252-0538-00L, Spring 2024

Shape Modeling and Geometry Processing

Smoothing and Mesh Parameterization

INTERACTIVE GEOMETRY LAB

April 2, 2025

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Smoothing as optimization

igl

ETH zürich

Minimizing a some energy

- General optimization approach:
  - Define some energy $E(\mathbf{p})\colon \mathbb{R}^N \to \mathbb{R}$
  - Minimize $E(\mathbf{p})$, many solvers available.

- Example
  - $E_{\text{smooth}}(\mathbf{p})$ measureing smoothness
  - $E_{\text{error}}(\mathbf{p})$ measuring error to original surface
  - $E_1(\mathbf{p}) + E_2(\mathbf{p})$ easy to combine

Minimizing a smoothness energy

$$\Delta_{\mathcal{M}} \mathbf{p} = -H\mathbf{n}$$

goal: $H = 0$ or $H = \text{const}$

- Just Minimize H? 🤔
- Problem ⚠️: Avoid trivial solution $\mathbf{p} = 0$

Let's regularize💡!

weighting factor

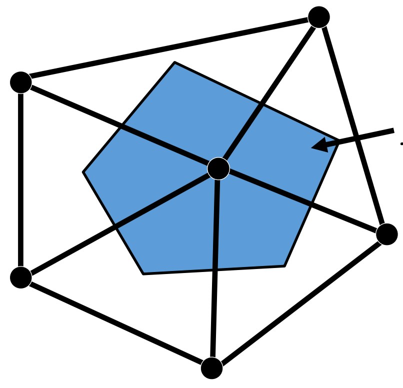
$$\min_{\tilde{\mathbf{p}}} \int_{\mathcal{M}} \|\Delta_{\mathcal{M}} \tilde{\mathbf{p}}\|^2 + w\|\tilde{\mathbf{p}} - \mathbf{p}\|^2$$

small H — stay close to original surface

$E_{\text{smooth}}(\mathbf{p})$ — $E_{\text{error}}(\mathbf{p})$

Minimizing a smoothness energy

$$\min_{\tilde{\mathbf{p}}} \int_{\mathcal{M}} \|\Delta_{\mathcal{M}} \tilde{\mathbf{p}}\|^2 + w\|\tilde{\mathbf{p}} - \mathbf{p}\|^2$$



- Discretize:

$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L\tilde{\mathbf{p}}_i\|^2 + w\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2\right)$$

- Minimize!

$$\frac{\partial E}{\partial \mathrm{p}} = 0$$

remember: $\frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^\top A \mathbf{x}) = (A + A^\top)\mathbf{x}$

Good source: “The Matrix Cookbook”

Minimizing a smoothness energy

Goal:

$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L\tilde{\mathbf{p}}_i\|^2 + w\|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2\right) \qquad \tilde{\mathbf{p}} = [\tilde{\mathbf{x}}\ \tilde{\mathbf{y}}\ \tilde{\mathbf{z}}] = \begin{pmatrix} \tilde{p}_{1x} & \tilde{p}_{1y} & \tilde{p}_{1z} \\ \tilde{p}_{2x} & \tilde{p}_{2y} & \tilde{p}_{2z} \\ \vdots & \vdots & \vdots \\ \tilde{p}_{nx} & \tilde{p}_{ny} & \tilde{p}_{nz} \end{pmatrix} \in \mathbb{R}^{n\times 3}$$

$$E(\tilde{\mathbf{p}}) = \sum_{\mathbf{v}\in\{\mathbf{x},\mathbf{y},\mathbf{z}\}} (L\tilde{\mathbf{v}})^{\mathsf{T}} M\,(L\tilde{\mathbf{v}}) + w(\tilde{\mathbf{v}} - \mathbf{v})^{\mathsf{T}} M(\tilde{\mathbf{v}} - \mathbf{v})$$

$$\frac{\partial E}{\partial \tilde{\mathbf{p}}} = 2L^{\mathsf{T}} M L\tilde{\mathbf{p}} + 2wM(\tilde{\mathbf{p}} - \mathbf{p}) \stackrel{!}{=} 0$$

$$\Rightarrow \left(L^{\mathsf{T}} M L + wM\right)\tilde{\mathbf{p}} = wM\mathbf{p}$$

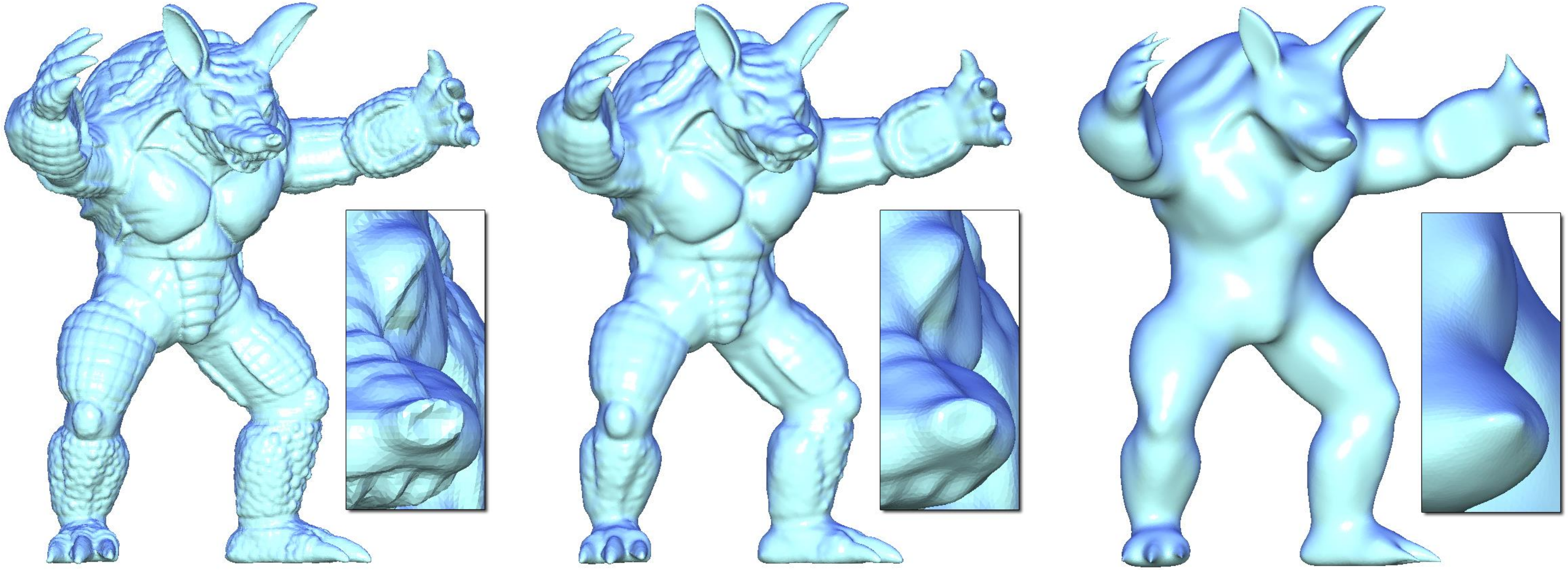
$$(I - dt\,\lambda L)\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)}$$

compare with previous implicit Euler!

With: $L = M^{-1}L_w \;\Rightarrow\; L^{\mathsf{T}} M L = L_w M^{-1} L_w$

Results

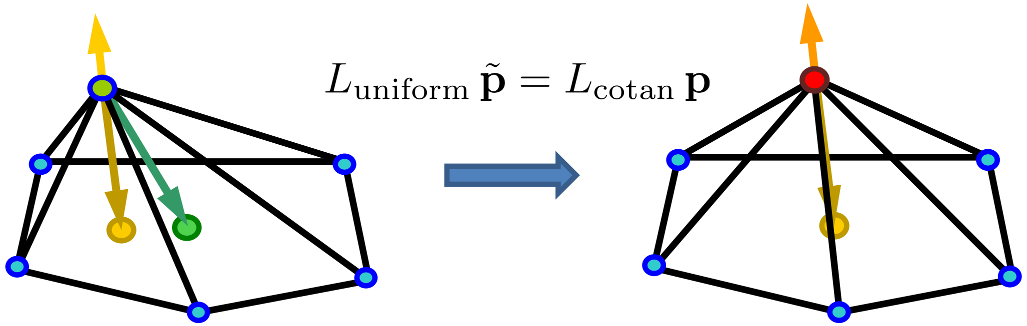
$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\|L\tilde{\mathbf{p}}_i\|^2 + w \|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2 \right)$$



original $\qquad w = 0.2 \qquad w = 0.02$

Customize the energy functional

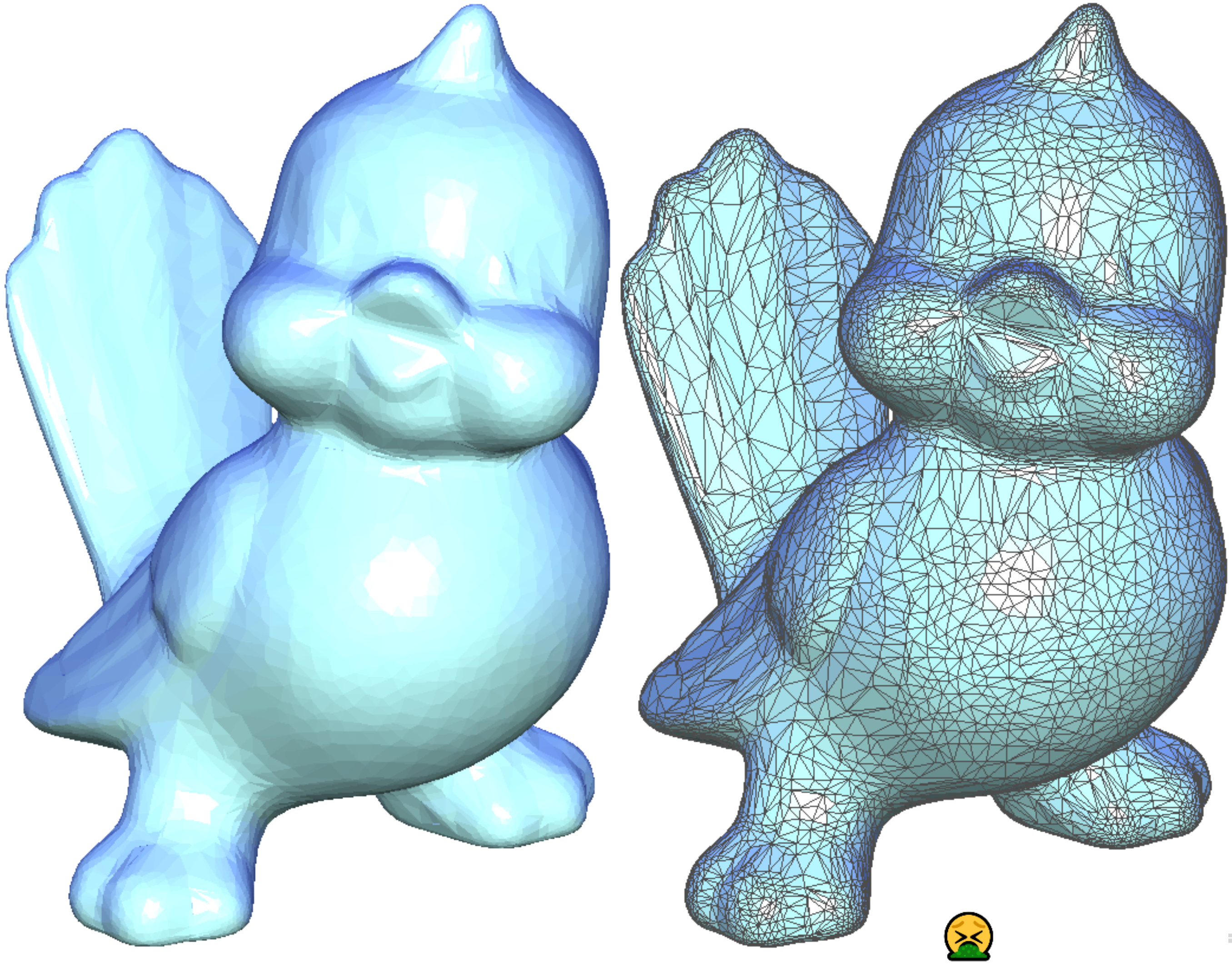
- Can do *tangential* smoothing!
  - Improve the shapes of mesh triangles without changing the surface shape



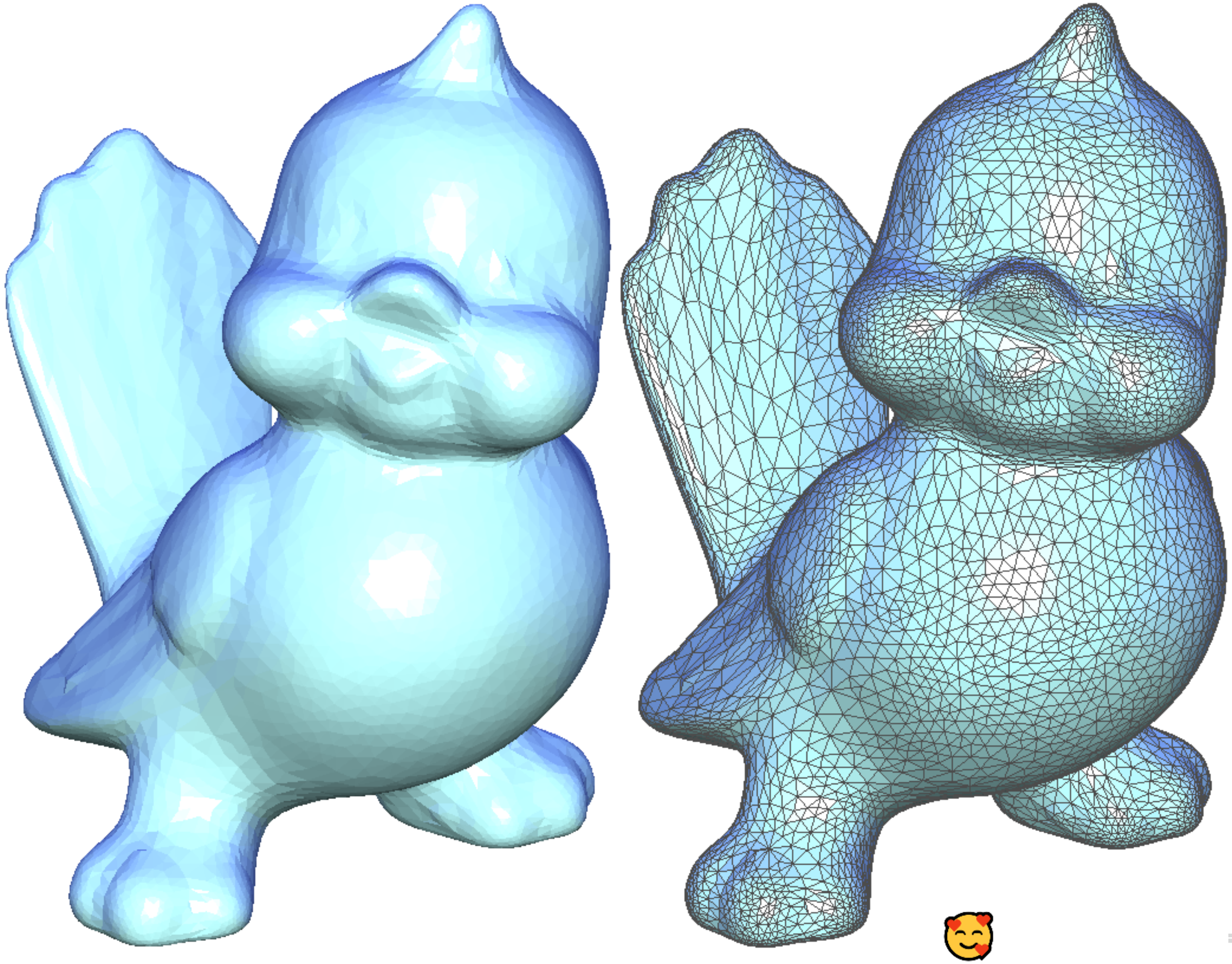
$$L_{\text{uniform}}\,\tilde{\mathbf{p}} = L_{\text{cotan}}\,\mathbf{p}$$

$$\min_{\tilde{\mathbf{p}}} \sum_{i=1}^{n} A_i \left(\| L_{\text{uniform}}\,\tilde{\mathbf{p}}_i - L_{\text{cotan}}\,\mathbf{p}_i \|^2 + w \|\tilde{\mathbf{p}}_i - \mathbf{p}_i\|^2 \right)$$

Original



Triangle Shape Optimization



Smoothing as filtering

Fourier analysis

Joseph Fourier (1768 - 1830)

- *Fourier:*
  The space of all periodic functions forms a vector space with a basis of $\cos(\ )$ and $\sin(\ )$ functions.

$f: [0,2\pi] \to \mathbb{R},$ is periodic $\Rightarrow \exists\ a_n, b_n \in \mathbb{R}, n \in \mathbb{N}$ such that

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi n x) + b_n \sin(2\pi n x)$$

Fourier analysis

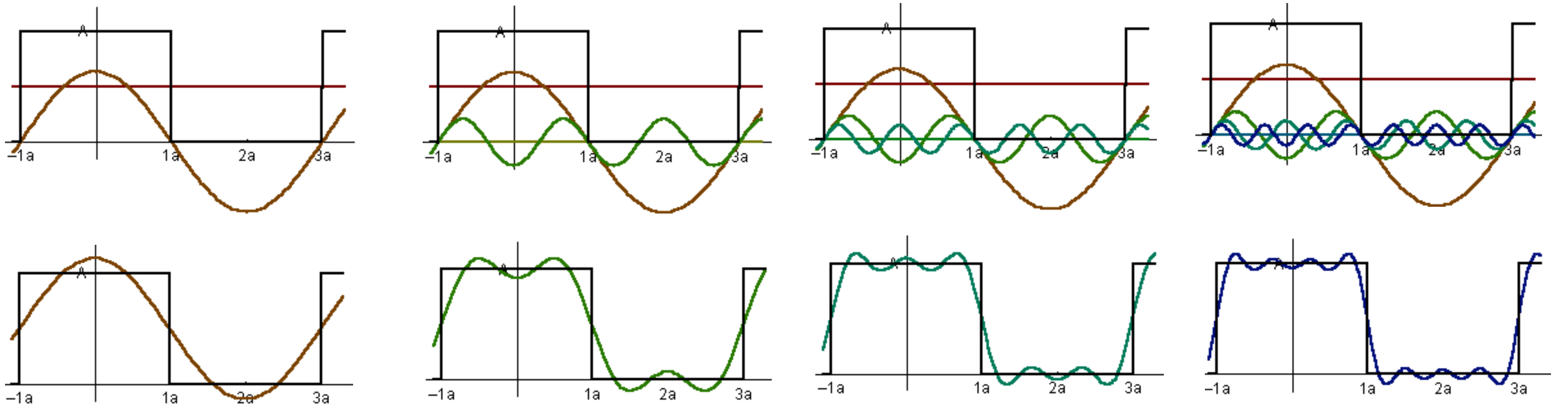
Joseph Fourier (1768 – 1830)

$$f(x) = a_0 + \sum_{i=1}^{\infty} a_i \cos(2\pi n x) + b_i \sin(2\pi i x)$$

$$f(x) = a_0 + a_1 \cos(x) \quad + a_2 \cos(3x) \quad + a_3 \cos(5x) \quad + a_4 \cos(7x) + \ldots$$

basis functions

weighted sum



Fourier analysis

$e^{ix} = \cos(x) + i\,\sin(x)$

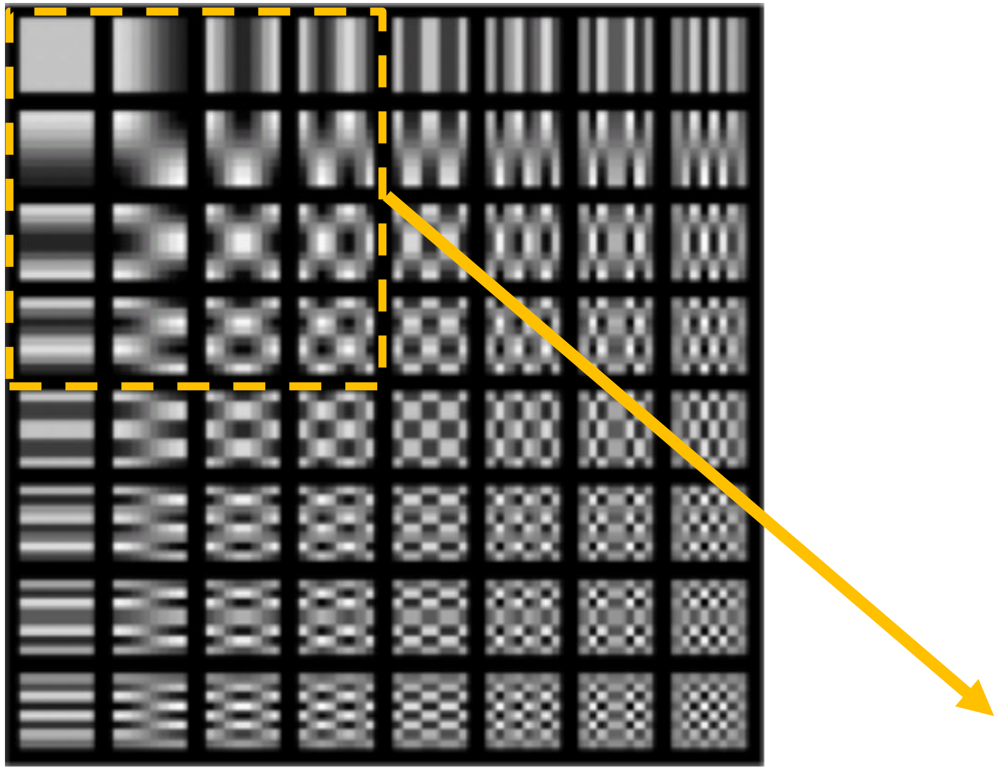
$$F(\omega) = \int_{-\infty}^{\infty} f(x) \mathrm{e}^{-2\pi i \omega x}\, dx$$

Frequency function — Spatial function

Inverse:

$$f(x) = \int_{-\infty}^{\infty} F(\omega) \mathrm{e}^{2\pi i \omega x}\, d\omega$$

Also works on rectangular 2D domains



Fourier Discrete Cosine Transform basis functions for 8x8 grayscale images

$$\cos(2\pi\omega_h)\cos(2\pi\omega_v)$$

All frequencies

High detail. Lots of data.

