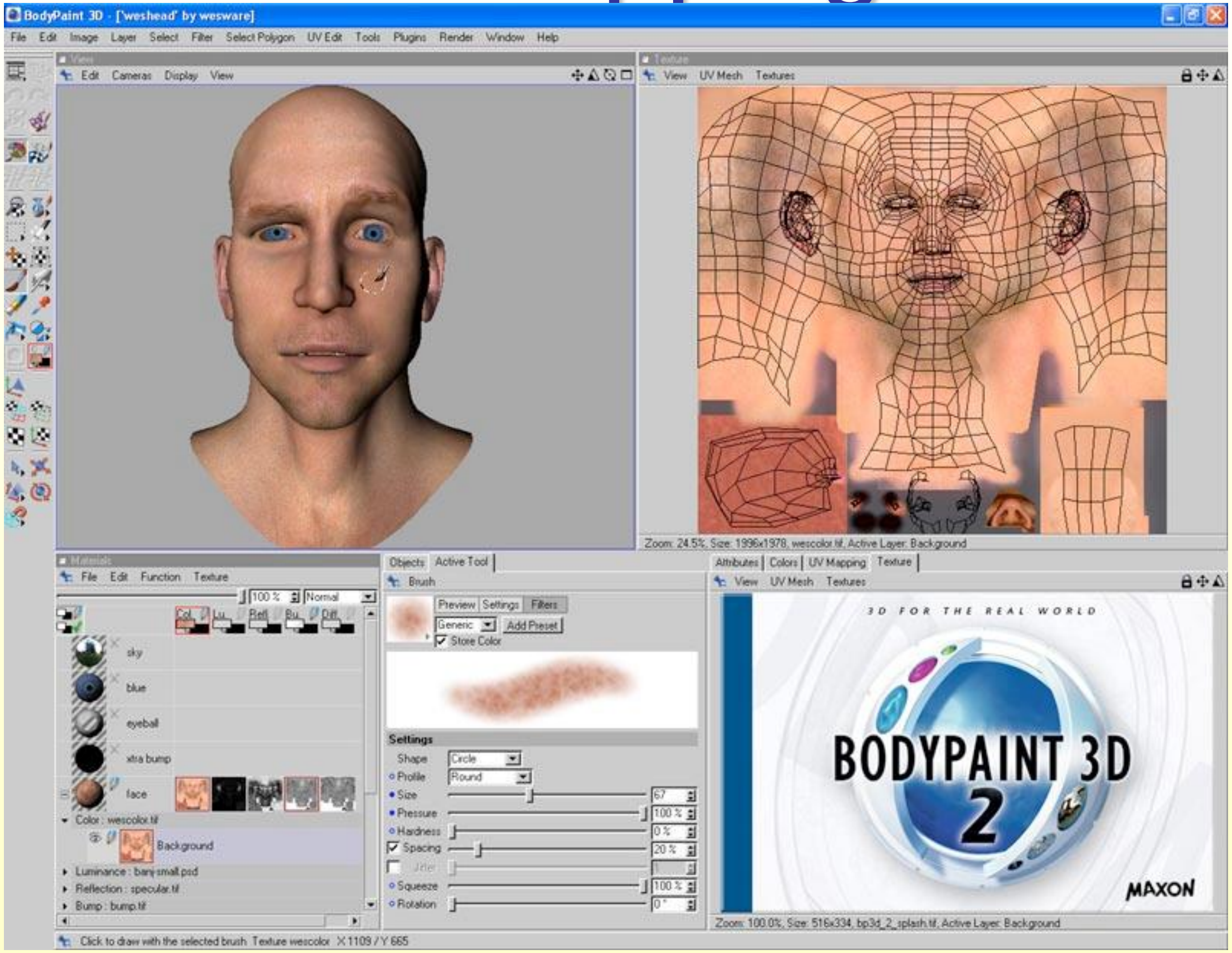
Parameterization Problem

Given a surface (mesh) S in R^3 and a domain Ω (e.g. plane):

Find a bijective map $U\colon \Omega \leftrightarrow S$.



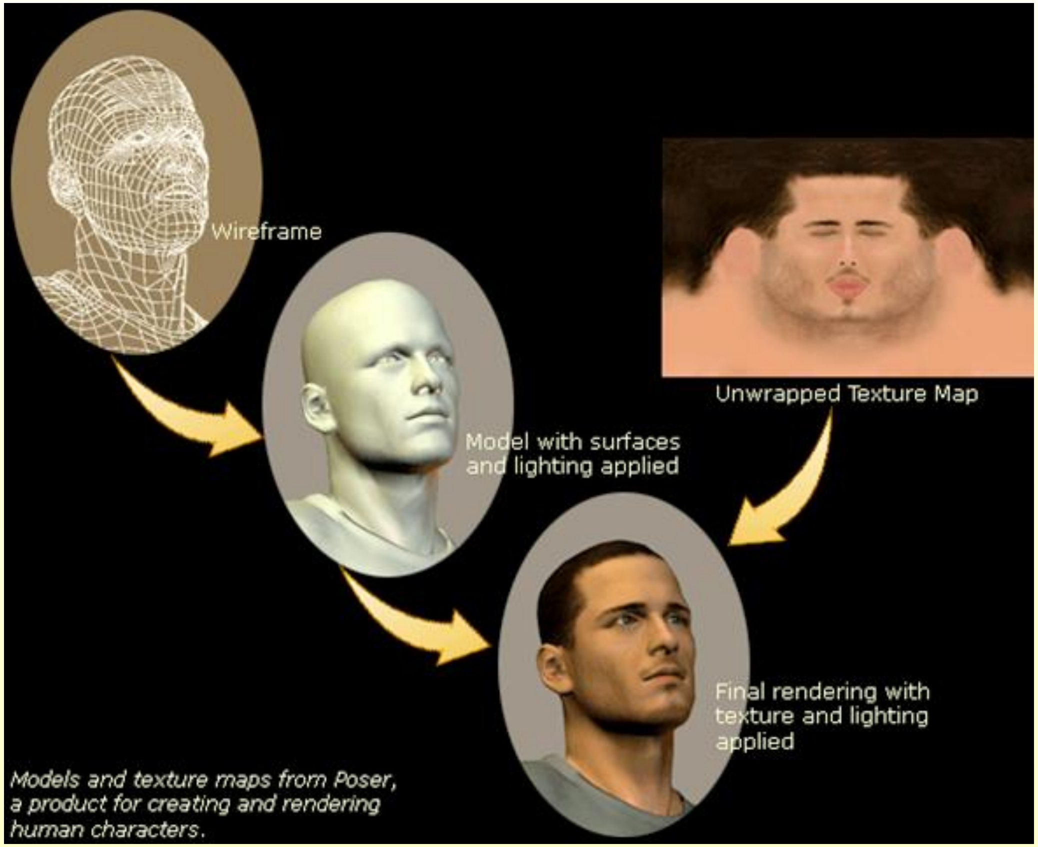
Parameterization for Texture Mapping



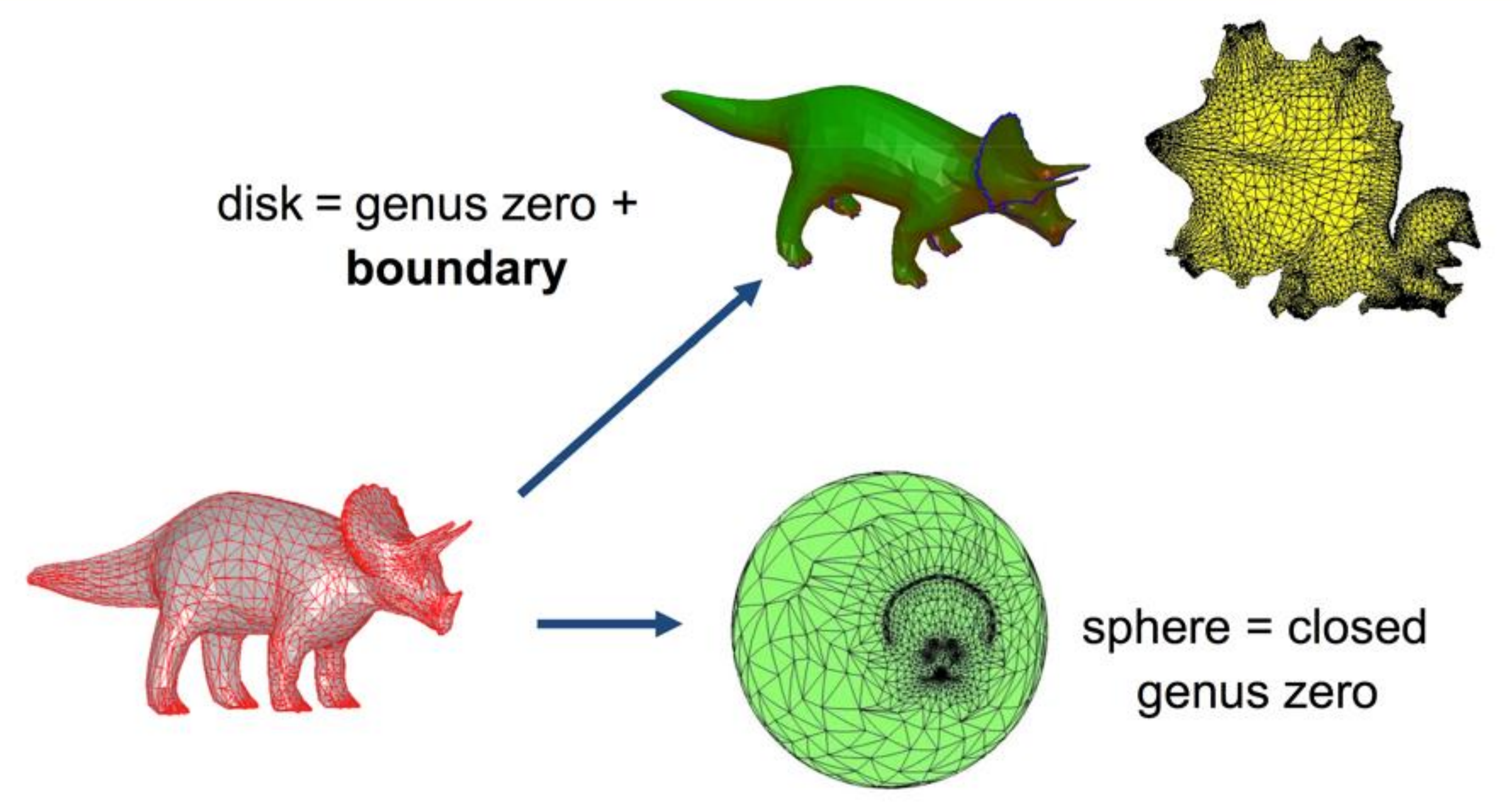
Bodypaint 3D

Parameterization for Texture Mapping

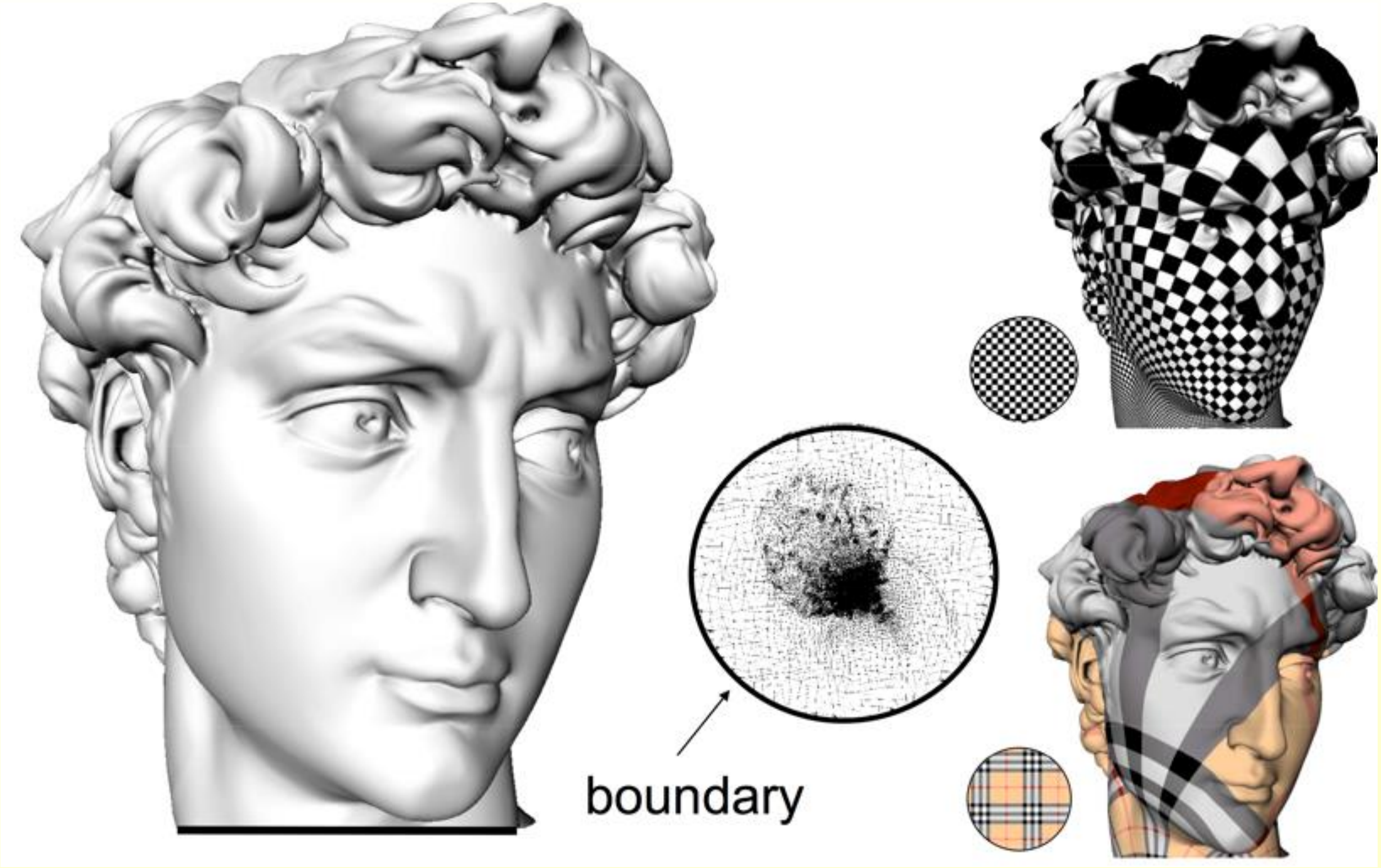
Rendering workflow:



Parameterization – Typical Domains

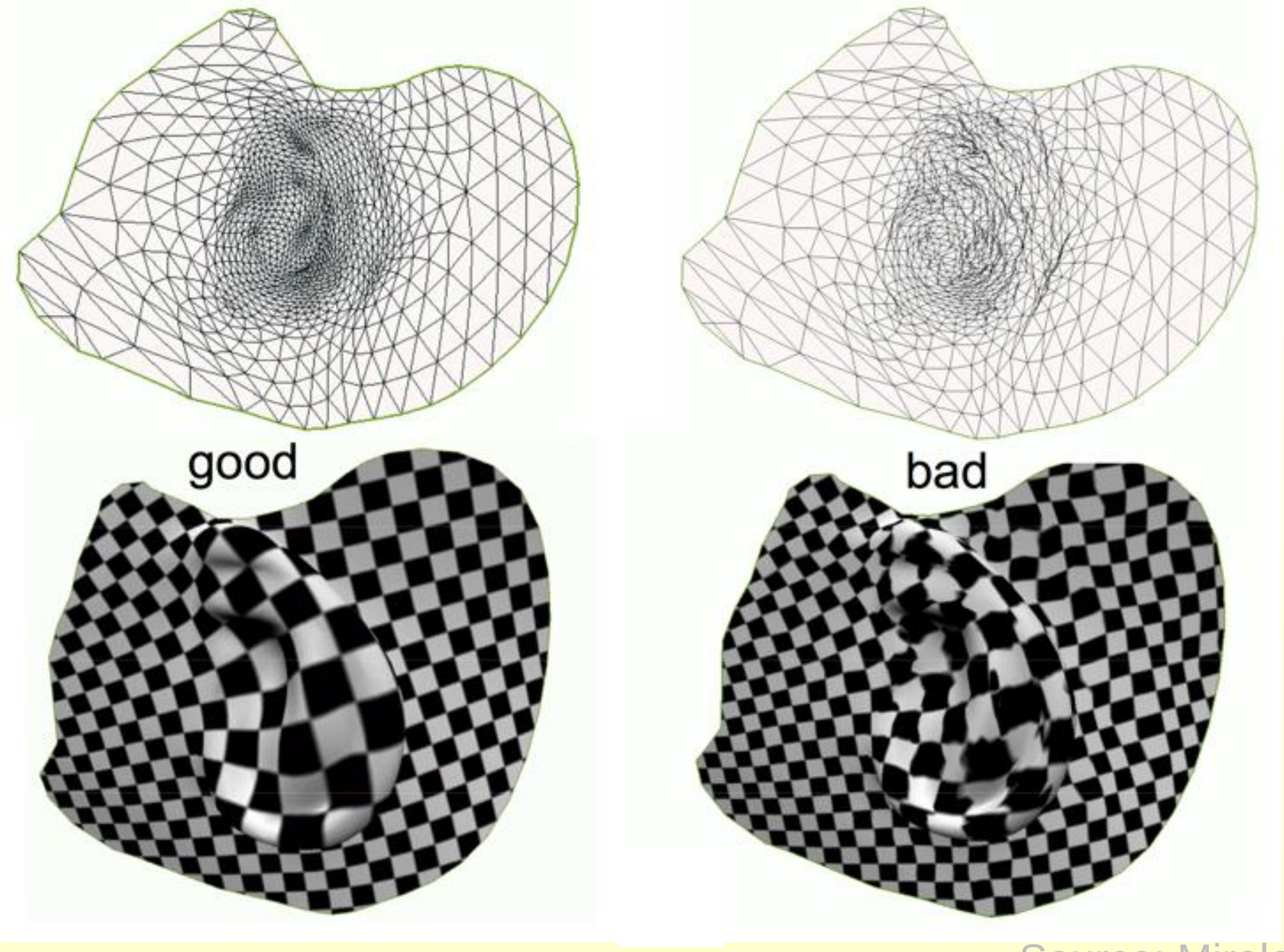


Parameterization – Boundary Problem



Source: Mirela Ben-Chen

Parameterization – Many Possibilities



Source: Mirela Ben-Chen

Parameterization – Applications

Recall Mesh simplification:

- Approximate the geometry using few triangles

Idea:

- Decouple geometry from appearance

~600k triangles

~600 triangles

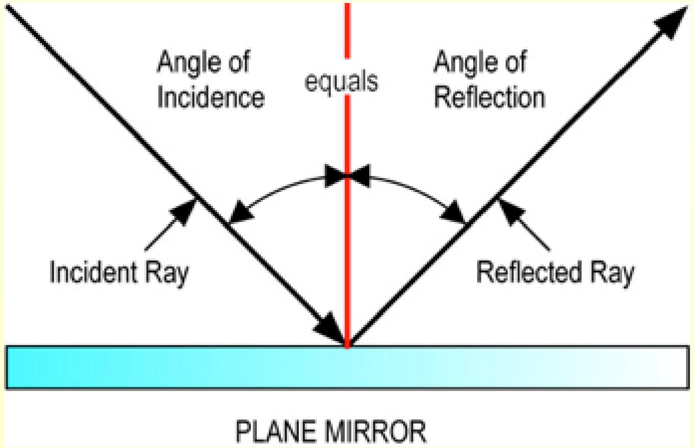
Parameterization – Applications

Recall Mesh simplification:

- Approximate the geometry using few triangles

Idea:

- Decouple geometry from appearance



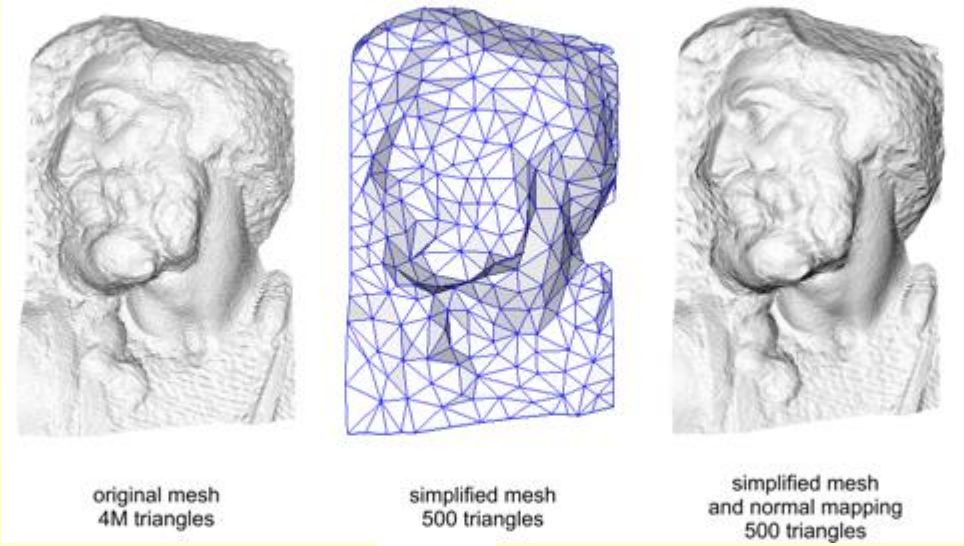
Observation: appearance (light reflection) depends on the geometry + normal directions.