Only low frequencies

☆ Blurs the image! ☆

Cut off high freq. detail = smoothing.

Extend Fourier to meshes?

- Which basis functions shall we take? 🤔
- Fourier basis functions are eigenfunctions of the (standard) Laplace operator:

$$\Delta f = \lambda f$$

Eigen equation

$$\Delta\left(e^{2\pi i\omega x}\right) = \frac{\partial^2}{\partial x^2} e^{2\pi i\omega x} = -(2\pi\omega)^2 e^{2\pi i\omega x}$$

"matrix times vector" "scalar times vector"

- On meshes: let's use the eigenvectors of the Laplace-Beltrami matrix $\Delta_{\mathcal{M}}$! 💡

Spectral analysis on meshes

- Take your favorite L-B matrix L
- Compute eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ with the k smallest eigenvalues
- Reconstruct the smoothed mesh geometry from these eigenvectors:

$$\mathbf{x} = [x_1, \ldots, x_n]^\mathsf{T} \qquad \mathbf{y} = [y_1, \ldots, y_n]^\mathsf{T} \qquad \mathbf{z} = [z_1, \ldots, z_n]^\mathsf{T}$$

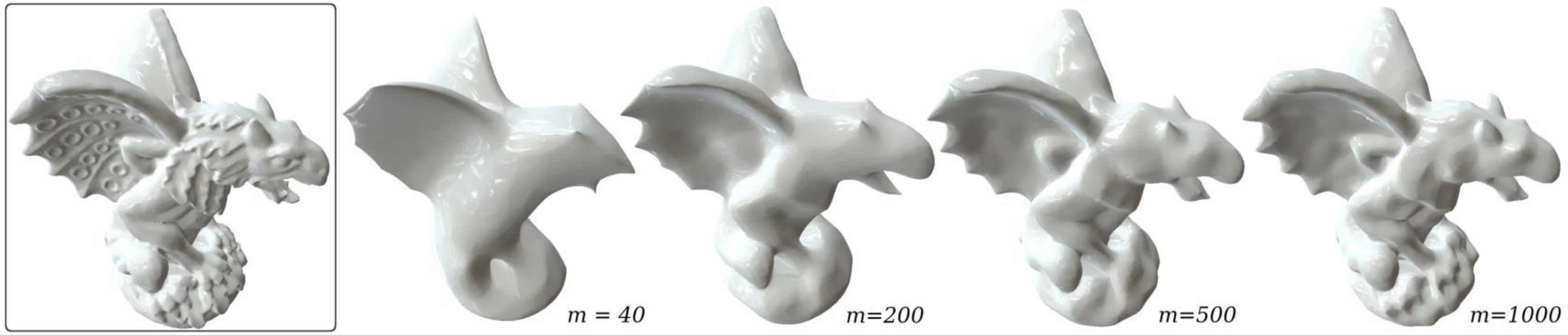
$$\tilde{\mathbf{x}} = \sum_{i=1}^{k} \left(\mathbf{x}^\mathsf{T}\mathbf{e}_i\right)\mathbf{e}_i \qquad \tilde{\mathbf{y}} = \sum_{i=1}^{k} \left(\mathbf{y}^\mathsf{T}\mathbf{e}_i\right)\mathbf{e}_i \qquad \tilde{\mathbf{z}} = \sum_{i=1}^{k} \left(\mathbf{z}^\mathsf{T}\mathbf{e}_i\right)\mathbf{e}_i$$

$$\tilde{\mathbf{p}} = [\tilde{\mathbf{x}}\ \tilde{\mathbf{y}}\ \tilde{\mathbf{z}}] \in \mathbb{R}^{n \times 3}$$

Spectral analysis on meshes

too expensive for large meshes

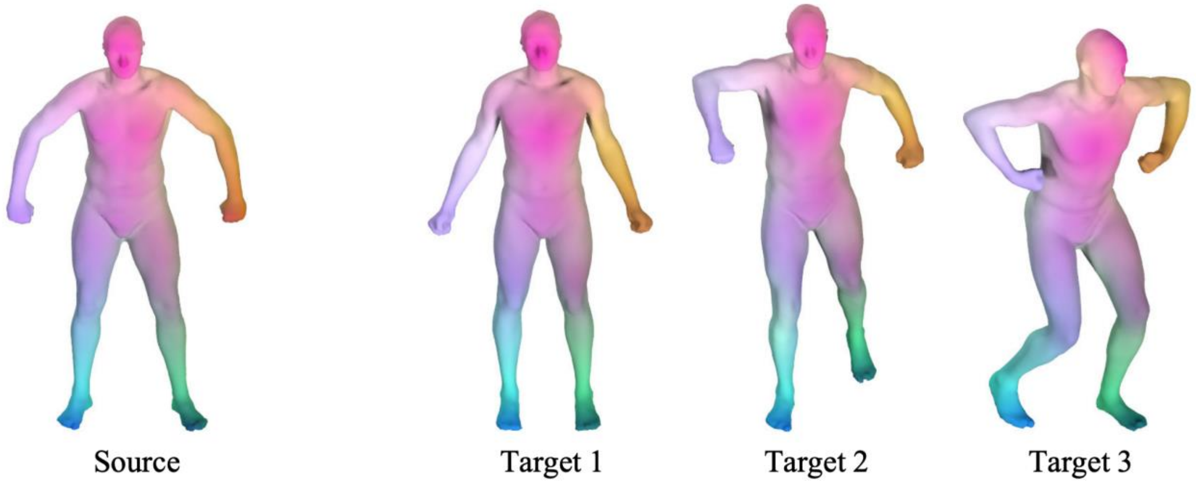
- Take your favorite L-B matrix L
- Compute eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ with the k smallest eigenvalues
- Reconstruct the smoothed mesh geometry from these eigenvectors:



Spectral analysis on meshes

- Take your favorite L-B matrix L
- Compute eigenvectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k$ with the k smallest eigenvalues

The (truncated) spectral basis is useful in other applications,
such as shape matching and functional maps!



"How can I map the right arm to the right arm and left leg to the left leg correctly between two distinct meshes?"

Parametrizations

Parameterization – What is it?

Geometry Acquisition Pipeline

Scanning
results in range images

⇨

Registration
bring all range images to one coordinate system

⇨

Stitching/reconstruction
Integration of scans into a single mesh

⇨

Postprocess
Topological filtering
Geometric filtering
Remeshing
Compression

Geometry Acquisition Pipeline

Scanning
results in range images

⇨

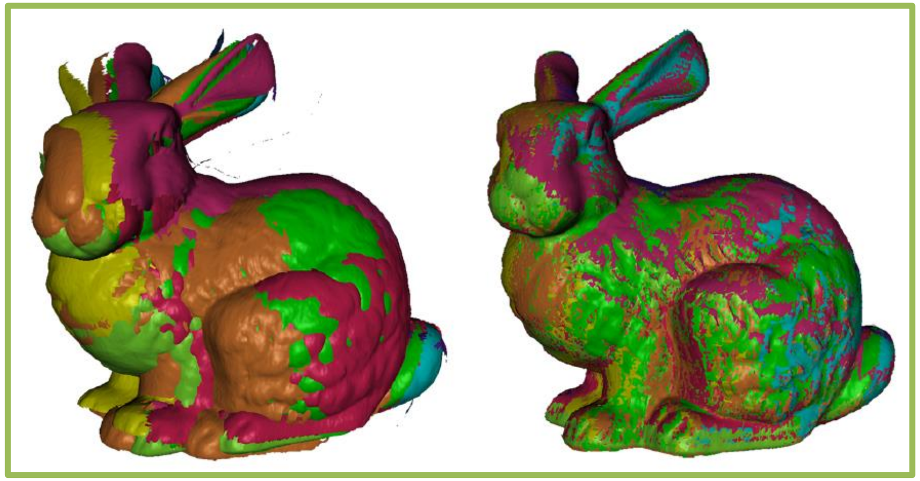
Registration
bring all range images to one coordinate system

⇨

Stitching/reconstruction
Integration of scans into a single mesh

⇨

Postprocess
Topological filtering
Geometric filtering
Remeshing
Compression



Geometry Acquisition Pipeline

Scanning
results in range images

⇨

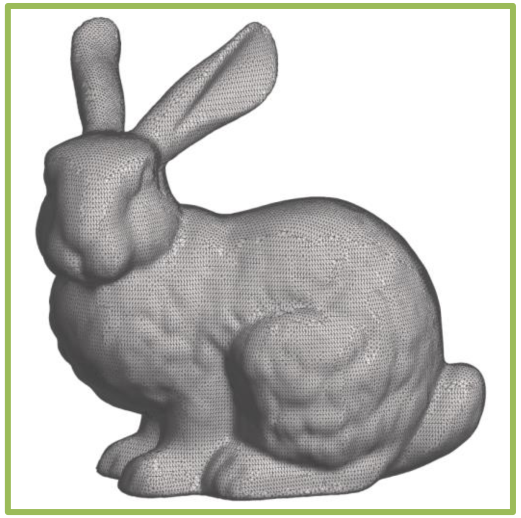
Registration
bring all range images to one coordinate system

⇨

Stitching/reconstruction
Integration of scans into a single mesh

⇨

Postprocess
Topological filtering
Geometric filtering
Remeshing
Compression



Geometry Acquisition Pipeline

Scanning
results in range images

⇨

Registration
bring all range images to one coordinate system

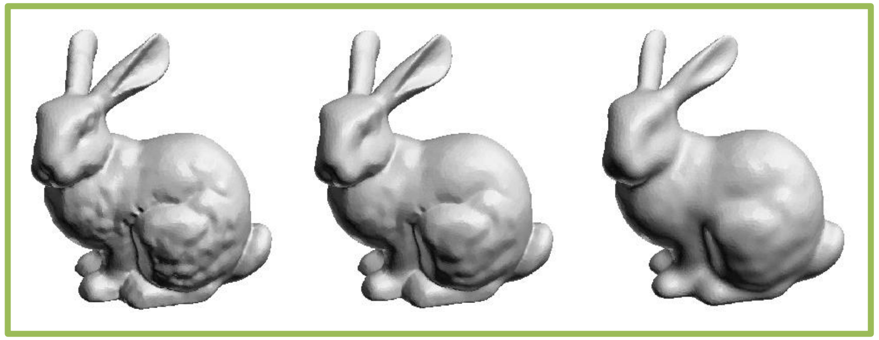
⇨

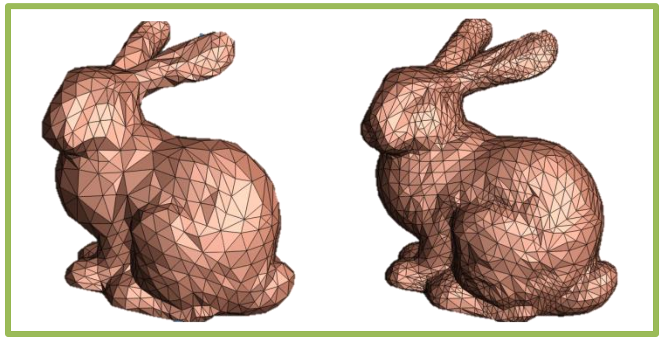
Stitching/reconstruction
Integration of scans into a single mesh

⇨

Postprocess
Topological filtering
Geometric filtering
Remeshing
Compression

Now we would like to add texture to the bunny! 🎨

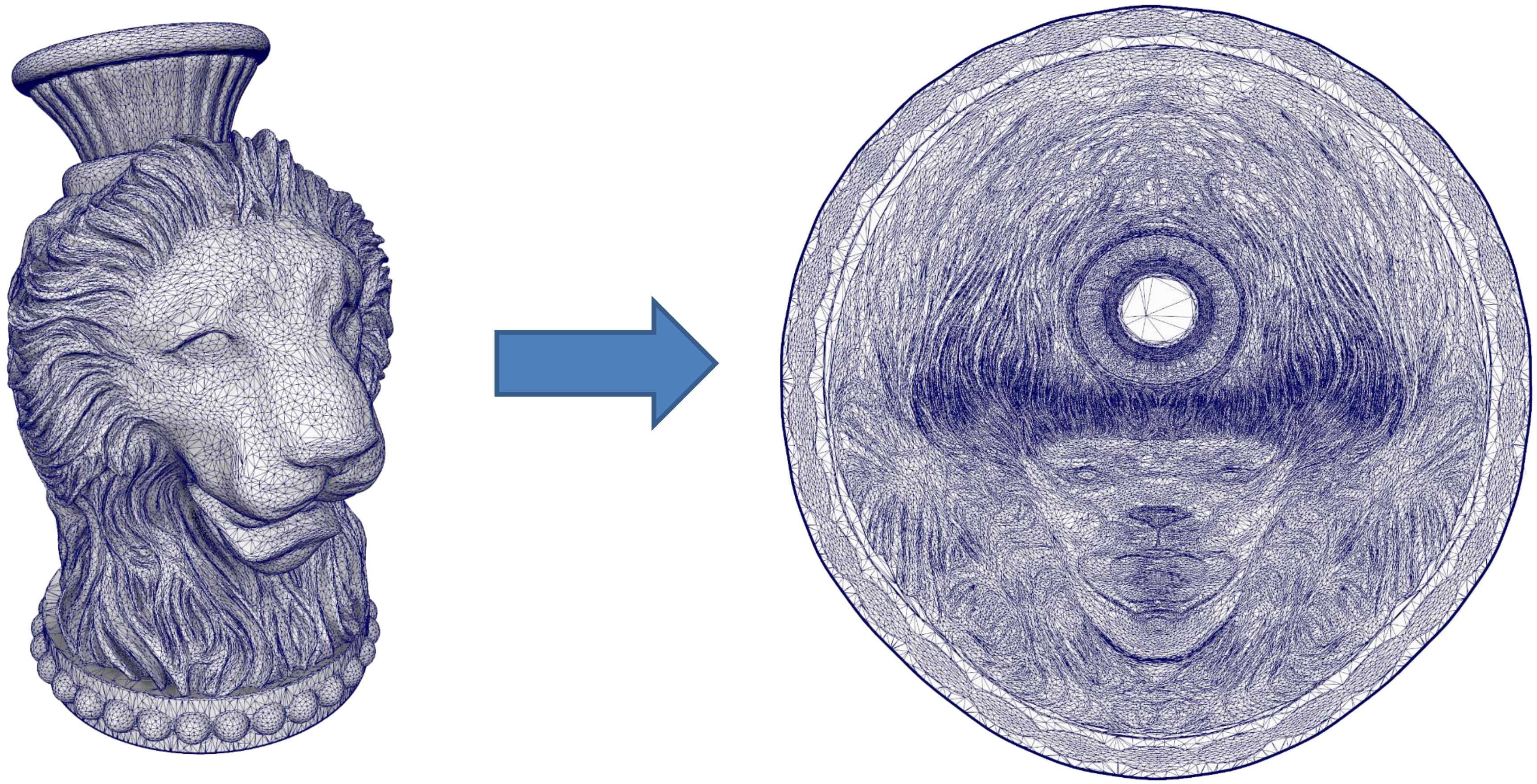




Surface Parameterization

3D space (x,y,z)

2D parameter domain (u,v)

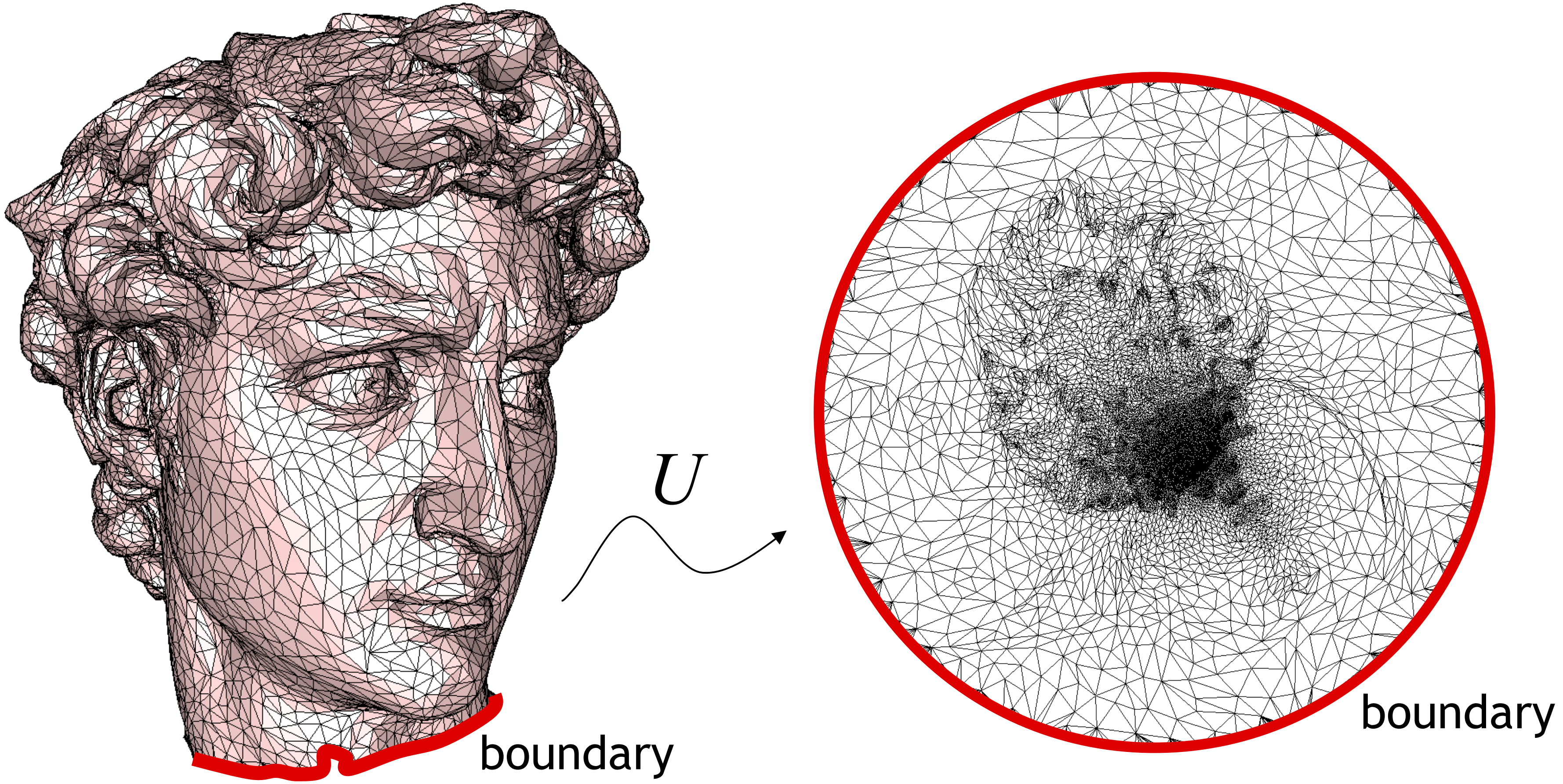


Surface Parameterization

2D projections are fantastic
for adding texture 🖌️

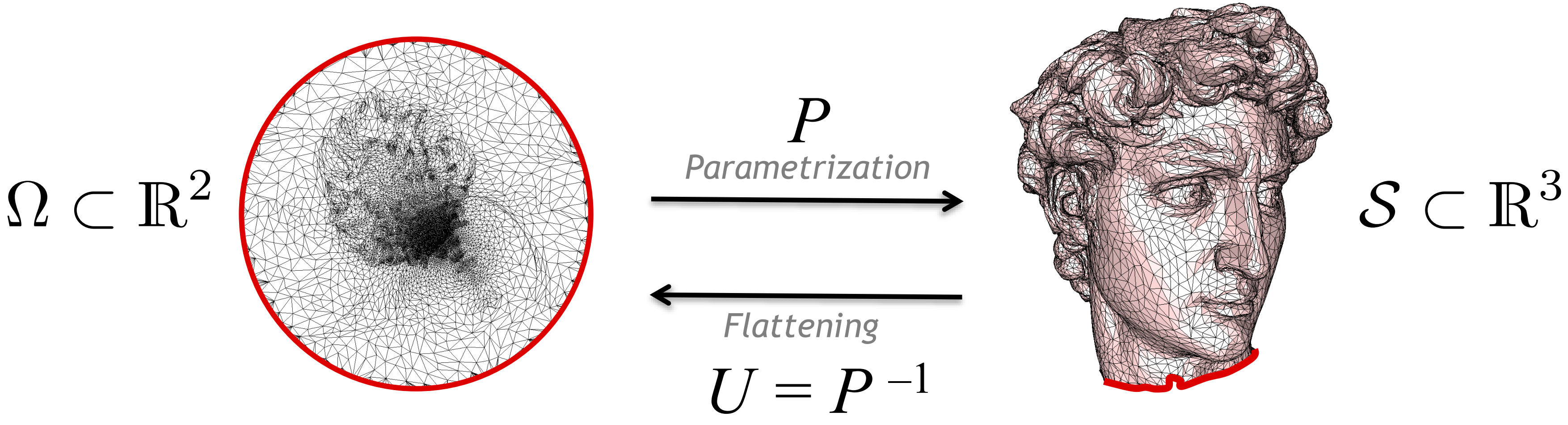
Mesh Parameterization

3D space (x,y,z) 2D parameter domain (u,v)



Mesh Parameterization – Definition

- Mapping P between a 2D domain Ω and the mesh S embedded in 3D
- Each mesh vertex has a corresponding 2D position: $U(\mathbf{v}_i) = (u_i, v_i)$
- Inside each triangle, the mapping is **affine** (linear barycentric coordinate interpolation)



„U“ maps to uv texture coordinates

Parameterization – What is it good for??

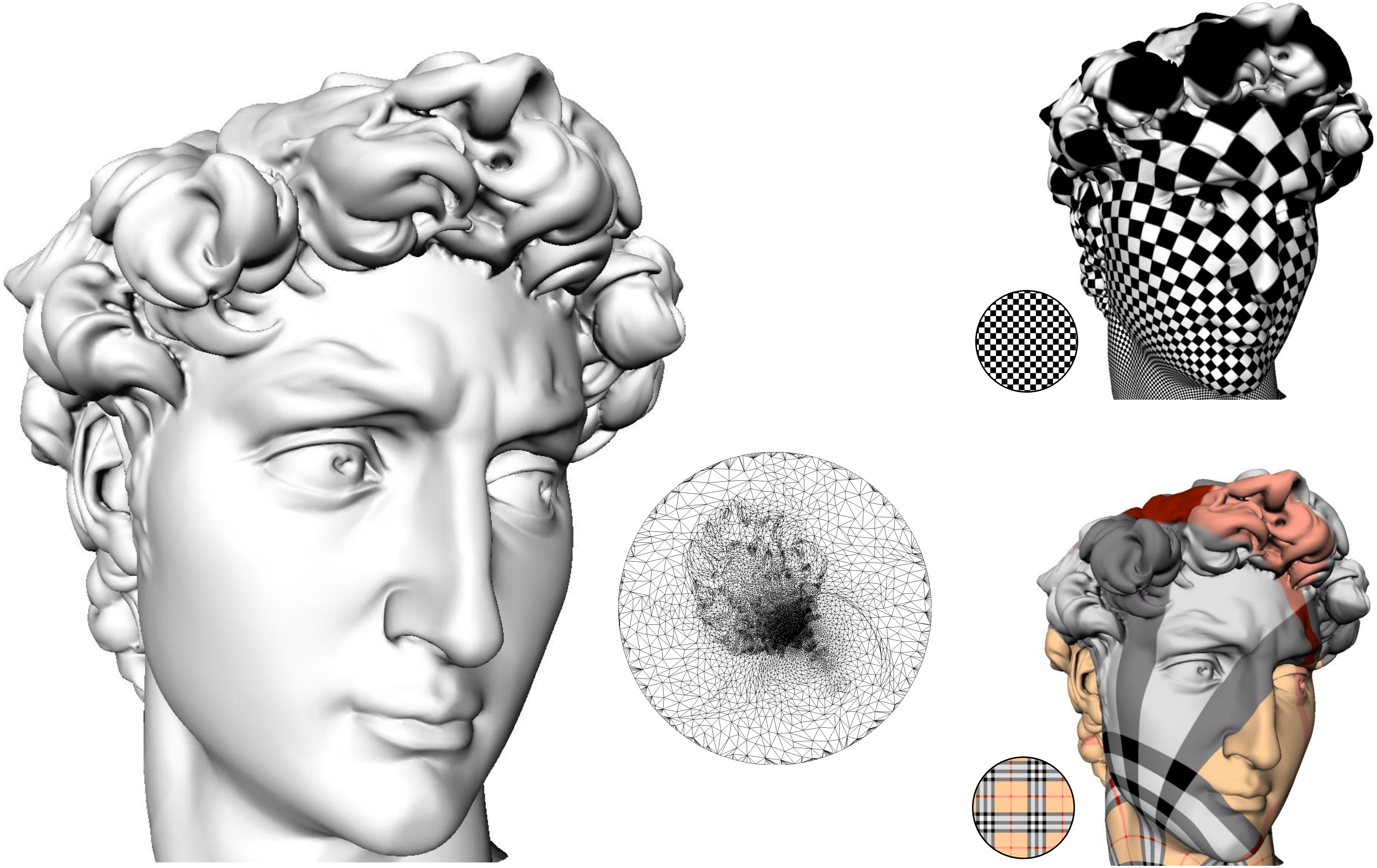
Why Parameterization?

Allows us to do many things in 2D and then map those actions onto the 3D surface

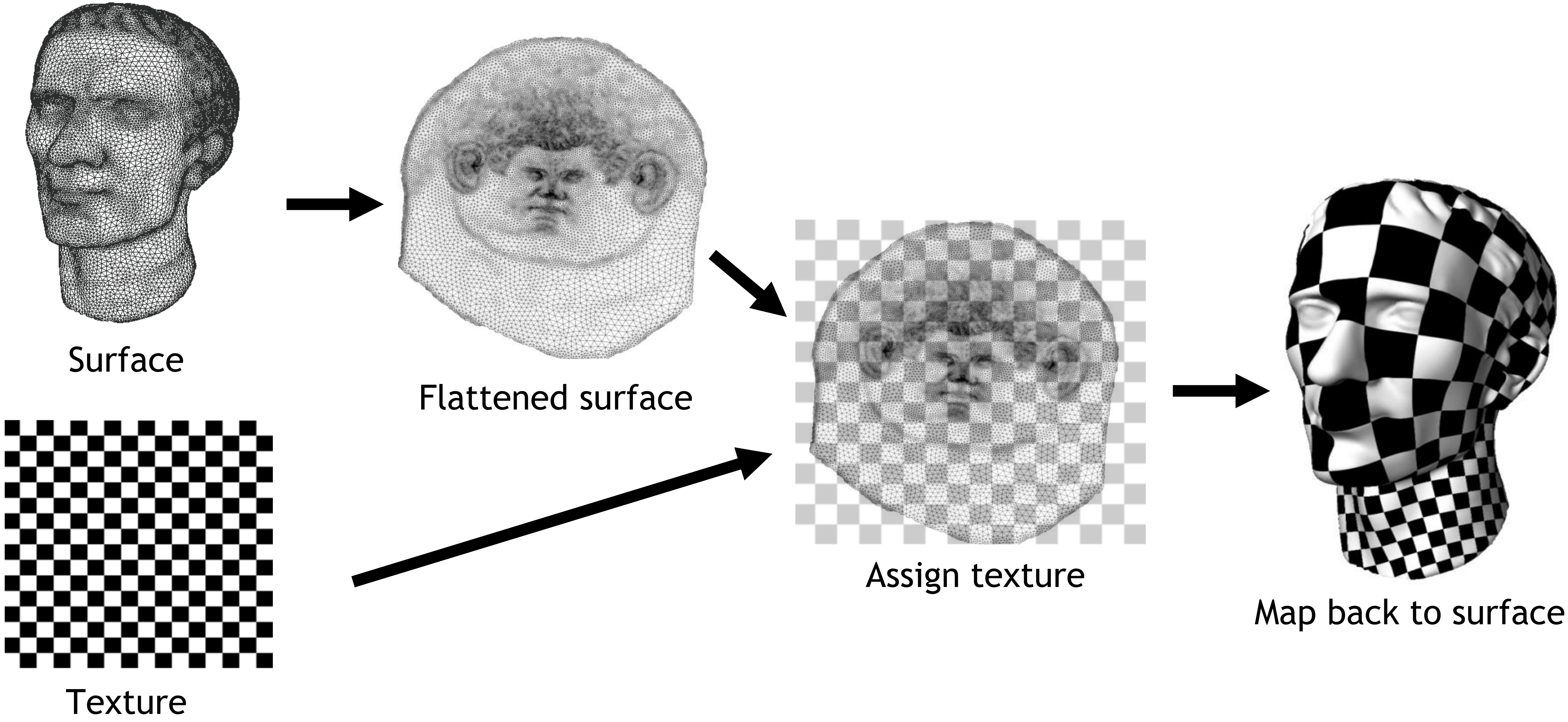
Mesh parameterization allows to use some notions from continuous surface theory

It is often easier to operate in the 2D domain

Main Application: Texture Mapping



Main Application: Texture Mapping



Texture Mapping

Lots of detail with little geometry ⭐

Texture Mapping

Common Goals
- Minimize image distortion 🖼️
- Maximize available space use 📦

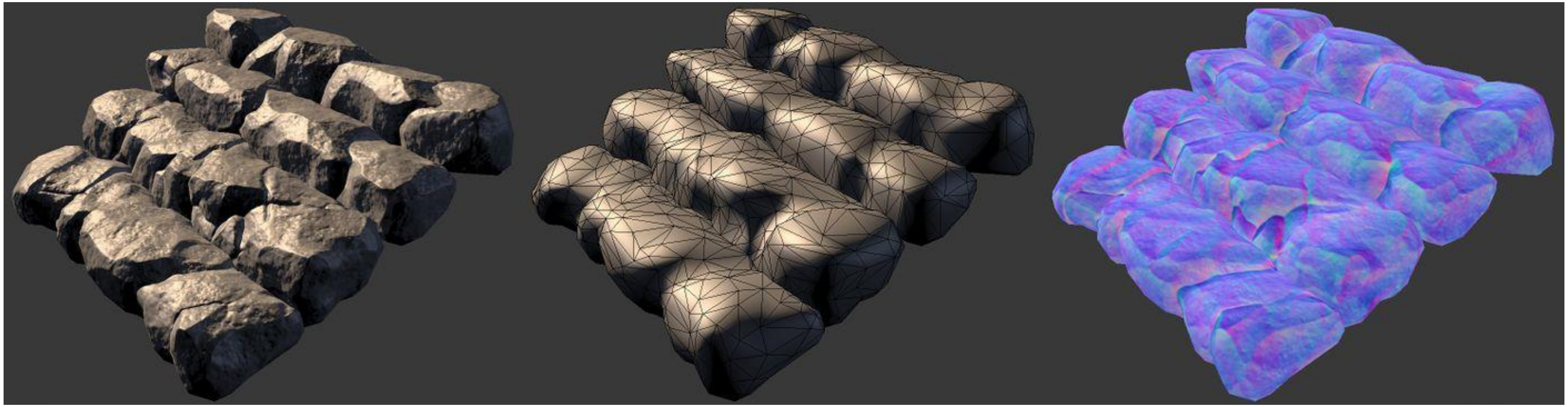
Image from Vallet and Levy, techreport INRIA

Garment Modeling

- The sewing pattern is a parameterization of the garment

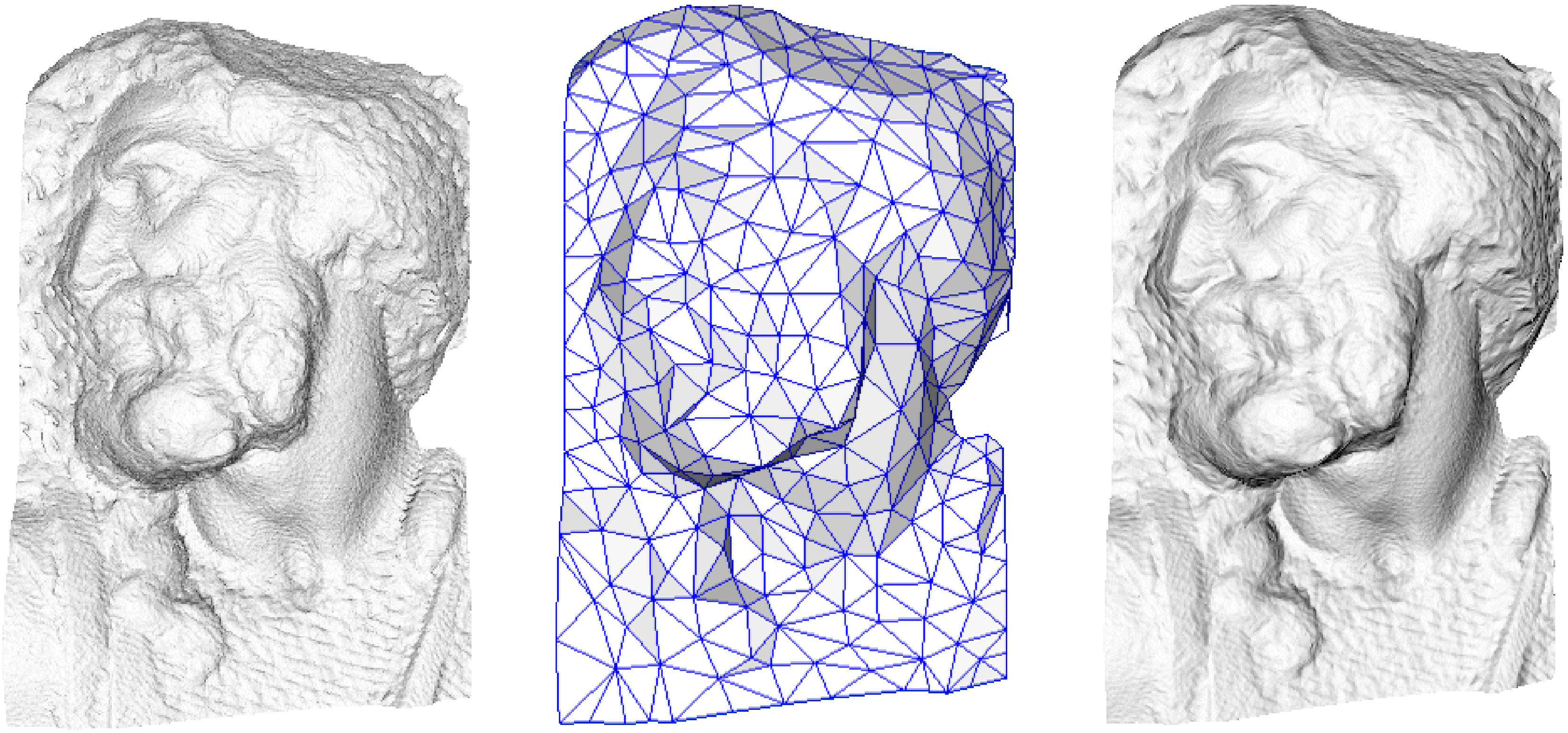
Normal/Bump Mapping

RGB encode X,Y,Z components of the normal vector 💡



A normal mapped model, the mesh without the map, and the normal map alone. Image by [Eric Chadwick](#).

Normal/Bump Mapping

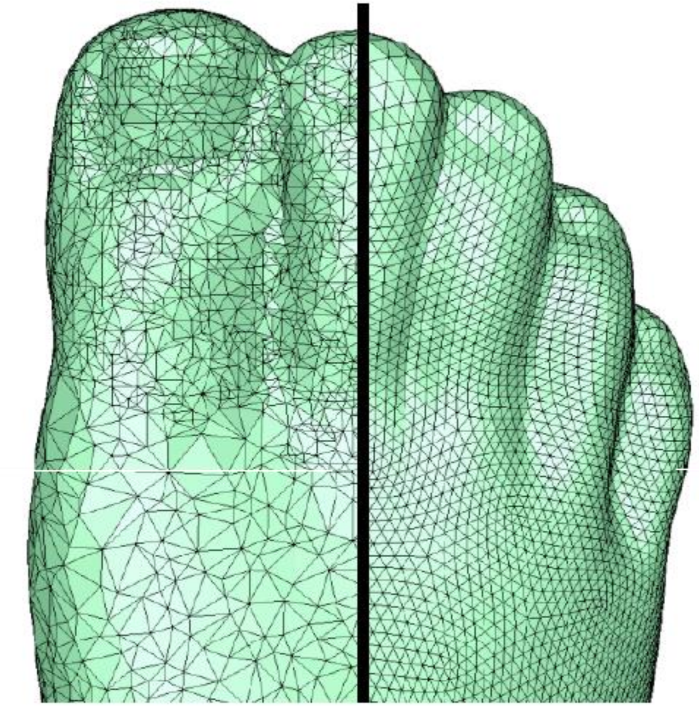


original mesh
4M triangles

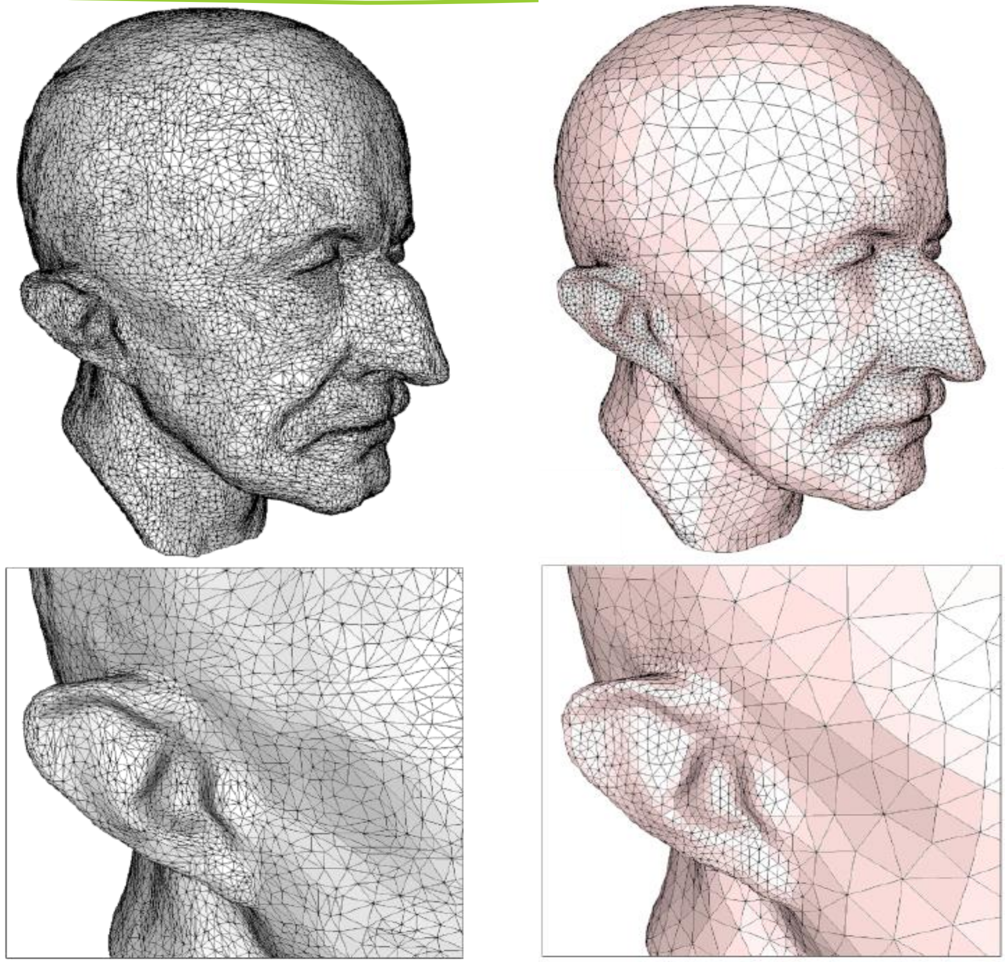
simplified mesh
500 triangles

simplified mesh
and normal mapping
500 triangles

Remeshing

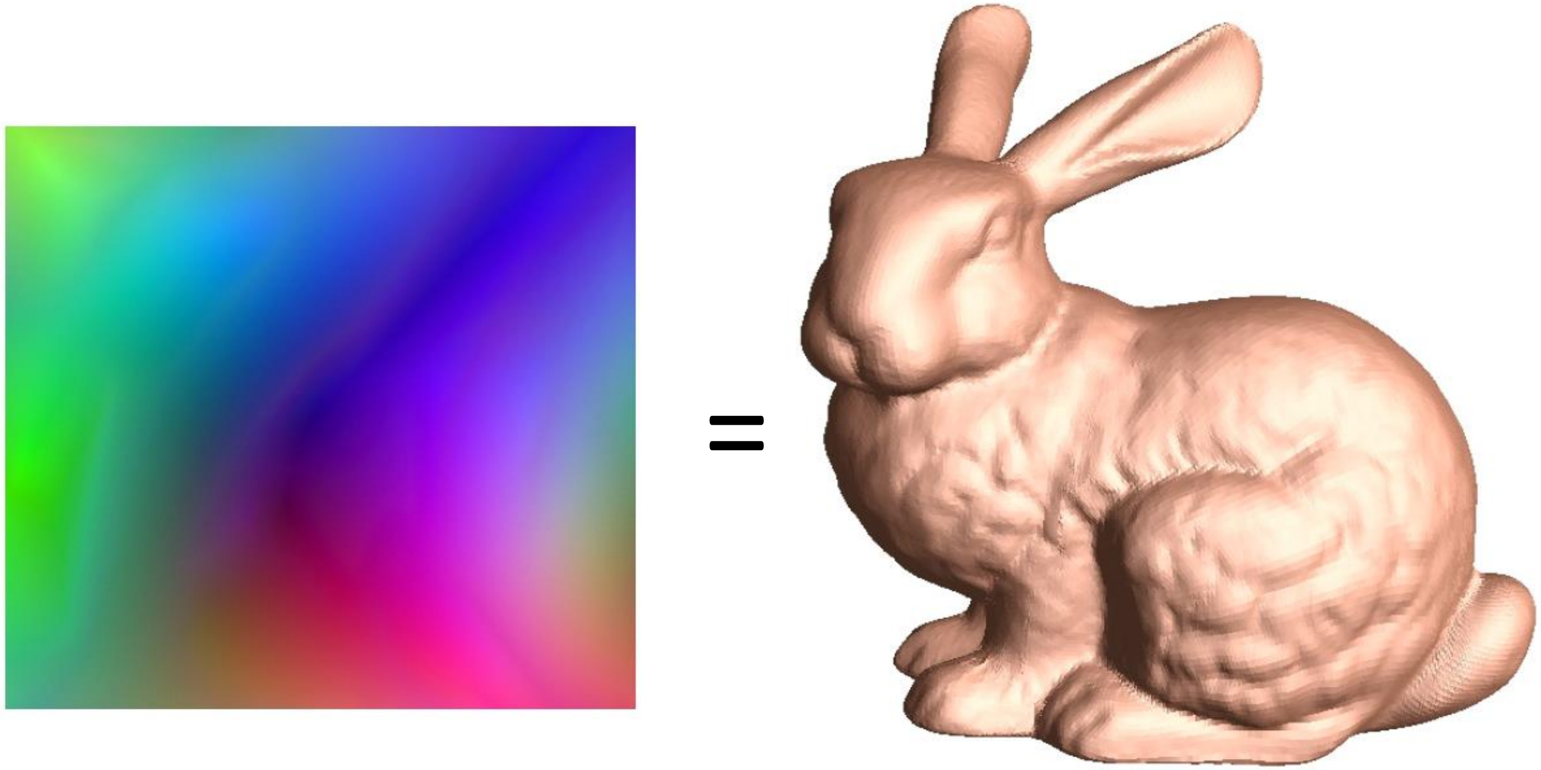


Remeshing can be easier on the flattened domain.