Parameterization – Applications

Normal Mapping

Idea:

- Decouple geometry from appearance
- Encode a normal field inside each triangle

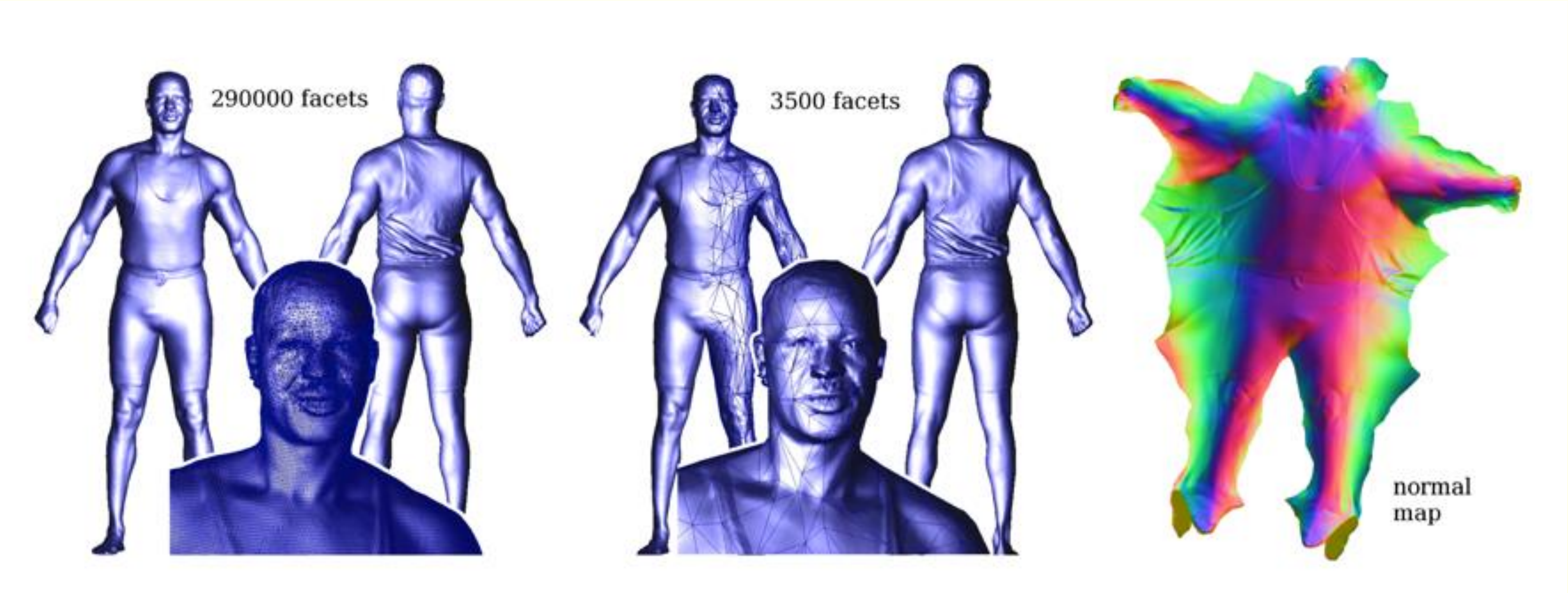


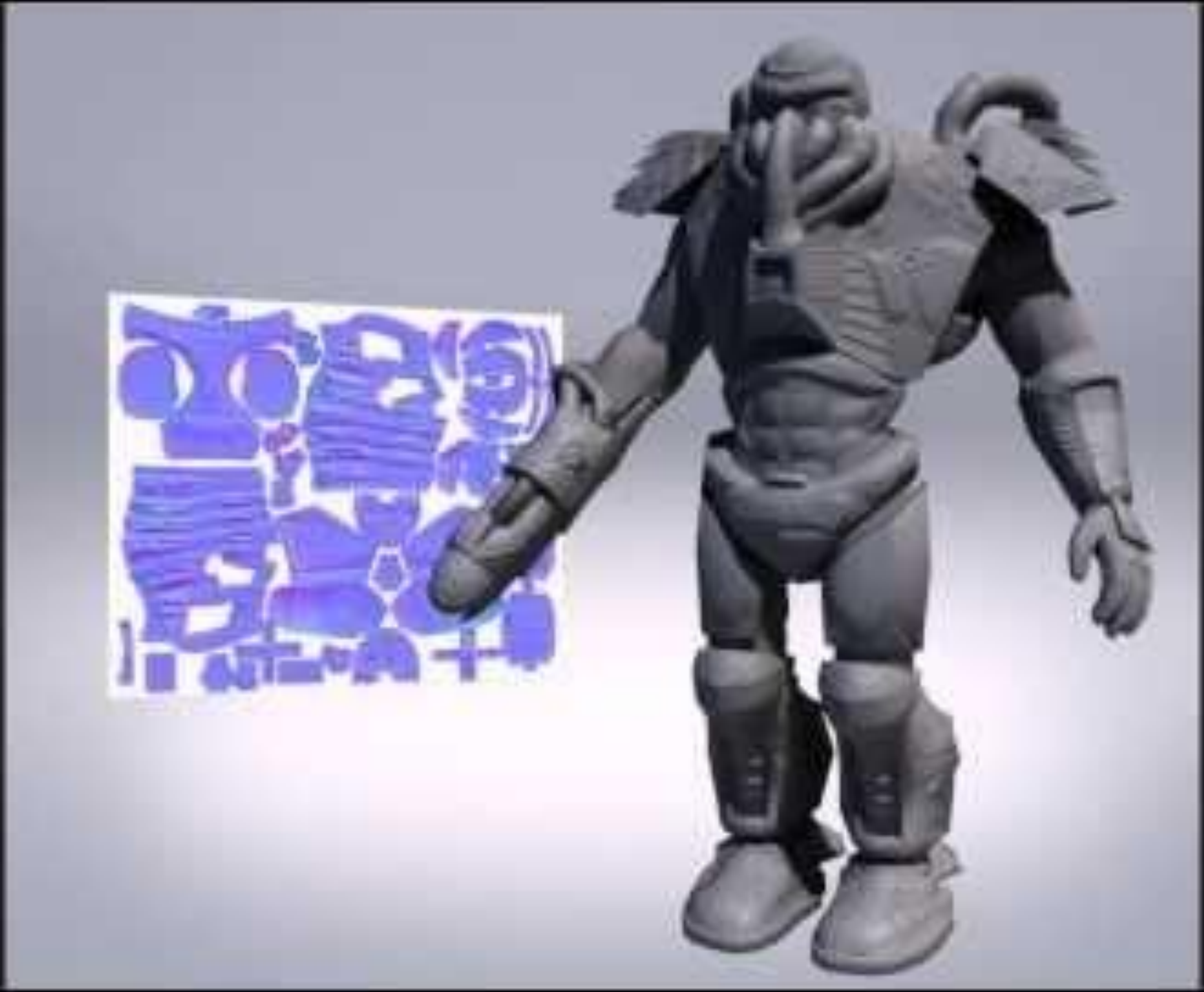
Cohen et al., '98
Cignoni et al. '98

Parameterization – Applications

Normal Mapping with parameterization:

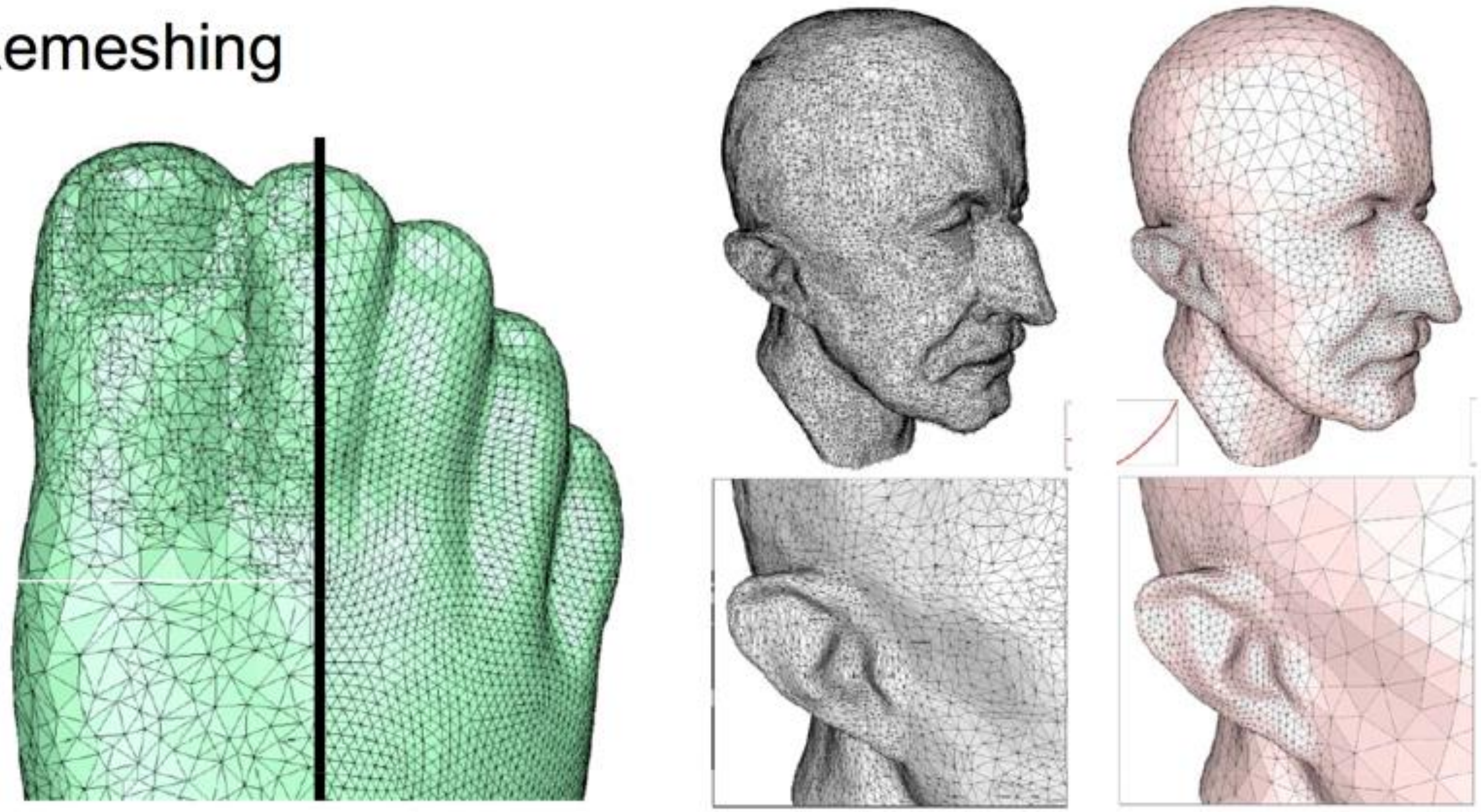
- Store normal field as an RGB texture.





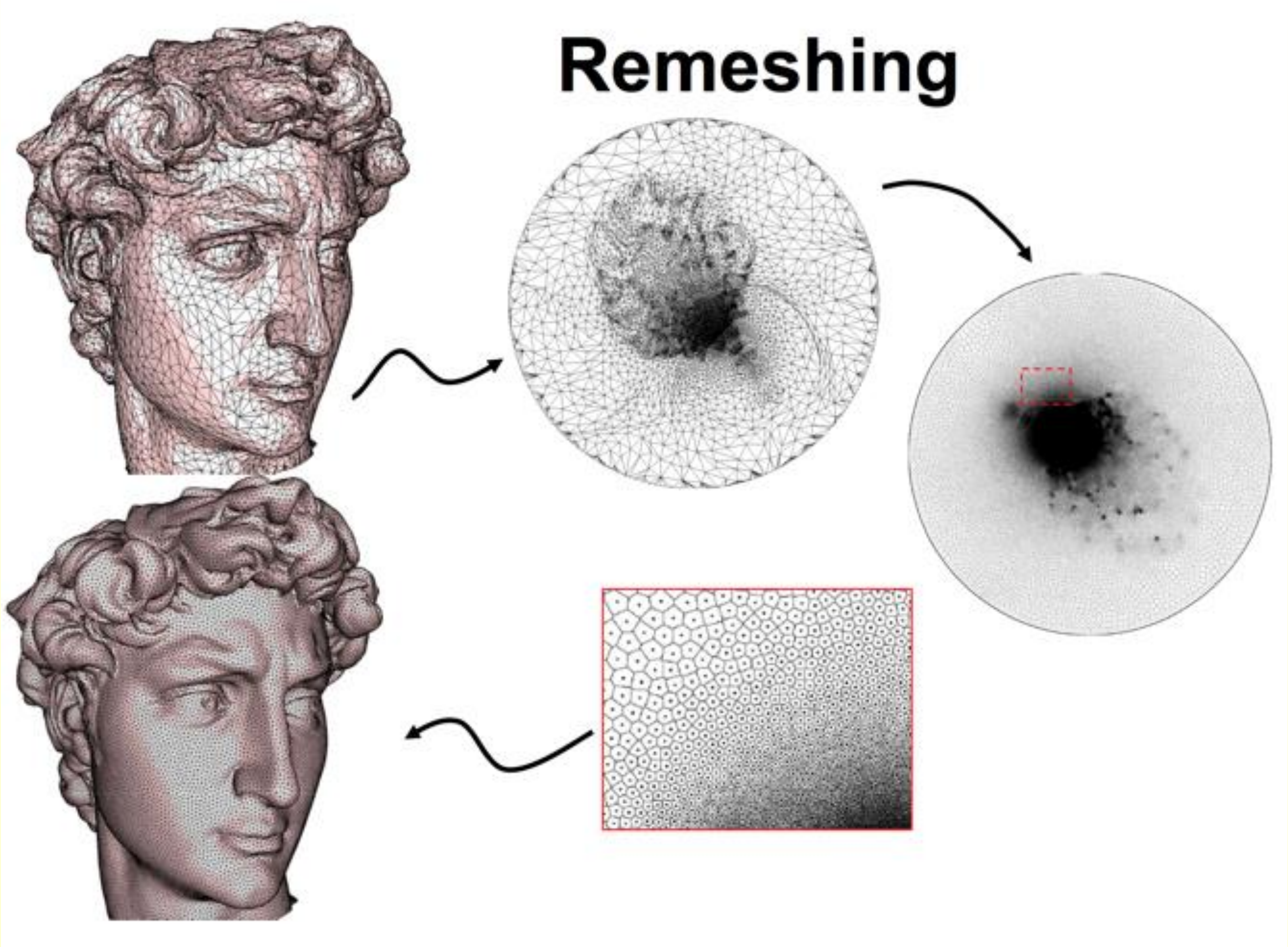
Parameterization – Applications

- Remeshing



source: Mirela Ben-Chen

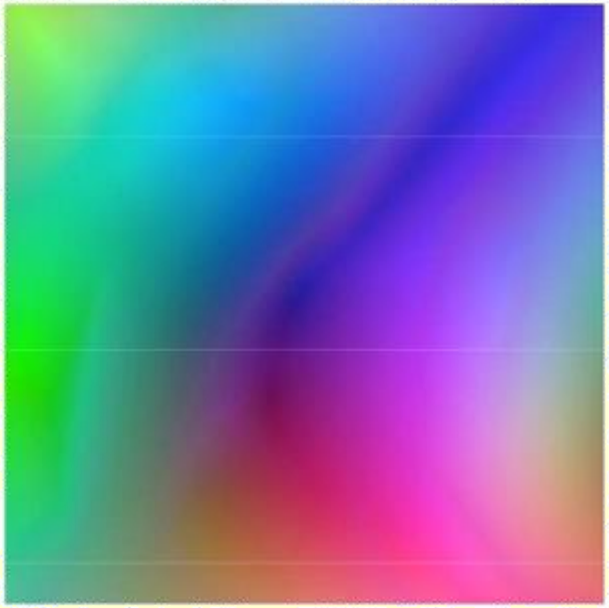
Parameterization – Applications



source: Mirela Ben-Chen

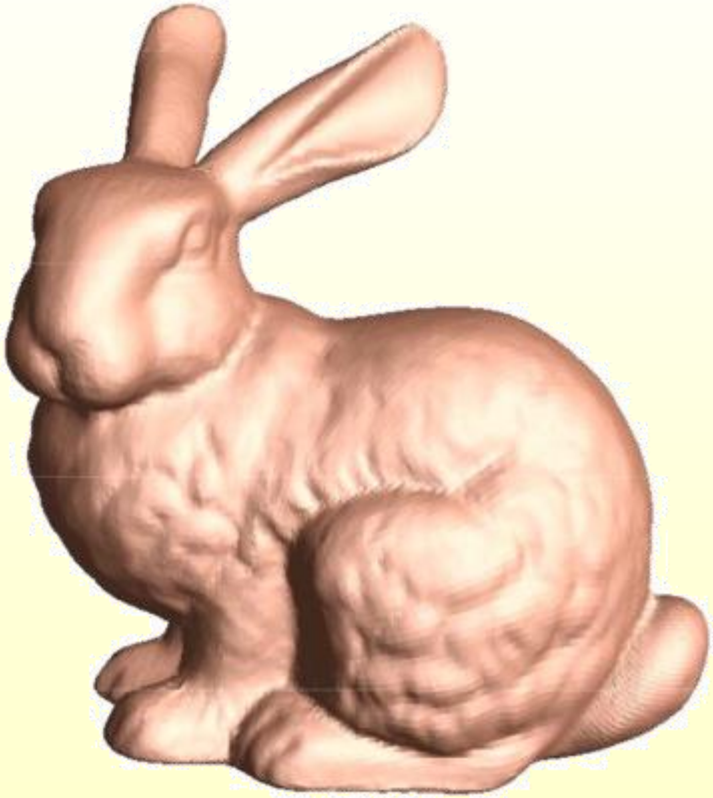
Parameterization – Applications

- Compression



Stanford Bunny

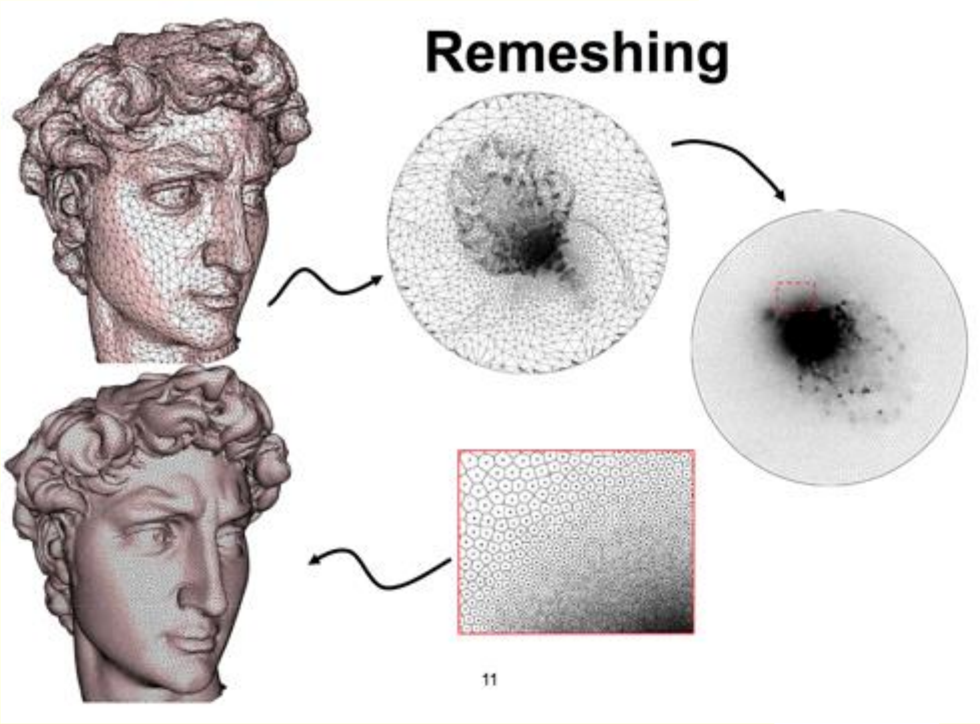
=



Gu, Gortler, Hoppe. Geometry Images. SIGGRAPH 2002

Parameterization – Applications

General Idea: Things become easier in a canonical domain (e.g. on a plane).