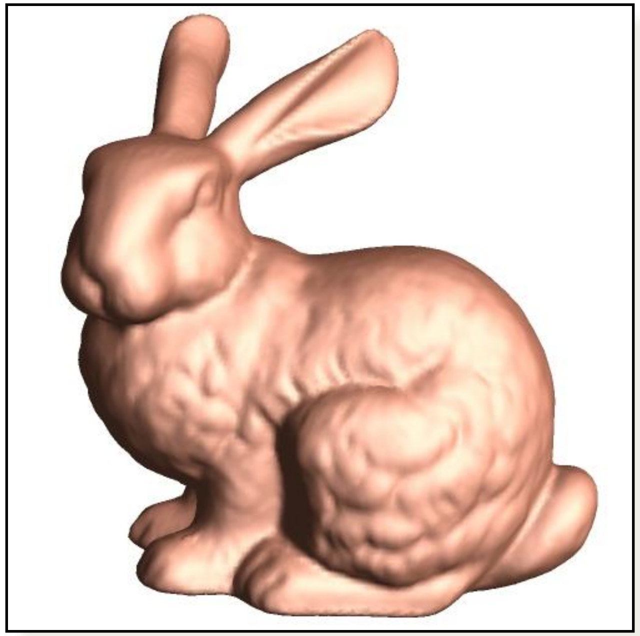
"Interactive Geometry Remeshing", Alliez et al., SIGGRAPH 2002

Compression

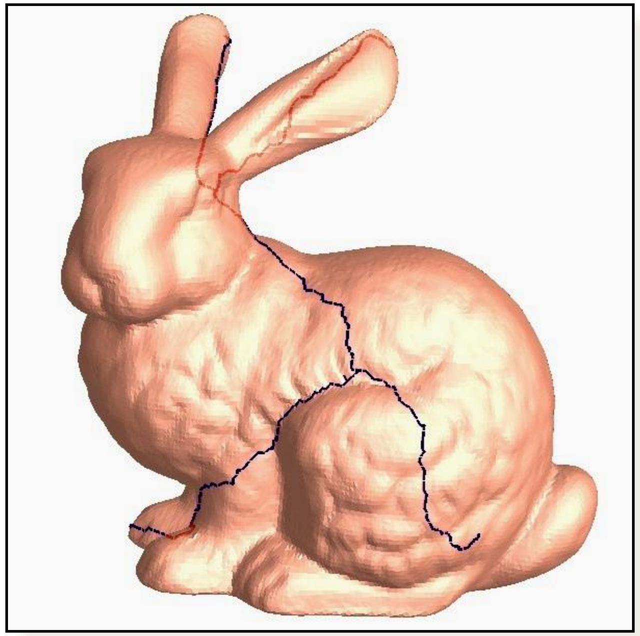


“Geometry images”, Gu et al., SIGGRAPH 2002
http://research.microsoft.com/en-us/um/people/hoppe/proj/gim/

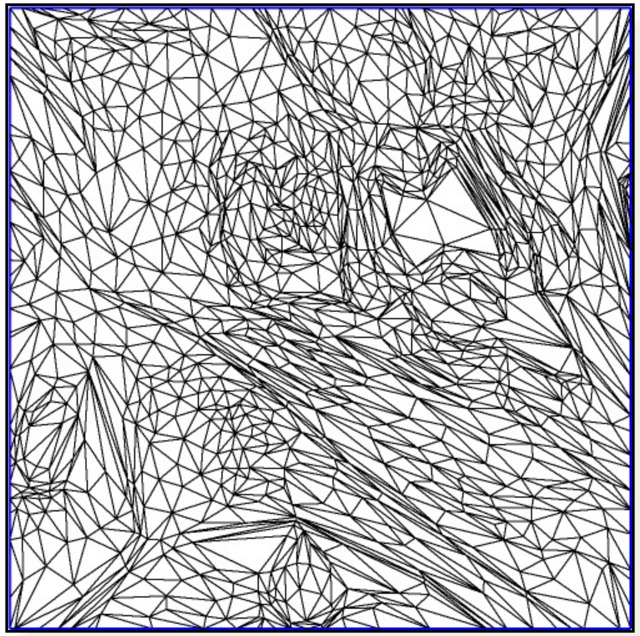
Geometry Images



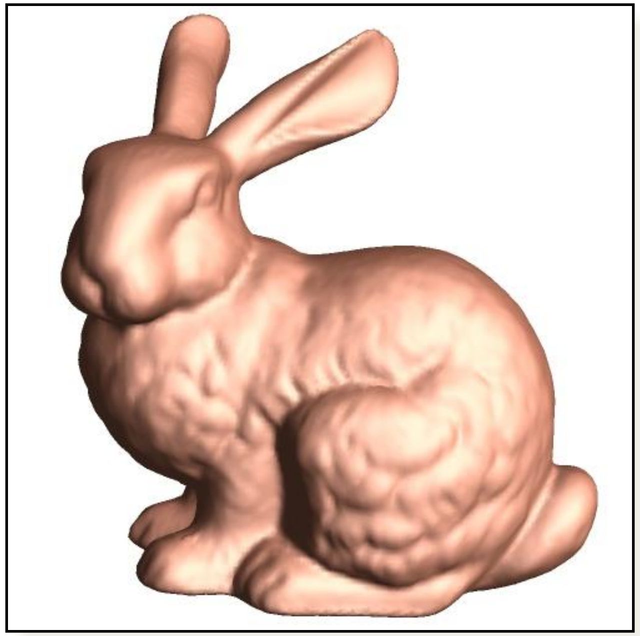
cut



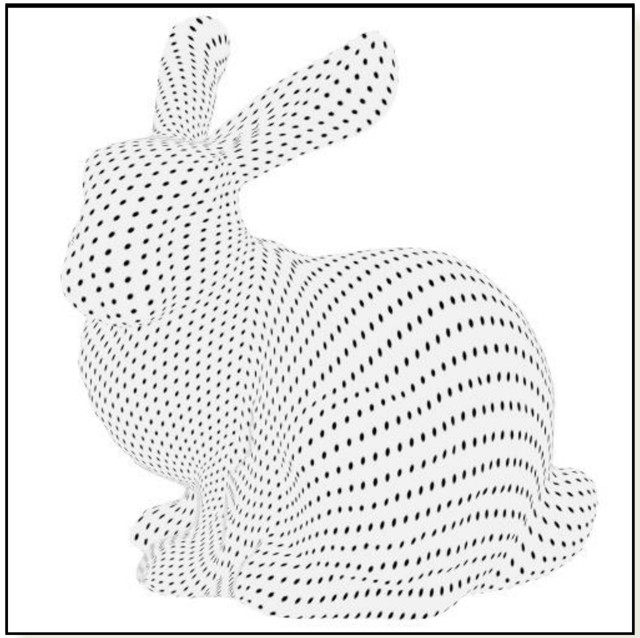
parametrize

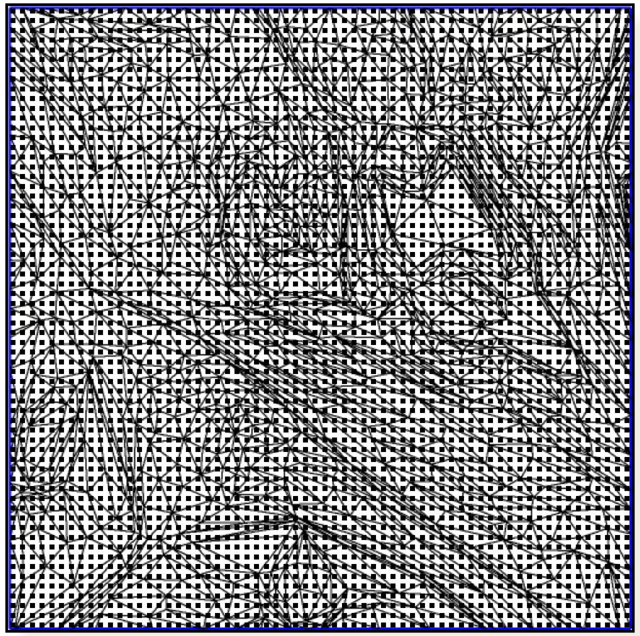


Geometry Images



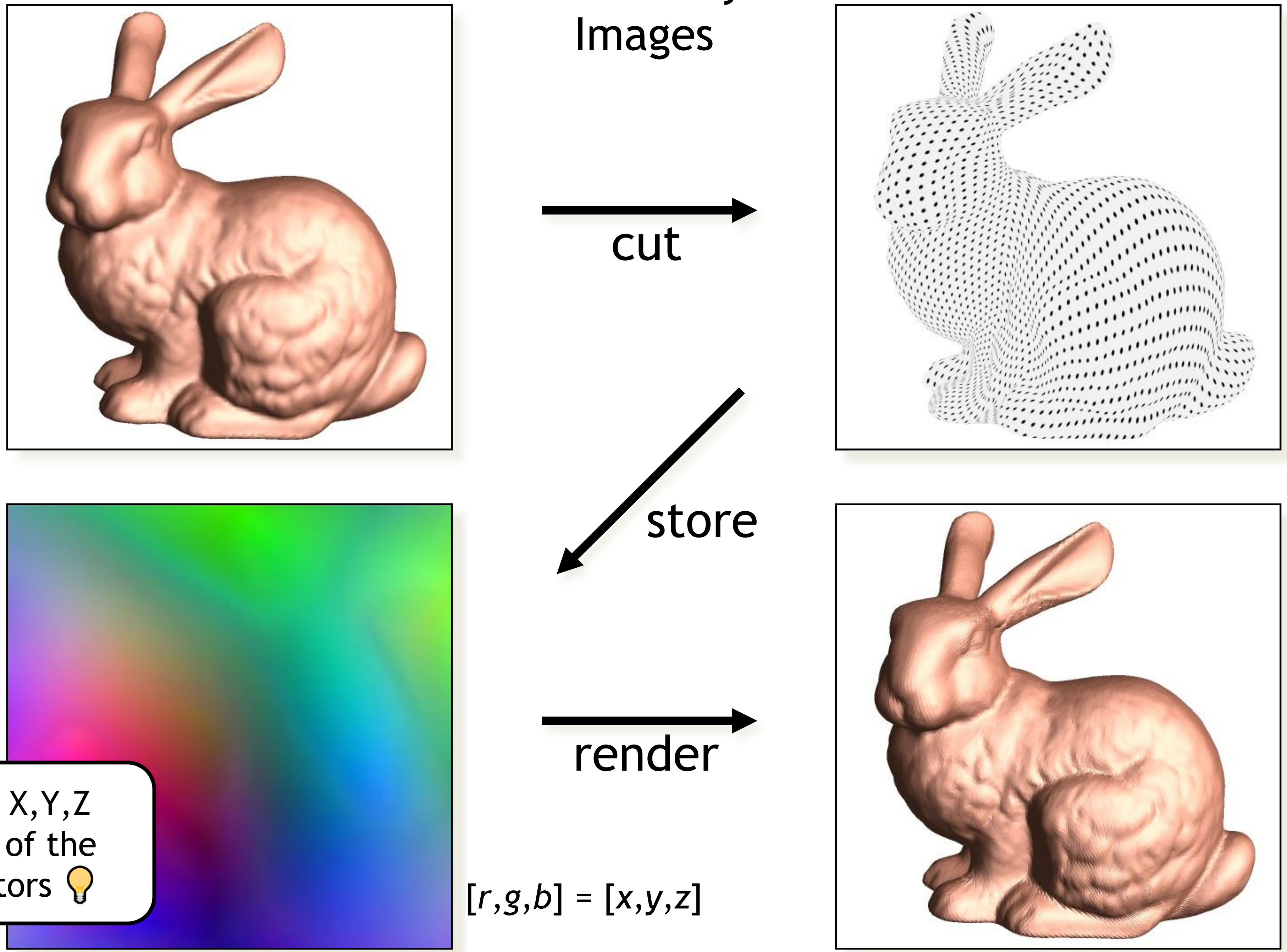
cut





sample

Geometry Images



RGB encode X,Y,Z components of the position vectors 💡

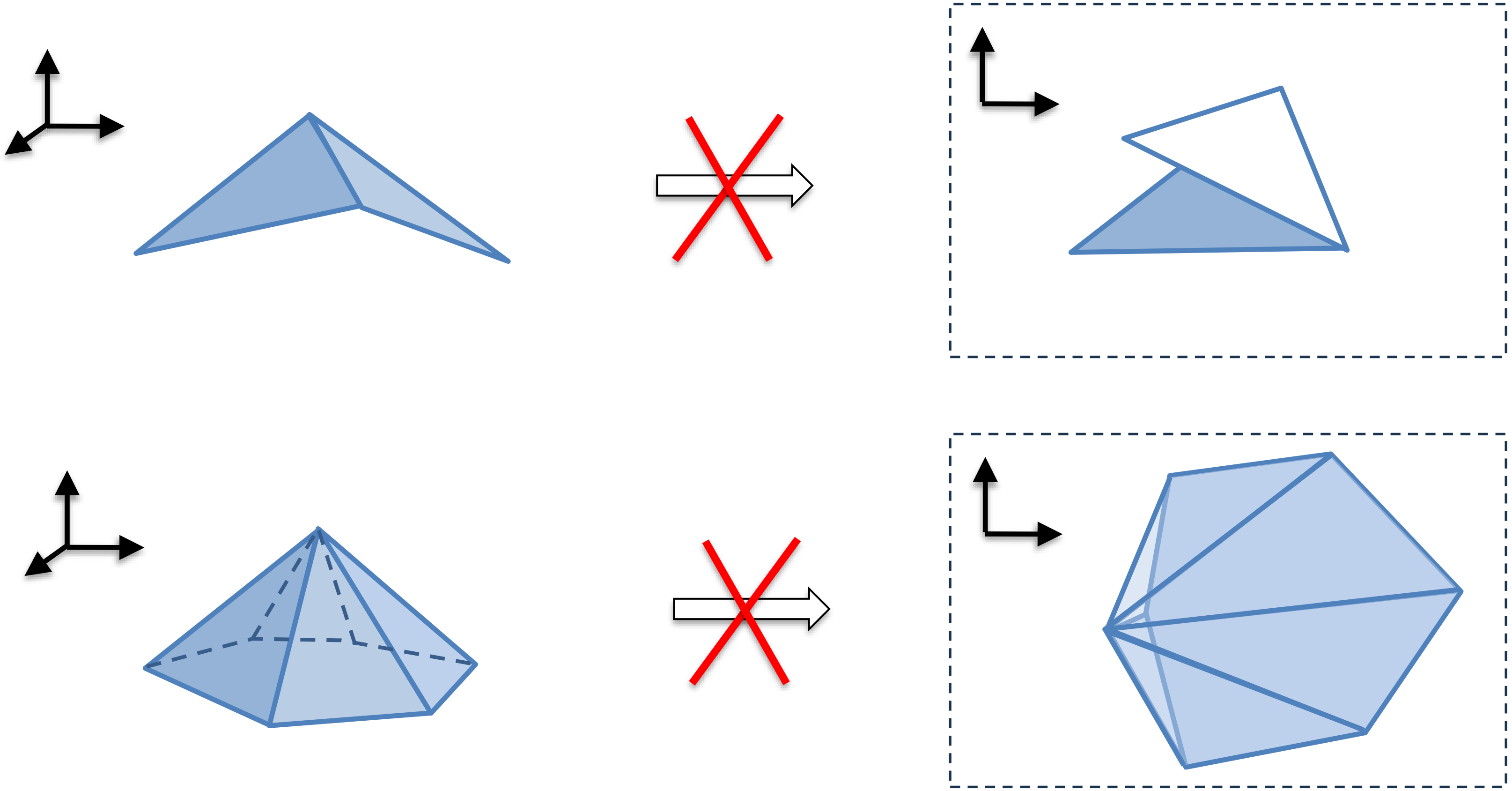
Parameterization Properties?

What are “good” parameterizations?

How do we define “good” mathematically?