Other Applications:

- Surface Fitting
- Editing
- Mesh Completion
- Mesh Interpolation
- Morphing and Transfer
- Shape Matching
- Visualization

…

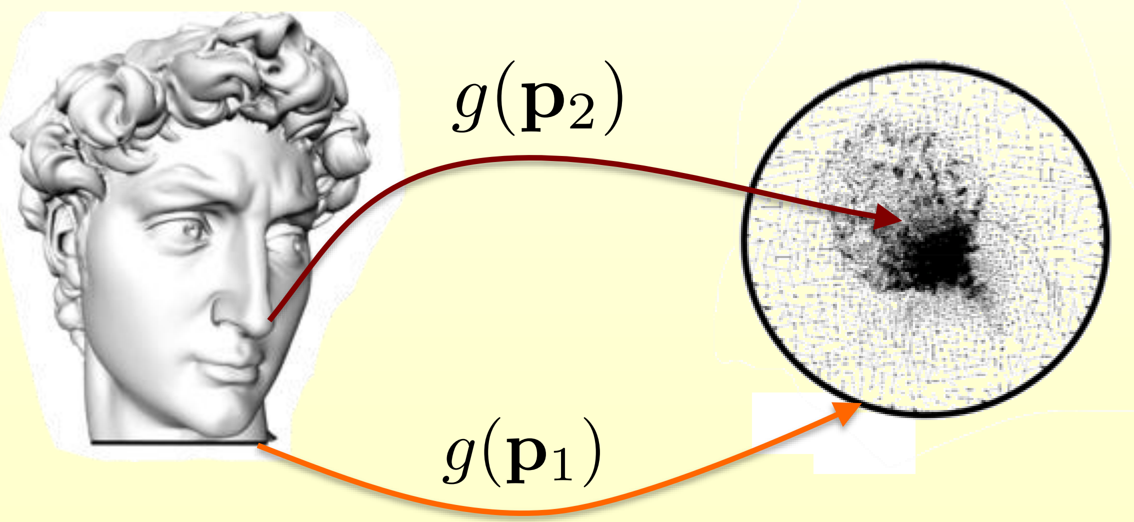
Parameterization onto the plane

General problem:

- Given a mesh (T, P) in 3D find a bijective mapping

$$g : P \to \mathbf{R}^2$$

$$g(\mathbf{p}_i) = \mathbf{u}_i = (u_i, v_i)$$

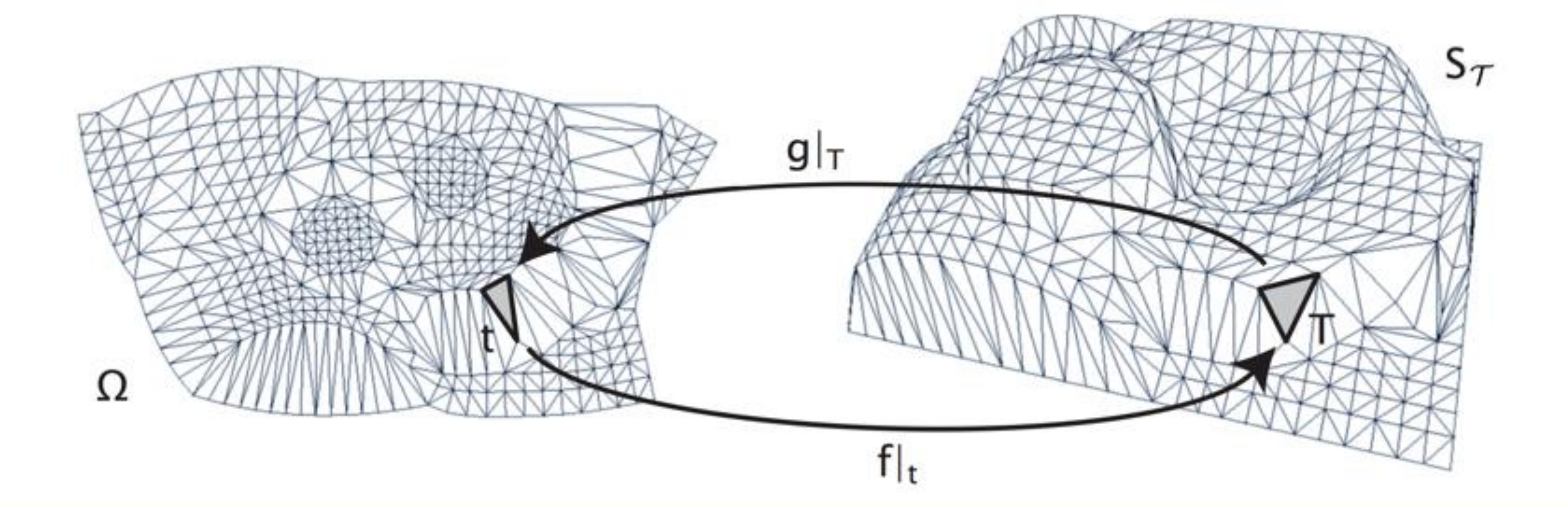


Parameterization onto the plane

General problem:

- Given a mesh (T, P) in 3D find a bijective mapping

$$g : P \rightarrow \mathbf{R}^2$$
$$g(\mathbf{p}_i) = \mathbf{u}_i = (u_i, v_i)$$



Parameterization onto the plane

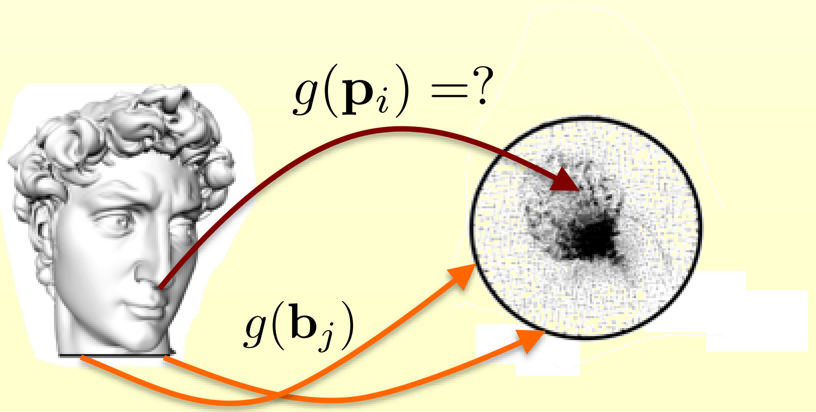
Simplified problem:

- Given a mesh (T, P) in 3D find a bijective mapping

$$g : P \to \mathbf{R}^2$$
$$g(\mathbf{p}_i) = \mathbf{u}_i = (u_i, v_i)$$

under some boundary constraints:

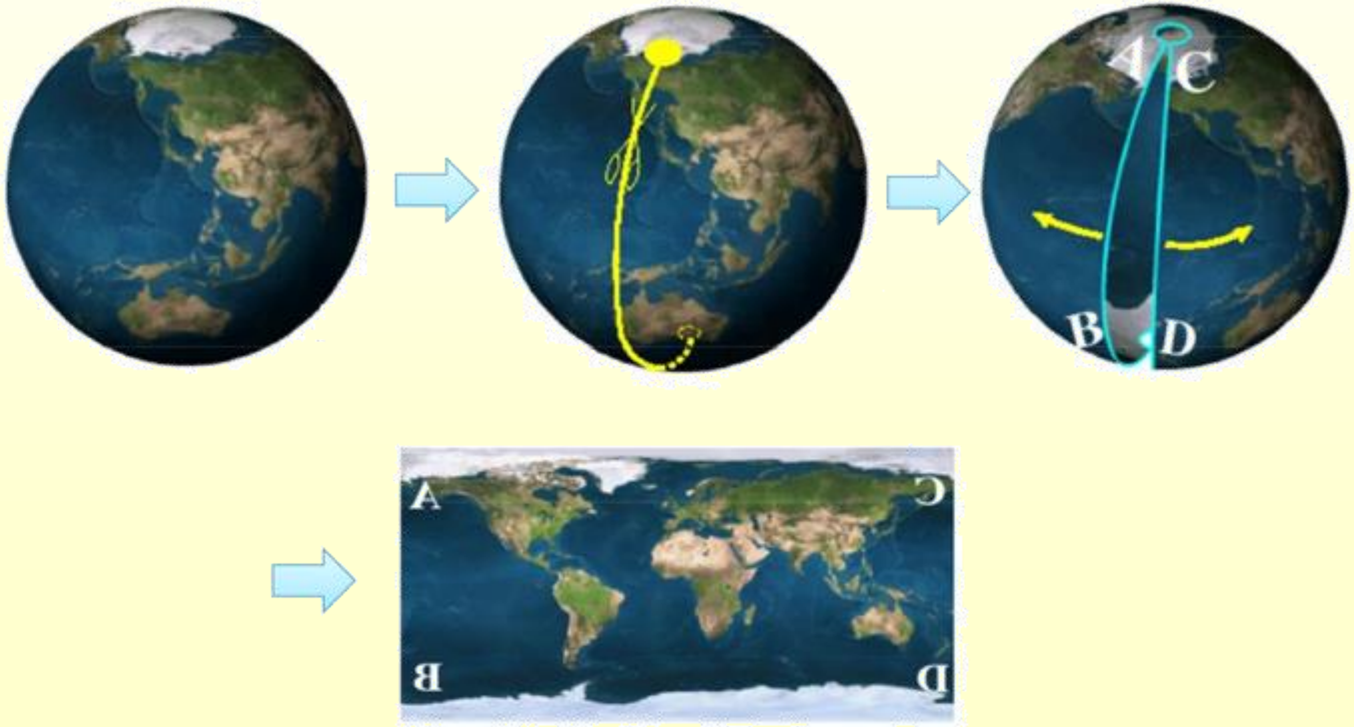
$$g(\mathbf{b}_j) = \mathbf{u}_j \text{ for some } \{\mathbf{b}_j\}$$



Parameterization onto the plane

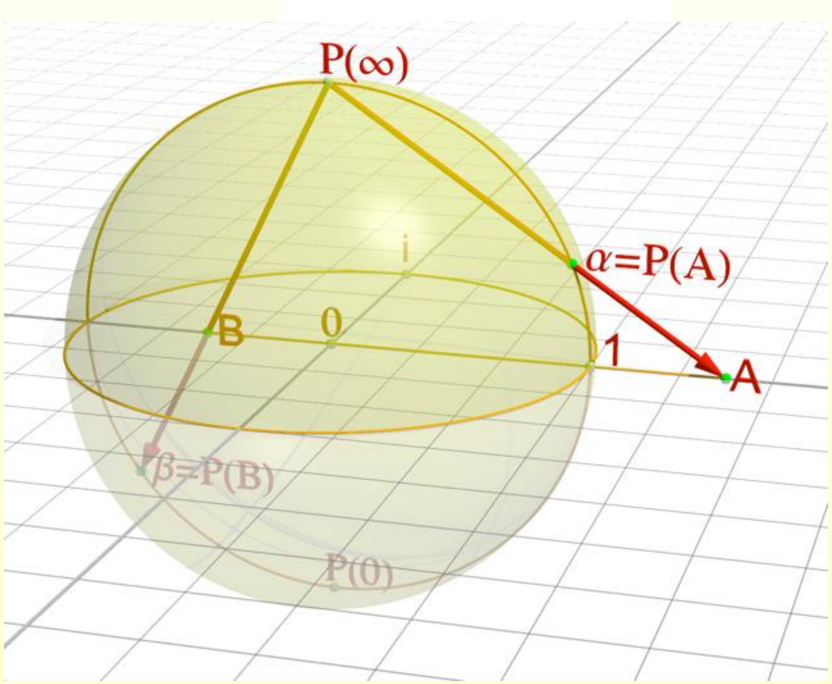
Recall a related problem.

Mapping the Earth: find a parameterization of a 3d object onto a plane.



Mapping the earth

Stereographic projection

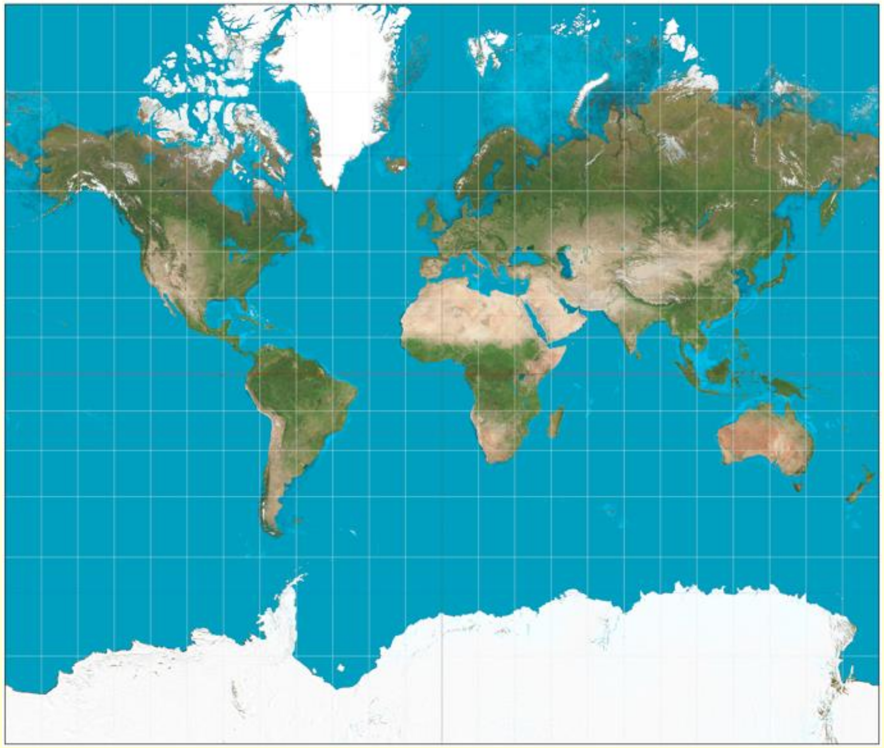


Maps circles to circles

Hipparchus (190–120 B.C.)

Mapping the earth

Mercator

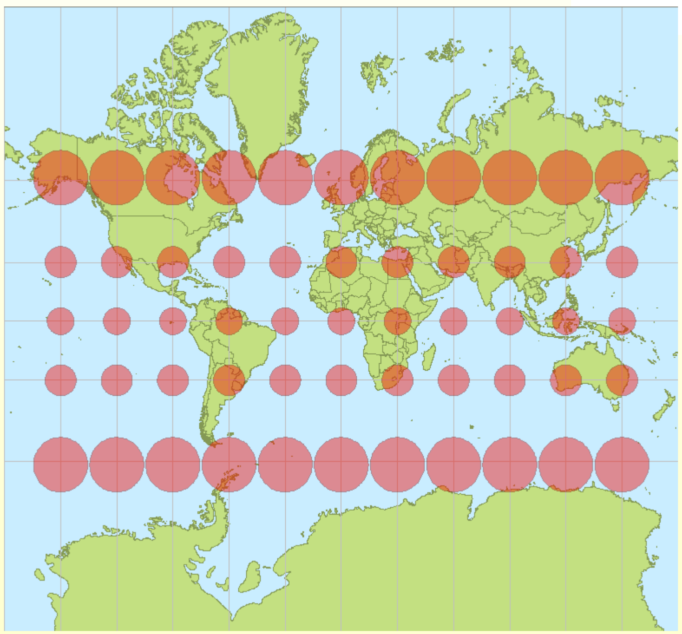


Maps loxodromes to lines

Gerardus Mercator (1569)

Mapping the earth

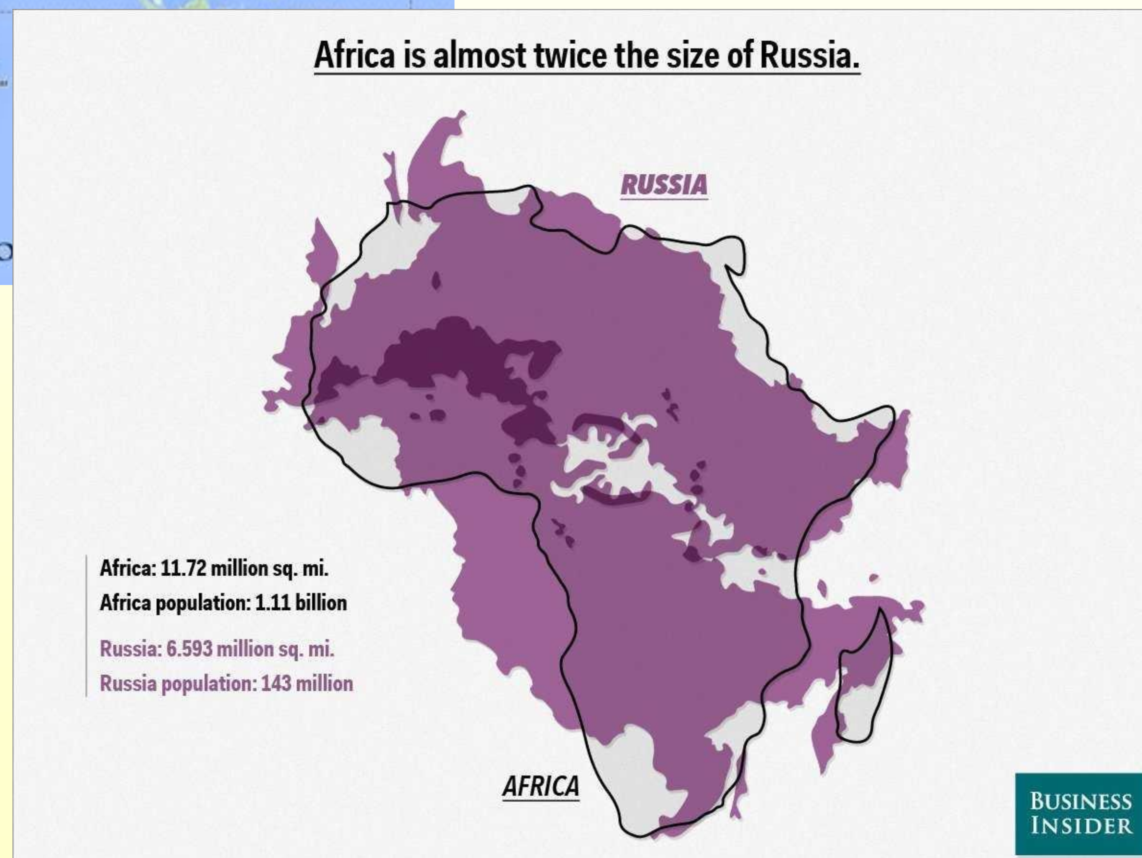
Mercator (preserves angles, but distorts areas)

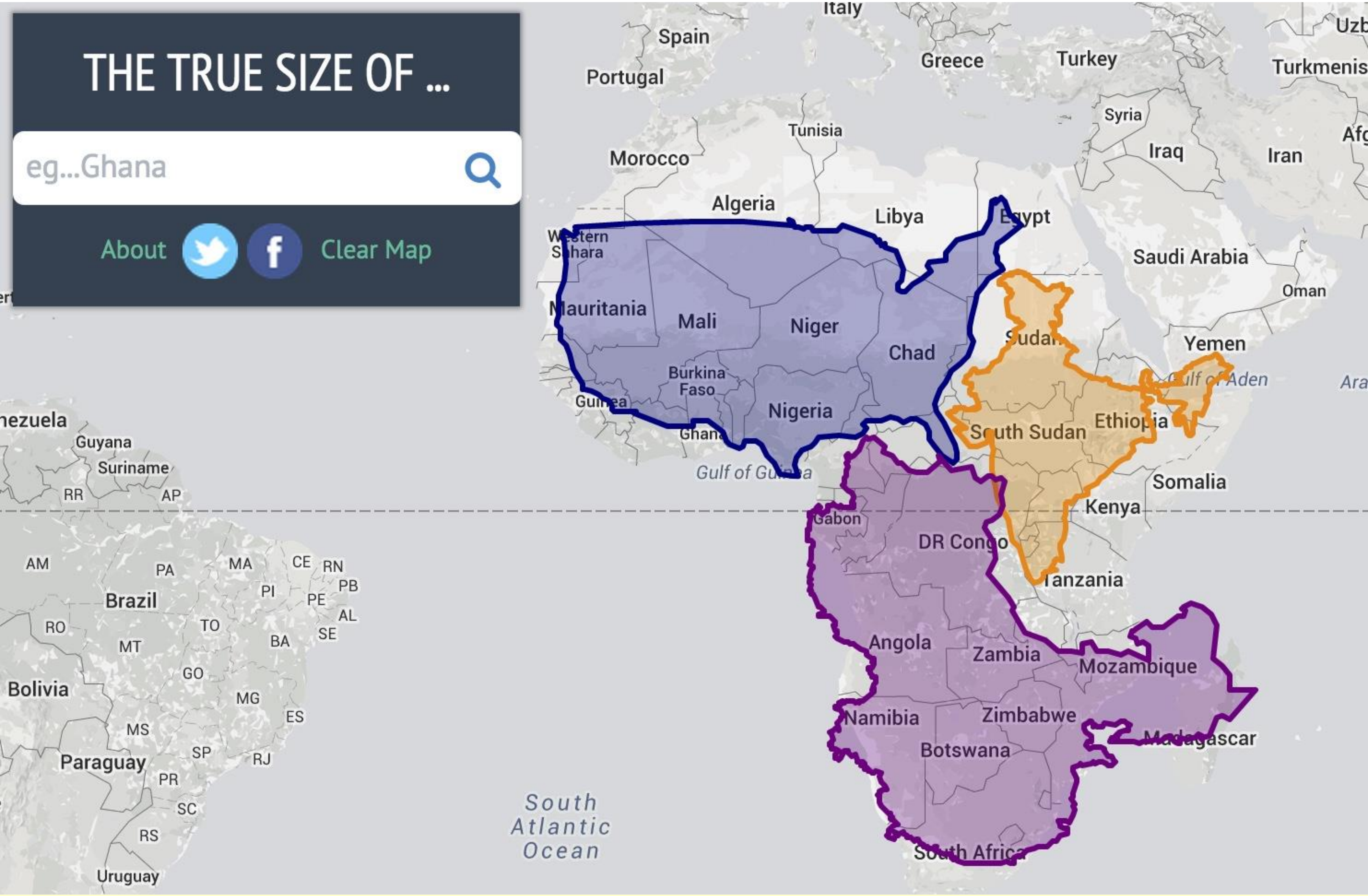


Maps loxodromes to lines

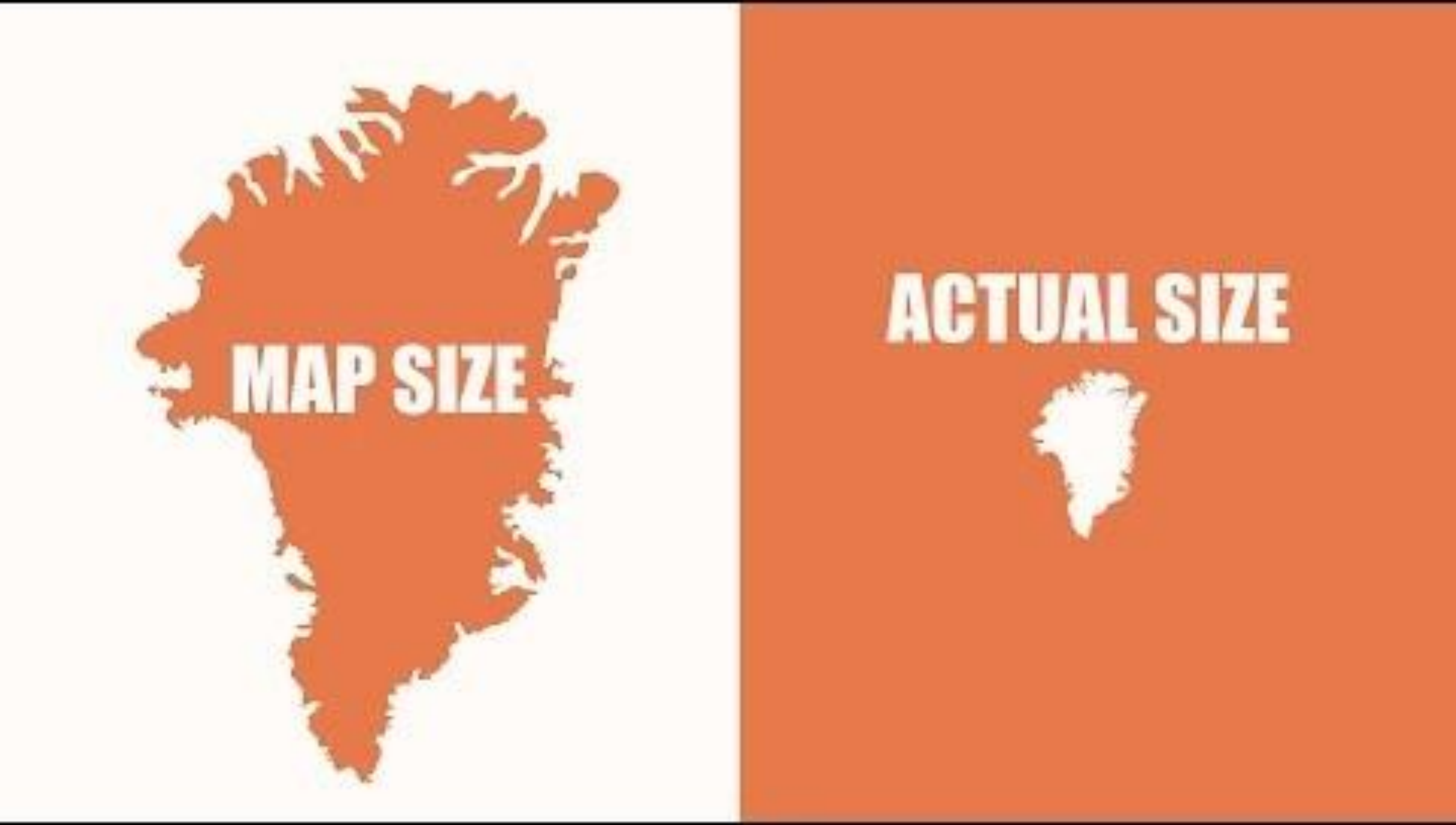
Gerardus Mercator (1569)