Bijectivity

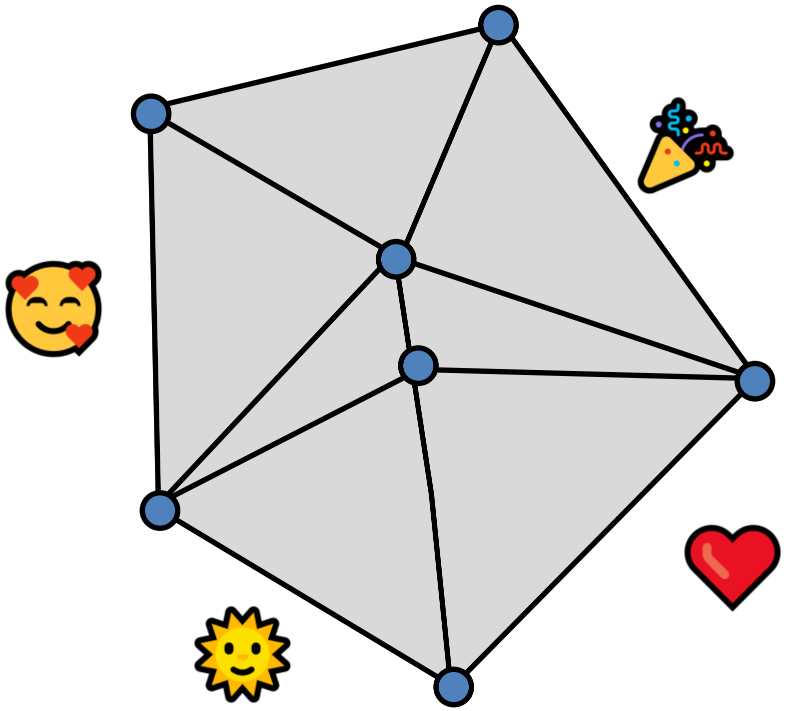
- Locally bijective: No triangles fold over



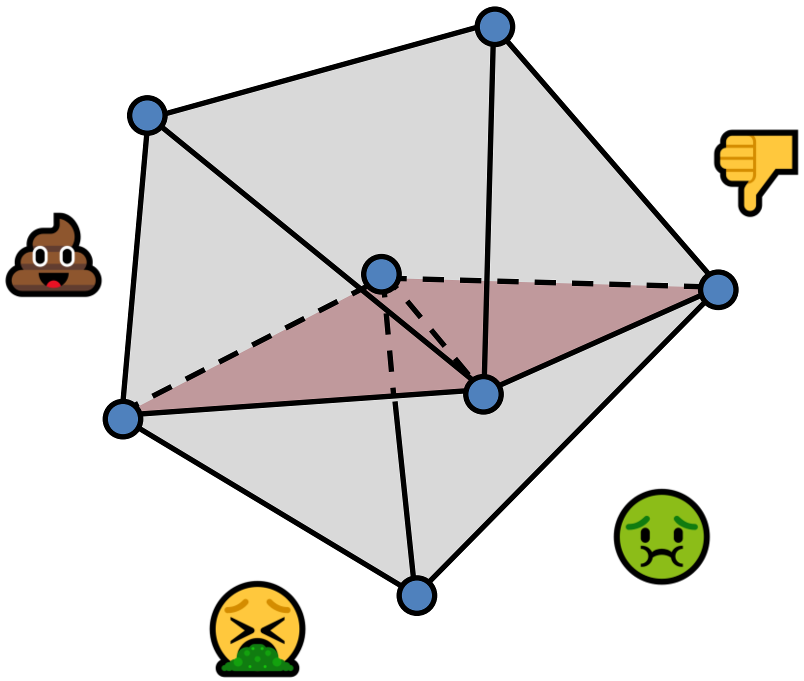
Many applications will straight up crash if this happens anywhere ⚠

Local Foldovers

No foldover

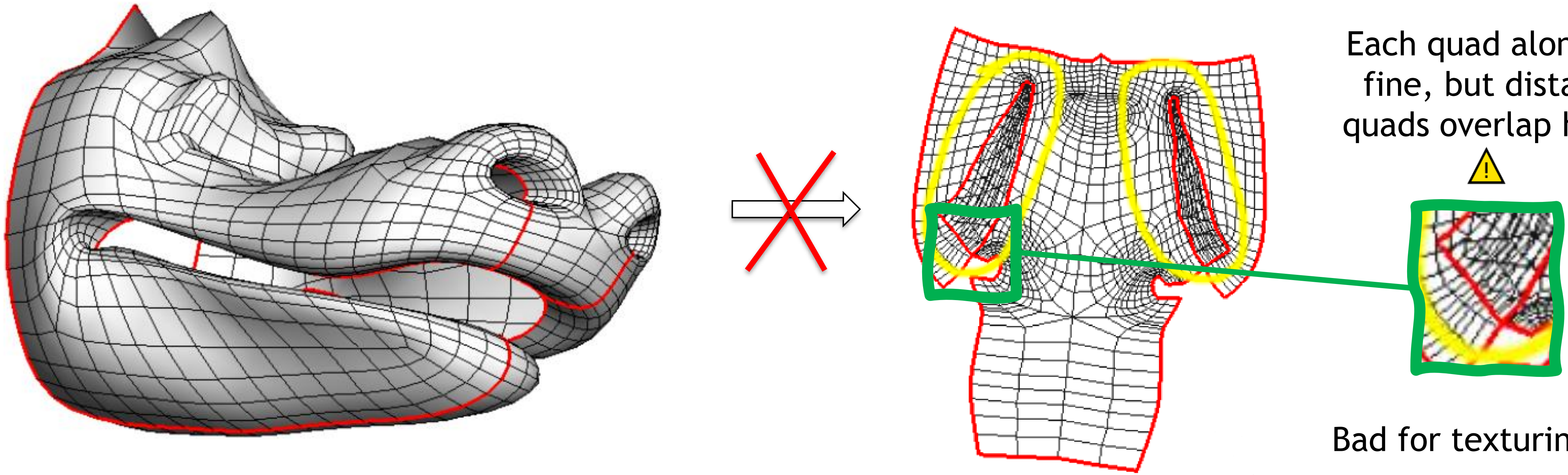


Foldover



Bijectivity

- Globally bijective: locally bijective + no “distant” areas overlap

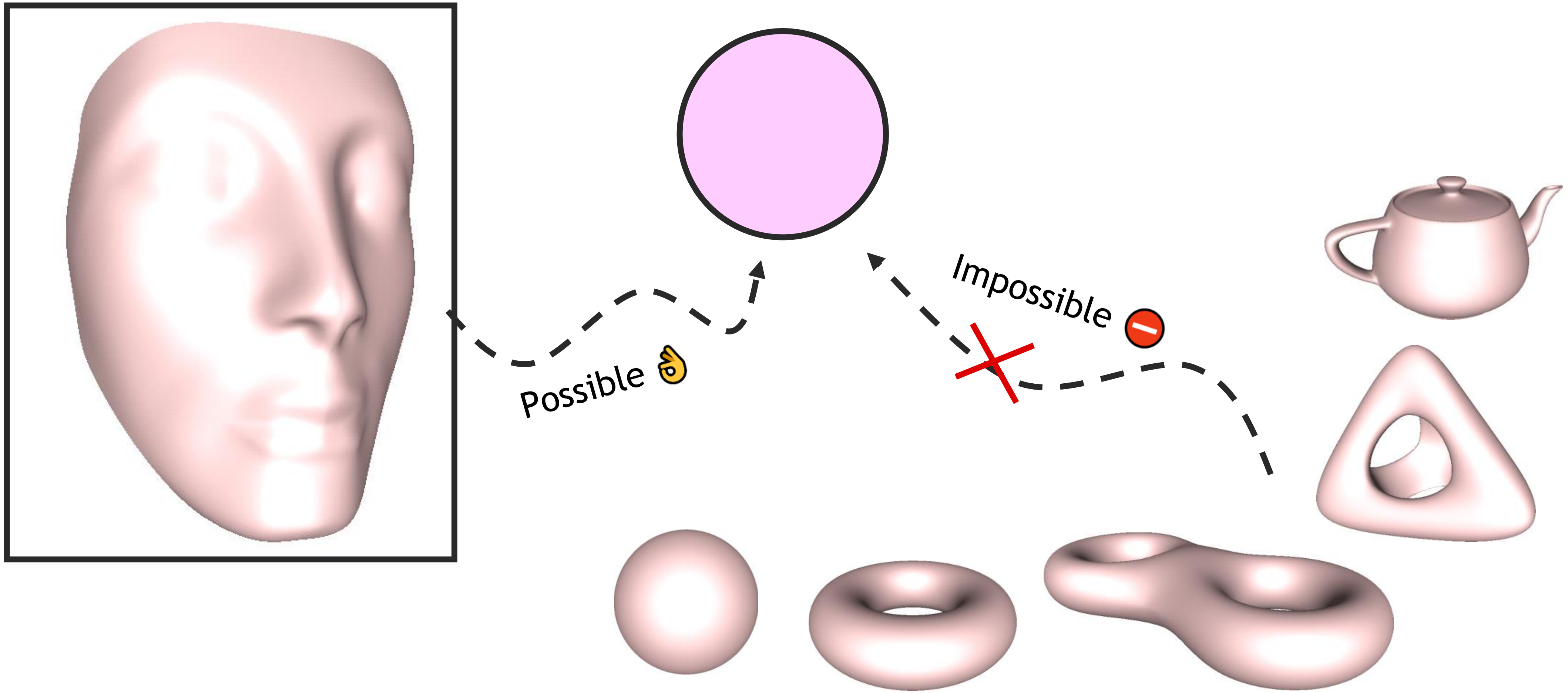


Each quad alone is fine, but distant quads overlap here ⚠

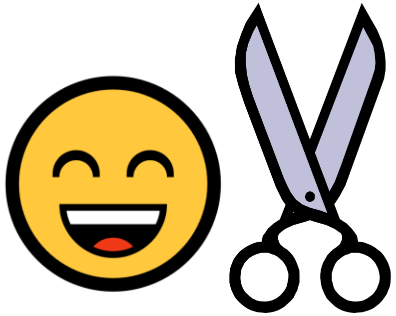
Bad for texturing 👎

image from “Least Squares Conformal Maps”, Lévy et al., SIGGRAPH 2002

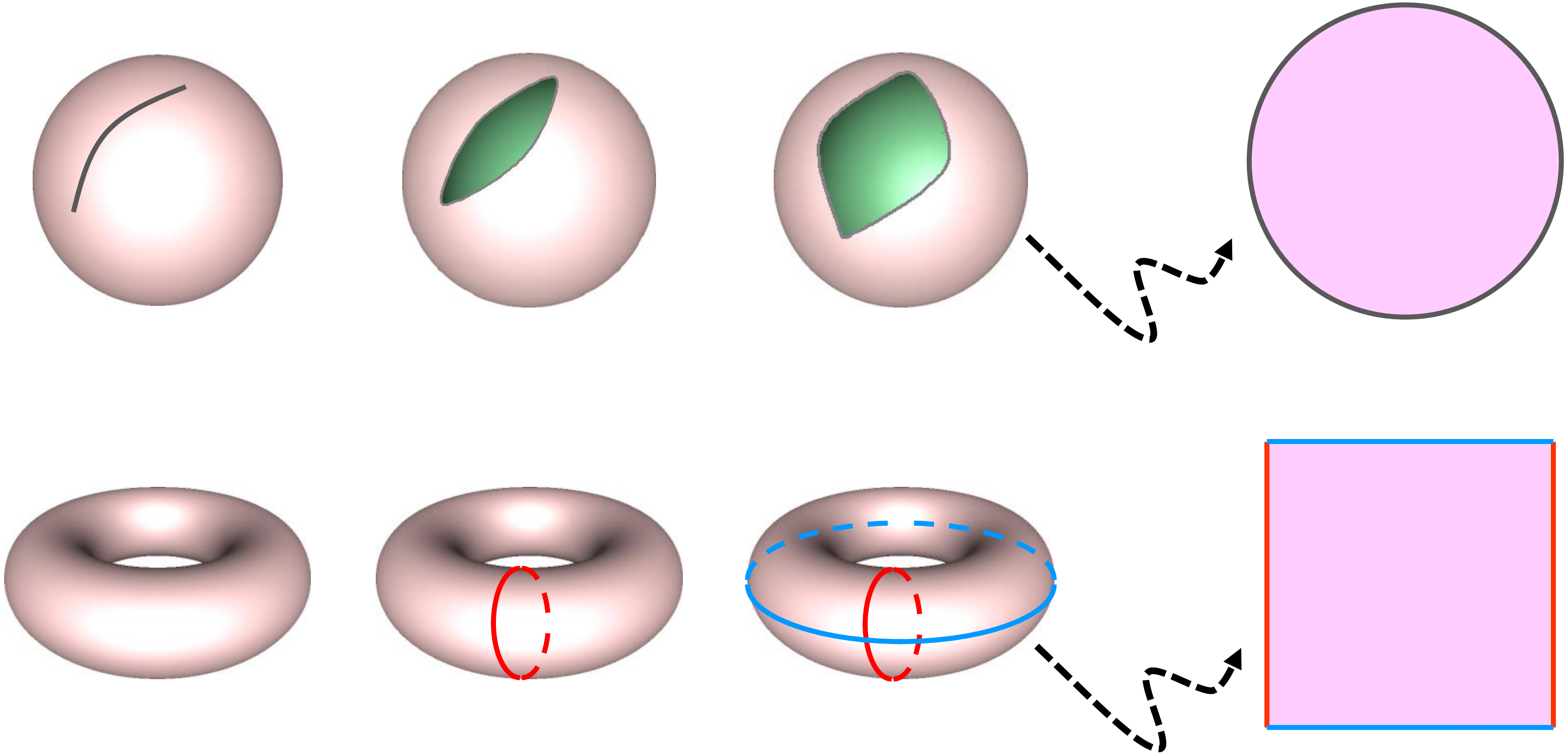
Bijectivity: Non-Disk Domains



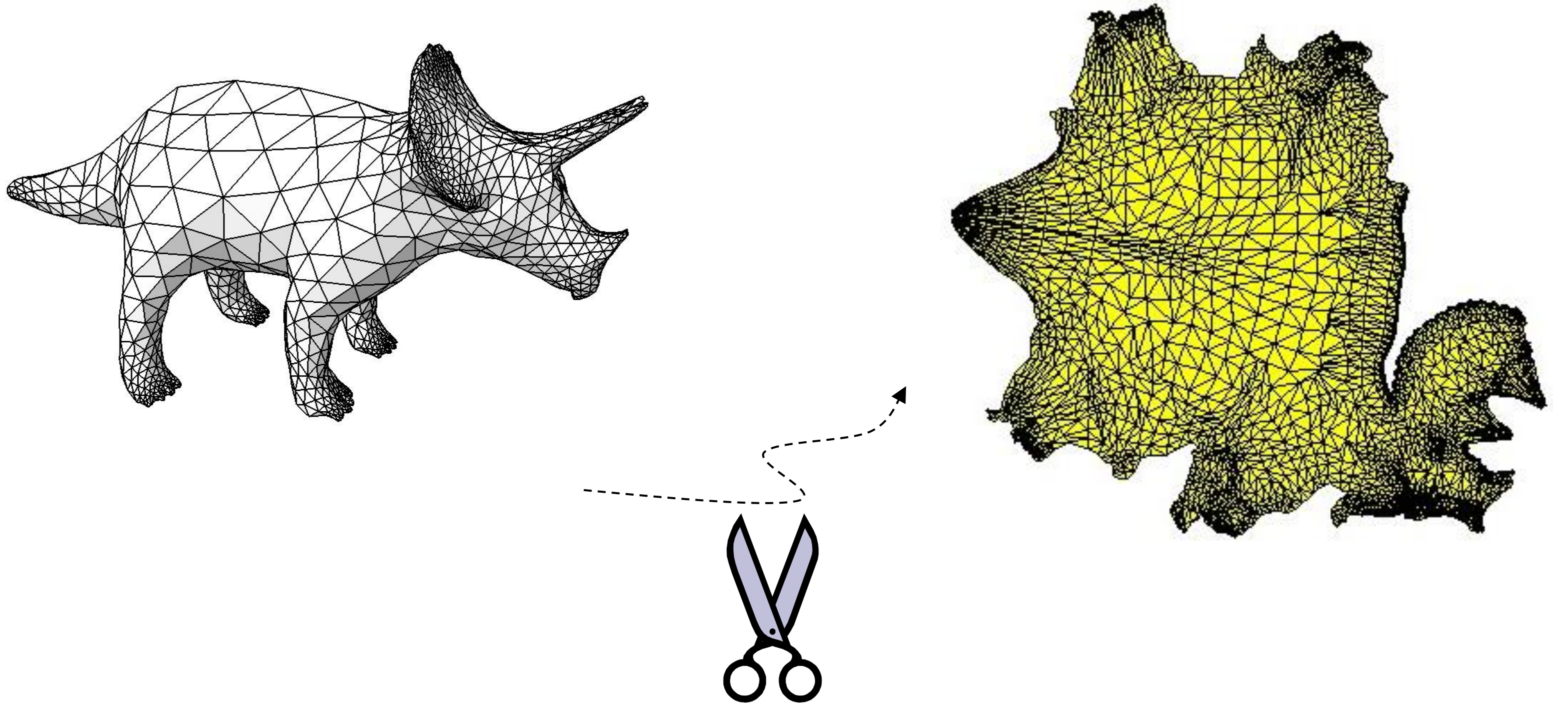
Topological Cutting



Topologists love cutting!



Topological Cutting



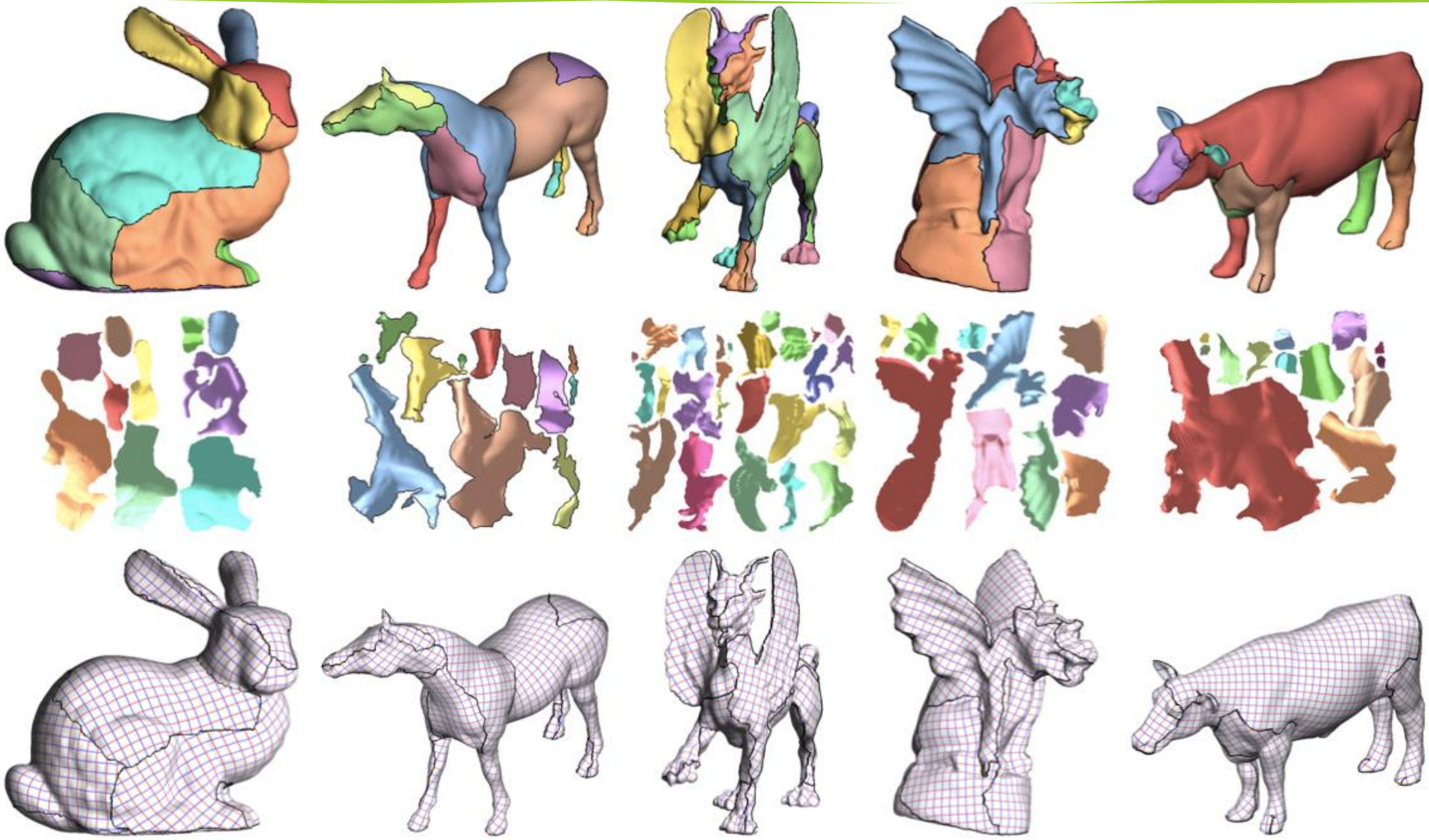
A. Sheffer, J. Hart:
Seamster: Inconspicuous Low-Distortion Texture Seam Layout, IEEE Vis 2002
http://www.cs.ubc.ca/~sheffa/papers/VIS02.pdf

Segmentation

Segmented surface

uv segmentation

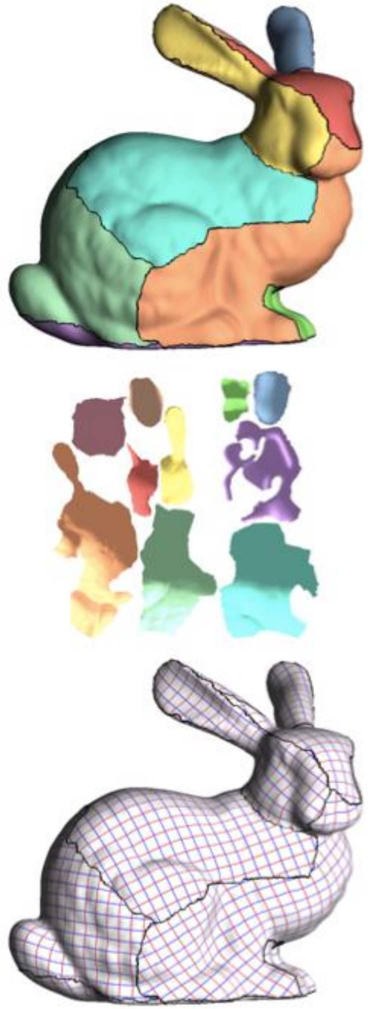
textured



Segmentation

- Find patches that align to mesh features and are close to being developable surfaces

"low Gauss curvature"

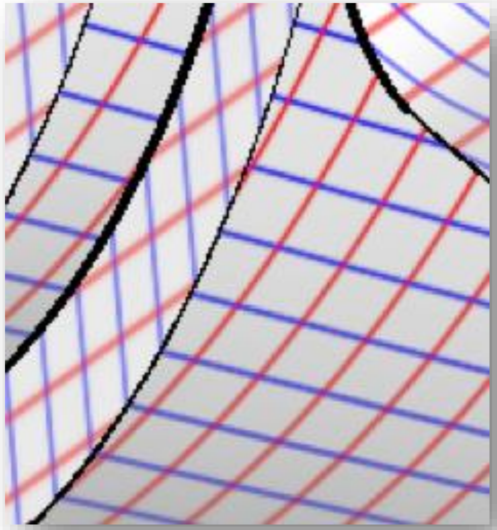


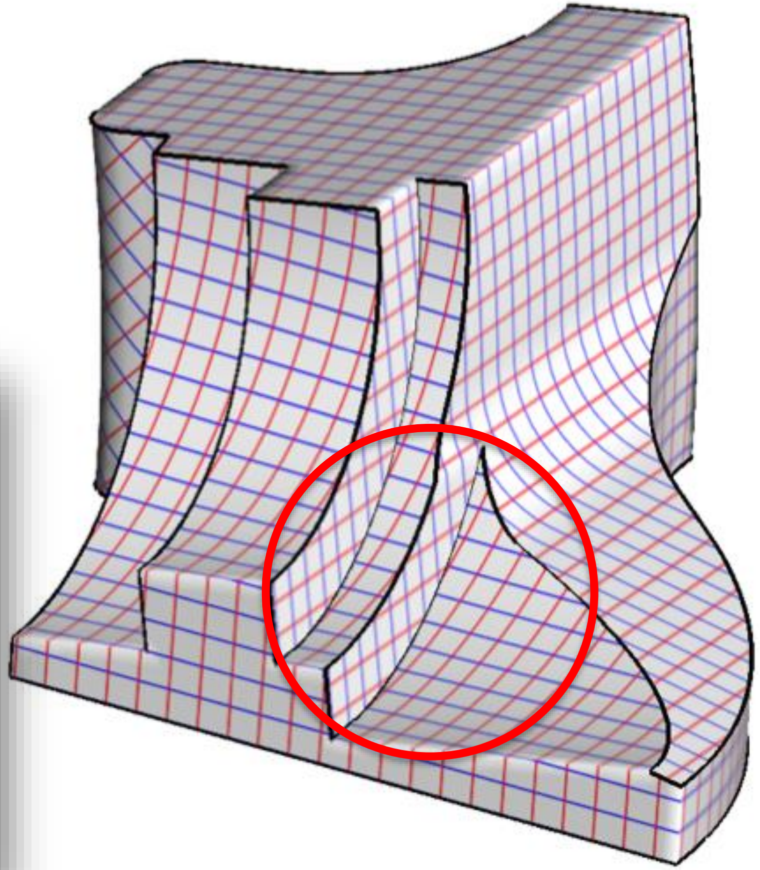
D-Charts: **Quasi-Developable Mesh Segmentation**,
D. Julius,V. Kraevoy, A. Sheffer, EUROGRAPHICS 2005

Good Cuts/Segmentations?