SIZE OF RUSSIA COMPARED TO AFRICA
Arctic Ocean
+Jakub
Greenland
Iceland
Finland
Sweden
Norway
United Kingdom
Poland
Germany
France
Spain
Italy
Turkey
Russia
Kazakhstan
Mongolia
China
Japan
South Korea
Afghanistan
Pakistan
India
Thailand
Iraq
Iran
Saudi Arabia
Egypt
Libya
Algeria
Niger
Chad
Nigeria
DR Congo
Angola
Namibia
Tanzania
Kenya
Ethiopia
South Africa
Madagascar
Canada
United States
Mexico
Venezuela
Colombia
Peru
Brazil
Bolivia
Chile
Argentina
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Southern Ocean
Google

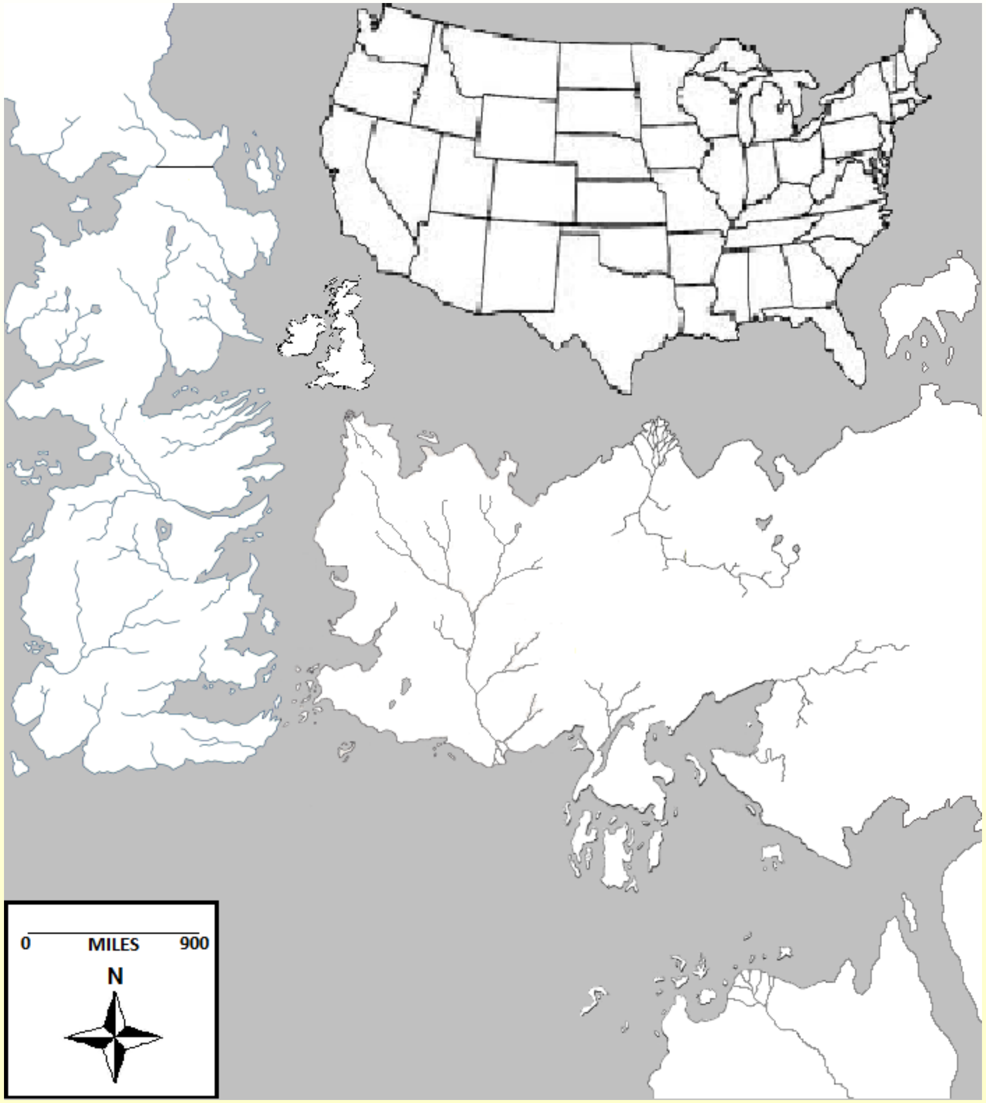

Africa is almost twice the size of Russia.
RUSSIA
AFRICA
Africa: 11.72 million sq. mi.
Africa population: 1.11 billion
Russia: 6.593 million sq. mi.
Russia population: 143 million
BUSINESS INSIDER



http://thetruesize.com


MAP SIZE
ACTUAL SIZE

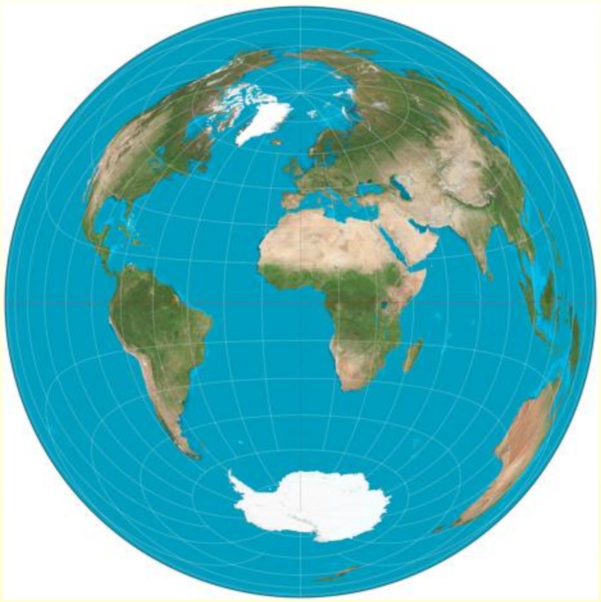
The size of Westeros compared to the USA

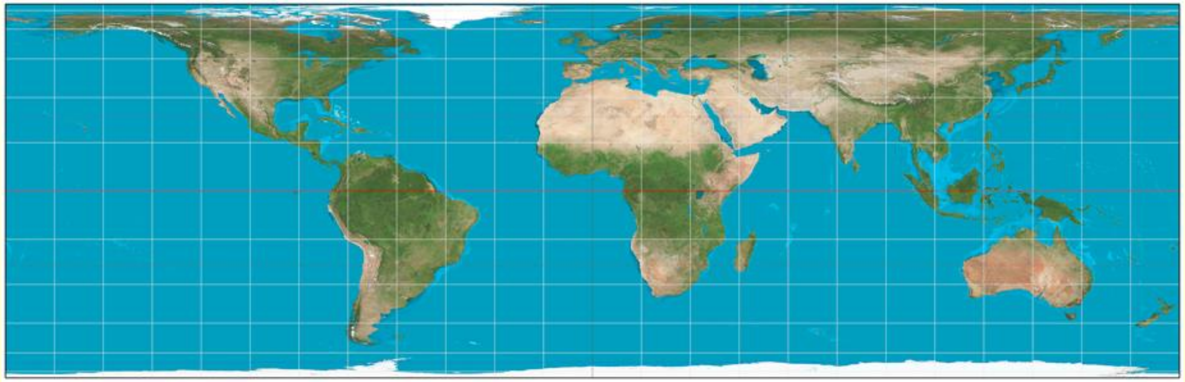


http://thewertzone.blogspot.com/2012/06/size-of-westeros-compared-to-usa.html

Mapping the earth

Lambert (preserves areas, but distorts angles)

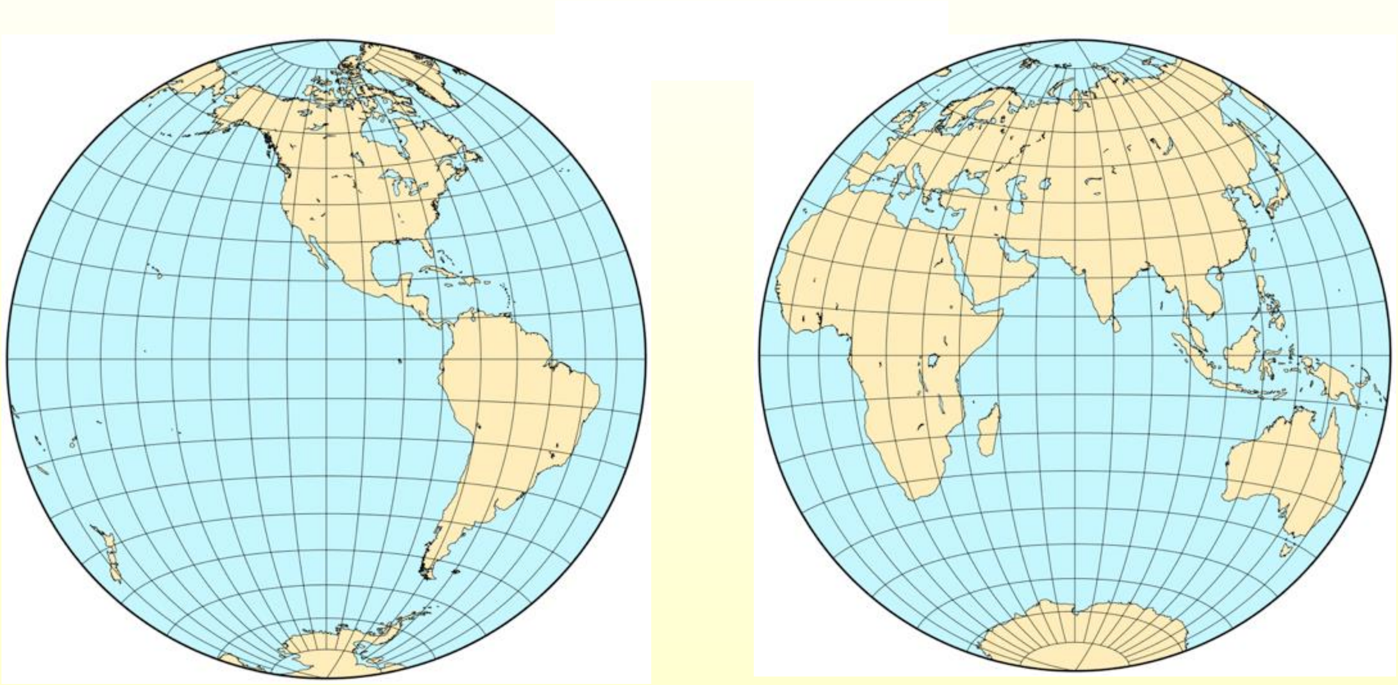




Johann Heinrich Lambert (1772)

Mapping the earth

Lambert (preserves areas, but distorts angles)



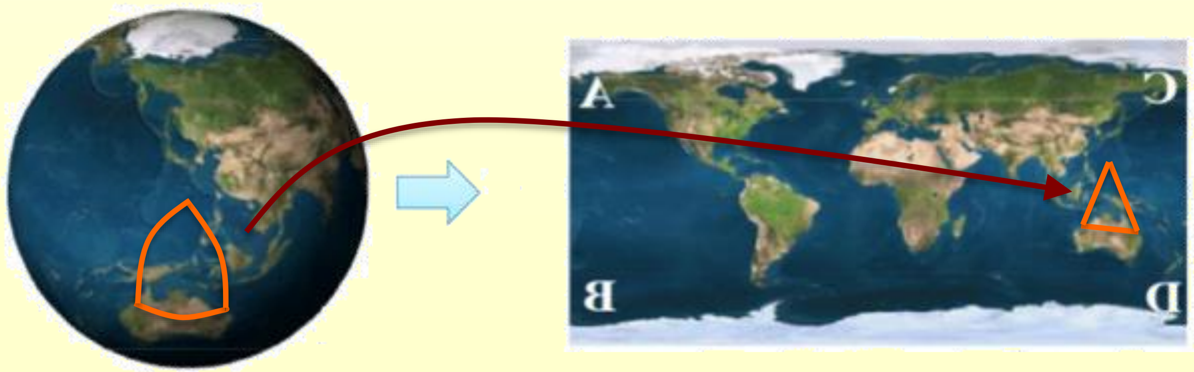
Johann Heinrich Lambert (1772)

Different kinds of Parameterization

Various notions of distortion:

1. Equiareal: preserving areas (up to scale)
2. Conformal: preserving angles of intersections
3. Isometric: preserving geodesic distances (up to scale)

Theorem: Isometric = Conformal + Equiareal



Different kinds of Parameterization

Intrinsic properties:

Those that depend on angles and distances on the surface. E.g.

Intrinsic: geodesic distances

Extrinsic: coordinates of points in space

Remark:

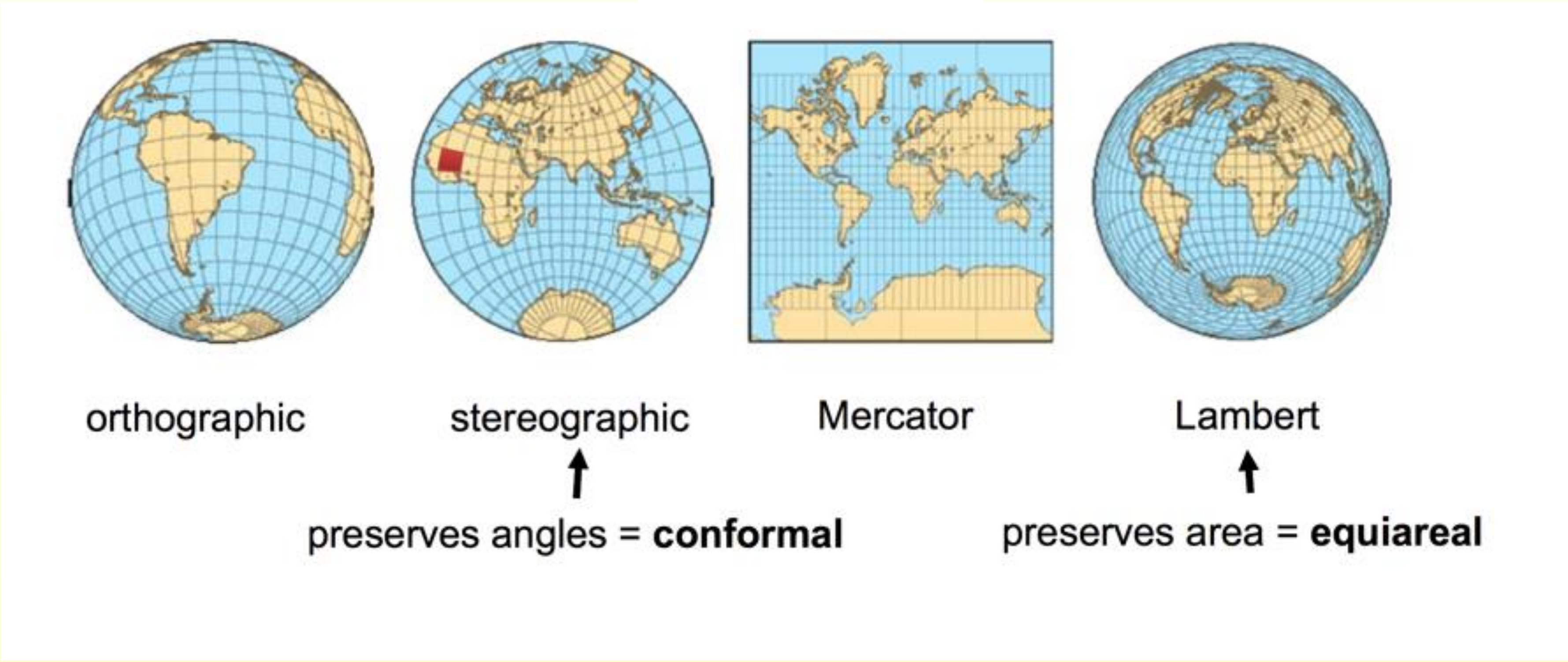
Intrinsic properties are preserved by isometries.

Bad news:

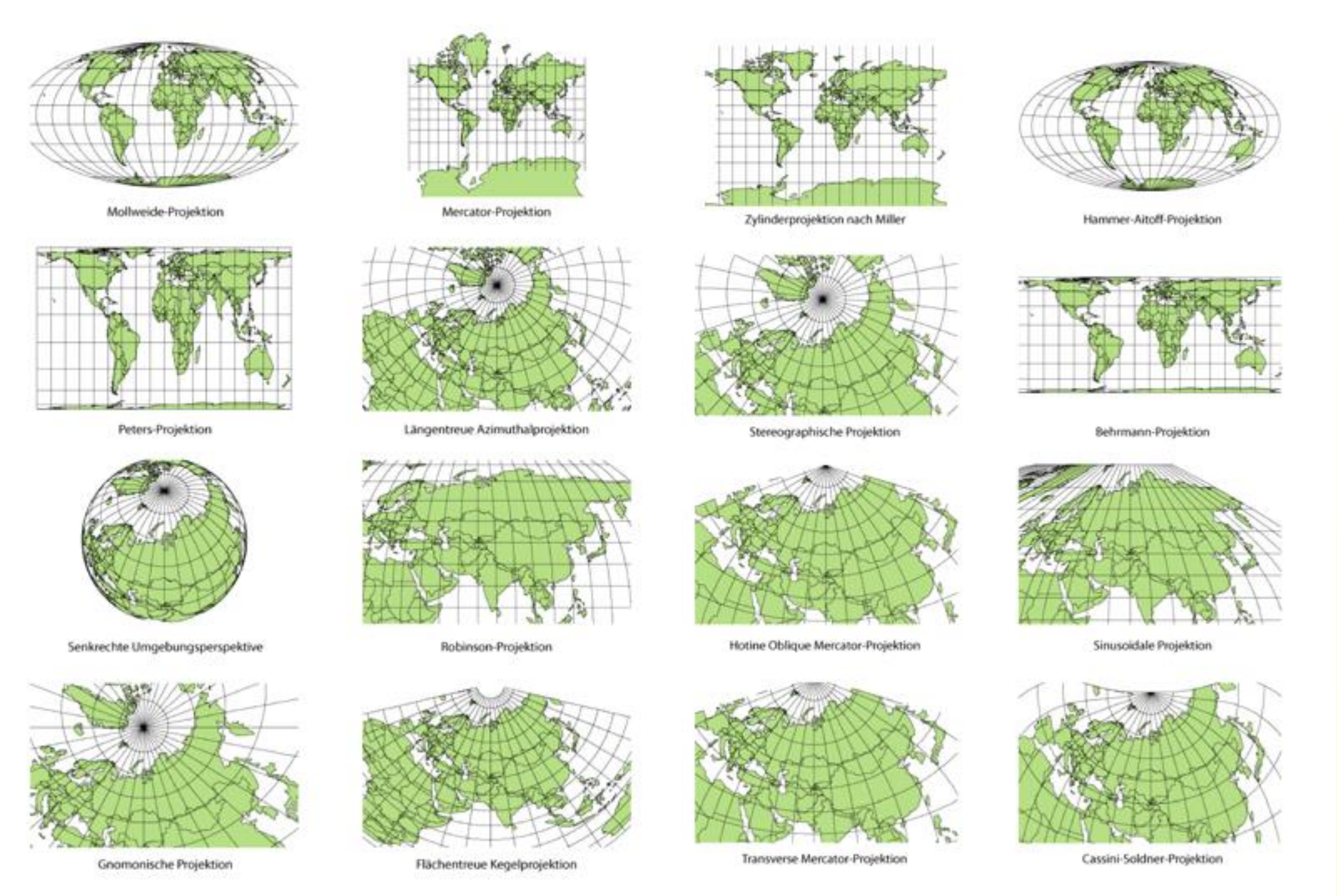
Gauss's Theorema Egregium: curvature is an intrinsic property.

→ There is no isometric mapping between a sphere and a plane.

Different kinds of Parameterization

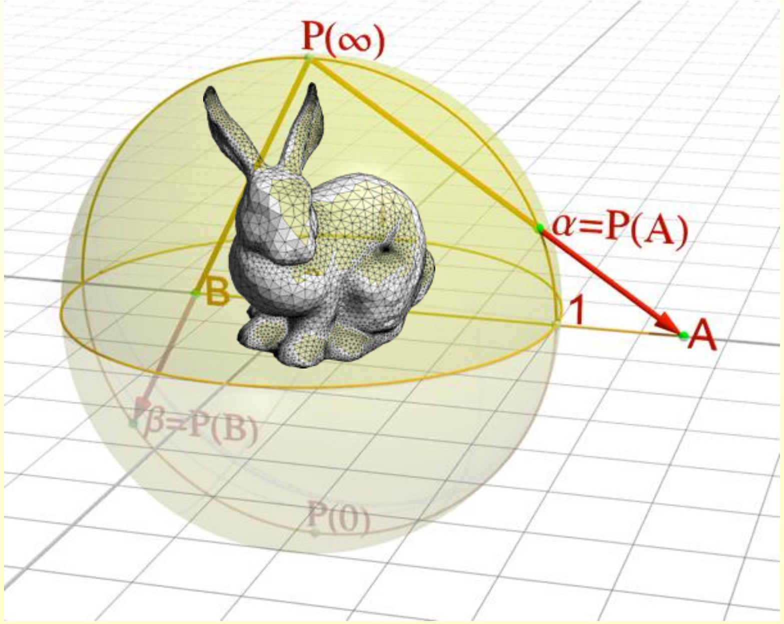


Different kinds of Parameterization



Different kinds of Parameterization

Since we are dealing with a triangle mesh, we first need to ensure a *bijective* map

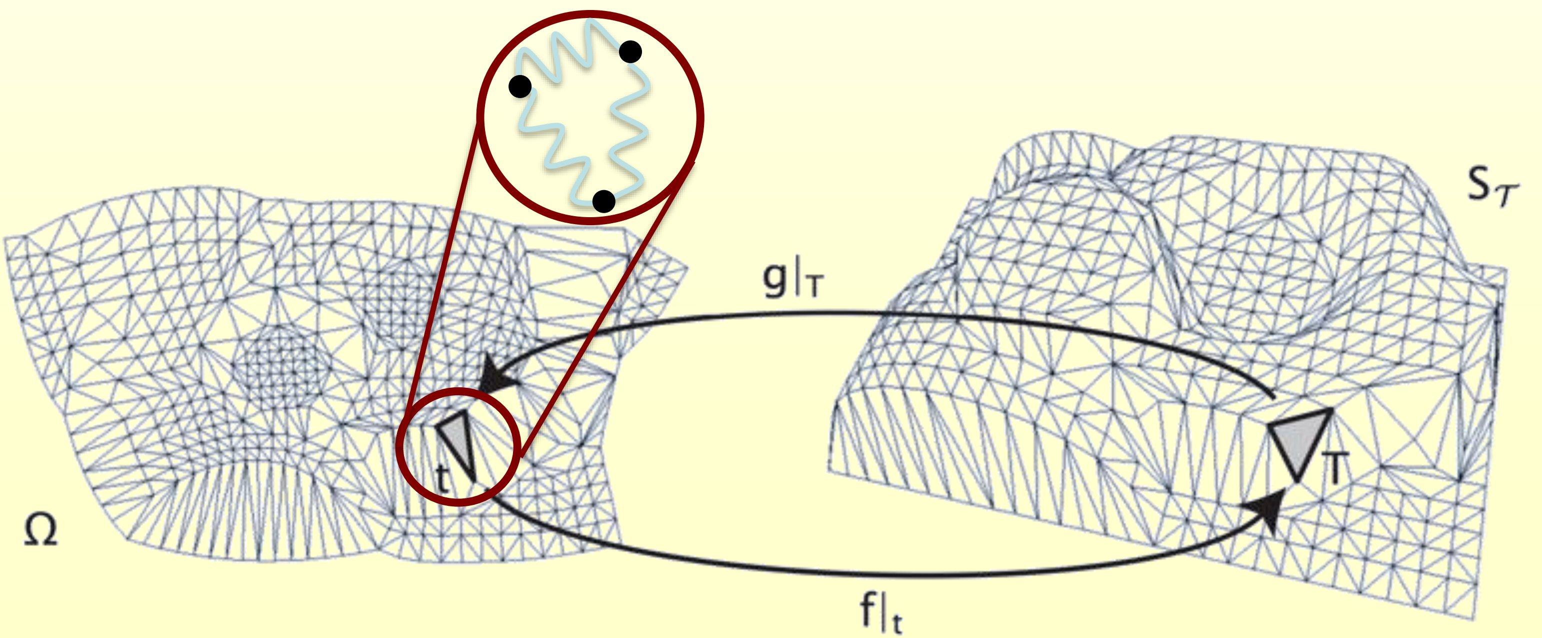


Spring Model for Parameterization

Given a mesh (T, P) in 3D find a bijective mapping $g(\mathbf{p}_i) = \mathbf{u}_i$

given constraints: $g(\mathbf{b}_j) = \mathbf{u}_j$ for some $\{\mathbf{b}_j\}$

Model: imagine a **spring** at each edge of the mesh.
If the boundary is fixed, let the interior points find an **equilibrium**.

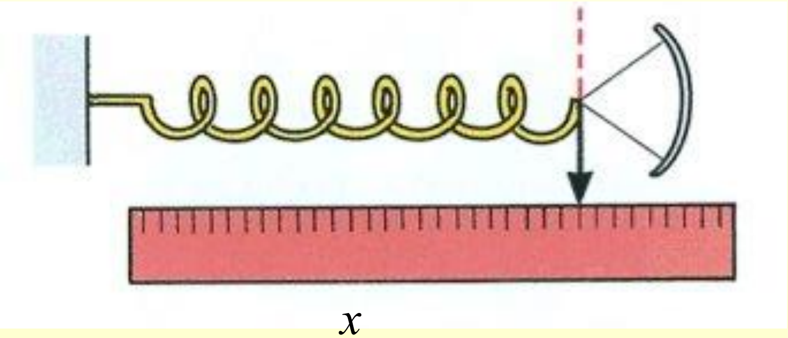


Spring Model for Parameterization

Recall: potential energy of a spring stretched by distance x:

$$E(x) = \frac{1}{2}kx^2$$

k: spring constant.



x

Spring Model for Parameterization

Given an embedding (parameterization) of a mesh, the potential energy of the whole system:

$$E = \sum_{e} \frac{1}{2} D_e \|\mathbf{u}_{e1} - \mathbf{u}_{e2}\|^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j \in \mathcal{N}_i} \frac{1}{2} D_{ij} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

Where $D_e = D_{ij}$ is the spring constant of edge e between i and j

Goal: find the coordinates $\{\mathbf{u}_i\}$ that would minimize E.

Note: the boundary vertices prevent the degenerate solution.

Parameterization with Barycentric Coordinates

Finding the optimum of:

$$E = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}_i}\frac{1}{2}D_{ij}\|\mathbf{u}_i - \mathbf{u}_j\|^2$$

$$\frac{\partial E}{\partial \mathbf{u}_i} = 0 \Rightarrow \quad \sum_{j\in\mathcal{N}_i} D_{ij}(\mathbf{u}_i - \mathbf{u}_j) = 0$$

$$\Rightarrow \quad \mathbf{u}_i = \sum_{j\in\mathcal{N}_i}\lambda_{ij}\mathbf{u}_j, \text{ where } \lambda_{ij} = \frac{D_{ij}}{\sum_{j\in\mathcal{N}_i} D_{ij}}$$

I.e. each point $\mathbf{u}_i$ must be an **convex combination** of its neighbors.

Hence: barycentric coordinates.

Parameterization with Barycentric Coordinates

To find the solution in practice:

1. Fix the boundary points $\mathbf{b}_i, i \in \mathcal{B}$
2. Form linear equations

$$\mathbf{u}_i = \mathbf{b}_i, \qquad \text{if } i \in \mathcal{B}$$

$$\mathbf{u}_i - \sum_{j \in \mathcal{N}_i} \lambda_{ij} \mathbf{u}_j = 0, \qquad \text{if } i \notin \mathcal{B}$$

1. Assemble into *two* linear systems (one for each coordinate):

$$LU = \bar{U}, \quad LV = \bar{V} \qquad L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\lambda_{ij} & \text{if } j \in \mathcal{N}_i, \ i \notin \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

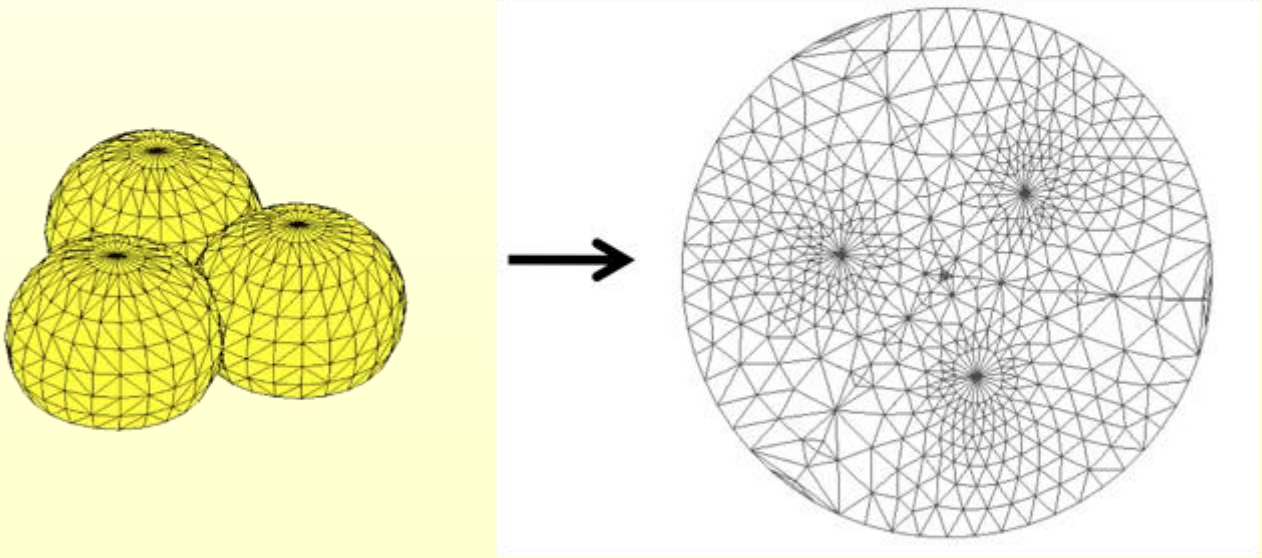
1. Solution of the linear system gives the coordinates:
   Note: system is very sparse, can solve efficiently. $\mathbf{u}_i = (u_i, v_i)$

Parameterization with Barycentric Coordinates

Does this work?

- **Theorem** (Maxwell-Tutte)

  If $G = <V,E>$ is a 3-connected planar graph (triangular mesh) then any **barycentric** drawing is a valid embedding.



Laplacian Matrix

Our system of equations (forgetting about boundary):

$$\mathbf{u}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij} \mathbf{u}_j , \text{ where } \lambda_{ij} = \frac{D_{ij}}{\sum_{j \in \mathcal{N}_i} D_{ij}}$$

$$LU = 0 \qquad L_{ij} = \begin{cases} 1 & \text{if } i = j \\ -\lambda_{ij} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

L is not symmetric

Alternatively, if we write it as:

$$\mathbf{u}_i \sum_{j \in \mathcal{N}_i} D_{ij} = \sum_{j \in \mathcal{N}_i} D_{ij} \mathbf{u}_j$$

We get:

$$LU = 0 \qquad L_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} D_{ij} & \text{if } i = j \\ -D_{ij} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

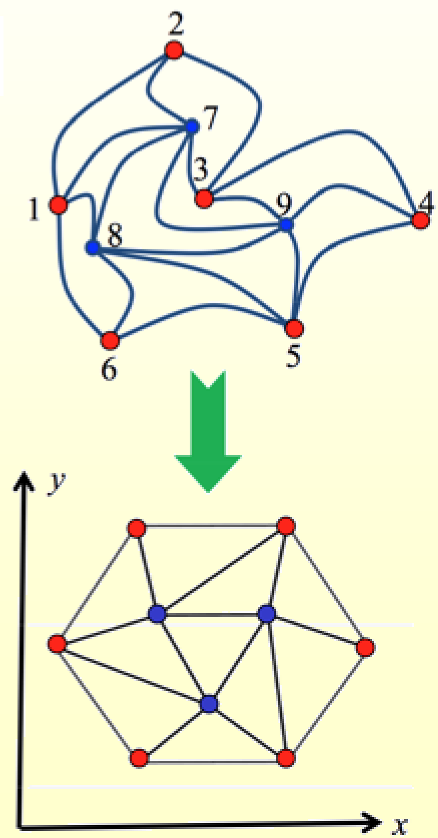
L is symmetric

Parameterization with Barycentric Coordinates

Example:

Uniform weights:

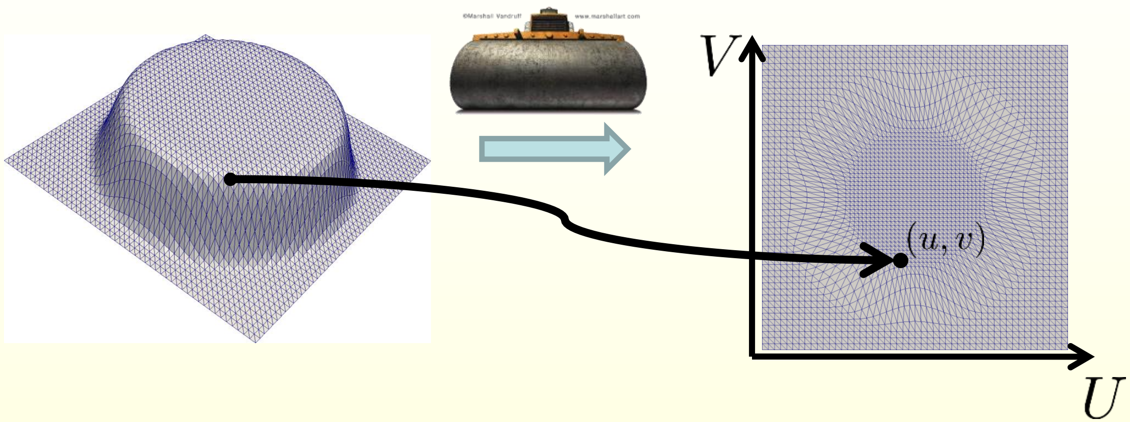
$$D_{ij} = 1$$



Laplacian Matrix

$$W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & -5 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & -5 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & -5 \end{pmatrix} \quad b_x = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad b_y = \begin{pmatrix} 2 \\ 3 \\ 3 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Alternative: Simple realization



Goal: Assign (*u*,*v*) coordinate to each mesh vertex.

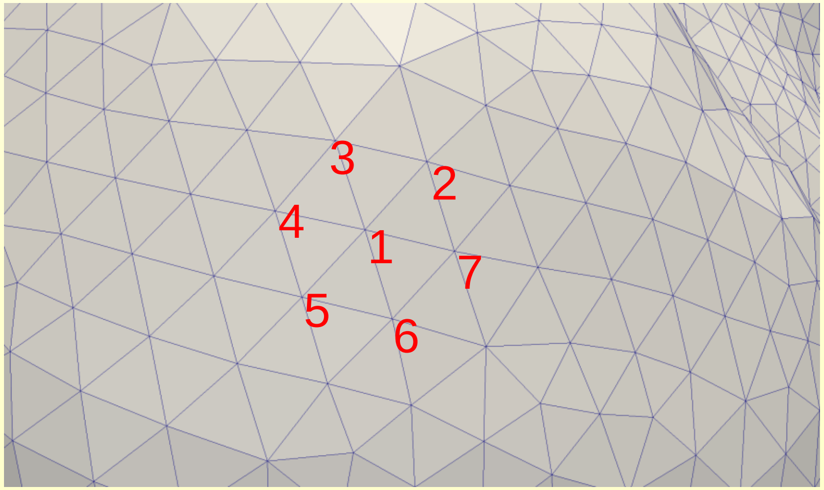
1. Fix (*u*,*v*) coordinates of boundary.
2. Want interior vertices to be at the center of mass of neighbors:

$$u_i = \frac{1}{|N(i)|} \sum_{j \in N(i)} u_j \qquad v_i = \frac{1}{|N(i)|} \sum_{j \in N(i)} v_j$$

Iterative Algorithm

1. Fix (*u*,*v*) coordinates of boundary.
2. Initialize (u,v) of interior points (e.g. using naïve).
3. While not converged: for each interior vertex, set:

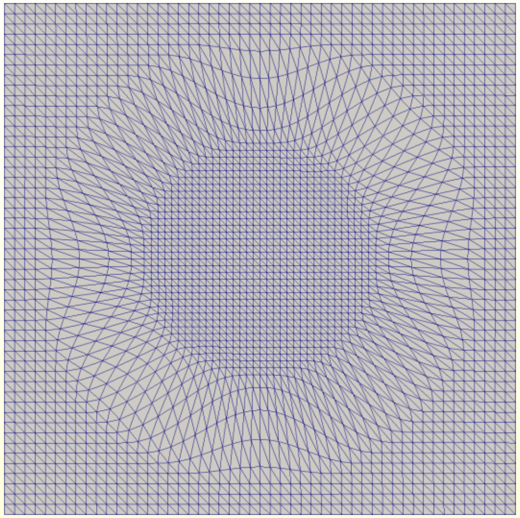
$$u_i \leftarrow \frac{1}{|N(i)|} \sum_{j \in N(i)} u_j \qquad v_i \leftarrow \frac{1}{|N(i)|} \sum_{j \in N(i)} v_j$$



$$u_1 \leftarrow \frac{u_2 + u_3 + u_4 + u_5 + u_6 + u_7}{6}$$

$$v_1 \leftarrow \frac{v_2 + v_3 + v_4 + v_5 + v_6 + v_7}{6}$$

What do you think?

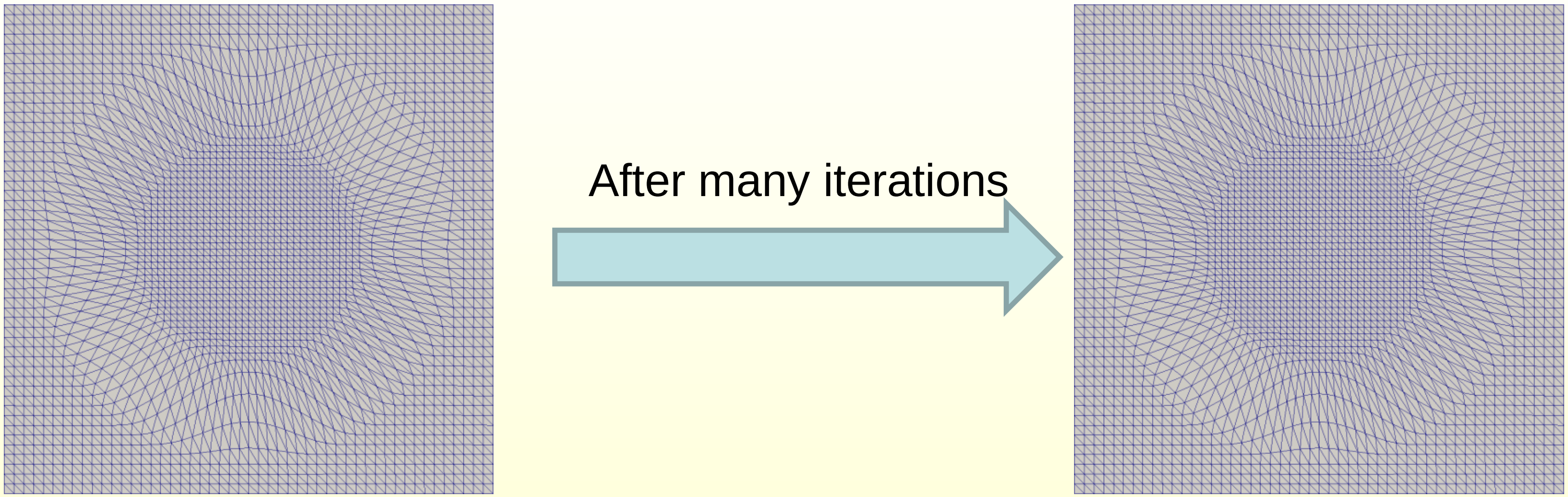


After many iterations

?

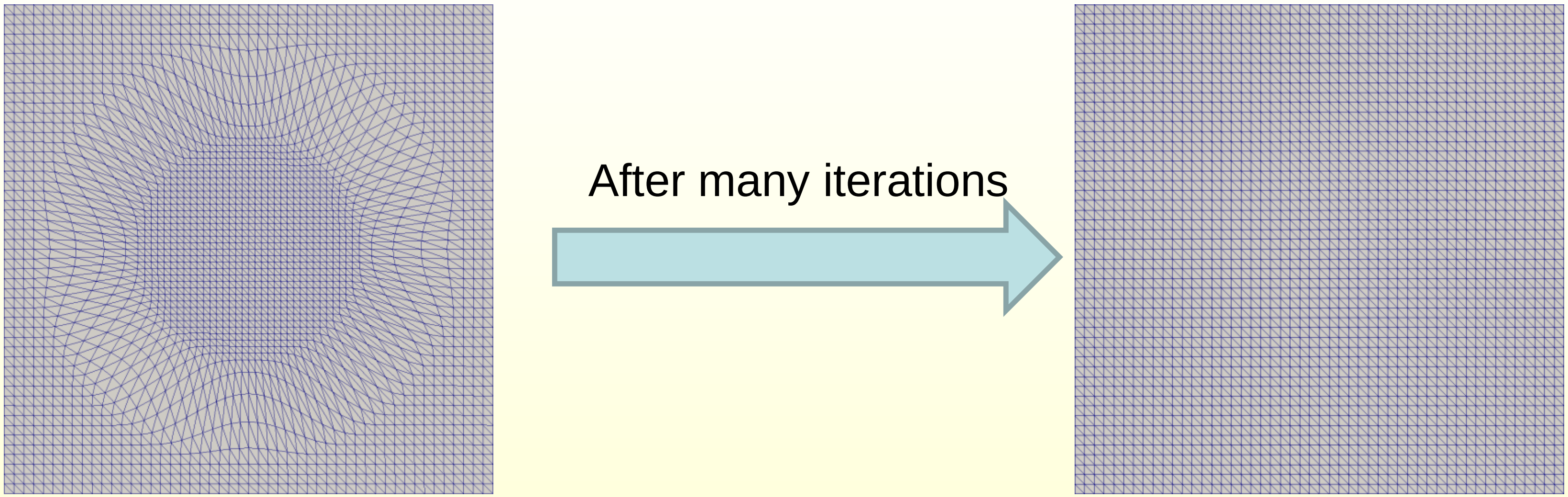
Some random planar mesh

Expectation



It is already planar: best parameterization = itself

Reality… Why? How to avoid?



Converges to a somewhat uniform grid!

Triangle shapes and sizes are not preserved!