- Hide seams
- Small number/length of seams

- Or: make parameterization continuous across seams!
  - Good for remeshing
  - Good for garment textures

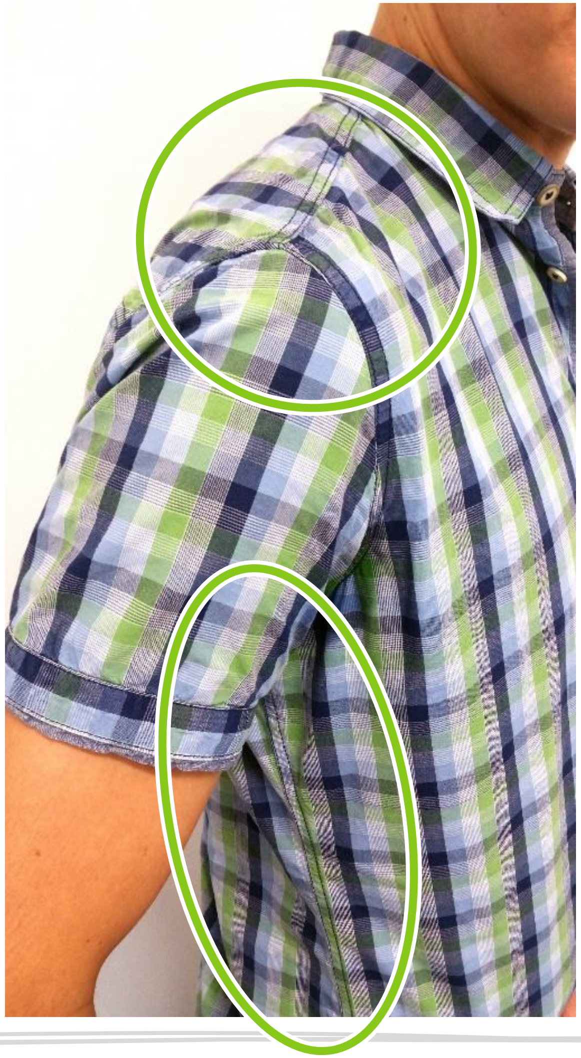




Good Cuts/Segmentations?

- Hide seams
- Small number/length of seams

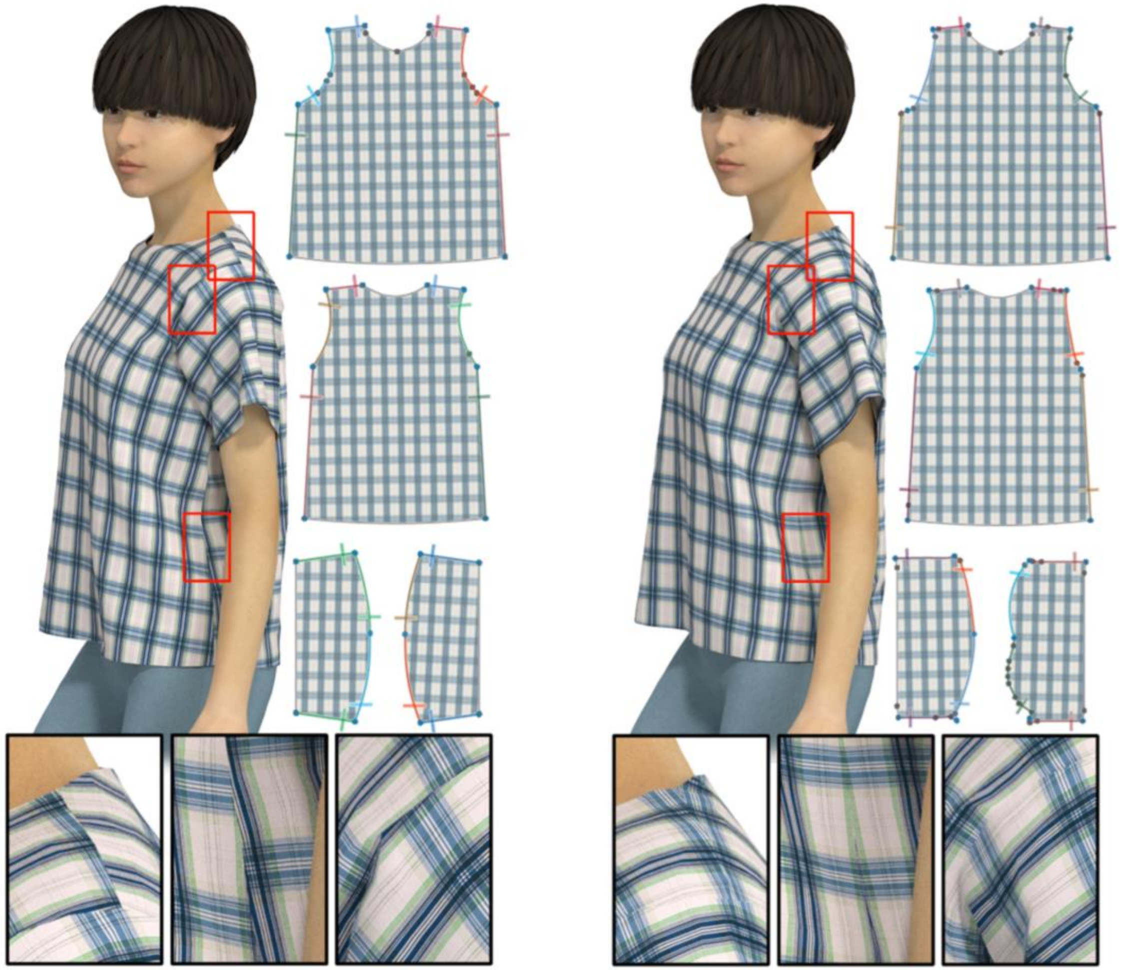
- Or: make parameterization continuous across seams!
  - Good for remeshing
  - Good for garment textures



Good Cuts/Segmentations?

- Hide seams
- Small number/length of seams

- Or: make parameterization continuous across seams!
  - Good for remeshing
  - Good for garment textures



“Wallpaper Pattern Alignment along Garment Seams“, Wolff and S.-H., SIGGRAPH 2019

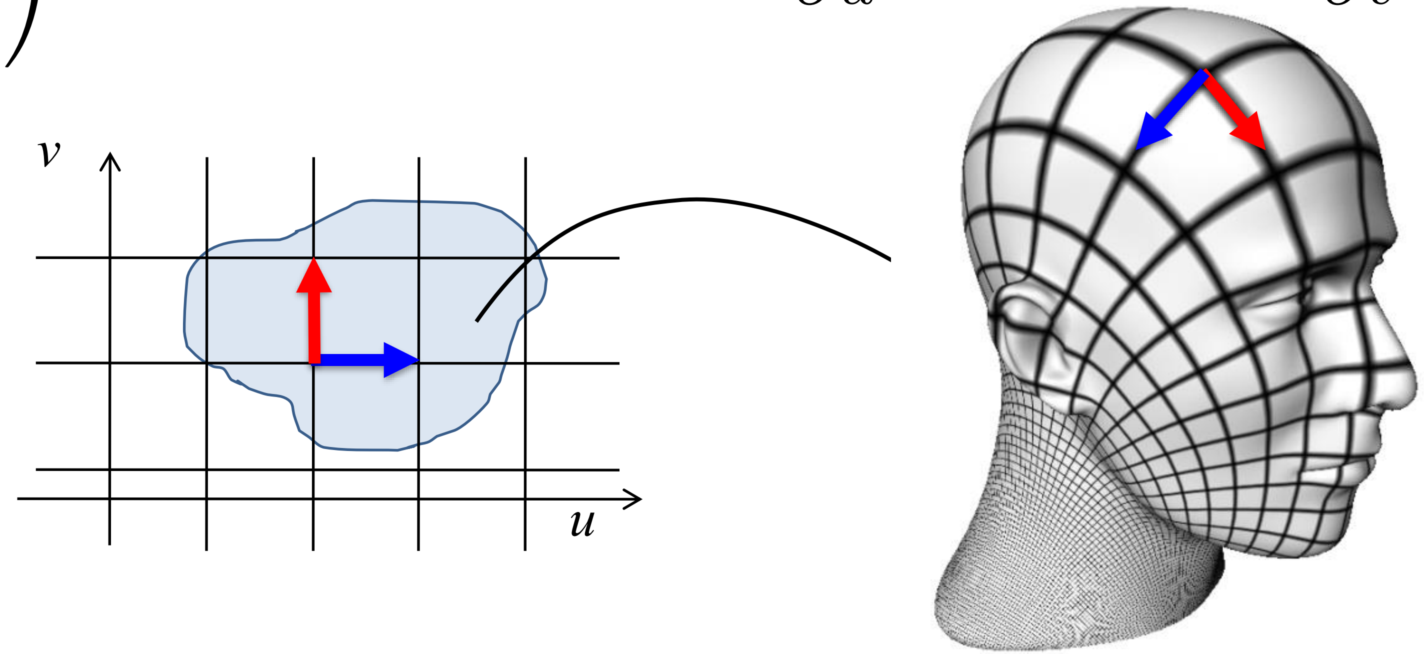
How to Measure Distortion?

Measures of Local Distortion

- Remember differential geometry

$$\mathbf{p}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix}, \quad (u,v) \in \mathbb{R}^2 \qquad \mathbf{p}_u = \frac{\partial \mathbf{p}(u,v)}{\partial u}, \quad \mathbf{p}_v = \frac{\partial \mathbf{p}(u,v)}{\partial v}$$

What happens to tangent vectors?

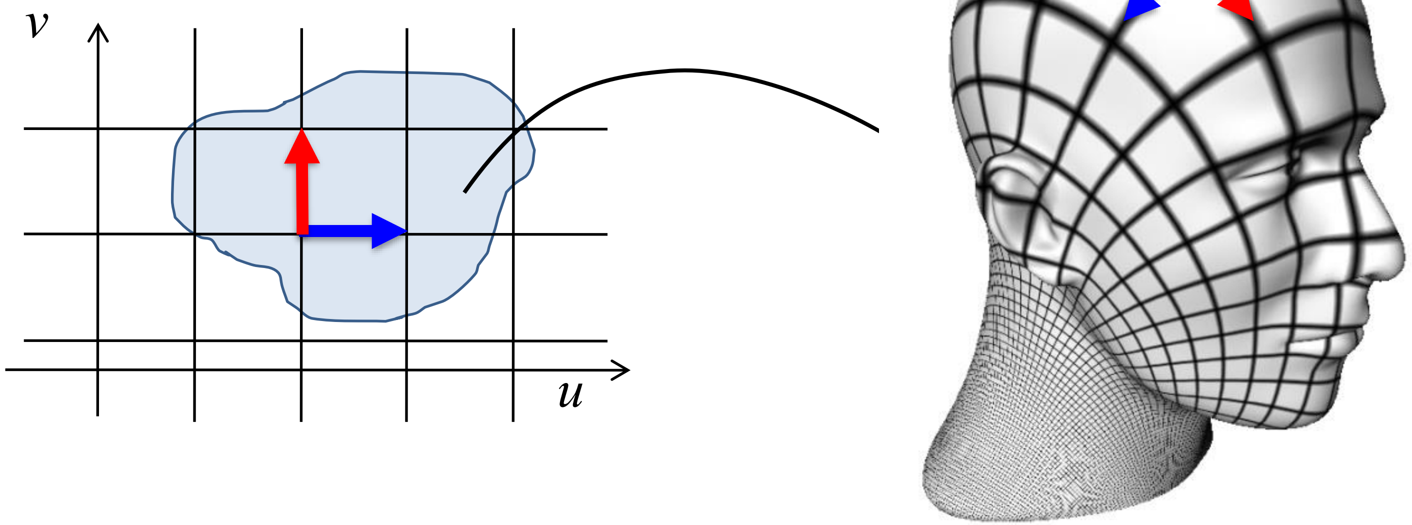


Measures of Local Distortion

- How do lengths and angles of tangents change?
  - First fundamental form!

$$\mathbf{I} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^\top \mathbf{p}_u & \mathbf{p}_u^\top \mathbf{p}_v \\ \mathbf{p}_u^\top \mathbf{p}_v & \mathbf{p}_v^\top \mathbf{p}_v \end{pmatrix}$$

What happens to tangent vectors?



Measures of Local Distortion

- How do lengths and angles of tangents change?
  - First fundamental form!

$$\mathbf{I} = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \mathbf{p}_u^\top \mathbf{p}_u & \mathbf{p}_u^\top \mathbf{p}_v \\ \mathbf{p}_u^\top \mathbf{p}_v & \mathbf{p}_v^\top \mathbf{p}_v \end{pmatrix}$$

Angle change (off-diagonal entries $\mathbf{p}_u^\top \mathbf{p}_v$) — Length change (diagonal entries $\mathbf{p}_u^\top \mathbf{p}_u$, $\mathbf{p}_v^\top \mathbf{p}_v$)

Area distortion = area element

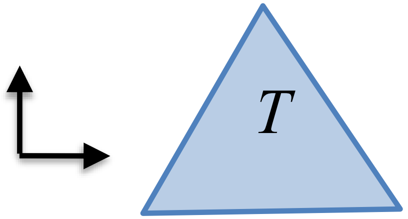
$$dA = \sqrt{EG - F^2}\,du\,dv = \det(\mathbf{I})$$

The eigenvalues of $\mathbf{I}$ indicate the maximal/minimal stretching of a tangent vector 💡

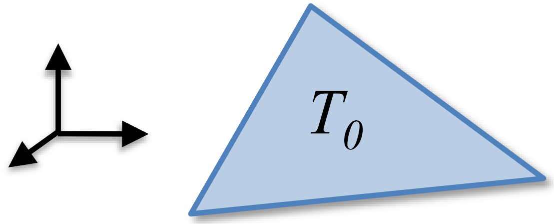
Distortion on Triangle Meshes?

Unique affine map

Triangle in 2D



Triangle in 3D



$$P : \mathbb{R}^2 \to \mathbb{R}^3, \quad P(\mathbf{u}) = A\mathbf{u} + \mathbf{c}$$

$$[P_u \;\; P_v] = A$$ 3×2 matrix (Jacobian) $\quad \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix}$

first fundamental form

$$\mathbf{I} = A^{\mathsf{T}} A$$

Singular Value Decomposition

$$A = U \begin{pmatrix} \Gamma & \\ & \sigma \end{pmatrix} V^{\mathsf{T}}$$

Γ, σ indicate distortion

Possible distortion measures:

$$E(T) = \sqrt{\Gamma^2 + \sigma^2}$$

Distortion magnitude

or

$$E(T) = \max\left\{\Gamma, \frac{1}{\sigma}\right\}$$

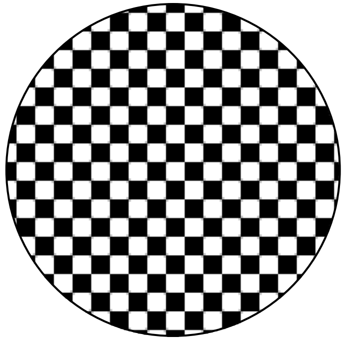
Max expansion vs compression

or …

Measures how much the triangle was deformed and stretched

How to compute good parameterizations? And quickly?

Distortion Minimization

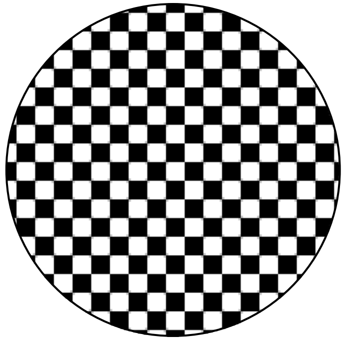

Texture map

$$\underset{(u_1,v_1),\dots,(u_n,v_n)}{\operatorname{argmin}} \sum_T A_T \, E(T)$$

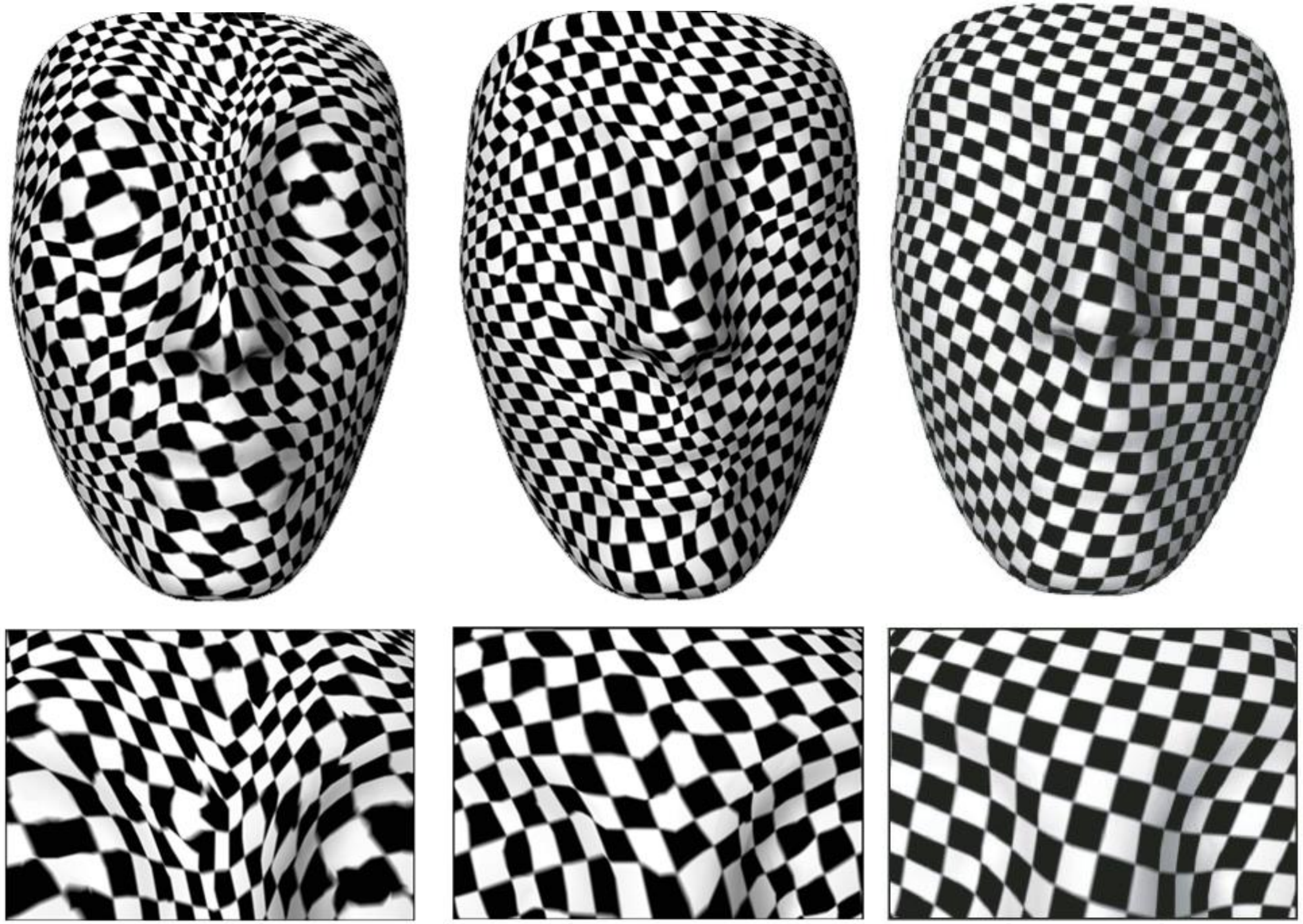
Integral of E over all triangles

Area of Trianlge T

Distortion Minimization



Texture map



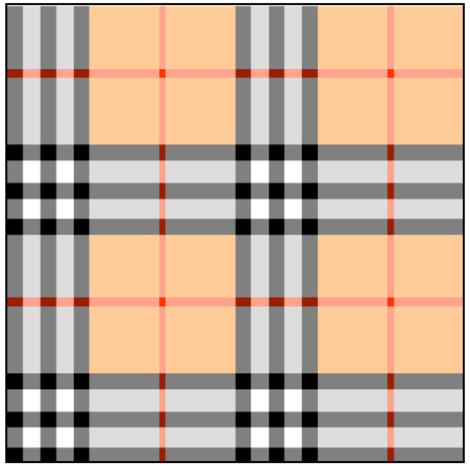
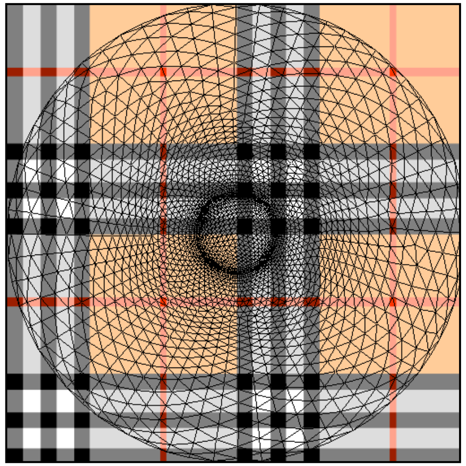
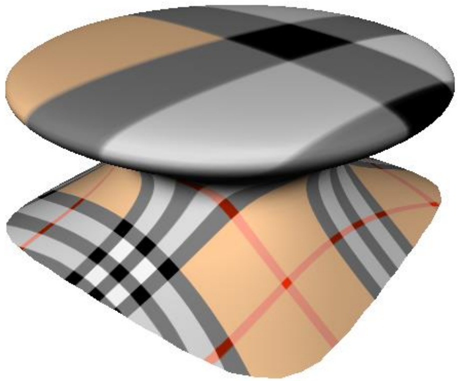
Kent et al ‘92 | Floater 97 | Sander et al ‘01