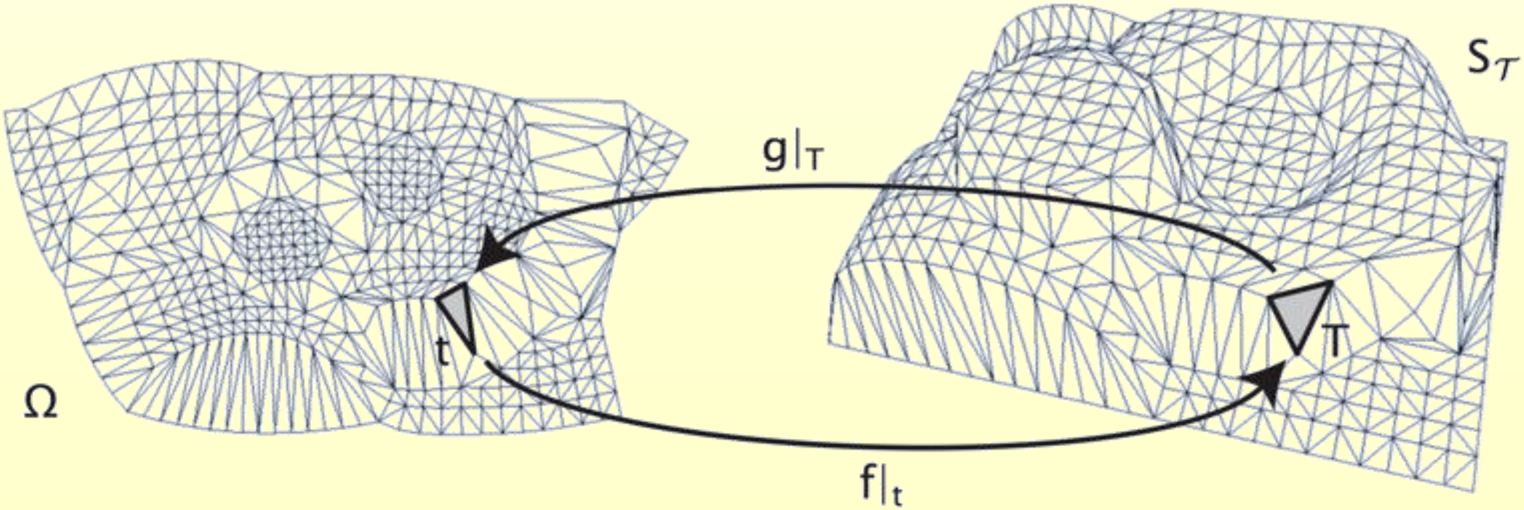
Parameterization with Barycentric Coordinates

Linear Reproduction:

- If the mesh is already **planar** we want to recover the original coordinates.

Problem:

- Uniform weights do not achieve linear reproduction
- Same for weights proportional to distances.



Parameterization with Barycentric Coordinates

Linear Reproduction:

- If the mesh is already **planar** we want to recover the original coordinates.

Problem:

- Uniform weights do not achieve linear reproduction
- Same for weights proportional to distances.

Solution:

- If the weights are **barycentric** with respect to **original points**:

$$\mathbf{p}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij} \mathbf{p}_j, \quad \sum_{j \in \mathcal{N}_i} \lambda_{ij} = 1$$

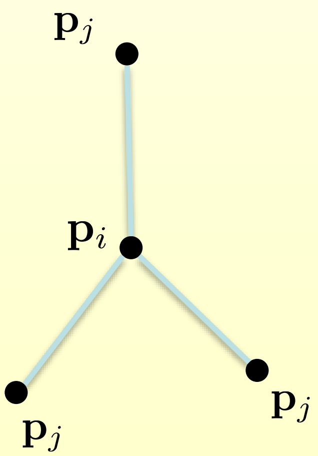
The resulting system will recover the planar coordinates.

Parameterization with Barycentric Coordinates

Solution:

- Barycentric coordinates with respect to original points:

$$\mathbf{p}_i = \sum_{j \in \mathcal{N}_i} \lambda_{ij} \mathbf{p}_j, \quad \sum_{j \in \mathcal{N}_i} \lambda_{ij} = 1$$



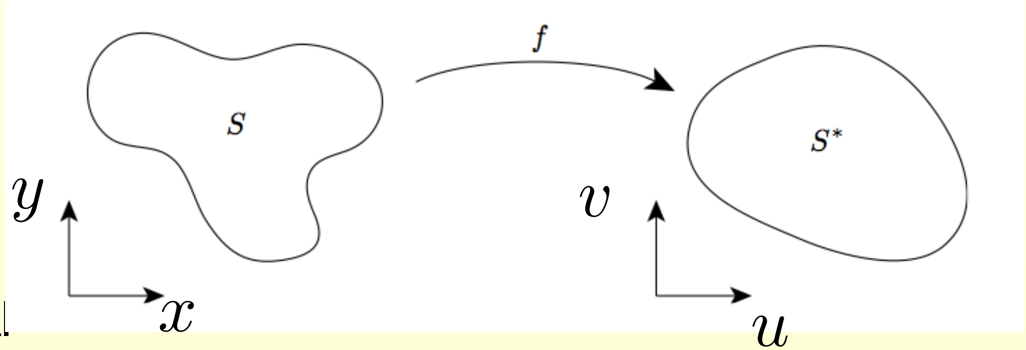
- If a point $\mathbf{p}_i$ has 3 neighbors, then the barycentric coordinates are **unique.**
- For more than 3 neighbors, many possible choices exist.

Conformal Mappings

Some good news.

Riemann Mapping Theorem:

Any surface topologically equivalent to a disk, **can be** conformally mapped to a unit disk.



Cauchy-Riemann equations:

If a map (x,y) → (u,v) is conformal then u(x,y) and v(x,y) satisfy:

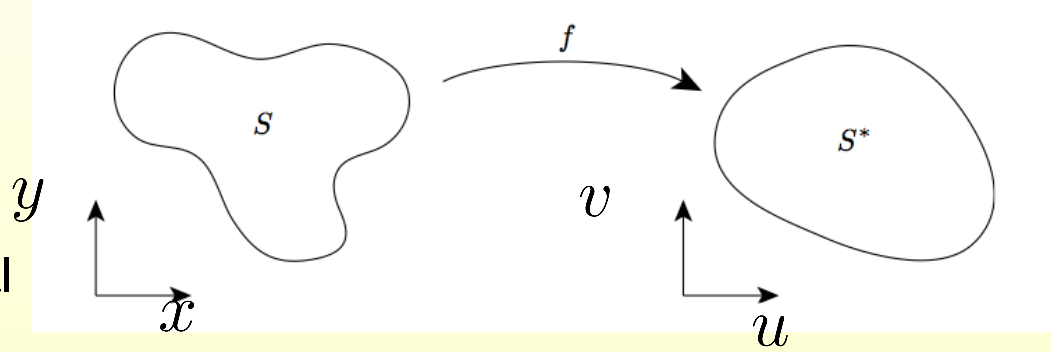
$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y},$$
$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

Conformal Mappings

Some good news.

Riemann Mapping Theorem:

Any surface topologically equivalent to a disk, **can be** conformally mapped to a unit disk.



Cauchy-Riemann equations:

If a map (x,y) → (u,v) is conformal then both u and v are harmonic:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) u = 0$$

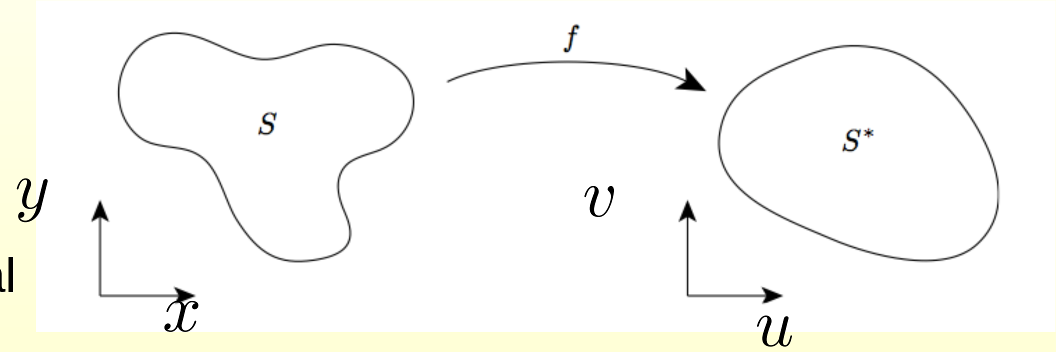
$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) v = 0$$

Conformal Mappings

Some good news.

Riemann Mapping Theorem:

Any surface topologically equivalent to a disk, **can be** conformally mapped to a unit disk.



Cauchy-Riemann equations:

If a map (x,y) → (u,v) is conformal then both u and v are harmonic:

$$\Delta u = 0$$

$$\Delta v = 0$$

Conformal Mappings

Some good news.

Riemann Mapping Theorem:

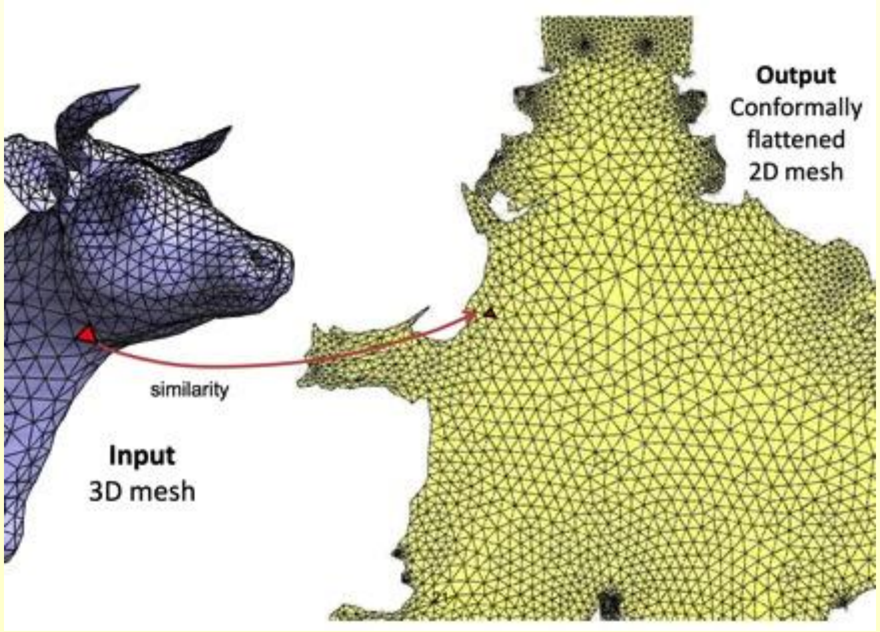
Any surface topologically equivalent to a disk, **can be** conformally mapped to a unit disk.

If a map $S \rightarrow$ (u,v) is conformal then both u and v are harmonic:

$$\Delta_S u = 0$$

$$\Delta_S v = 0$$

Δ_S : Laplace-Beltrami operator.



Harmonic Mappings

Recap:

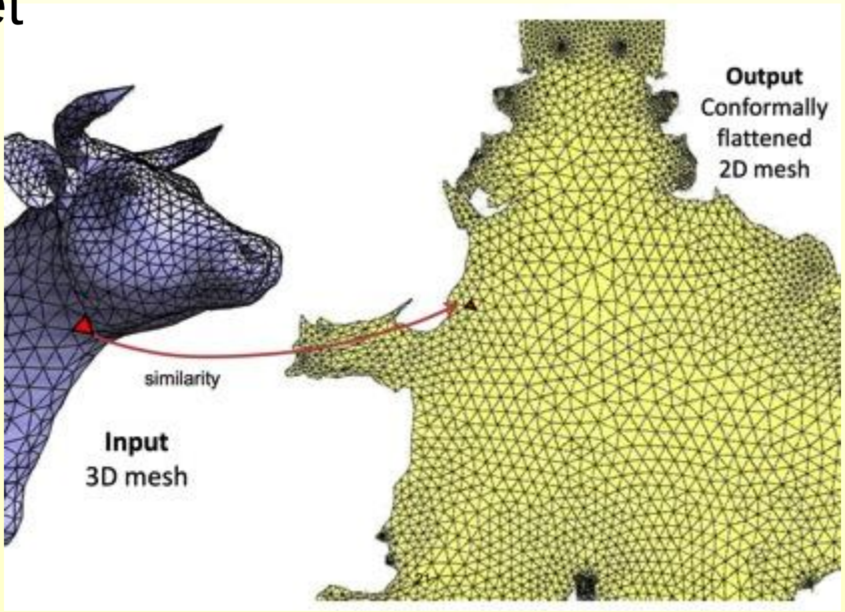
Isometric => Conformal => Harmonic

Harmonic mappings easiest to compute, but may not preserve angles. May not be bijective.

Harmonic maps minimize Dirichlet energy:

$$E_D(f) = \frac{1}{2} \sum_S \|\nabla_S f\|^2$$

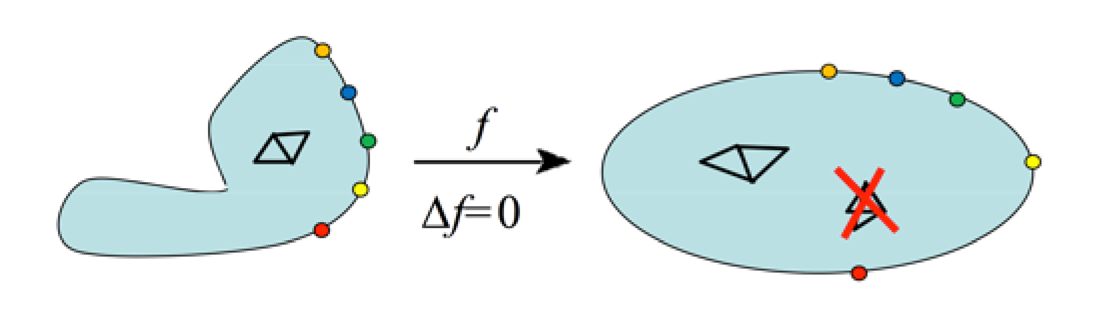
Given the boundary conditions.



Harmonic Mappings

Theorem (Rado-Kneser-Choquet):

If $f: S \rightarrow R^2$ is harmonic and maps the boundary ∂S onto the boundary ∂S^* of some convex region $S^* \subset R^2$, then f is **bijective**.



Recall the General Method:

To find the solution in practice:

1. Fix the boundary points $\mathbf{b}_i, i \in \mathcal{B}$

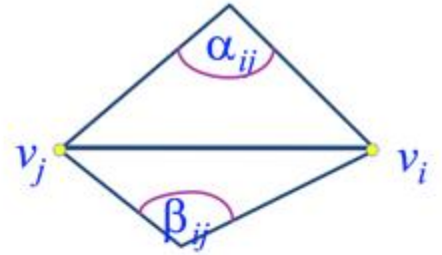
1. Assemble *two* linear systems (one for each coordinate):

$$LU = \bar{U}, \quad LV = \bar{V} \qquad L_{ij} = \begin{cases} 1 & \text{if } i = j,\ i \in \mathcal{B} \\ \sum_{j \in \mathcal{N}_i} D_{ij} & \text{if } i = j,\ i \notin \mathcal{B} \\ -D_{ij} & \text{if } j \in \mathcal{N}_i,\ i \notin \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

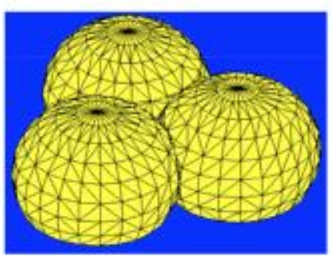
1. Solution of the linear system gives the coordinates: $\mathbf{u}_i = (u_i, v_i)$

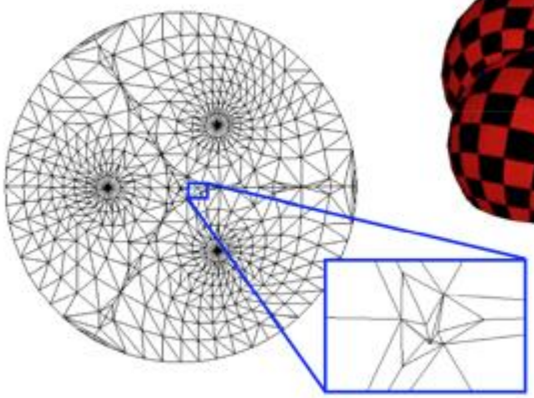
Barycentric Coordinates: Harmonic

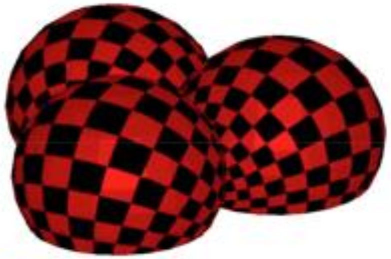
$$D_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2}$$



- Weights can be negative – not always valid
- Weights depend only on angles - close to conformal
- 2D reproducible

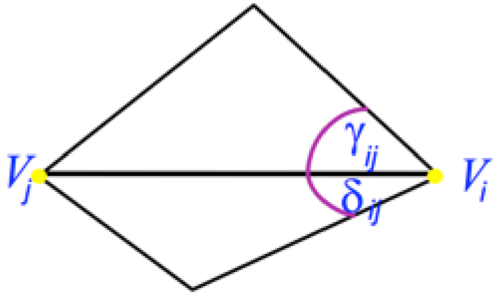




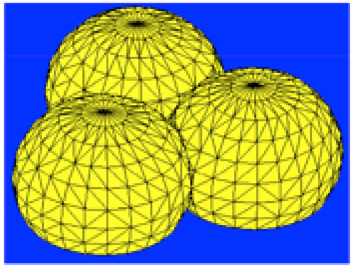


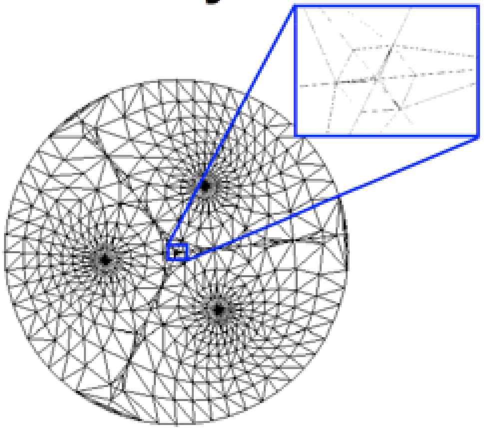
Barycentric Coordinates: Mean-value

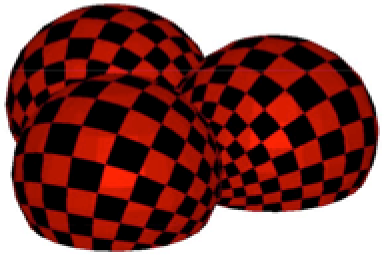
$$D_{ij} = \frac{\tan(\gamma_{ij}/2) + \tan(\delta_{ij}/2)}{2\,||V_i - V_j||}$$



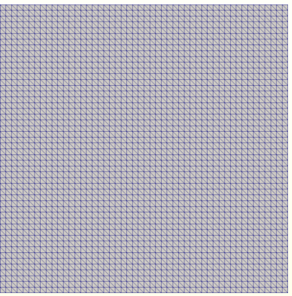
- Result visually similar to harmonic
- No negative weights – **always** valid
- 2D reproducible