Literature for previous slide

- Kent et al. '92, "Shape transformation for polyhedral objects", Computer Graphics Vol. 26(2), 47-54
  http://ijcc.org/ojs/index.php/ijcc/article/viewFile/87/78
- Floater '97, "Parametrization and smooth approximation of surface triangulations", Computer Aided Geometric Design Vol. 14 (1997), 231-250
  http://www.mn.uio.no/math/english/people/aca/michaelf/papers/param.pdf
- Mesh Parameterization: Theory and Practice, Hormann, Polthier, Sheffer;
  https://www.inf.usi.ch/hormann/papers/Hormann.2008.MPT.pdf

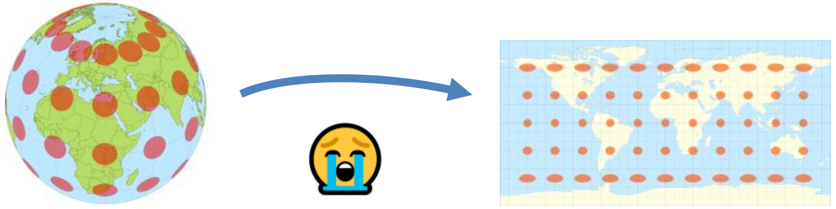
Area Distortion vs. Angle Distortion

- Is it possible to preserve both angles and areas at the same time? 🤔

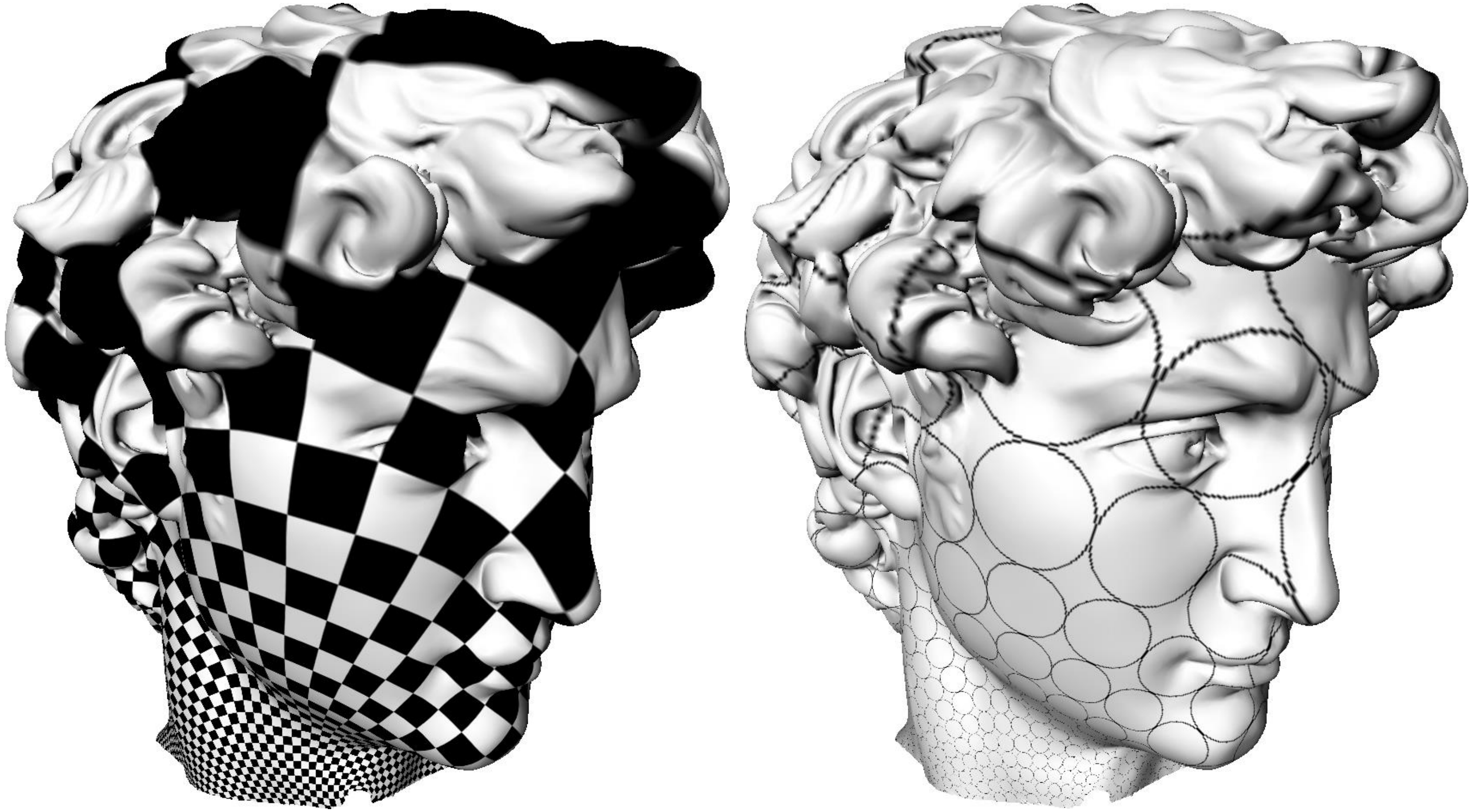
NO!

Not possible if $K \neq 0$.
(no perfect earth mapping)



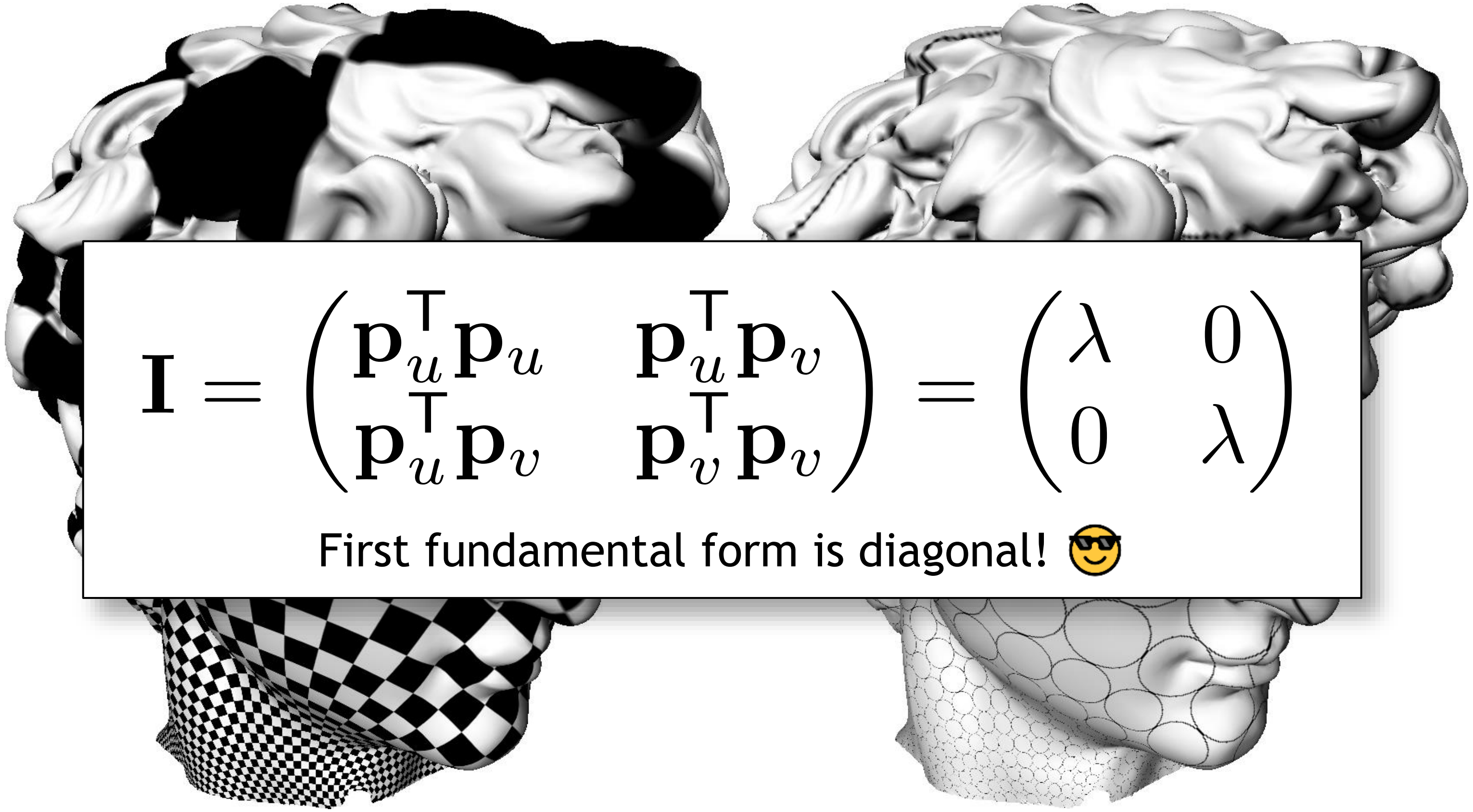
Conformal Parameterization

- Angle preservation; circles are mapped to circles



Conformal Parameterization

- Angle preservation; circles are mapped to circles



$$\mathbf{I} = \begin{pmatrix} \mathbf{p}_u^\top \mathbf{p}_u & \mathbf{p}_u^\top \mathbf{p}_v \\ \mathbf{p}_u^\top \mathbf{p}_v & \mathbf{p}_v^\top \mathbf{p}_v \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

First fundamental form is diagonal! 😎

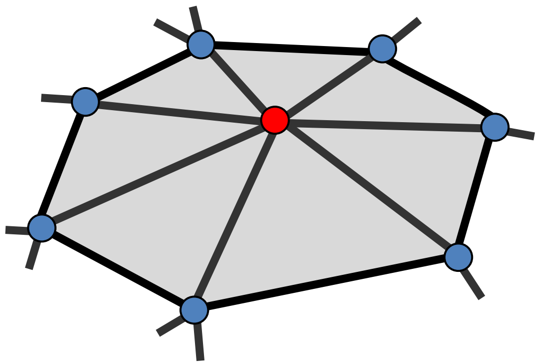
Discrete Conformal Parameterization

$$\mathbf{I} = \begin{pmatrix} \mathbf{p}_u^\top \mathbf{p}_u & \mathbf{p}_u^\top \mathbf{p}_v \\ \mathbf{p}_u^\top \mathbf{p}_v & \mathbf{p}_v^\top \mathbf{p}_v \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

- Next week: discrete conformal energy formulation $E(T)$ - **Least Squares Conformal Maps** (LSCM)
- Today: **harmonic mapping**, a simpler but more limited method

Harmonic Mapping – Idea

Want to flatten the mesh ⟷ no curvature ⟷ Laplace operator gives zero



$\mathbf{u} = (u,v)$ domain

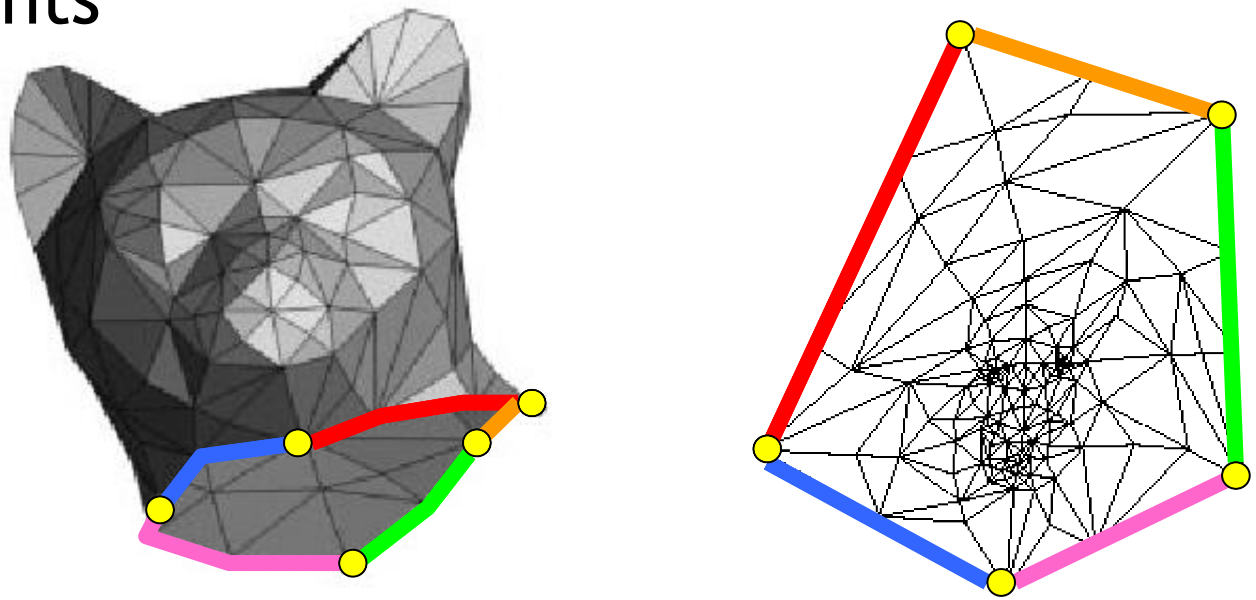
$$\Delta(\mathbf{u}) = 0$$

need boundary constraints
to prevent trivial solution;

which Laplacian operator?
(which weights?)

Convex Mapping (Tutte, Floater)

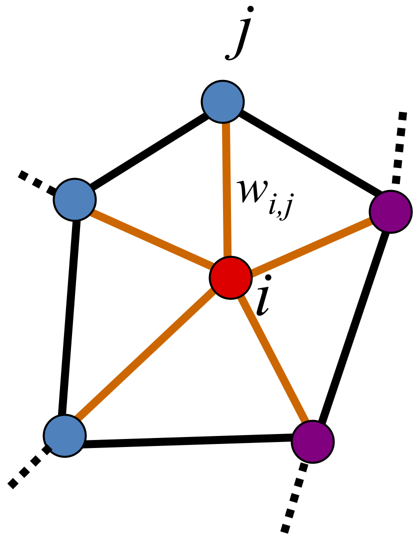
- Boundary vertices are fixed
- Convex weights



$\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ - inner vertices (free to move)
$\mathbf{u}_{n+1}, \ldots, \mathbf{u}_N$ - boundary vertices (fixed)

Convex Mapping (Tutte, Floater)

- Boundary vertices are fixed
- Convex weights



For the inner vertices $\mathbf{u}_1, \ldots, \mathbf{u}_n$, solve for $L(u_i) = 0$

$$L(\mathbf{u}_i) = \frac{1}{W_i} \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{u}_j - \mathbf{u}_i) = 0, \quad i = 1, \ldots, n$$

$$w_{ij} > 0$$

Convex Mapping (Tutte, Floater)

- Solve the linear system

$$L\mathbf{u} = 0 \qquad \mathbf{u} \in \mathbb{R}^{n\times 2}$$

⭐ The values of the boundary vertices are known and thus this system can be solved. ⭐

unknown

$$\begin{bmatrix} L_{II} & L_{IB} \\ L_{BI} & L_{BB} \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow L_{II}\mathbf{u}_I + L_{IB}\mathbf{u}_B = 0 \quad \Rightarrow L_{II}\mathbf{u}_I = -L_{IB}\mathbf{u}_B$$

known

Block matrices

First matrix block row

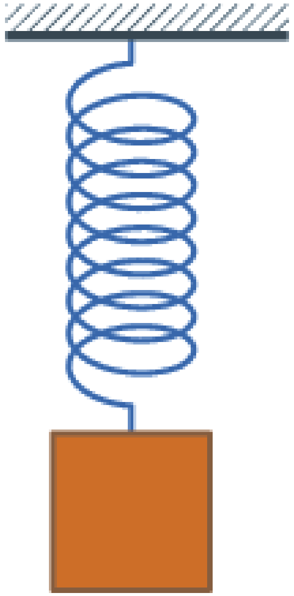
known

Solve for $\mathbf{u}_I$

Harmonic Mapping

Different approach for flattening:

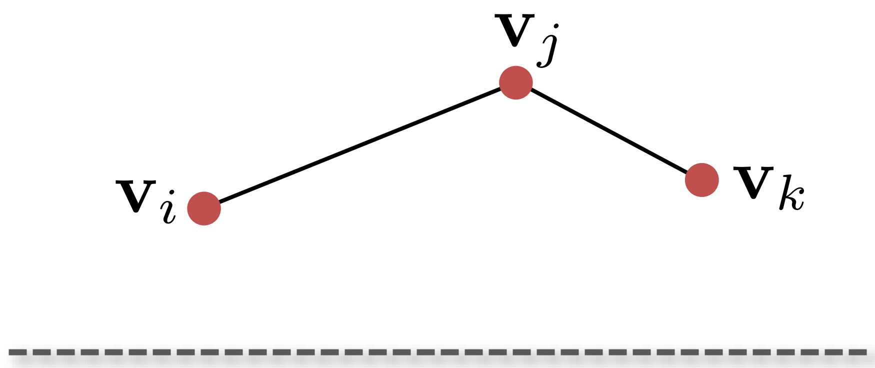
Mass spring system energy minimization.



https://en.m.wikipedia.org/wiki/File:Damped_spring.gif

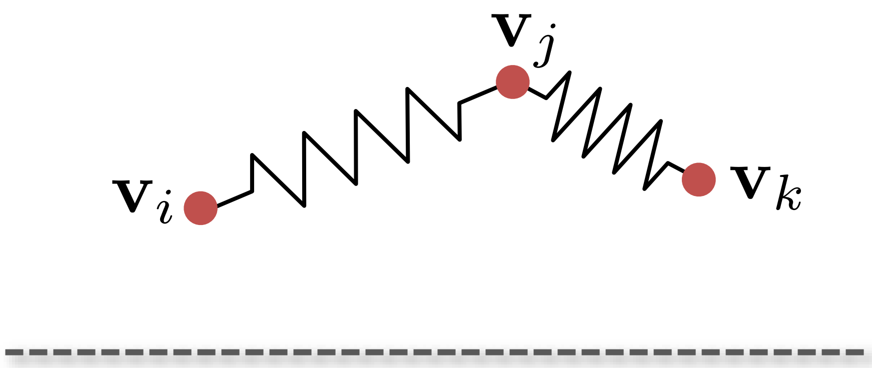
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



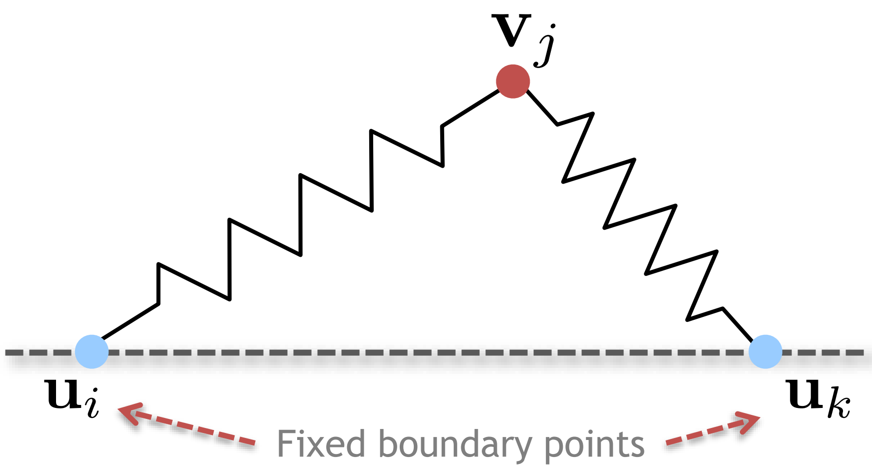
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