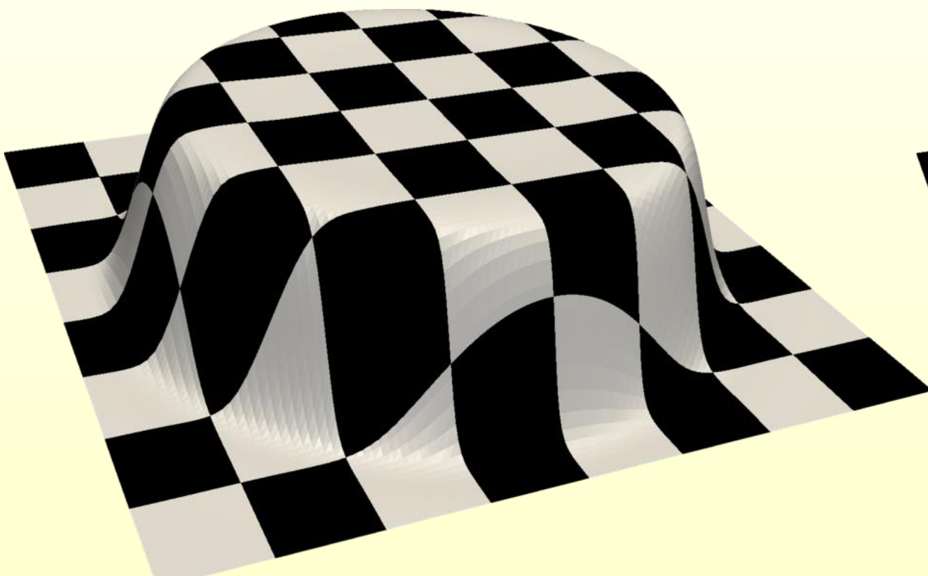




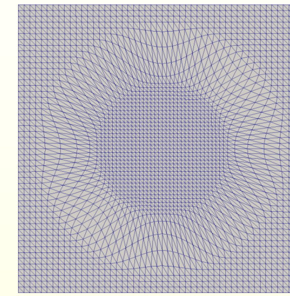


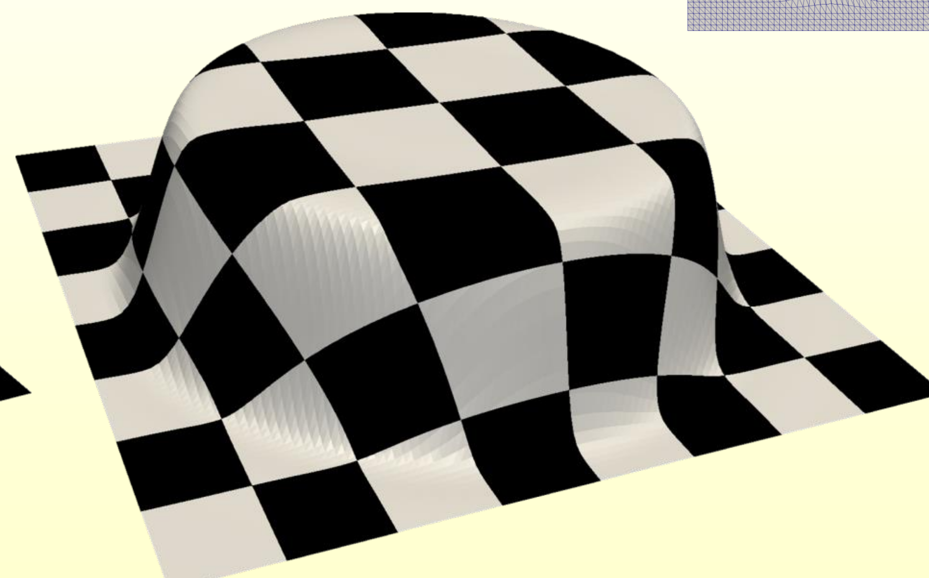
Results





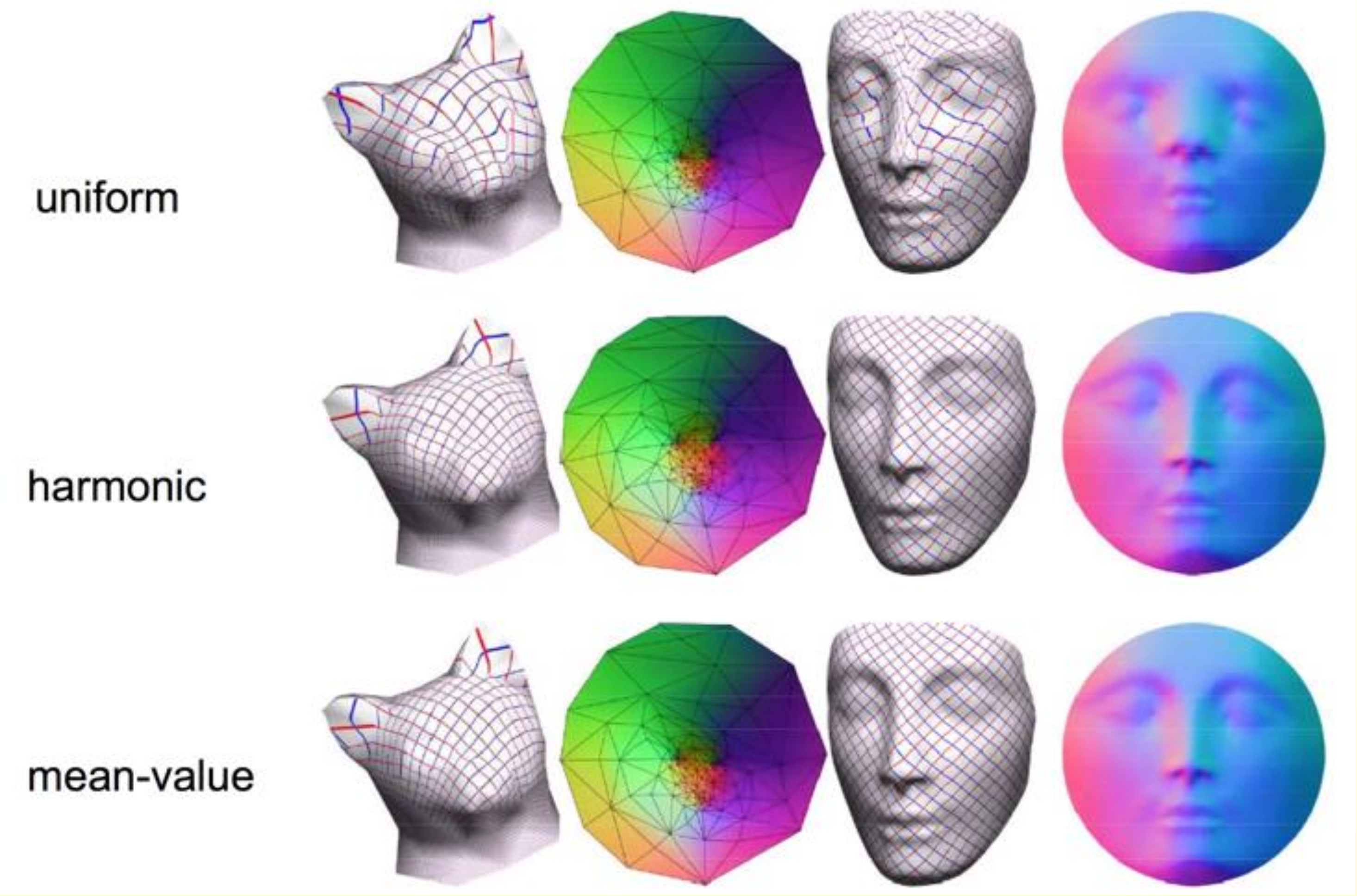
Naive





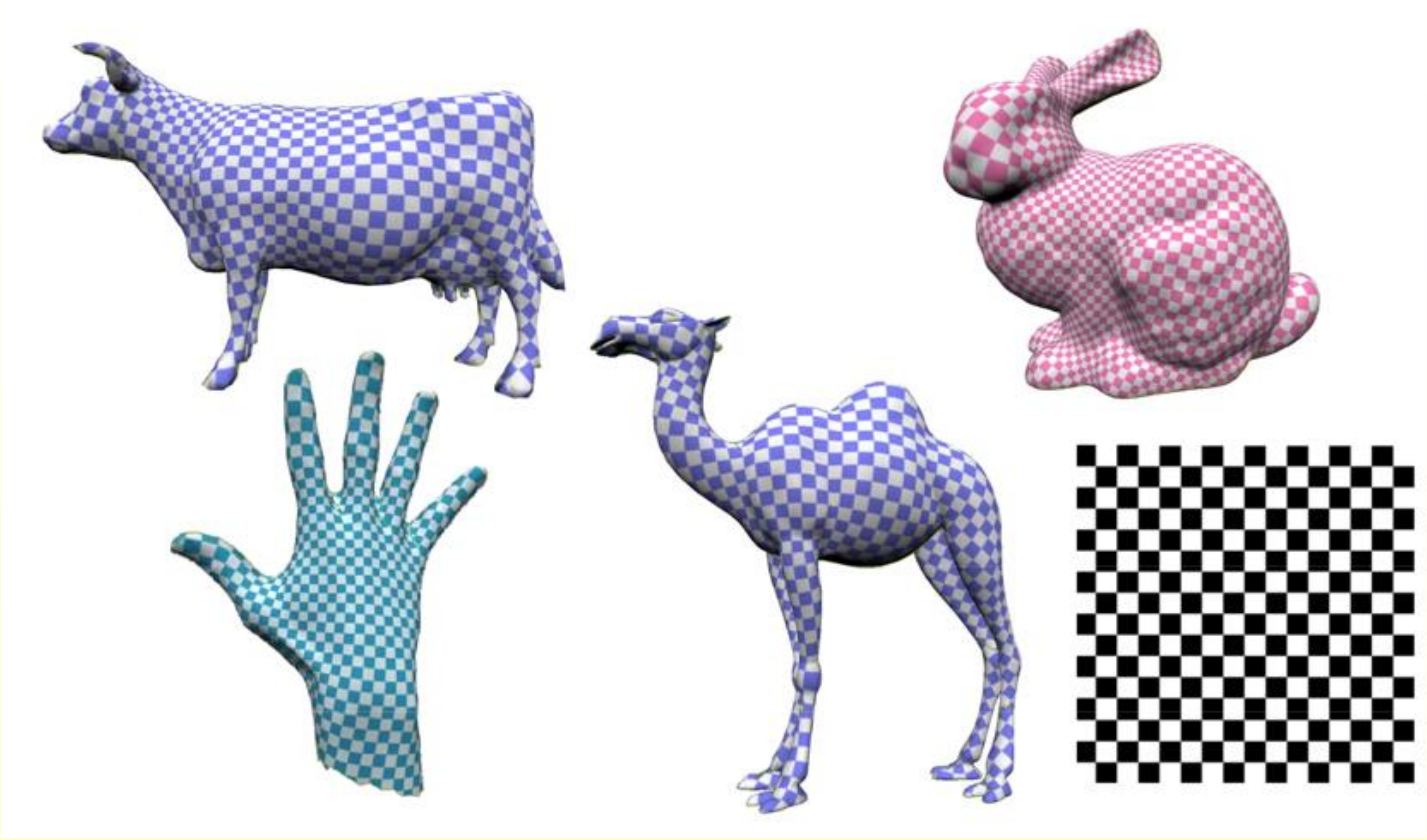
Harmonic

Barycentric Coordinates



Conformal Mappings

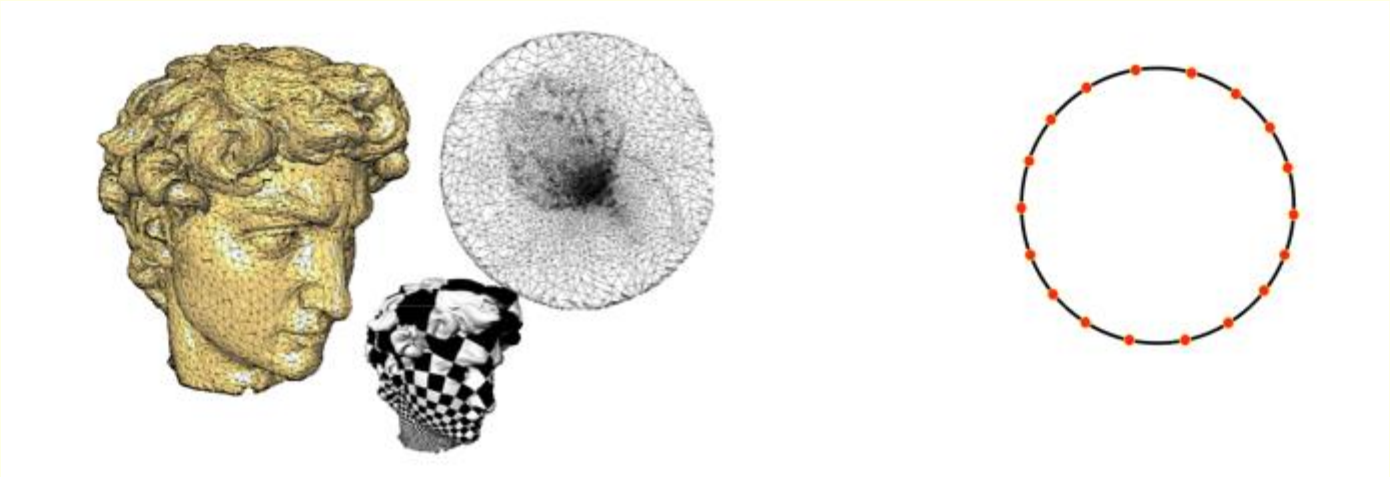
Most commonly used in practice.



Conformal Mappings

Fixing the boundary:

- Simple convex shape (triangle, square, circle)
- Distribute points on boundary
– Use chord length parameterization
- Fixed boundary can create high distortion



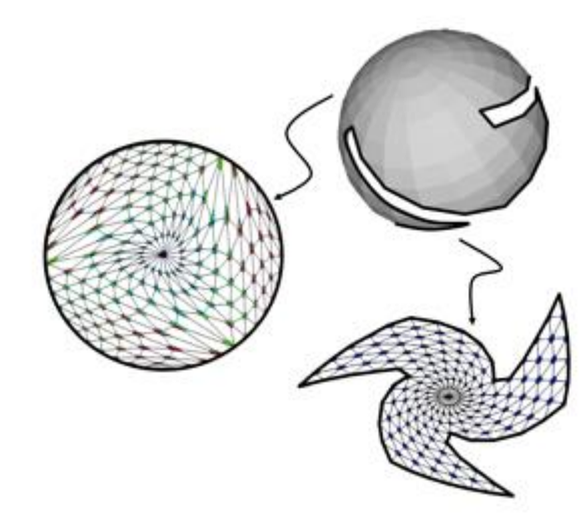
Conformal Mappings

Fixing the boundary:

- Simple convex shape (triangle, square, circle)
- Distribute points on boundary

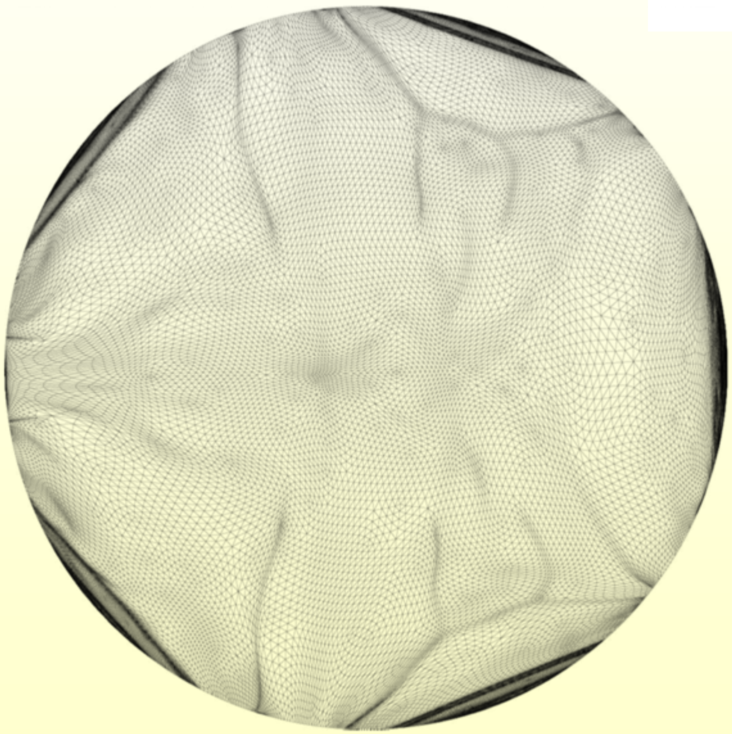
– Use chord length parameterization

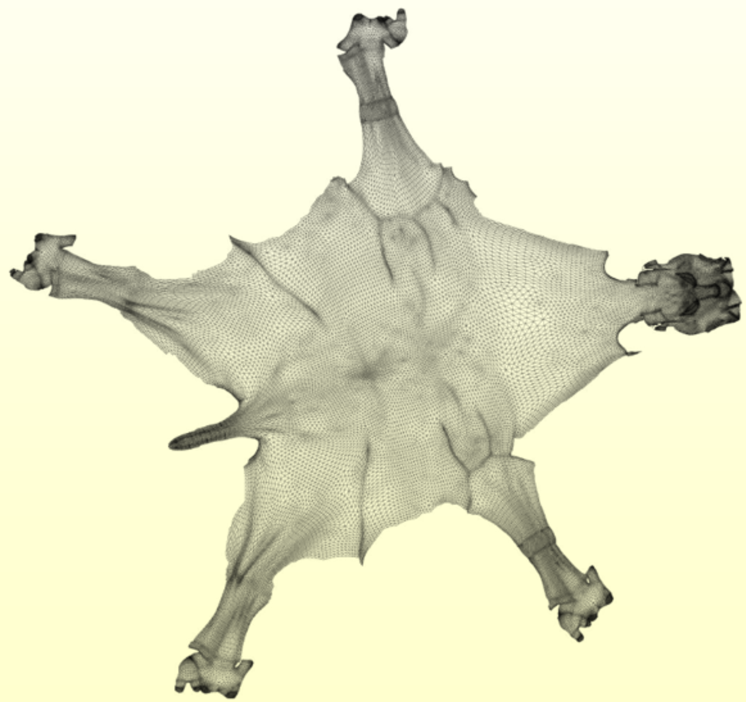
- Fixed boundary can create high distortion



“Free” boundary is better: harder to optimize for.

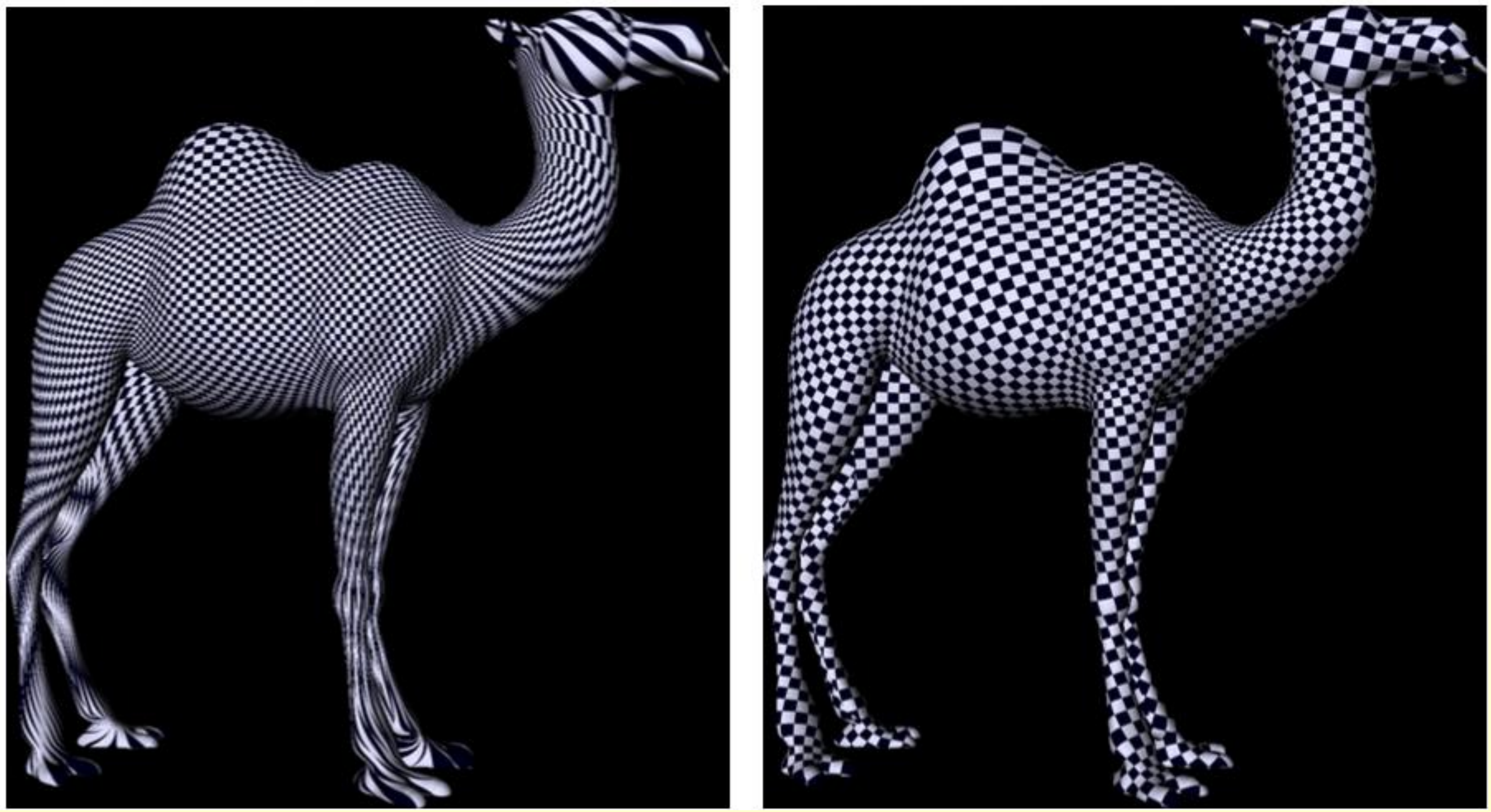
Fixed vs Free boundary





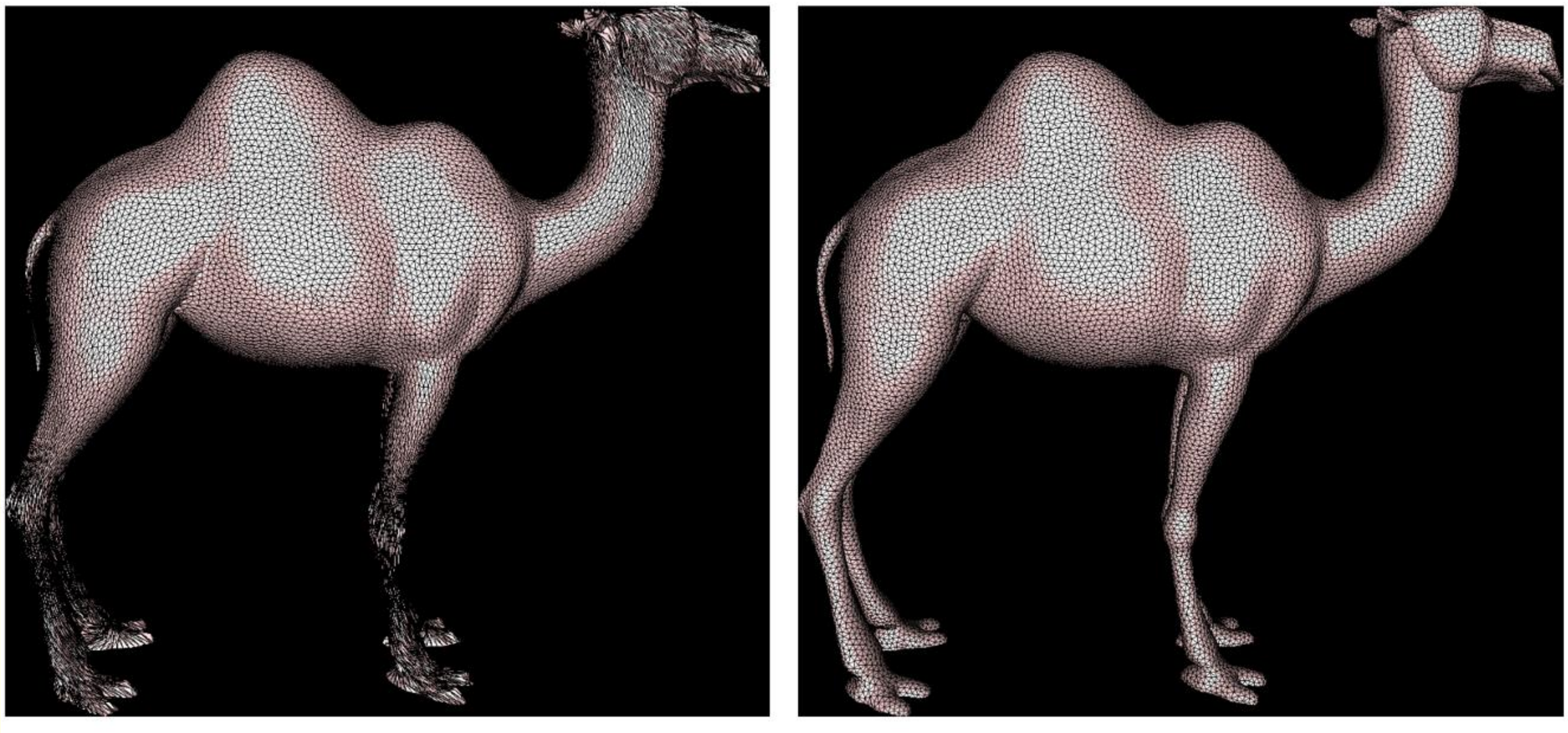
images by Mirela Ben-Chen

Fixed vs Free boundary



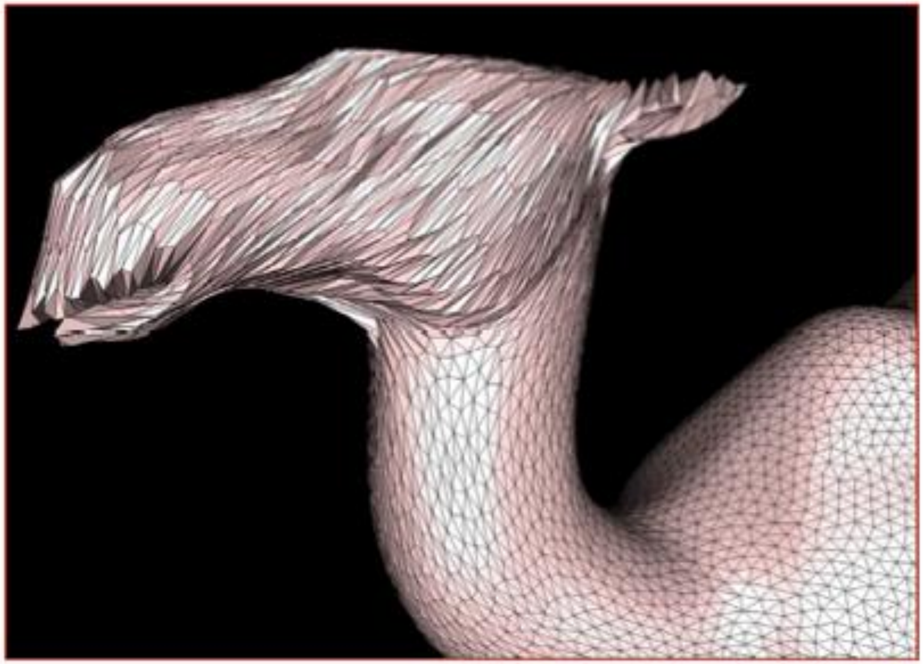
images by Mirela Ben-Chen

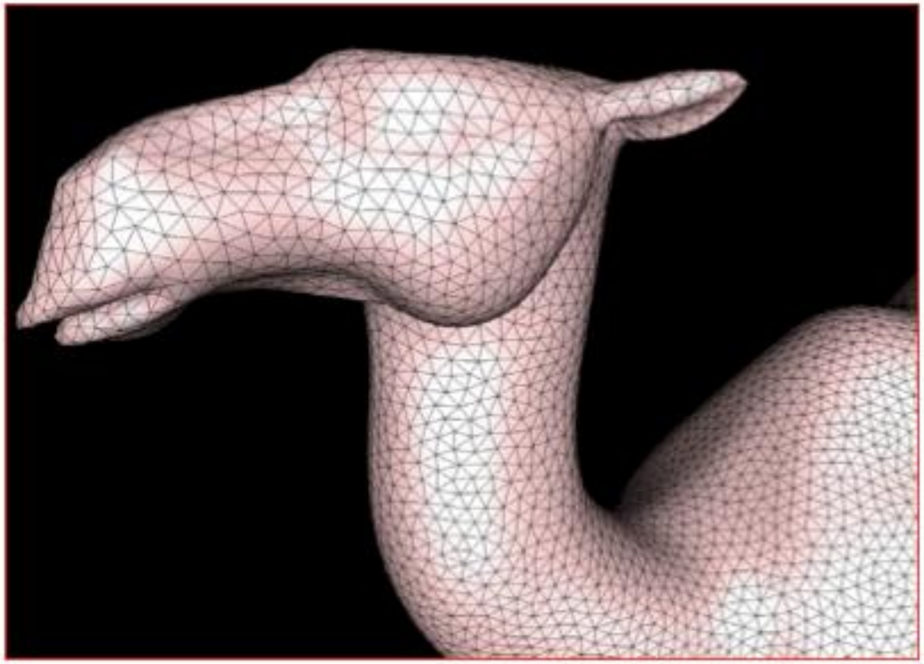
Fixed vs Free boundary



images by Mirela Ben-Chen

Fixed vs Free boundary



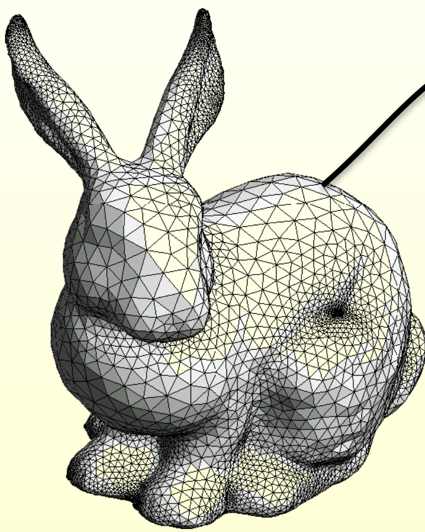


images by Mirela Ben-Chen

Free boundary methods

General approach:

f



$$E(f) = \sum_{t \in \text{triangles}} \text{distortion}(t|f)$$

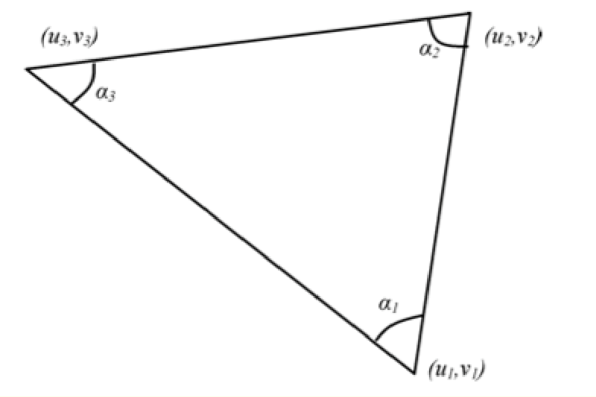
Let the coordinates of the vertices be unknowns, construct an energy that measures distortion.

$$(u_{\text{opt}}, v_{\text{opt}}) = \arg\min_{f=(u,v)} E(f)$$

given boundary conditions

Free boundary methods

For a any triangle:



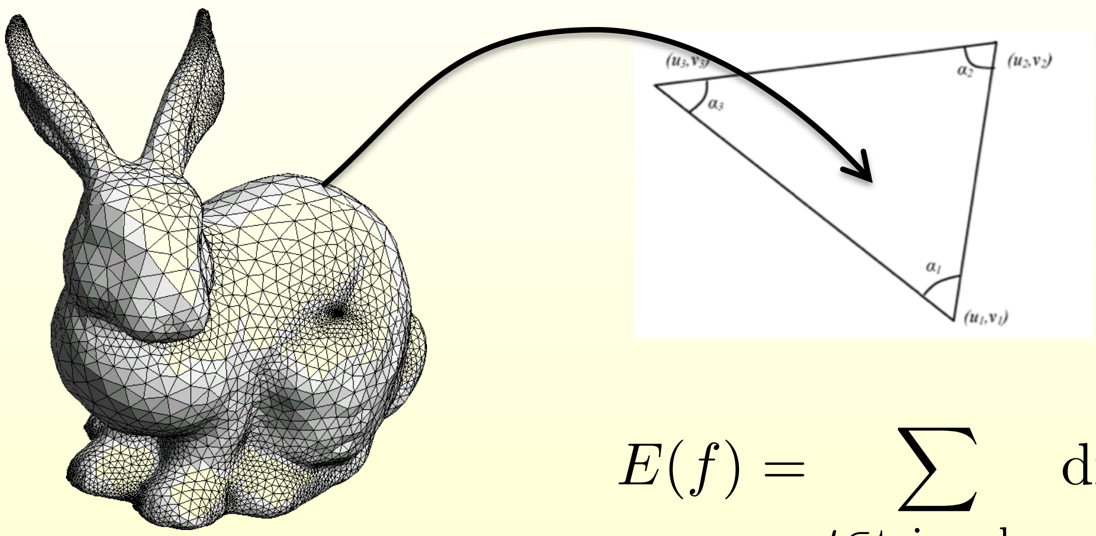
$$(u_3, v_3) - (u_1, v_1) = \frac{\sin \alpha_2}{\sin \alpha_3} R^{\alpha_1} [(u_2, v_2) - (u_1, v_1)]$$

If the mapping is conformal, the angles shouldn’t change. Keep the angles, let the coordinates be unknown. Leads to a least squares problem.

Least squares conformal maps for automatic texture atlas generation, Levy et al., SIGGRAPH 2002

Free boundary methods

More general:



$$E(f) = \sum_{t \in \mathrm{triangles}} \mathrm{distortion}(t|f)$$

$$\mathrm{distortion}(t|f) = H(J_f(t))$$

$J_f(t)$: Jacobian of the transformation

Free boundary methods

More generally:

$$\text{distortion}(t|f) = H(J_f(t))$$

$J_f(t)$: Jacobian of the transformation

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

1. Isometric mapping: $\sigma_1 = \sigma_2 = 1$
2. Conformal mapping: $\sigma_1 / \sigma_2 = 1$
3. Equiareal mapping: $\sigma_1 \sigma_2 = 1$

Free boundary methods

More generally:

$$\text{distortion}(t|f) = H(J_f(t))$$

$J_f(t)$: Jacobian of the transformation

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

$H(J_f(t)) = H(\sigma_1, \sigma_2)$, e.g.:

$$H_{\text{MIPS}}(\sigma_1, \sigma_2) = \frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}$$

Non-linear, difficult to optimize for.

MIPS: An efficient global parameterization method, Hormann and Greiner, Curve and Surface design, '99

Free boundary methods

More generally:

$$\mathrm{distortion}(t|f) = H(J_f(t))$$

$J_f(t)$: Jacobian of the transformation

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

Can show that:

$\sigma_1^2 + \sigma_2^2 \quad \text{and} \quad \sigma_1\sigma_2$ are *quadratic* in the target vertex coordinates.

Thus, e.g. $H(\sigma_1, \sigma_2) = (\sigma_1 - \sigma_2)^2$ leads to a linear system of equations.

Surface Parameterization: a Tutorial and Survey, Floater and Hormann, AMGM, 2005

Free boundary methods

More generally:

$$\text{distortion}(t|f) = H(J_f(t))$$

$J_f(t)$: Jacobian of the transformation

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

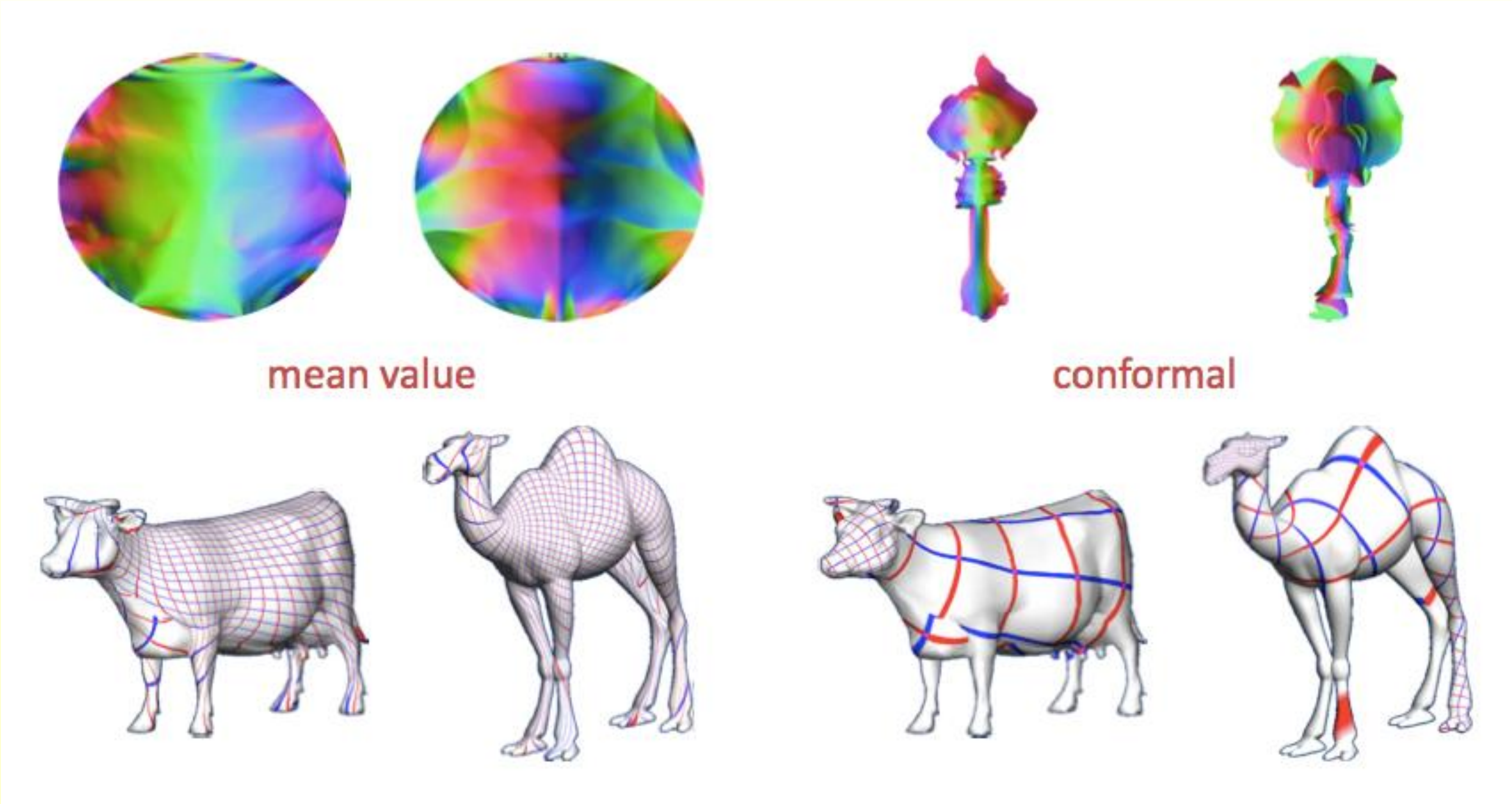
Can show that:

$\sigma_1^2 + \sigma_2^2 \quad \text{and} \quad \sigma_1\sigma_2$ are *quadratic* in the target vertex coordinates.

Thus, e.g. $H(\sigma_1, \sigma_2) = (\sigma_1 - \sigma_2)^2$ leads to a linear system of equations.

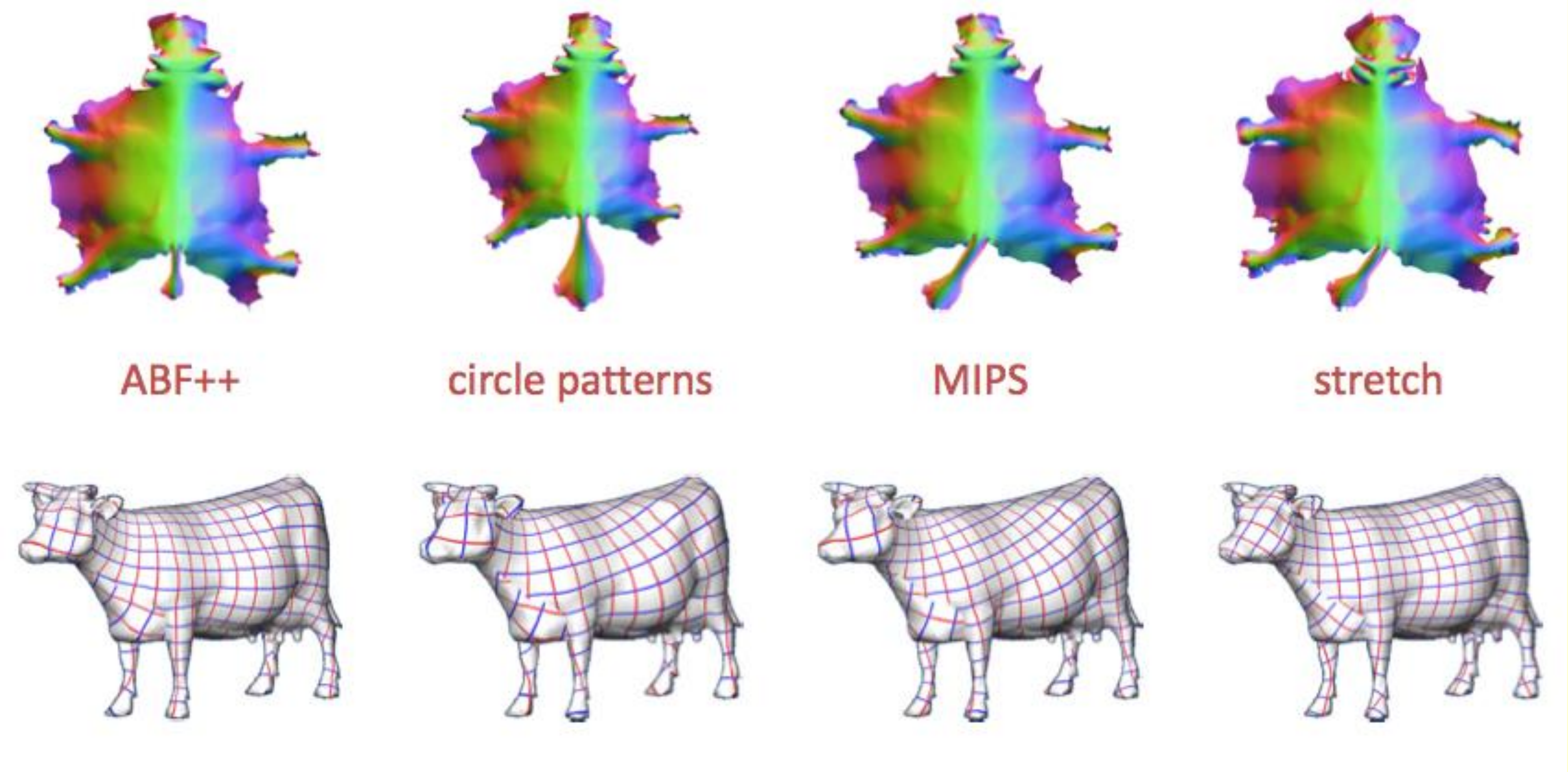
Some results

Linear Methods:



Some results

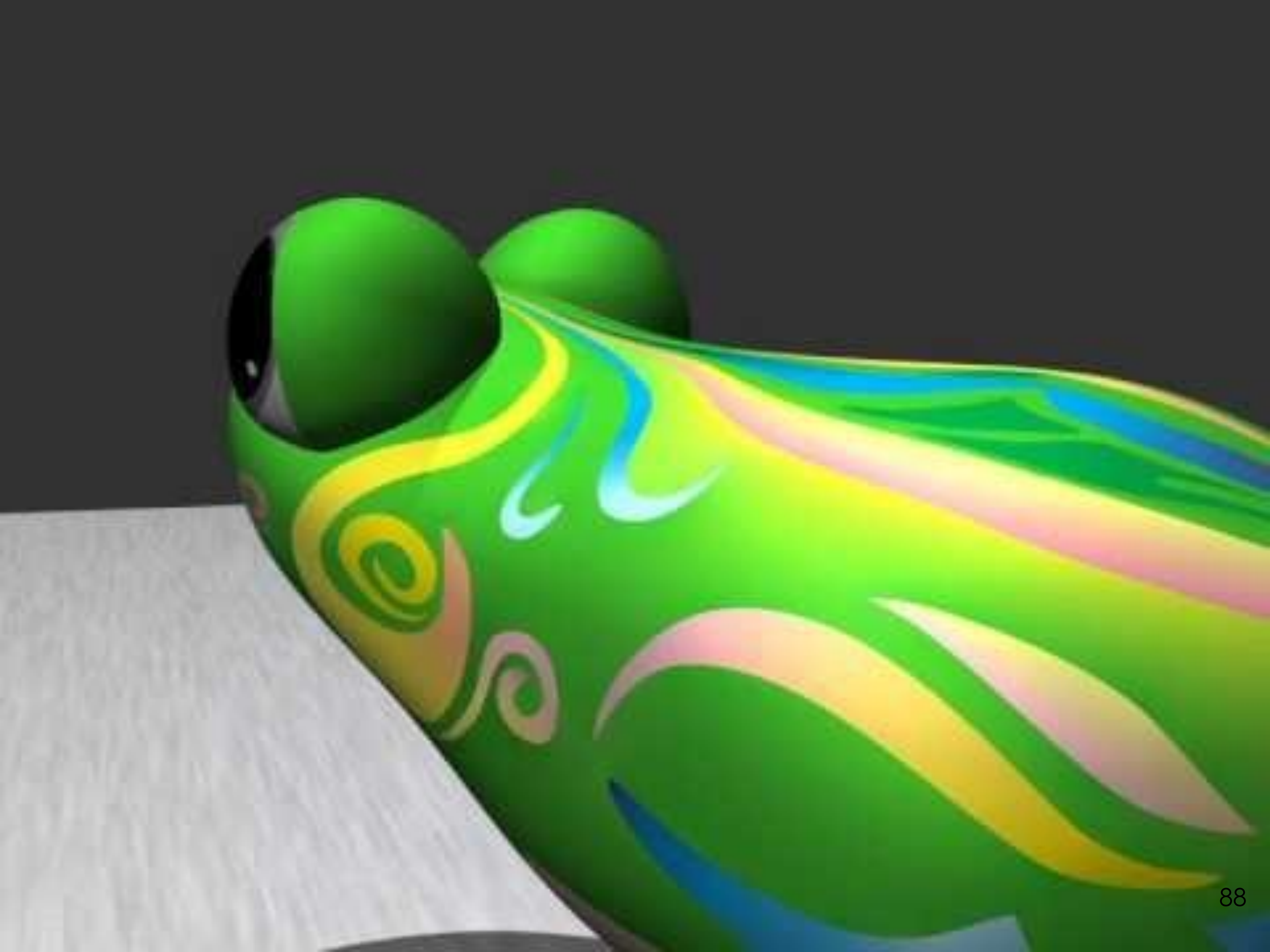
Non-linear Methods:



Conclusions

Surface parameterization:

- No perfect mapping method
- A very large number of techniques exists
- Conformal model:
  - Nice theoretical properties
  - Leads to a simple (linear) system of equations
  - Closely related to the Poisson equation and Laplacian operator
- More general methods
  - Can get smaller distortion using non-linear optimization
  - Very difficult to guarantee bijectivity in general



Breathing Type: Normal

Comparing Real vs. Animated Breathing


DYNAMIC MICROGEOMETRY


Rig Animation with a Tangible and
Modular Input Device