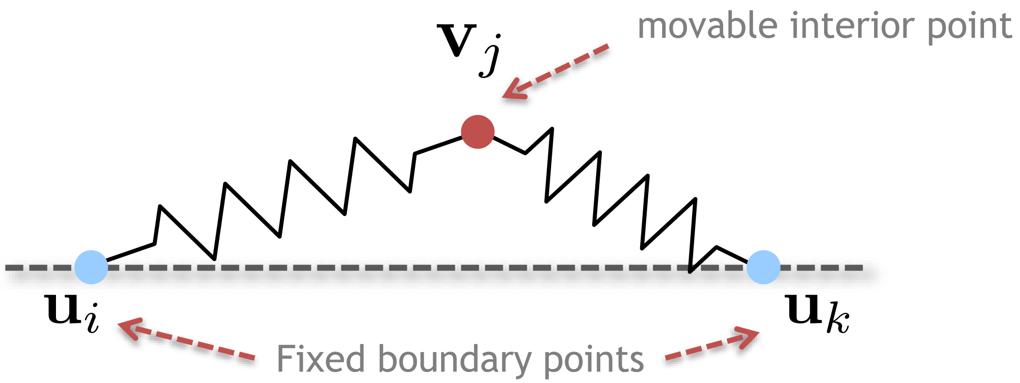
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



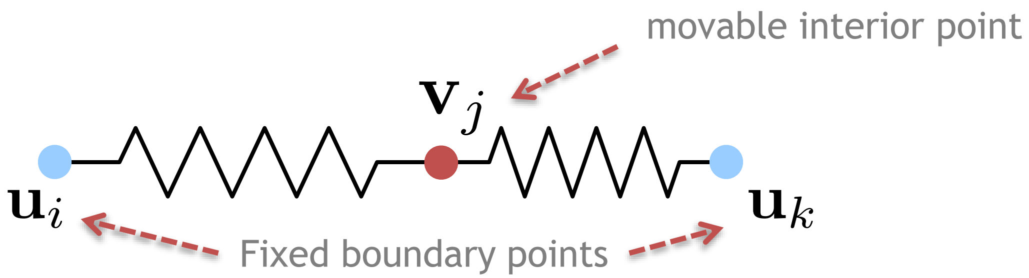
Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



Harmonic Mapping

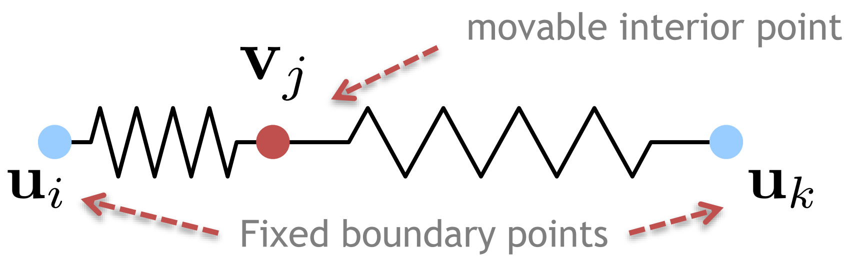
- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane



Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane

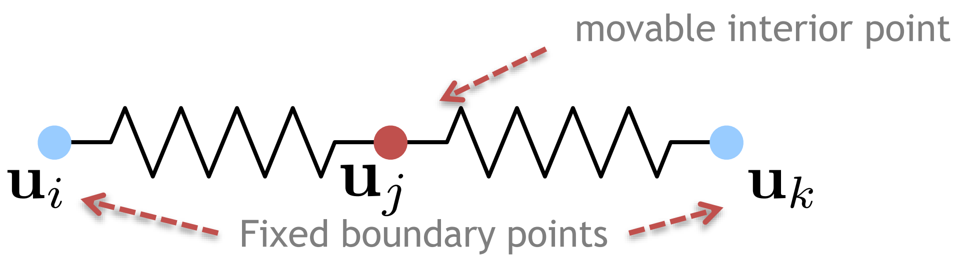
No equilibrium 🤢



Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane

Equilibrium 💖



Spring energy:

(linearized)

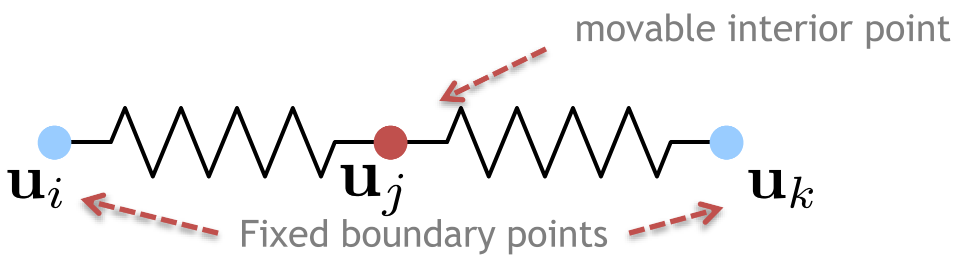
$$\frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

$$\mathbf{u}_i, \mathbf{u}_j \in \mathbb{R}^2$$

Harmonic Mapping

- Inner mesh edges as springs
- Find minimum-energy state where all vertices lie in the 2D plane

Equilibrium 💖



⭐ We can solve for the exact interior points that minimize E ⭐

Spring energy of flattened mesh:

(linearized)

$$E(\mathbf{u}_1, \ldots, \mathbf{u}_n) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

Minimizing Spring Energy

$$E(\mathbf{u}_1, \ldots, \mathbf{u}_n) = \sum_{(i,j)\in\mathcal{E}} \frac{1}{2} k_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2$$

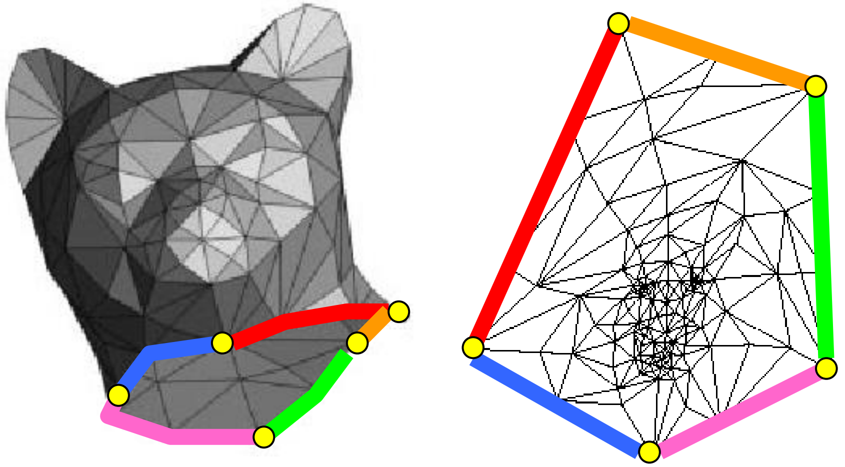
$$\frac{\partial E(\mathbf{u}_1, \ldots, \mathbf{u}_n)}{\partial \mathbf{u}_i} = \sum_{j\in\mathcal{N}(i)} k_{i,j}(\mathbf{u}_i - \mathbf{u}_j) = 0$$ (Derive only by $\mathbf{u}_i$)

$$\sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\mathbf{u}_i + \sum_{j\in\mathcal{N}(i)\setminus\mathcal{B}} k_{i,j}(\mathbf{u}_i - \mathbf{u}_j) = \sum_{j\in\mathcal{N}(i)\cap\mathcal{B}} k_{i,j}\mathbf{u}_j$$

Separate boundary and interior $\mathbf{u}_i$, $\mathbf{u}_j$

unknown flat vertex positions

known fixed boundary positions



$\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ - inner vertices (free to move)

$\mathbf{u}_{n+1}, \ldots, \mathbf{u}_N$ - boundary vertices (fixed)

Minimizing Spring Energy

- Sparse linear system of n equations to solve!

$$\sum_{j \in \mathcal{N}(i) \cap \mathcal{B}} k_{i,j} \mathbf{u}_i + \sum_{j \in \mathcal{N}(i) \setminus \mathcal{B}} k_{i,j} (\mathbf{u}_i - \mathbf{u}_j) = \sum_{j \in \mathcal{N}(i) \cap \mathcal{B}} k_{i,j} \mathbf{u}_j$$

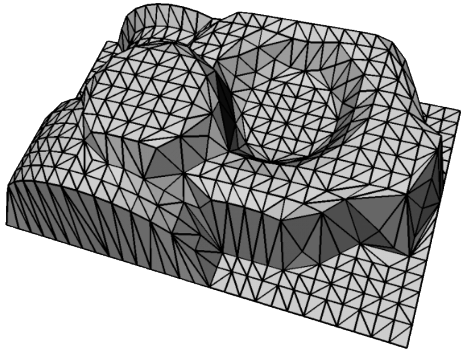
Matrix representation of the above equation:

$$\begin{pmatrix} \sum_j k_{i,j} & * & \cdots & -k_{i,j} \\ * & \sum_j k_{i,j} & * & \vdots \\ \vdots & * & \ddots & * \\ -k_{j,i} & \cdots & * & \sum_j k_{i,j} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{u}}_1 \\ \bar{\mathbf{u}}_2 \\ \vdots \\ \bar{\mathbf{u}}_n \end{pmatrix}$$

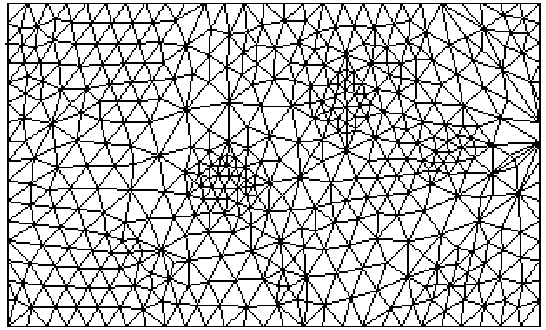
Interior (unknown)

Boundary (known)

Choice of spring constants $k_{i,j}$

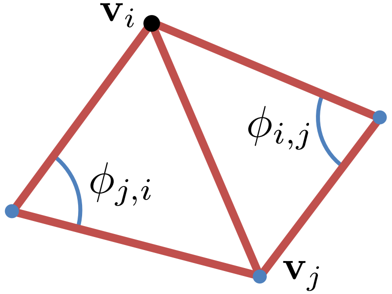


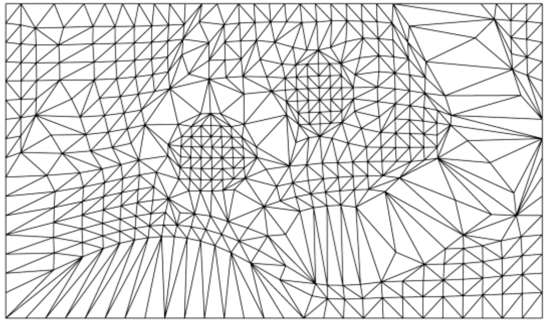
- Uniform $k_{i,j} = 1$



More regular triangles

- Cotan $k_{i,j} = \cot \phi_{i,j} + \cot \phi_{j,i}$





Less distortion

Tutte's Theorem

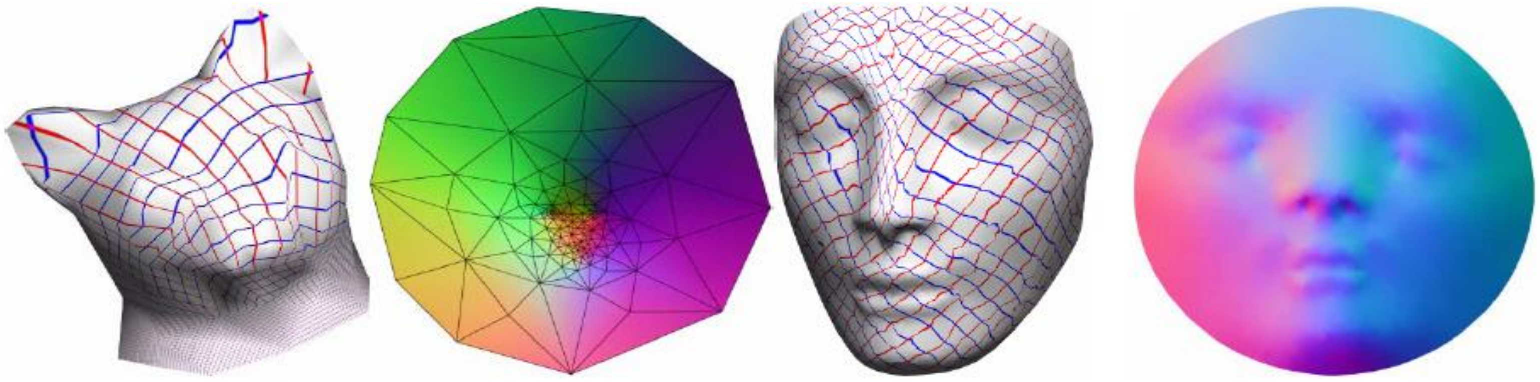
- If the weights are **nonnegative**, and the boundary is fixed to a **convex** polygon, the parameterization is **bijective.**

- Tutte'63 proved for uniform weights, Floater'97 extended to arbitrary nonnegative weights.

W.T. Tutte. "How to draw a graph". Proceedings of the London Mathematical Society, 13(3):743-768, 1963.

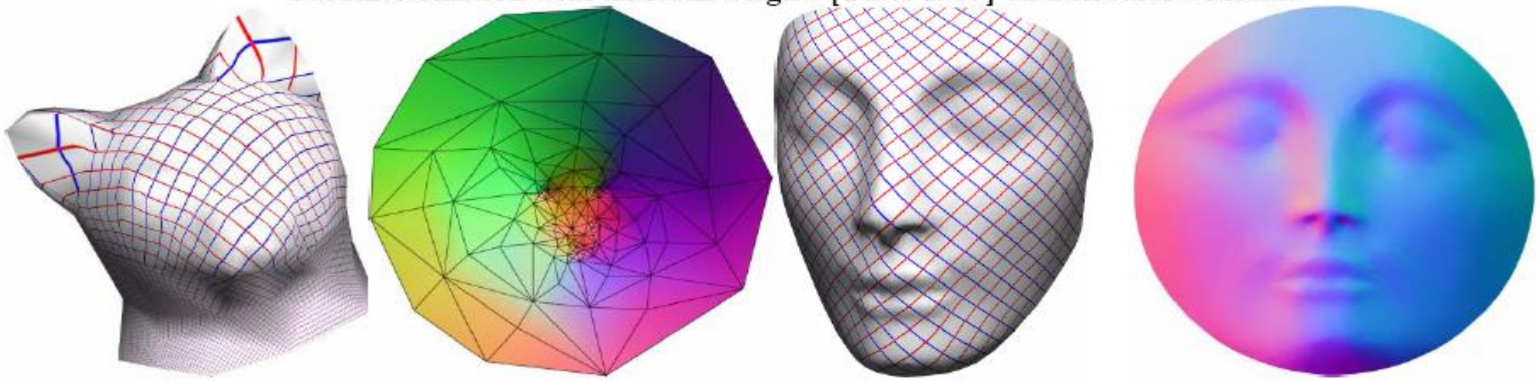
Comparison of Weights

uniform weights



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

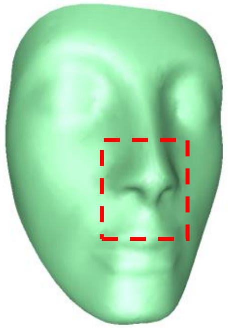
cotan weights



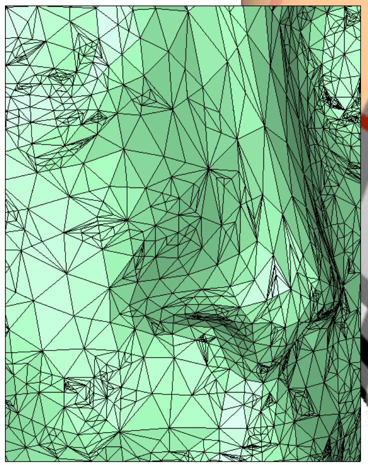
Parameterization with harmonic weights [Eck et al. 1995] on a circular domain.

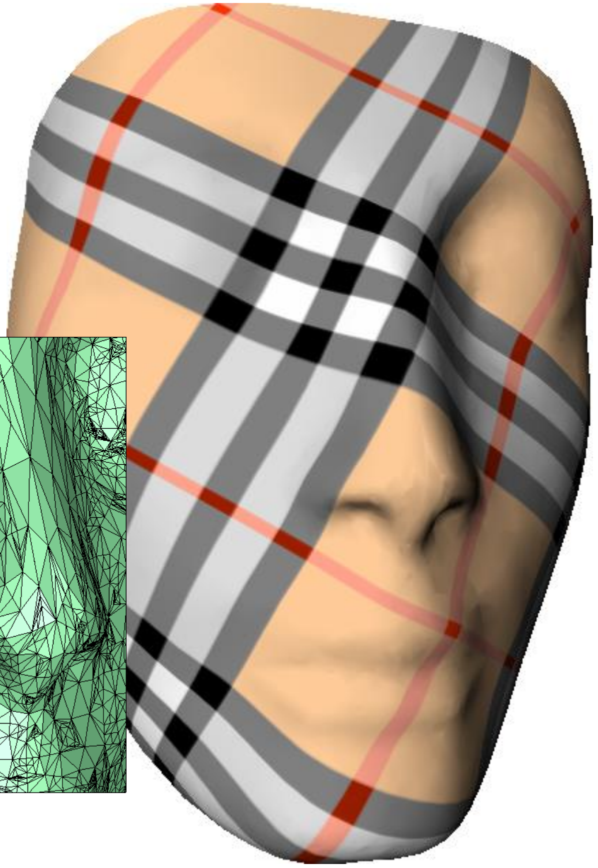
Eck et al. 1995, “Multiresolution analysis of arbitrary meshes”, SIGGRAPH 1995

Comparison of Weights



Terrible input mesh 👻





cotan 💖

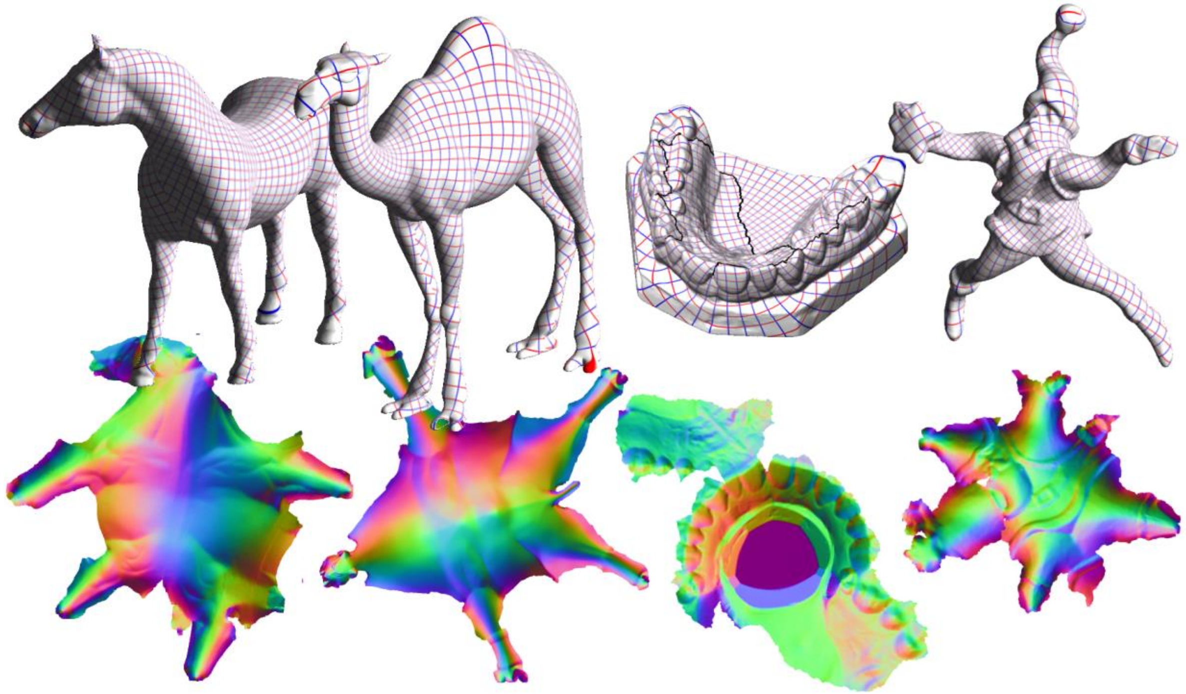
uniform 🤢

Discussion

- The results of cotan-weights mapping are better than those of uniform convex mapping
  - (local area and angles preservation).
- But: the mapping is not always legal
  - the cotan weights can be negative $\omega_{ij} < 0$ for badly-shaped triangles
  - this leads to self-intersections 💣 💥
- In any case:
  - sparse system to solve, so robust and efficient numerical solvers exist. 😇

Discussion

- Both mappings have the problem of fixed boundary – it constrains the minimization and causes distortion. Which boundary is „best“? Unlcear!
- More advanced methods do not require boundary conditions.



ABF++ method,
Sheffer et al. 2005
http://www.cs.ubc.ca/~sheffa/ABF++/abf.htm

Thank You!

INTERACTIVE GEOMETRY LAB

April 2, 2025

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich