252-0538-00L, Spring 2025

Shape Modeling and Geometry Processing

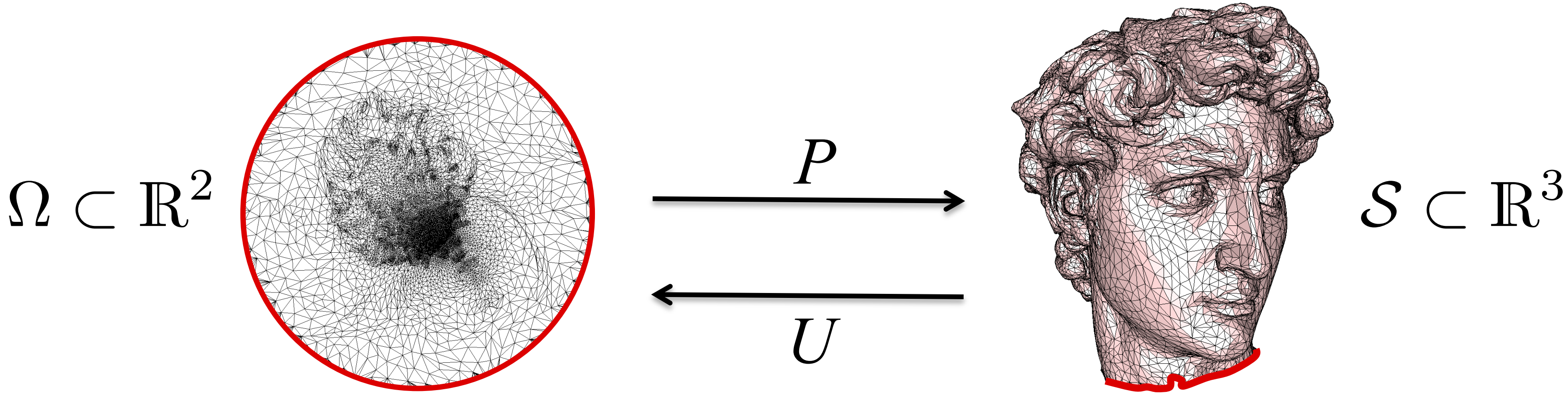
Least Squares Conformal Maps
As-Rigid-As-Possible Parameterization
Remeshing

INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

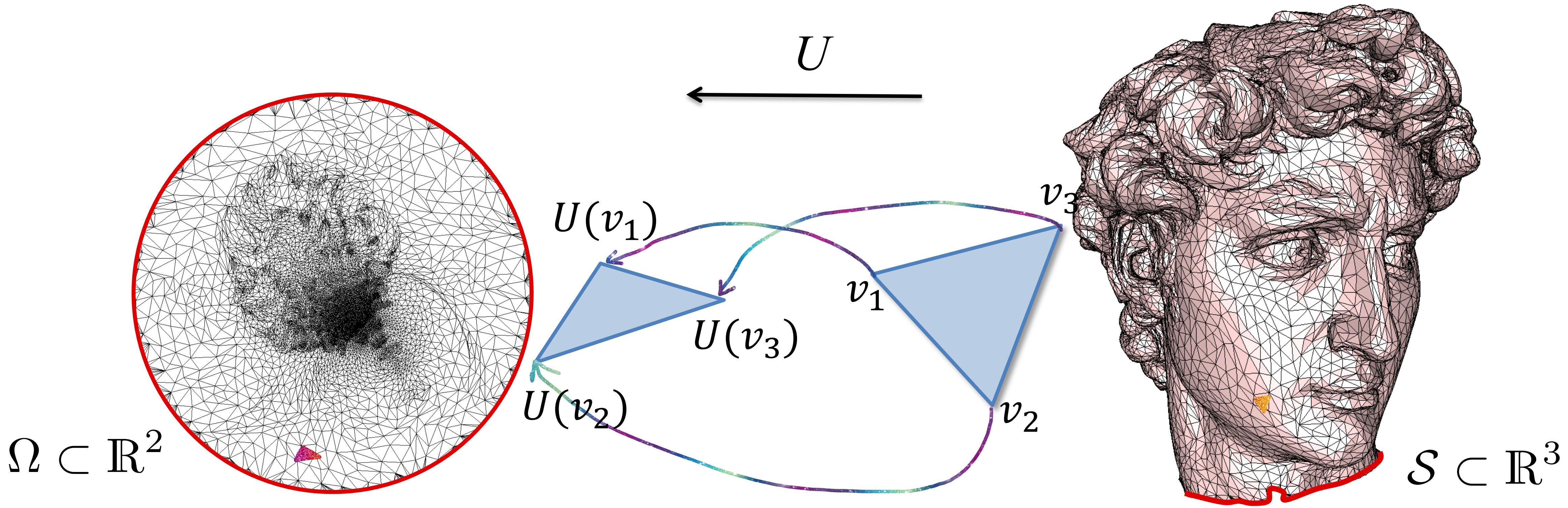
Parameterization – Definition

- Parameterization mapping P : 2D domain Ω → 3D mesh S
- Flattening mapping U : 3D mesh S → 2D domain Ω $\quad U(\mathbf{v}_i) = (u_i, v_i)$
- Inverse(P) = flattening U.
- (sometimes we also call U as the "parameterization" of input mesh S)



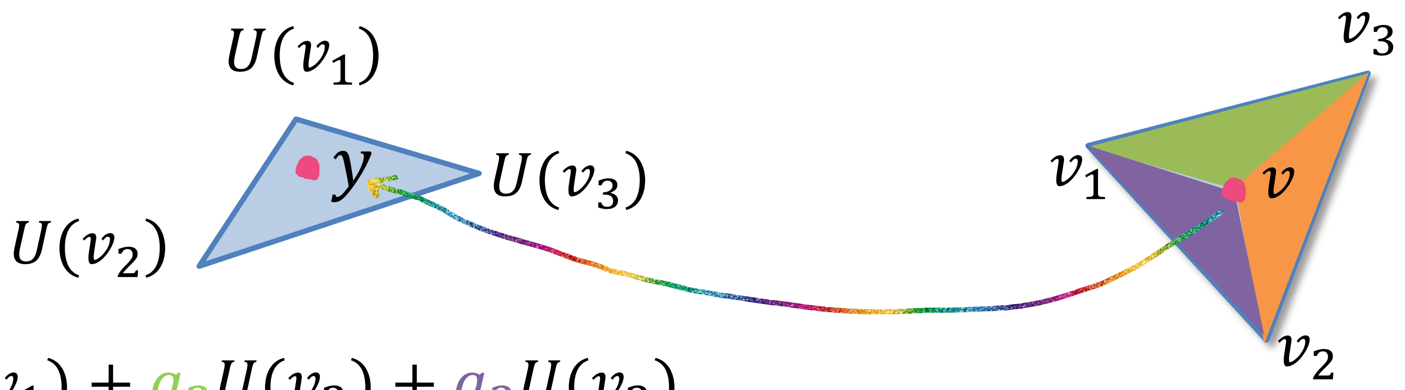
Parameterization – Definition

- Flatting mapping U: 3D mesh $S \rightarrow$ 2D domain Ω
- Each mesh vertex has a corresponding 2D position: $U(\mathbf{v}_i) = (u_i, v_i)$



Parameterization – Definition

- Flatting mapping U : 3D mesh $S \rightarrow$ 2D domain Ω
- Each mesh vertex has a corresponding 2D position: $U(\mathbf{v}_i) = (u_i, v_i)$
- Inside each triangle, the mappings P and U are affine (barycentric coordinates)

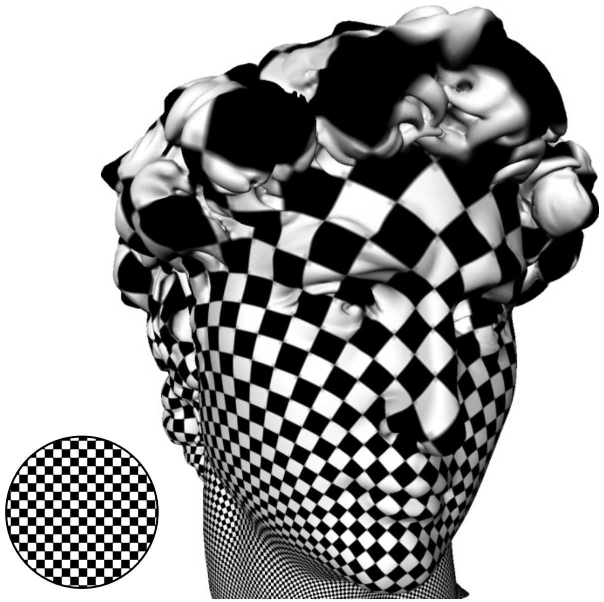


$$v = a_1 v_1 + a_2 v_2 + a_3 v_3$$

$$y = a_1 U(v_1) + a_2 U(v_2) + a_3 U(v_3)$$

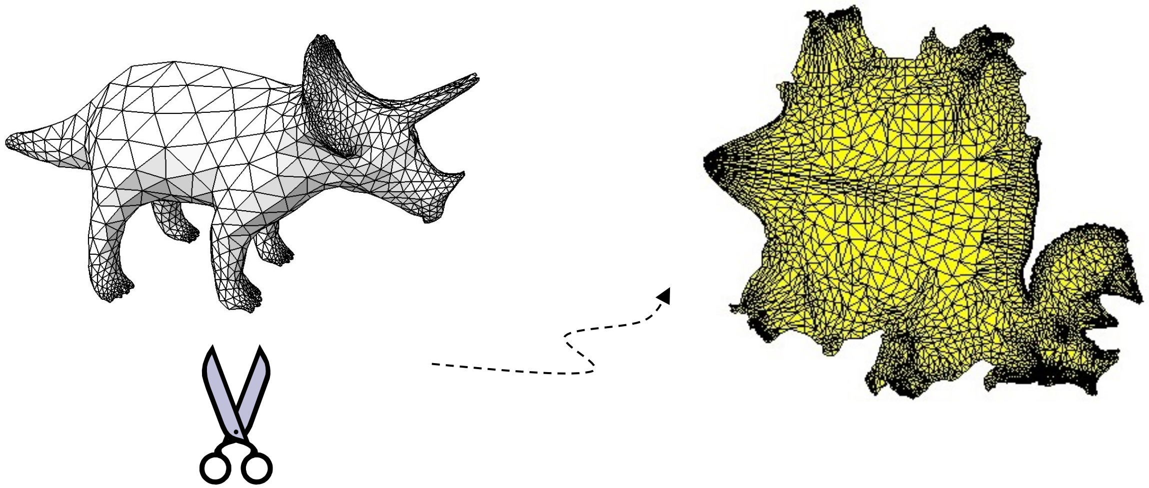
Parameterization – quality

- Minimize distortion E



minimize angular/conformal distortion

- Minimize/hide cuts

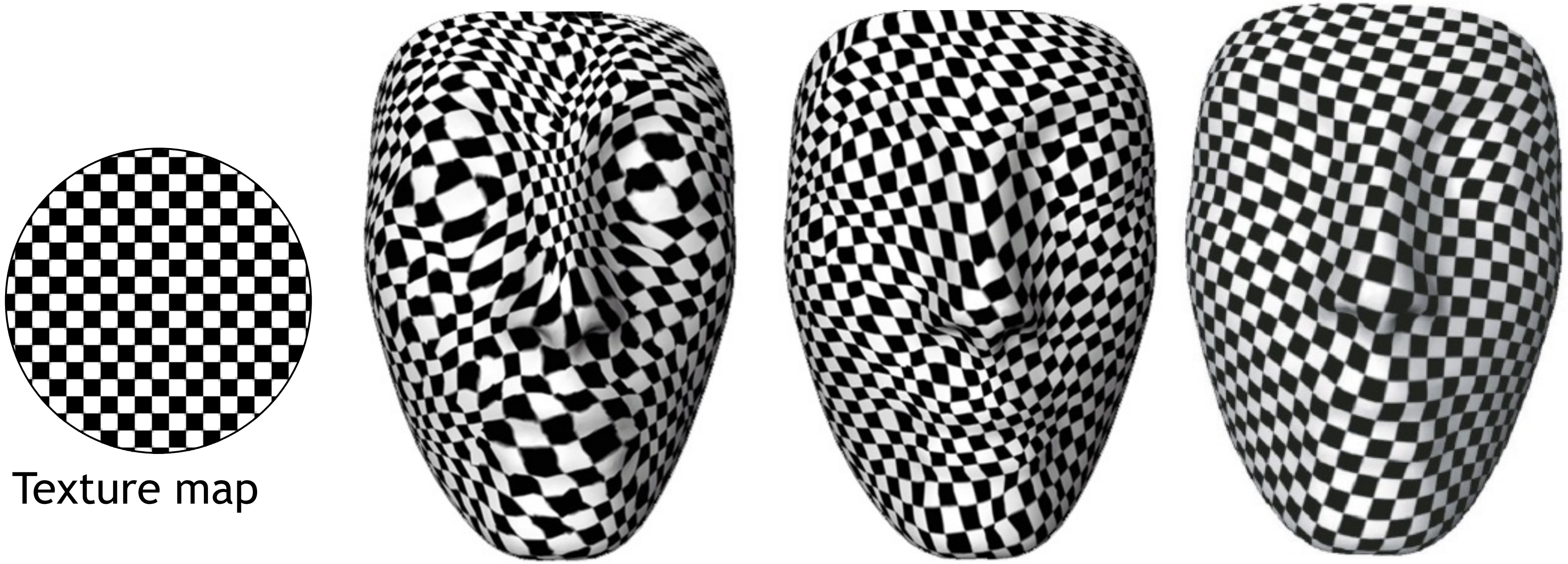


free boundary helps distortion minimization

Distortion on Triangle Meshes?

- Triangle in 3D $\Longleftrightarrow$ triangle in 2D
- Unique affine mapping

Distortion Minimization



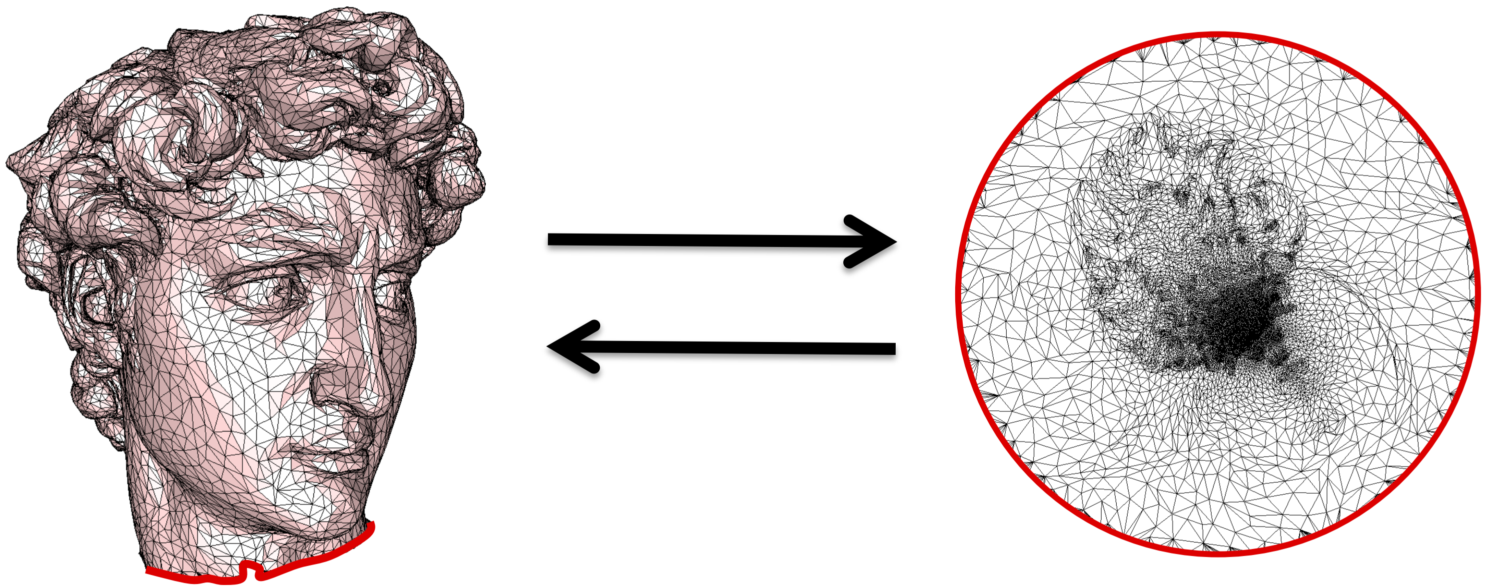
$$\underset{(u_1,v_1),\ldots,(u_n,v_n)}{\operatorname{argmin}} \sum_T A_T\, E(T)$$

triangle area in 3D

Recap: Harmonic mapping

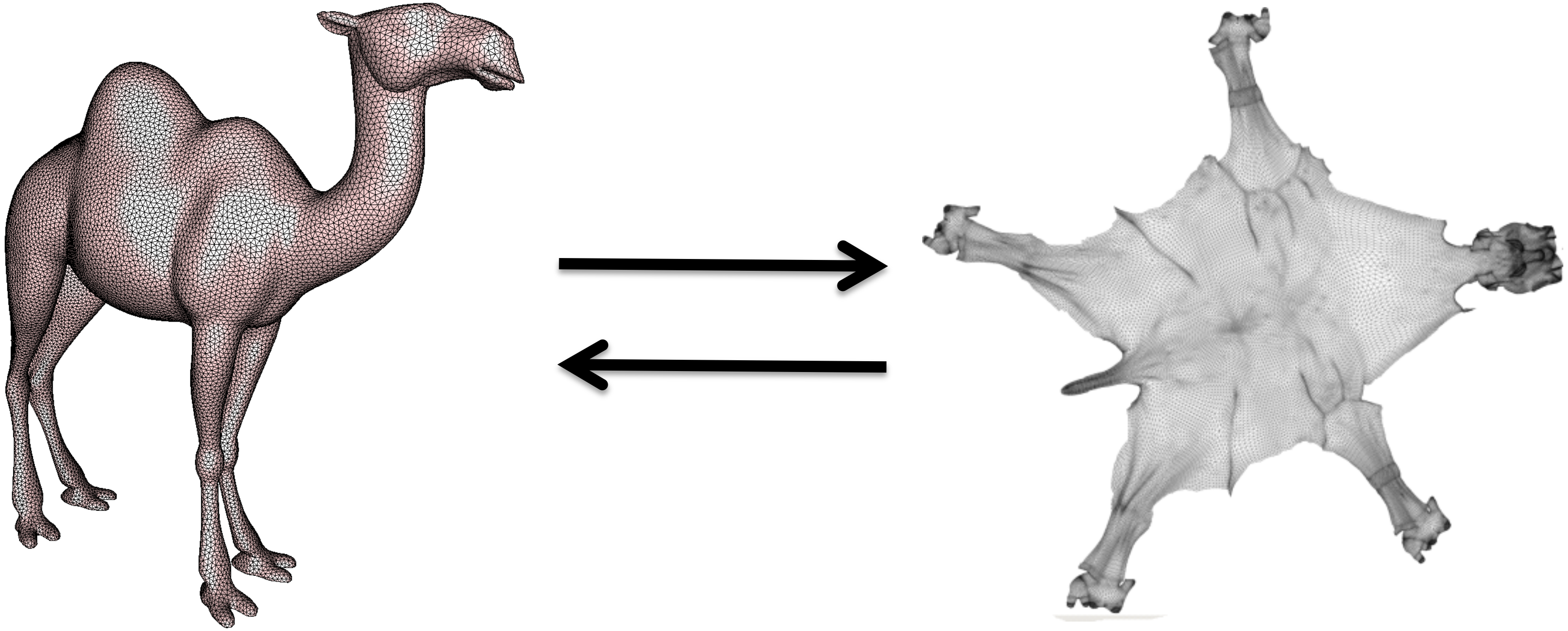
- Minimize linearized spring energy
- Boundary vertices must be fixed a priori

  ➔ Less freedom, more distortion

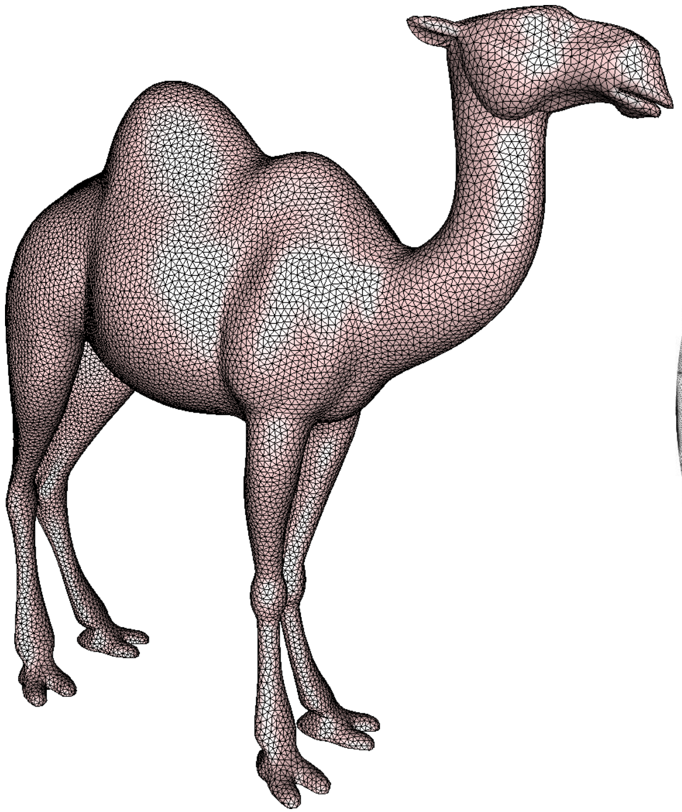


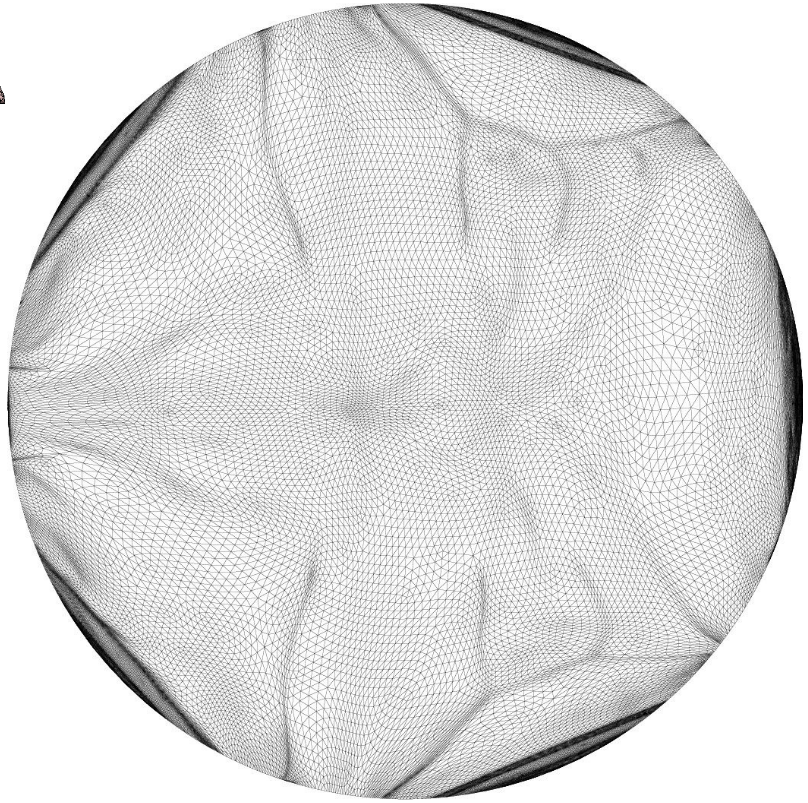
Today: free boundary param.

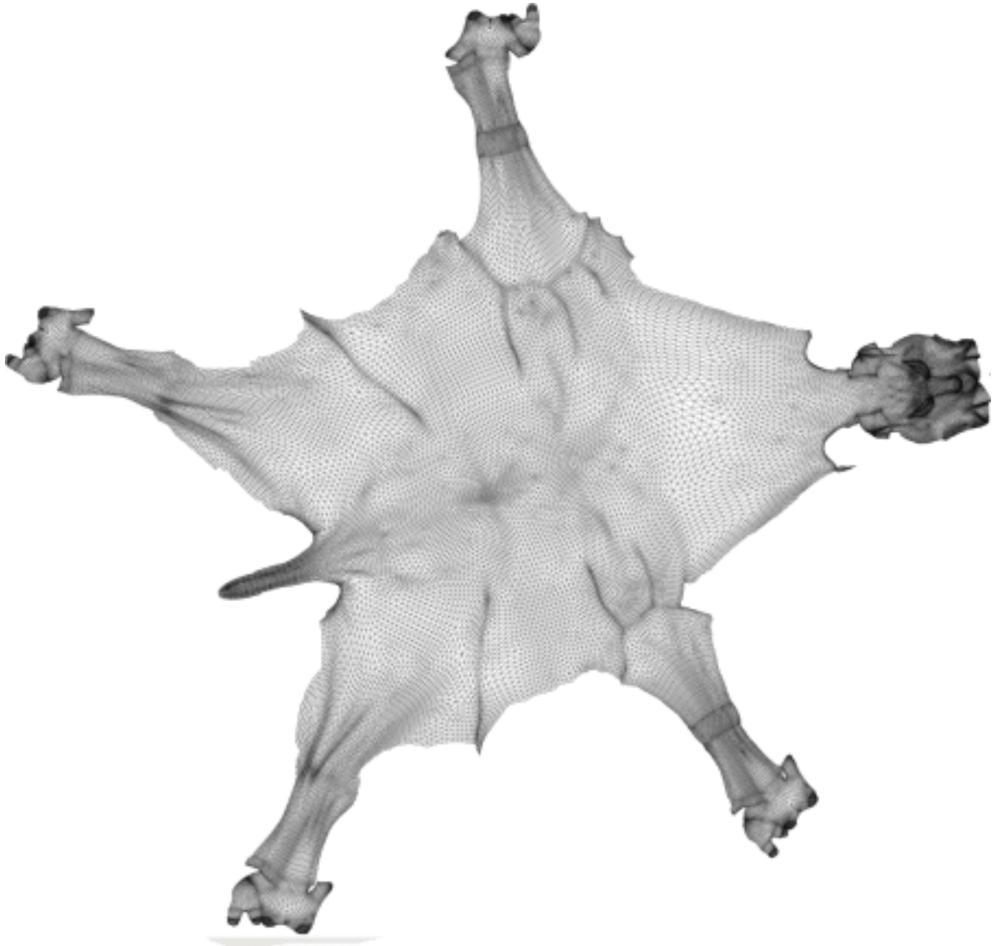
- Boundary vertices are free in optimization
  - ➔ "Good" boundary positions automatically obtained ☺



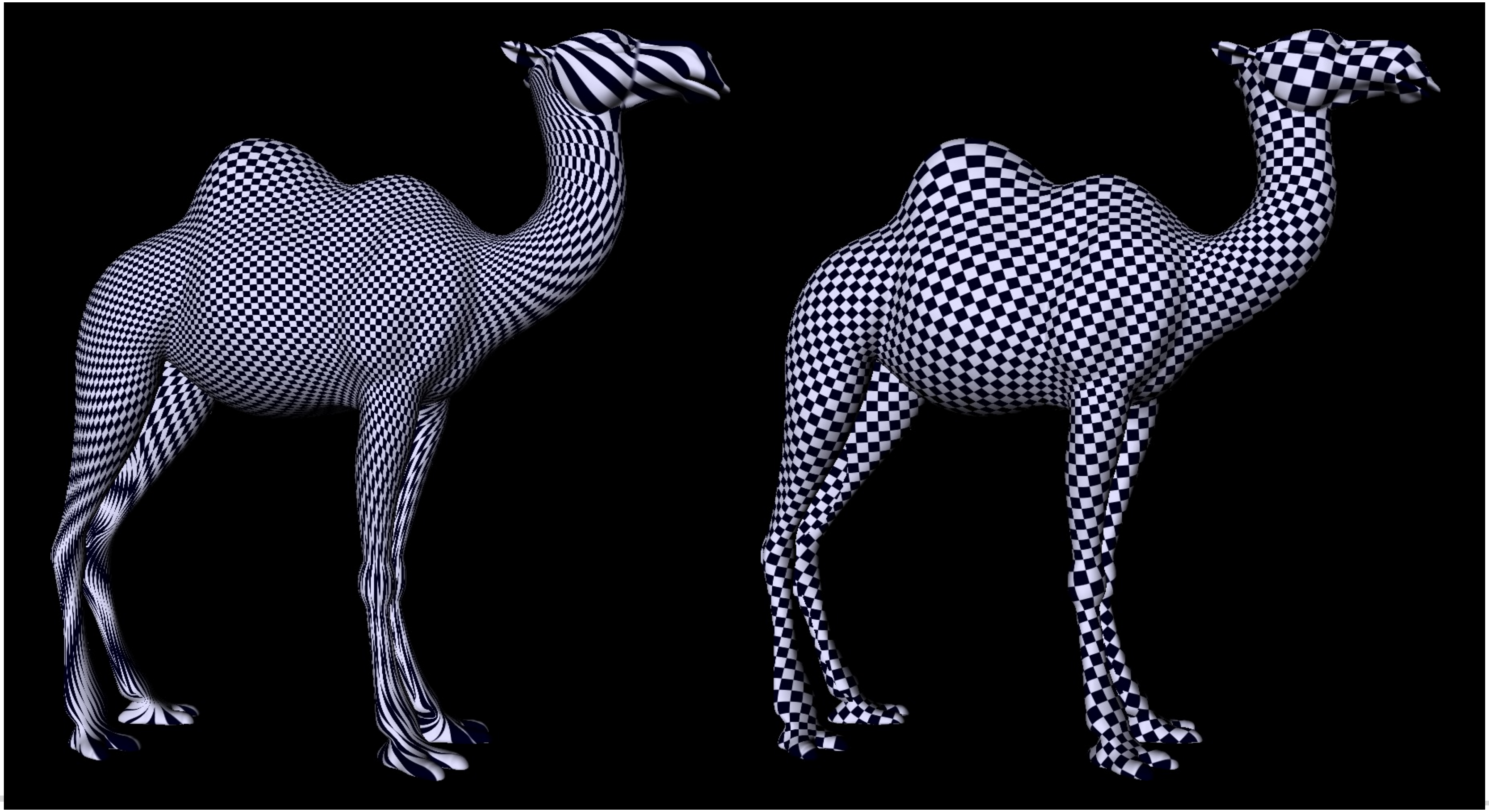
Fixed vs. free boundary







Fixed vs. free boundary



Literature for today

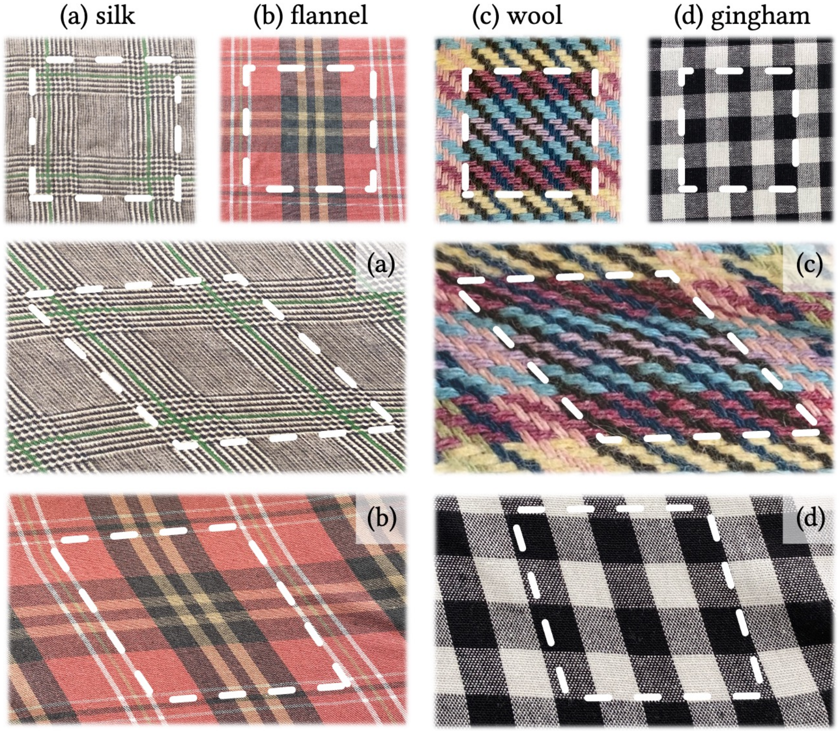
1. Angle-preserving parameterization(**LSCM**)
   - "Least Squares Conformal Maps", Lévy et al., SIGGRAPH 2002
2. Isometric parameterization (**ARAP**)
   - "A Local/Global Approach to Mesh Parameterization ",Liu et al., SGP 2008

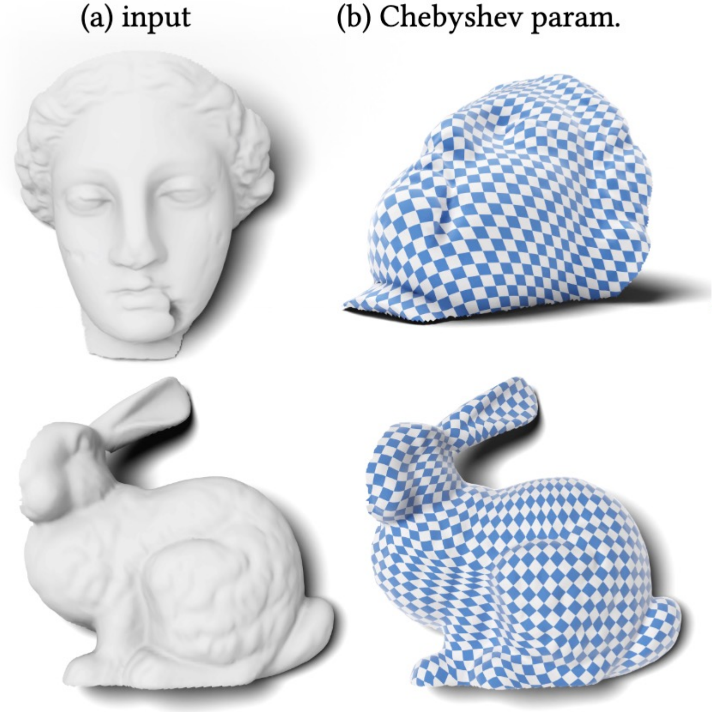
LSCM : "Conformal"

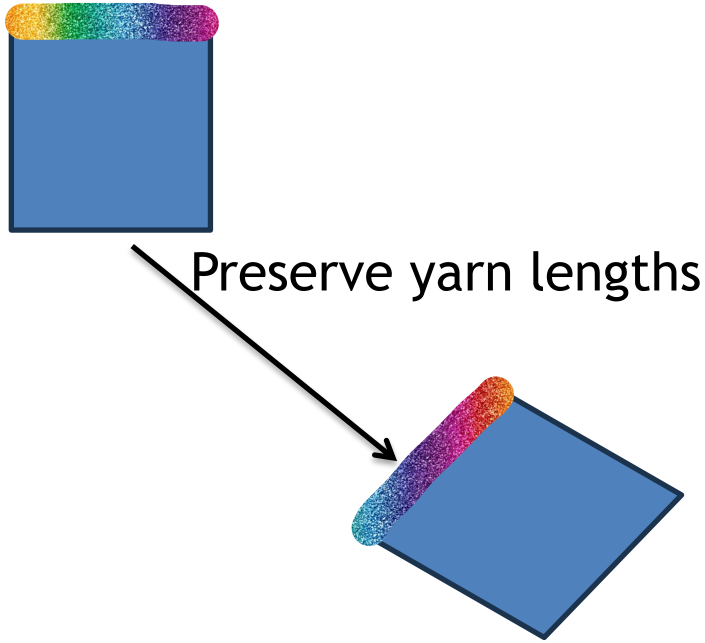
ARAP : "Isometric"

Literature for today

3. length-preserving along yarn directions
   - "Chebyshev Parameterization for Woven fabric Modeling", Oehri et al., SIGGRAPH Asia 2024







Discrete distortion energies

igl

ETH zürich

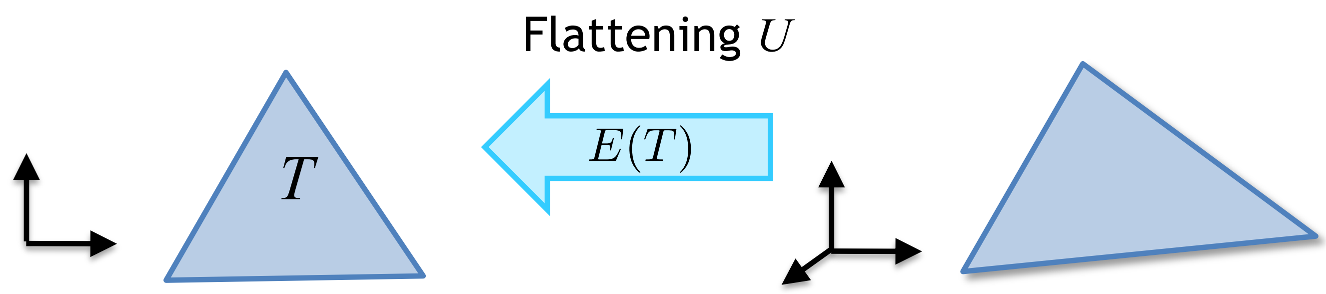
Notations

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \; E(X)$$

Variables: $X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^{\mathsf{T}}$

Distortion energy: $E(X) = \sum_{\text{triangles } T} A_T \, E(T)$

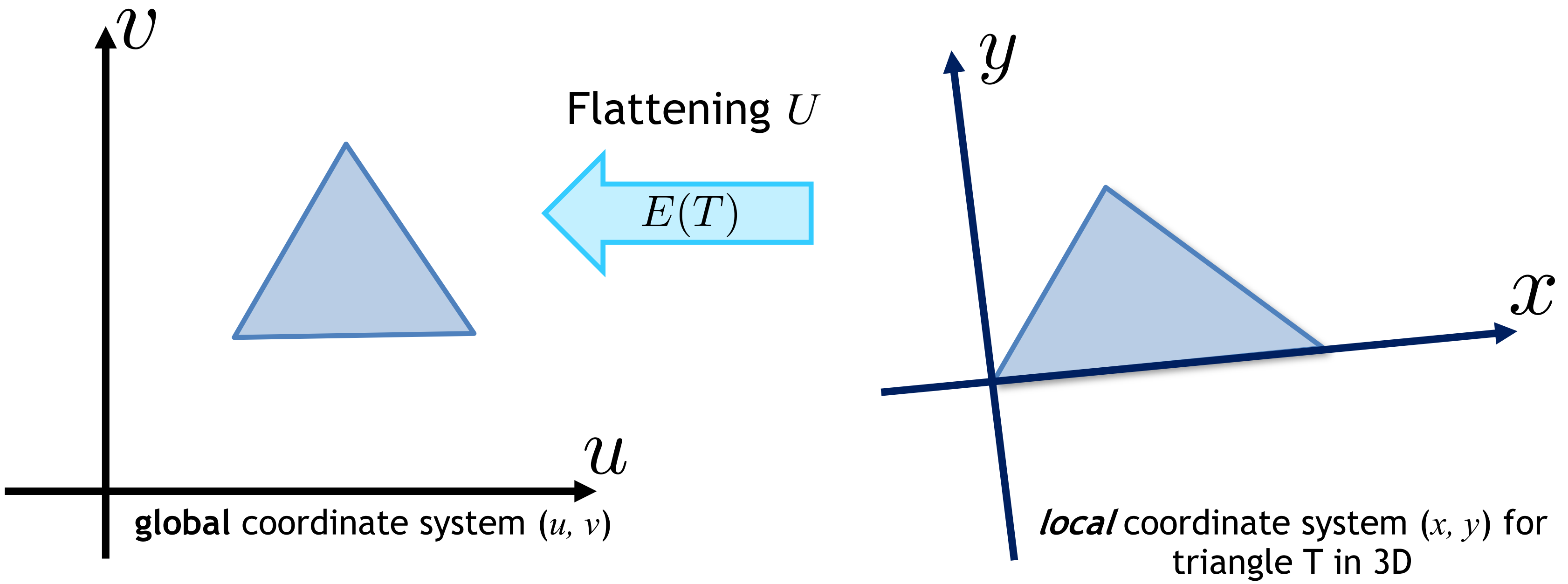
per-triangle distortion measure, value dependent on X



Local triangle basis

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T \, E(T)$$

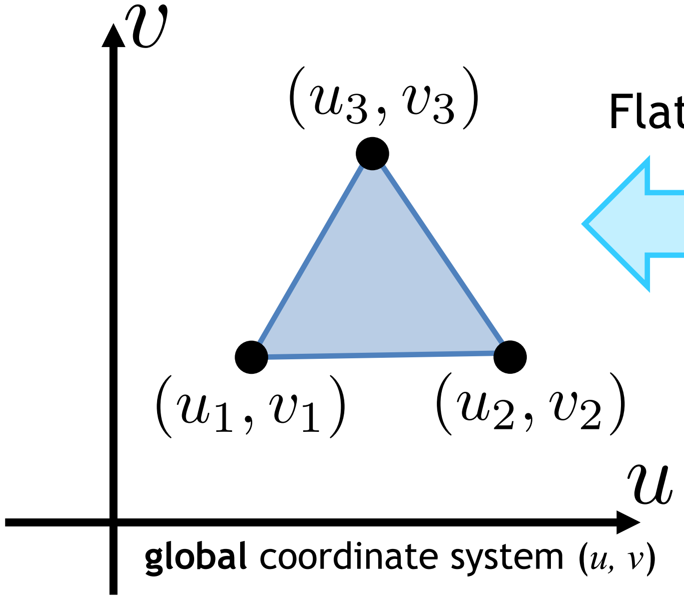
$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$



Local triangle basis

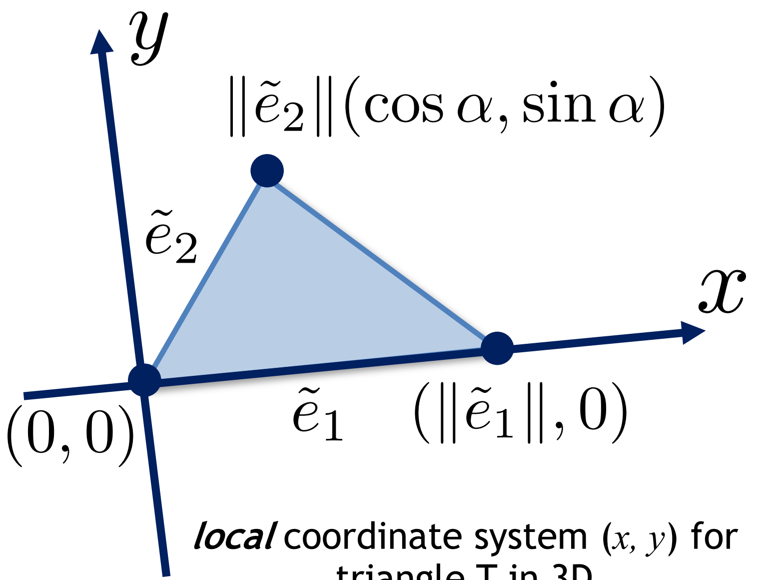
$$\underset{X \in \mathbb{R}^{2n}}{\text{argmin}} \sum_{\text{triangles } T} A_T \, E(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$



Flattening U

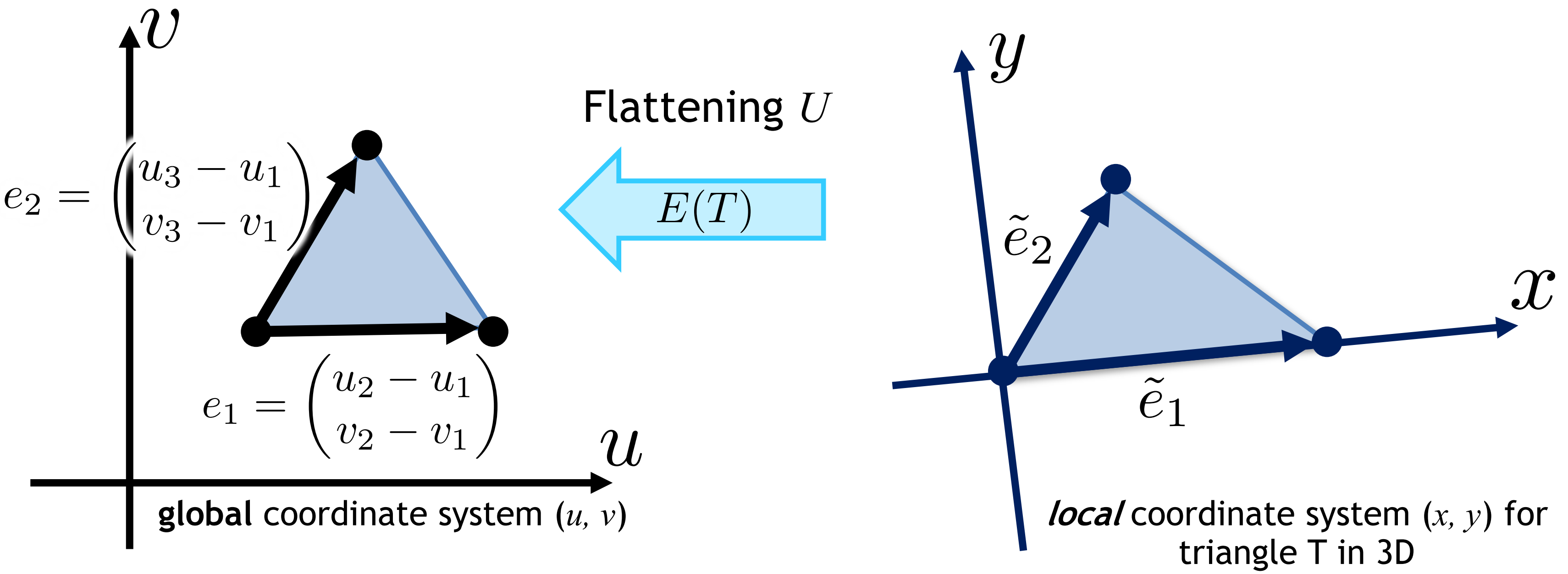
$E(T)$



Local triangle basis

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T E(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

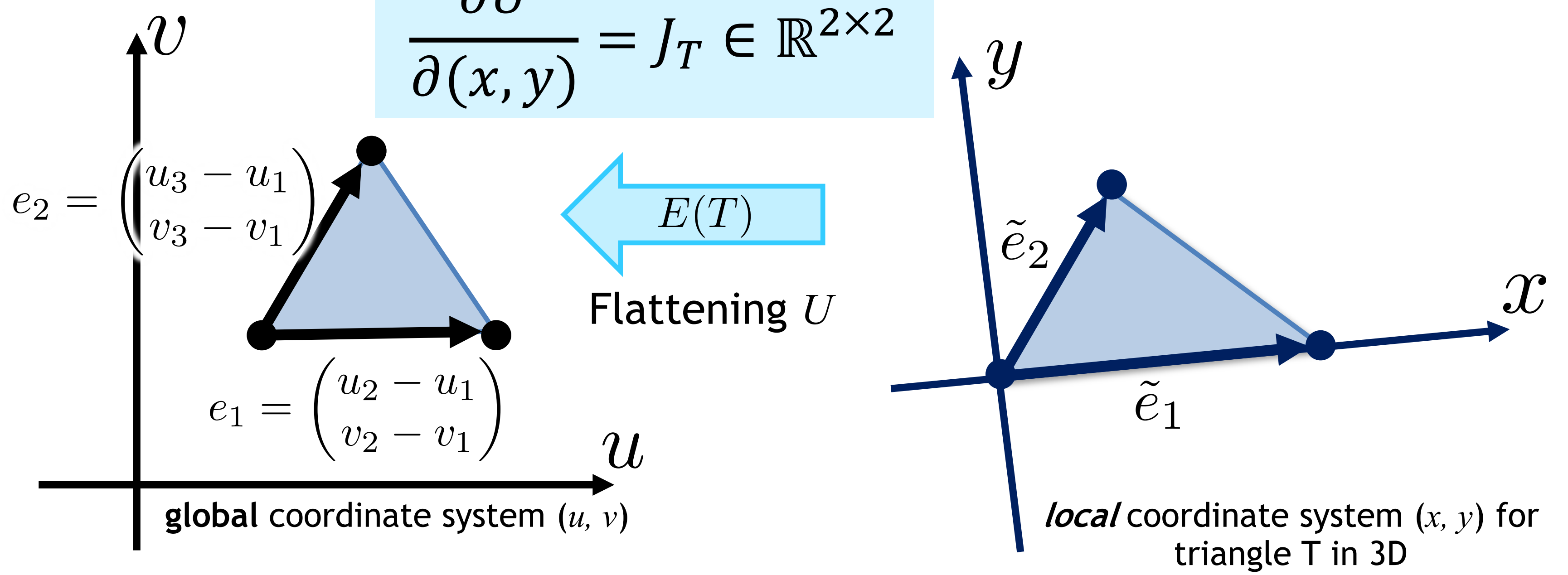


Mapping between triangles

$$\underset{X\in\mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T\, E(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$\frac{\partial U}{\partial(x,y)} = J_T \in \mathbb{R}^{2\times 2}$$



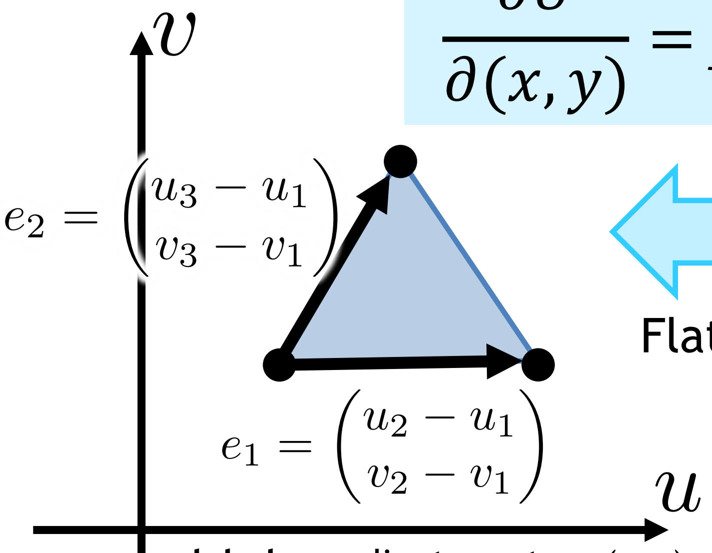
Mapping between triangles

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T \, E(T)$$

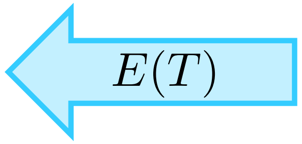
$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^{\mathsf{T}}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix}$$

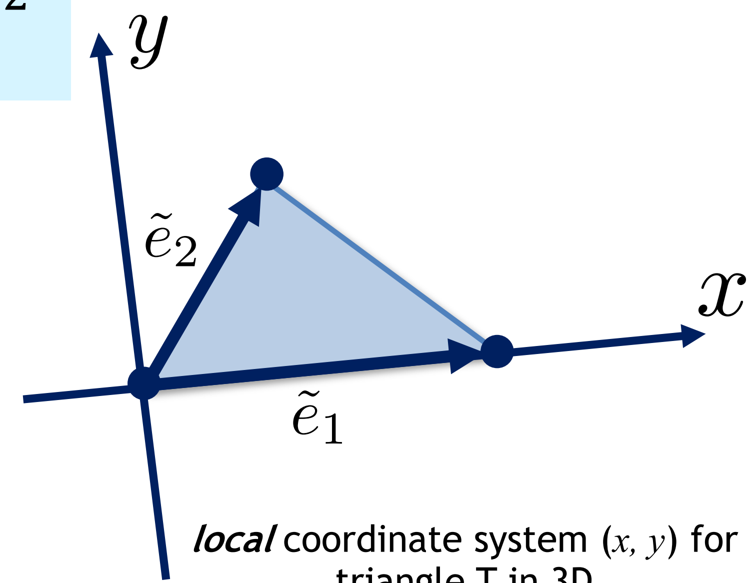
$$\frac{\partial U}{\partial(x, y)} = J_T \in \mathbb{R}^{2\times 2}$$



global coordinate system (u, v)



Flattening U



local coordinate system (x, y) for triangle T in 3D

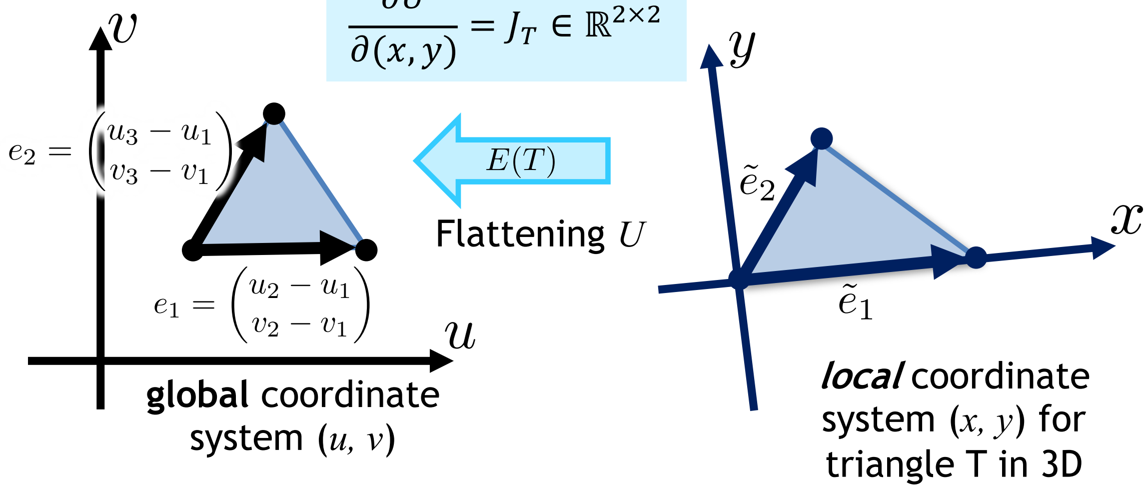
Mapping between triangles

$$\underset{X \in \mathbb{R}^{2n}}{\text{argmin}} \sum_{\text{triangles } T} A_T \, E(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \Rightarrow J_T = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

$$\frac{\partial U}{\partial (x,y)} = J_T \in \mathbb{R}^{2\times 2}$$



Mapping between triangles

$$\underset{X \in \mathbb{R}^{2n}}{\text{argmin}} \sum_{\text{triangles } T} A_T \, E(T)$$

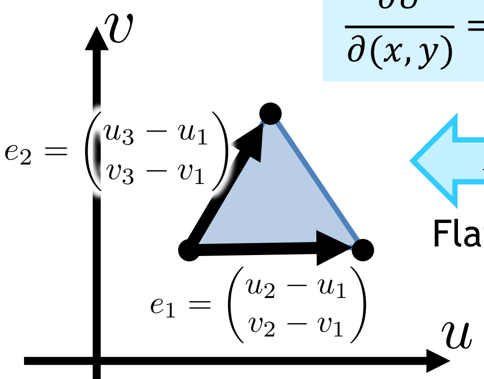
$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^{\mathsf{T}}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \;\Rightarrow\; J_T = \underbrace{\begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix}}_{\begin{pmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{pmatrix}} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

Linear in our unknowns X!

Constant 2x2 matrix, depends 3D triangle only

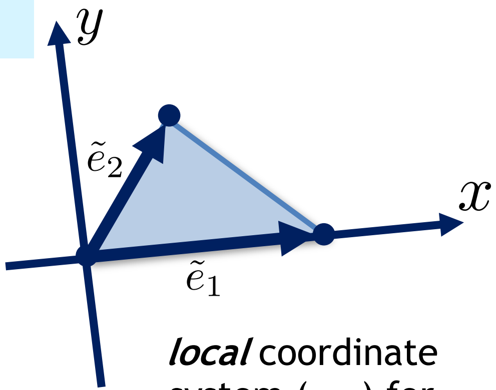
$$\frac{\partial U}{\partial(x, y)} = J_T \in \mathbb{R}^{2 \times 2}$$



global coordinate system (u, v)

$E(T)$

Flattening U



local coordinate system (x, y) for triangle T in 3D

Mapping between triangles

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T\, E(T)$$

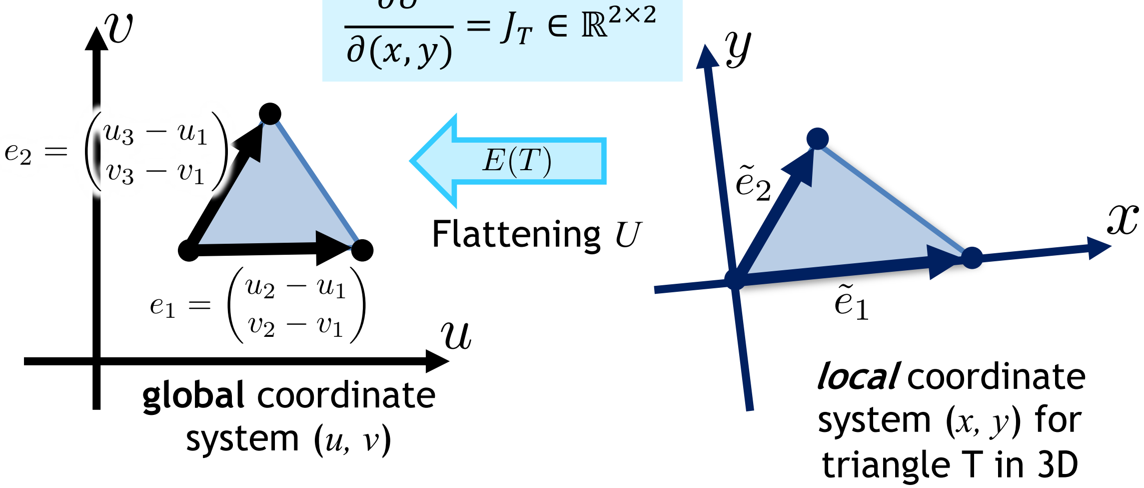
$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \Rightarrow J_T = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

Linear in our unknowns X! $\begin{pmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{pmatrix}$

Constant 2x2 matrix, depends 3D triangle only

$$\frac{\partial U}{\partial(x,y)} = J_T \in \mathbb{R}^{2\times 2}$$

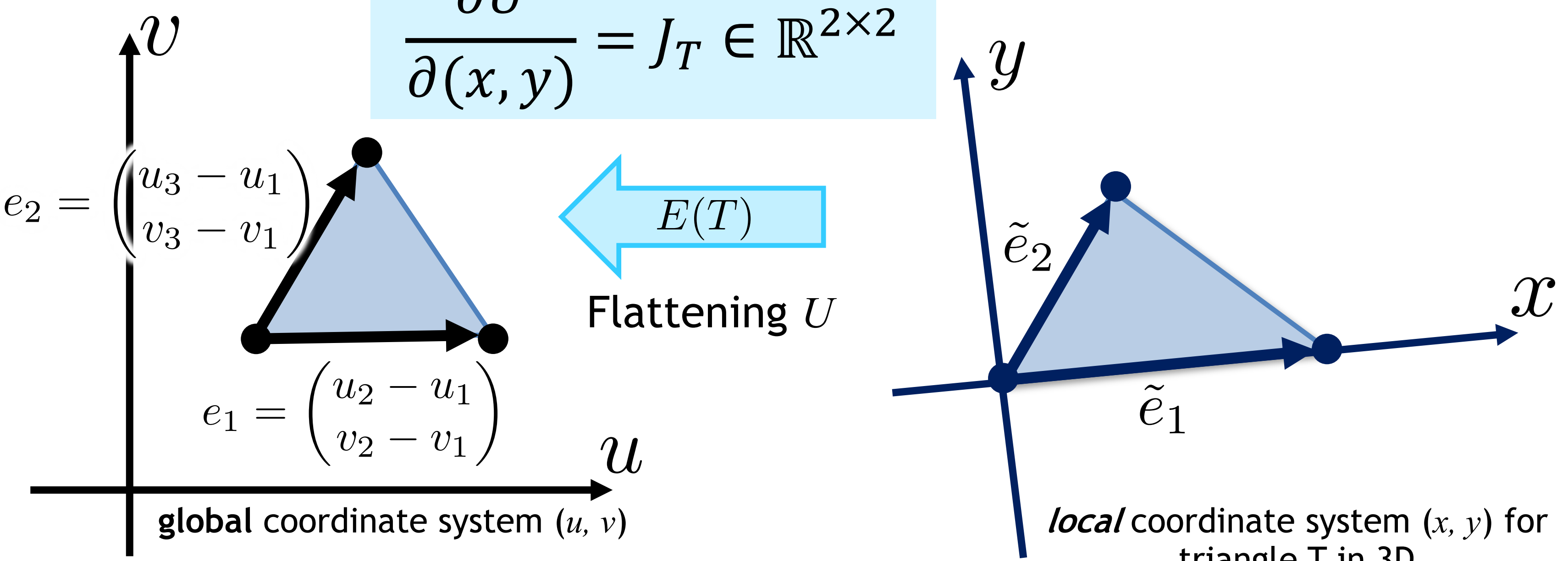


J_T **is linear** in unknowns X !

Jacobian J_T and triangle deformation

- Jacobian J_T is translation-invariant

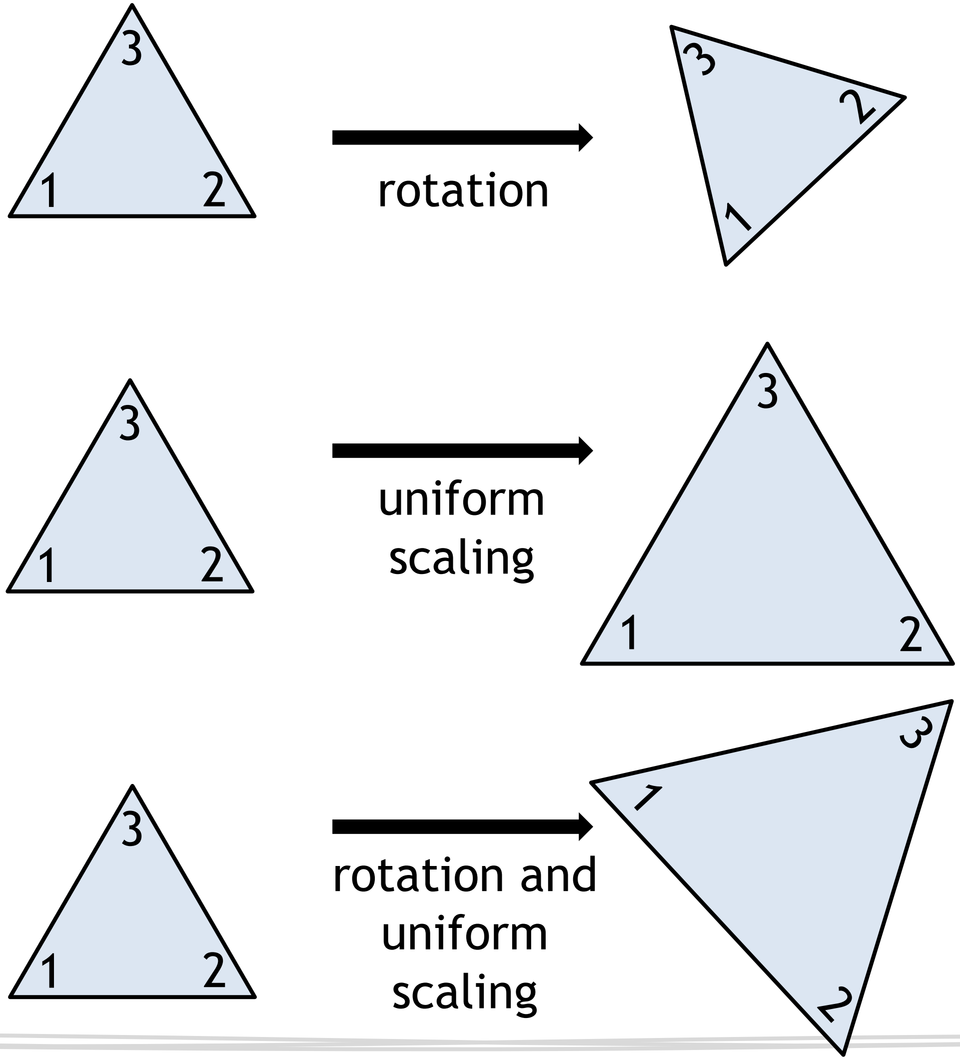
$$\frac{\partial U}{\partial(x,y)} = J_T \in \mathbb{R}^{2\times 2}$$



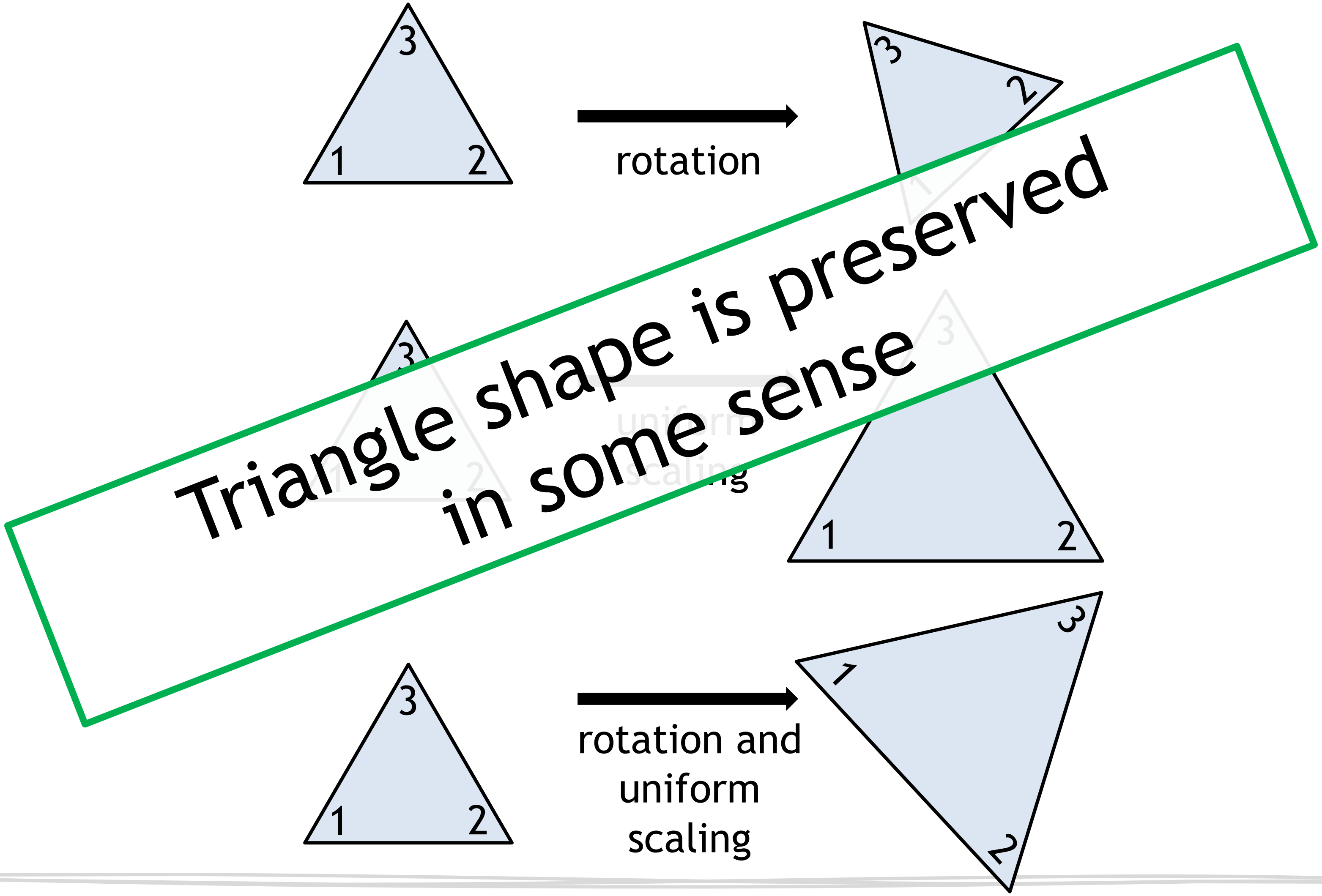
global coordinate system (u, v)

local coordinate system (x, y) for triangle T in 3D

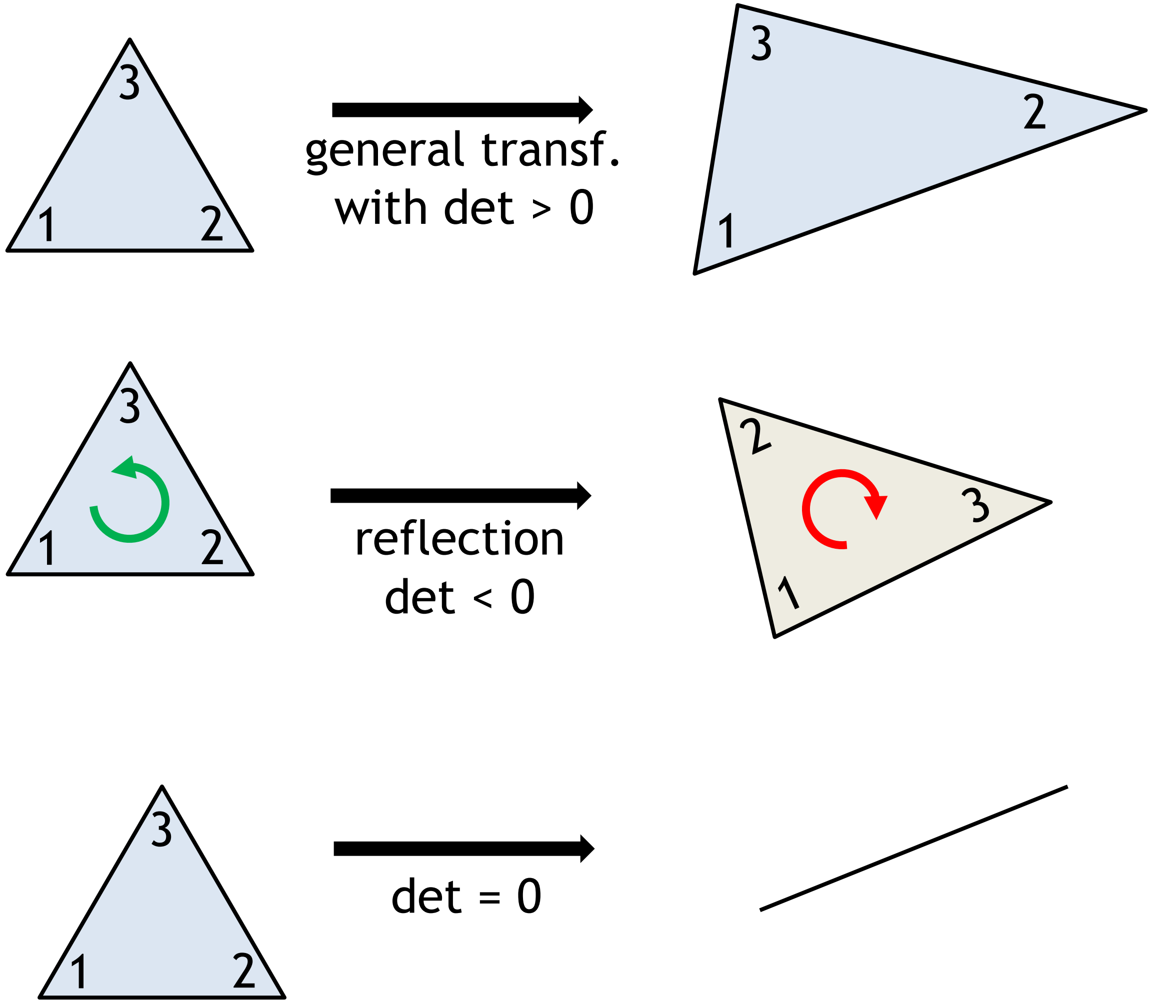
Jacobian J_T and triangle deformation



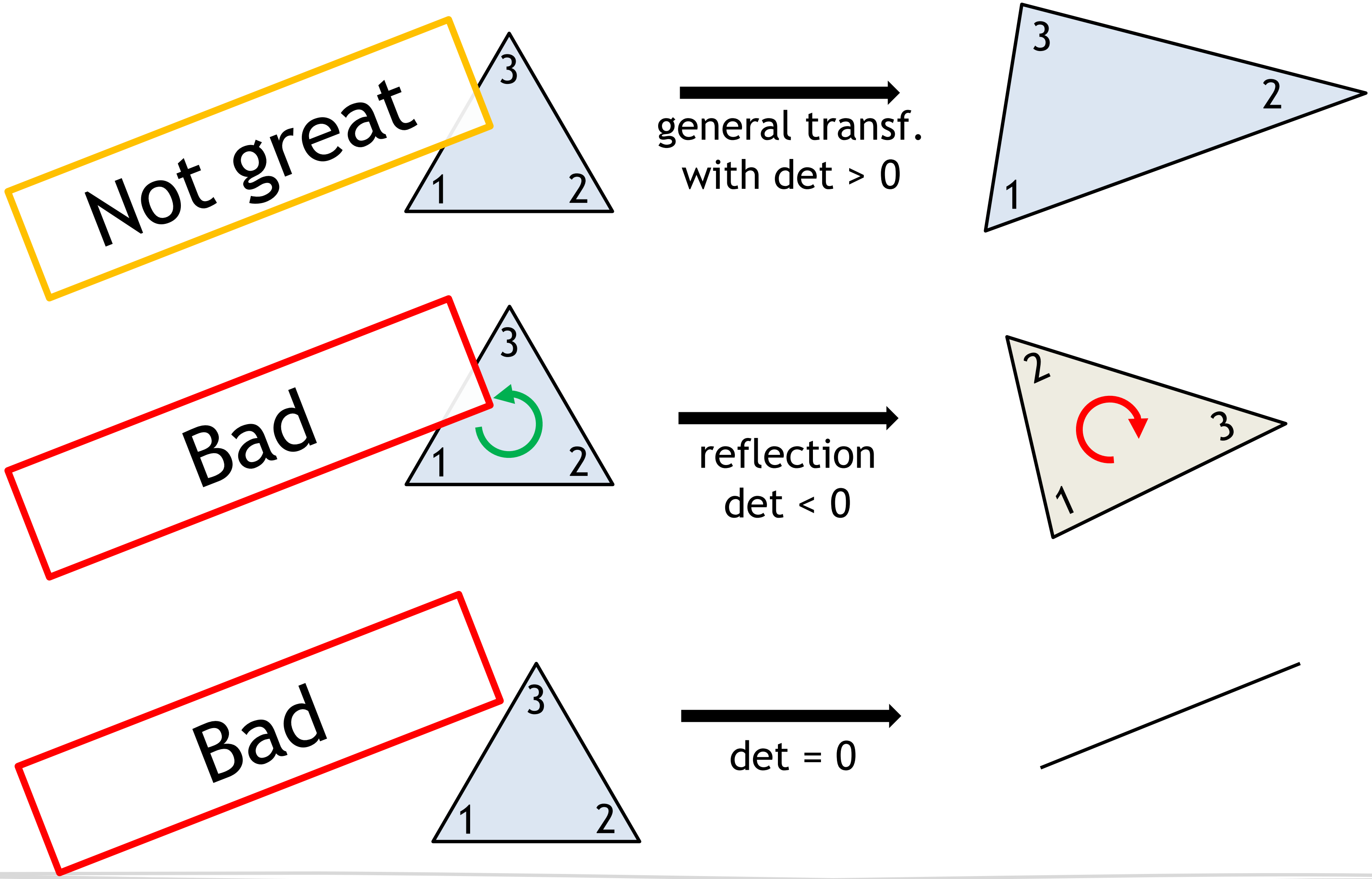
Jacobian J_T and triangle deformation



Jacobian J_T and triangle deformation



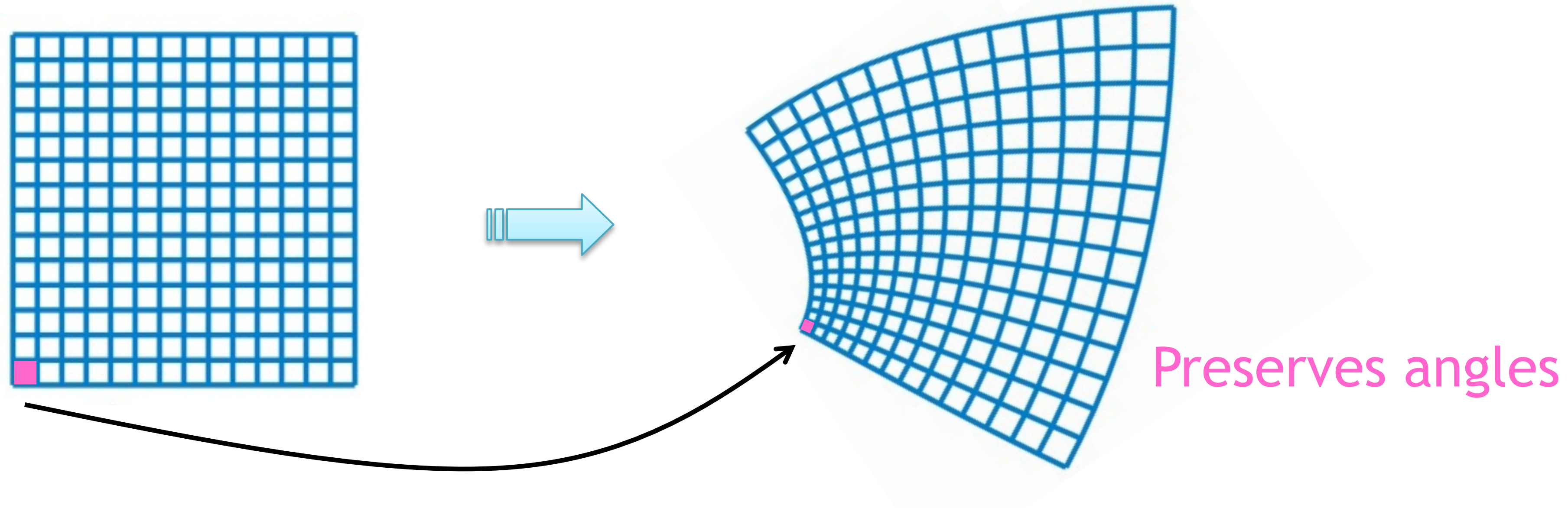
Jacobian J_T and triangle deformation



LSCM: Least Squares Conformal Maps

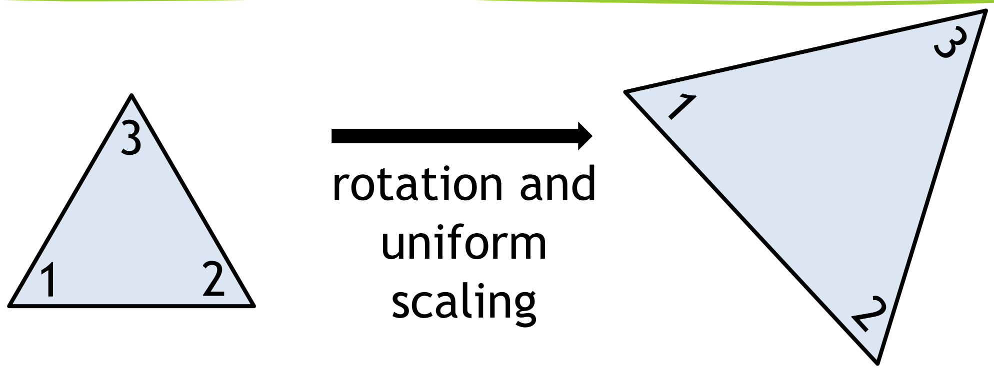
Conformal maps

- Smooth maps that preserve angles
- Straight lines are mapped to curves



Similarity = Rotation + Scale

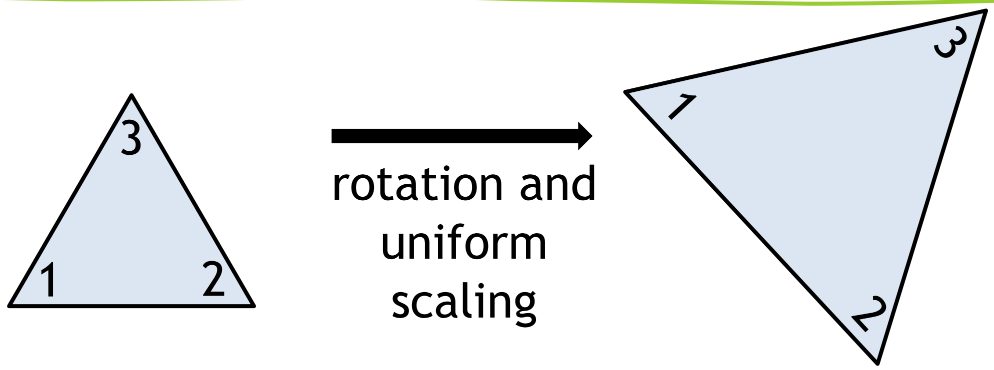
LSCM aims to preserve *triangle* angles



$$J_T = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$$

$$s \in \mathbb{R}^+$$

LSCM aims to preserve angles



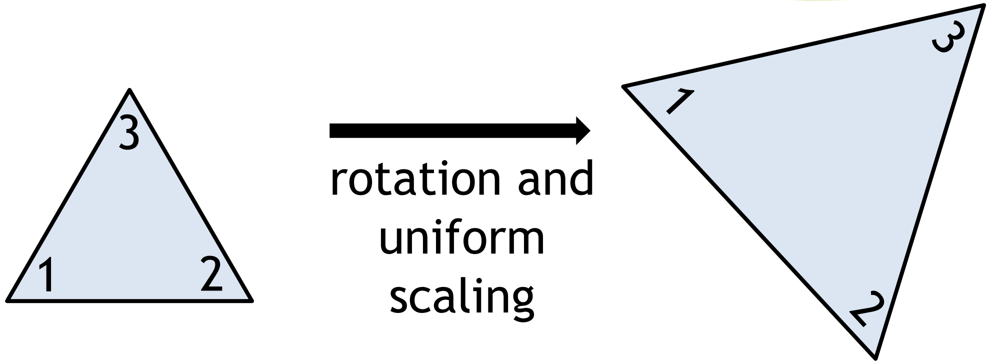
$$J_T = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} = \begin{pmatrix} s\cos\phi & s\sin\phi \\ -s\sin\phi & s\cos\phi \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

$s \in \mathbb{R}^+$

$a, b \in \mathbb{R}$

$$a^2 + b^2 = s^2(\cos^2\phi + \sin^2\phi) = s^2$$

LSCM aims to preserve angles



$$J_T = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} = \begin{pmatrix} s\cos\phi & s\sin\phi \\ -s\sin\phi & s\cos\phi \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

$s \in \mathbb{R}^+$

$a, b \in \mathbb{R}$

$$a^2 + b^2 = s^2(\cos^2\phi + \sin^2\phi) = s^2$$

Simple, linear "template" for J_T: $J_T = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, $a, b \in \mathbb{R}$, avoid $(a, b) = (0, 0)$

Recall: J_T is linear in unknown uv values

$$\operatorname*{argmin}_{X \in \mathbb{R}^{2n}} \sum_{\text{triangles } T} A_T\, E(T)$$

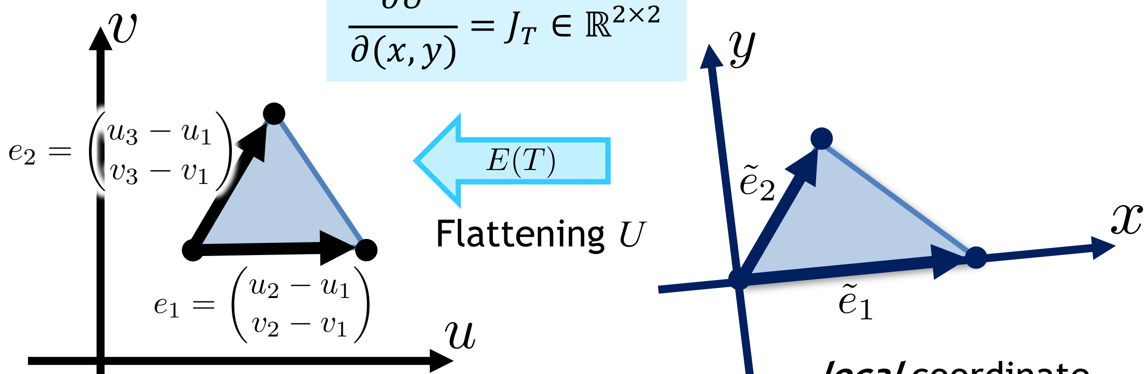
$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \Rightarrow J_T = \underbrace{\begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix}}_{\begin{pmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{pmatrix}} \underbrace{\begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}}$$

Linear in our unknowns X!

Constant 2x2 matrix, depends 3D triangle only

$$\frac{\partial U}{\partial(x,y)} = J_T \in \mathbb{R}^{2\times 2}$$



global coordinate system (u, v)

local coordinate system (x, y) for triangle T in 3D

J_T **is linear** in unknowns X **!**

Recall: J_T is linear in unknown uv values

$$\operatorname*{argmin}_{X \in \mathbb{R}^{2n}} \sum_{\text{triangles } T} A_T \, E(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \;\Rightarrow\; J_T = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

J_T is linear in unknowns X !

$$J_T = \begin{pmatrix} j_{11}(X) & j_{12}(X) \\ j_{21}(X) & j_{22}(X) \end{pmatrix}, \quad j_{11}(X), j_{12}(X), j_{21}(X), j_{22}(X) \text{ are linear in } X$$

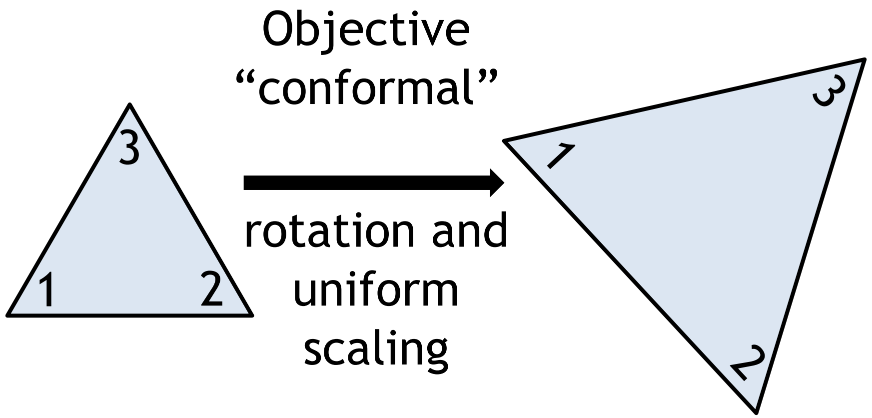
LSCM energy formulation

$$\underset{X\in\mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T\, E(T)$$

$$X = (u_1, v_1, u_2, v_2, \dots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \Rightarrow J_T = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

$J_T = \begin{pmatrix} j_{11}(X) & j_{12}(X) \\ j_{21}(X) & j_{22}(X) \end{pmatrix}$, $j_{11}(X), j_{12}(X), j_{21}(X), j_{22}(X)$ are linear in X



linear "template" for J_T: $J_T = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, $a, b \in \mathbb{R}$, avoid $(a,b) = (0,0)$

LSCM energy formulation

$$\underset{X \in \mathbb{R}^{2n}}{\operatorname{argmin}} \sum_{\text{triangles } T} A_T \, E_{\text{LSCM}}(T)$$

$$X = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)^\mathsf{T}$$

$$J_T \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix} = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \;\Rightarrow\; J_T = \begin{pmatrix} | & | \\ e_1 & e_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \tilde{e}_1 & \tilde{e}_2 \\ | & | \end{pmatrix}^{-1}$$

$J_T = \begin{pmatrix} j_{11}(X) & j_{12}(X) \\ j_{21}(X) & j_{22}(X) \end{pmatrix}$, $j_{11}(X), j_{12}(X), j_{21}(X), j_{22}(X)$ are linear in X

linear "template" for J_T: $J_T = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$, $a, b \in \mathbb{R}$, avoid $(a,b) = (0,0)$

$$E_{\text{LSCM}}(T) = (j_{11}(X) - j_{22}(X))^2 + (j_{12}(X) - (-j_{21}(X)))^2$$

$E_{\text{LSCM}}(T)$ is quadratic in X!

Minimizer is found by solving a sparse linear system in X.
Must avoid trivial solutions!

Null space of ∇E_{LSCM}

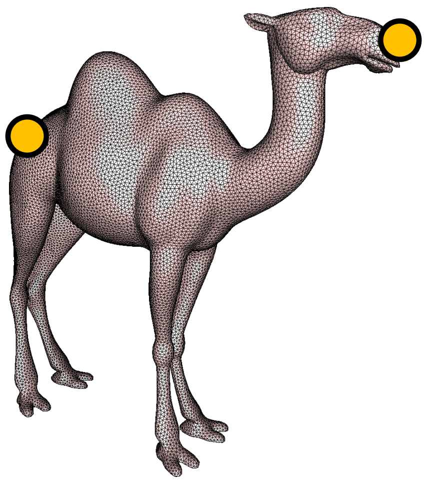
- As presented, E_{LSCM} attains its (global and only) minimum of value 0 if:
  - for a general input 3D mesh:
    all mesh vertices are assigned to a single (u,v) point
    $\rightarrow$ the null space is 2-dimensional
  - for a discrete-developable input 3D mesh (e.g. input put is already flat, or can be flattened without any distortion to any triangle). Then taking (u,v) coordinates that are a perfect unfolding of the mesh to the plane, or rotations, translations and *uniform scalings* thereof leads to zero energy.
    This is a rare and unlikely case.

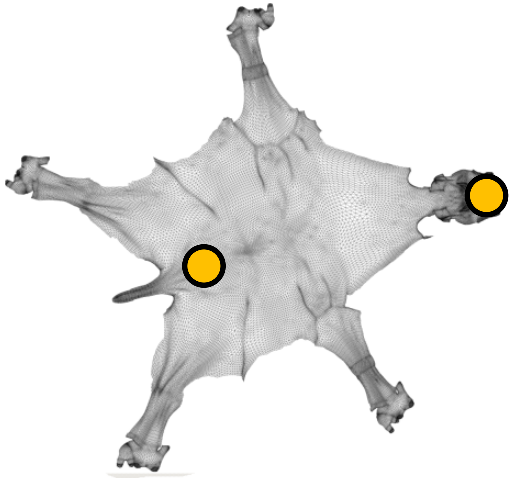
Null space of ∇E_{LSCM}

- As presented, E_{LSCM} attains its (global and only) minimum of value *0* if:
  - for a general input 3D mesh:
    all mesh vertices are assigned to a single (u,v) point
    $\rightarrow$ the null space is 2-dimensional
- This is a collapsed solution that must be avoided by fixing more than just 1 vertex (2 unknowns)!
- To avoid collapse to a single point, we need to fix the locations of 2 vertices to determine the degrees of freedom.

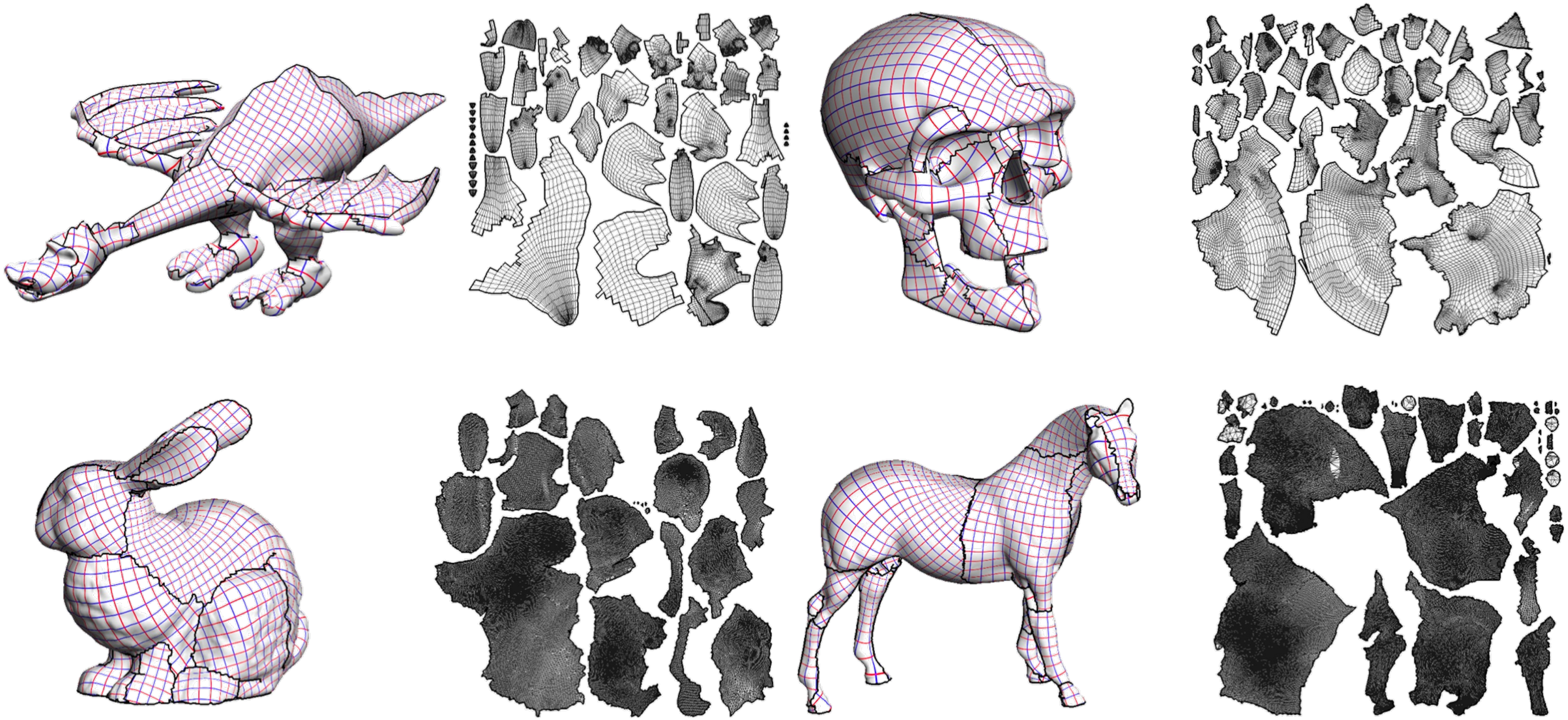
LSCM degrees of freedom

- [Levy et al. 2002] : fix 2 vertices in 2D that have the maximal shortest edge path between them.
- Leads to a sparse linear system in (2n-4) variables.
- System then has a unique and non-collapsed solution.
- The solution depends on the choice of vertices!

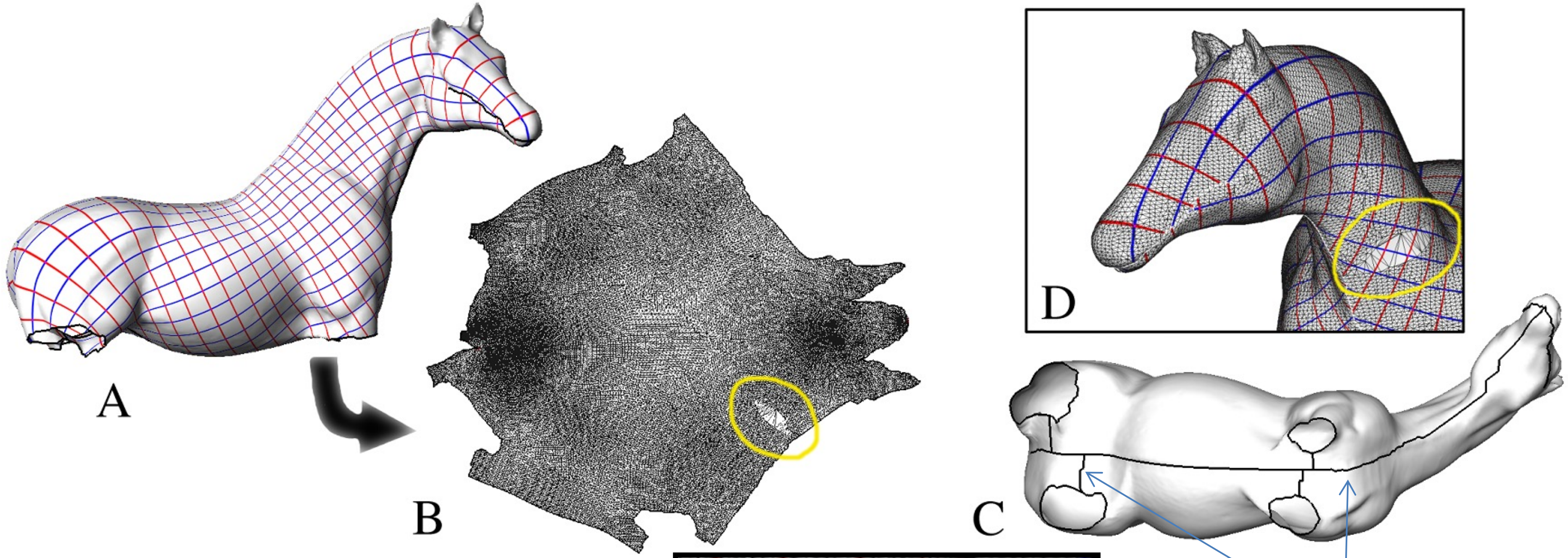




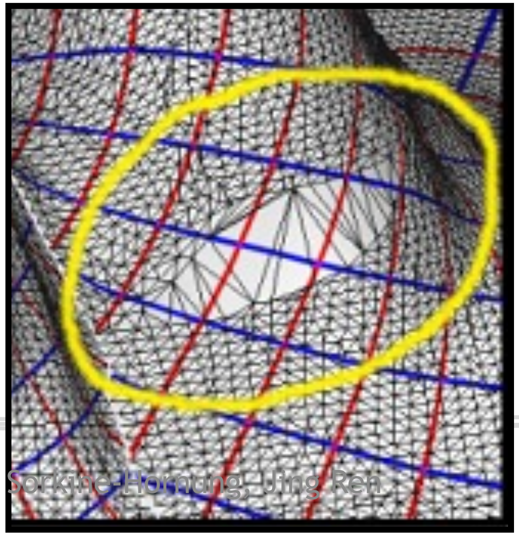
LSCM – Results



LSCM – Results



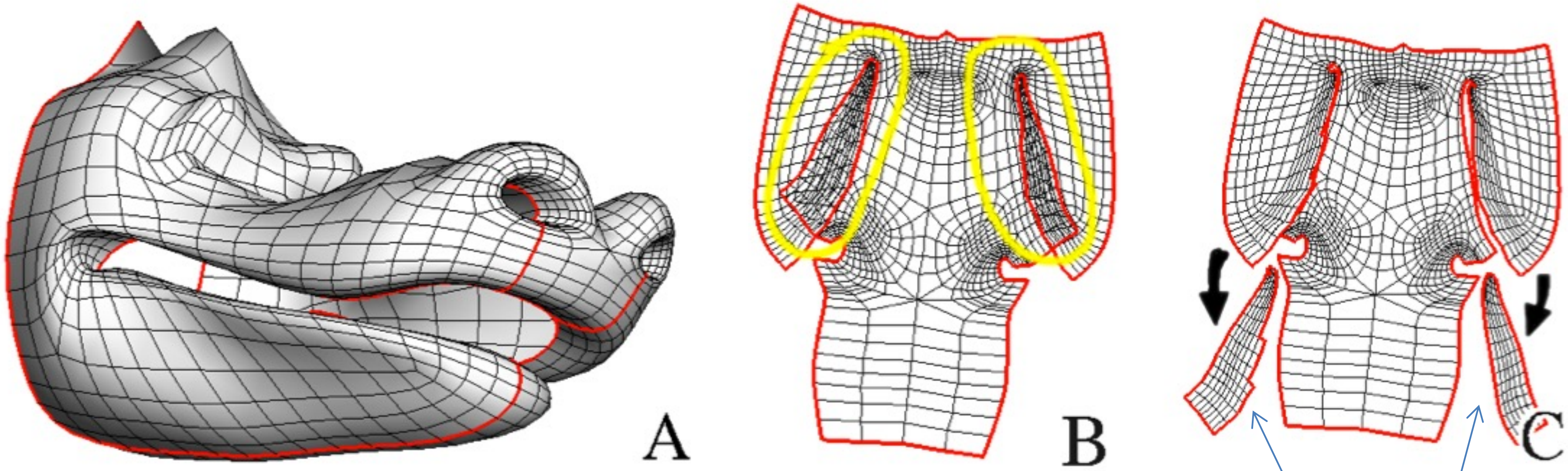
Manually placed cuts



Flipped triangles rarely occur

🡐 Robust against varying mesh density

Overlaps can still occur



Cut along overlapping boundaries

Triangle inversions and collapses can still occur (but rather rare)

- We get the J_T matrix template *only in least-squares sense*

$$J_T = \begin{pmatrix} j_{11}(X) & j_{12}(X) \\ j_{21}(X) & j_{22}(X) \end{pmatrix},\ j_{11}(X), j_{12}(X), j_{21}(X), j_{22}(X) \text{ are linear in } X$$

$$J_T = \begin{pmatrix} a & b \\ -b & a \end{pmatrix},\quad a, b \in \mathbb{R},\ \text{avoid } (a, b) = (0, 0)$$

$$E_{\mathrm{LSCM}}(T) = (j_{11}(X) - j_{22}(X))^2 + \big(j_{12}(X) - (-j_{21}(X))\big)^2$$

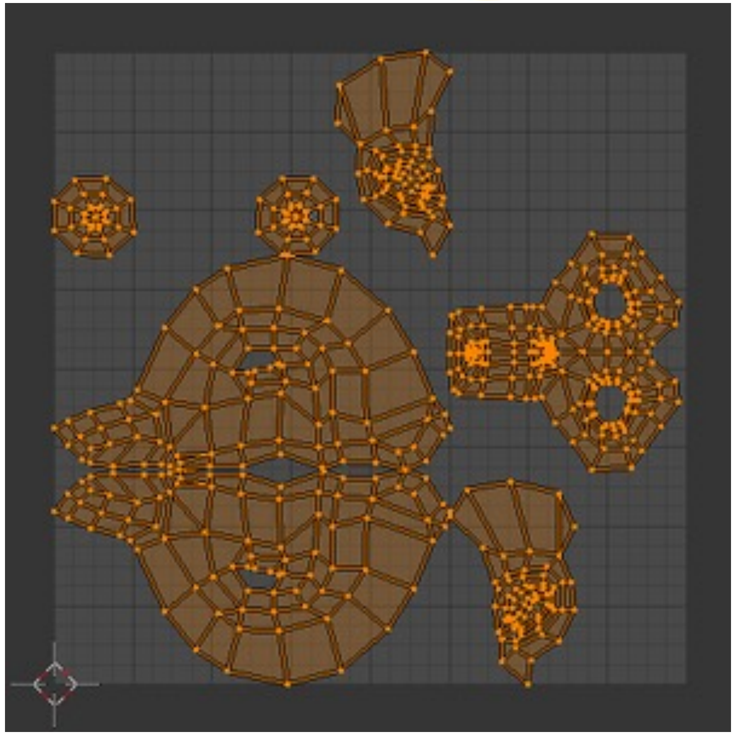
Texture mapped results

Provided in many packages

- Often called “UV-unwrapping”

- Blender
- Autodesk Maya, 3ds Max

- CGAL

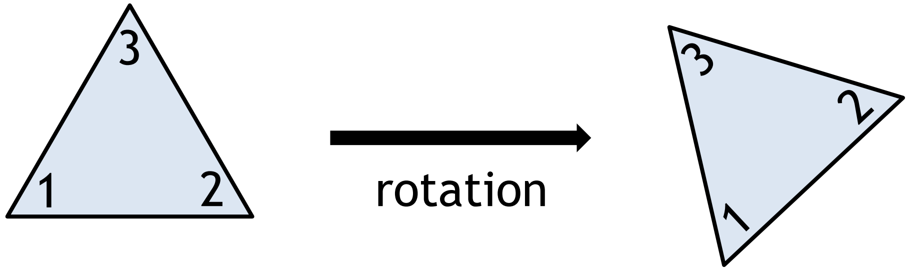


https://docs.blender.org/manual/en/latest/modeling/meshes/editing/uv.html

ARAP:
As-Isometric-As-Possible Parameterization

(Rigid)

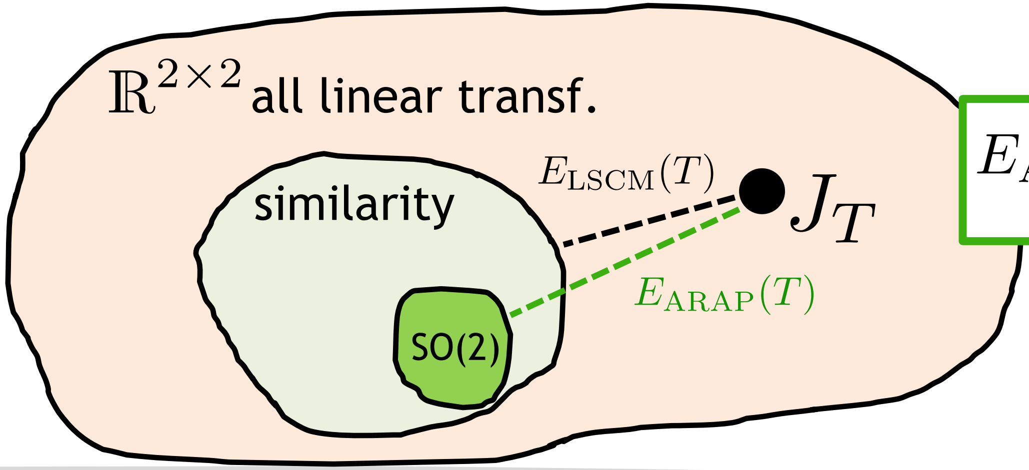
ARAP energy



$J_T \in \mathbb{R}^{2\times 2}$, linear function of X

$E_{\text{ARAP}}(T) =$ distance between J_T and a 2×2 rotation

= matrix in SO(2)



$$E_{\text{ARAP}}(T) = \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

Frobenius matrix norm: $\|(a_{ij})\|_F = \left(\sum_{i,j} a_{ij}^2\right)^{1/2}$

ARAP (local-global) algorithm

$$E_{\text{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- How to minimize efficiently? Local-global iterations!

1. Initialize X with LSCM or Tutte or smth
2. For each T, compute R_T using the current uv's, i.e. the current value of X
3. Fix all R_T's and solve for new X
4. If ~(stopping criterion) goto 2

ARAP (local-global) algorithm

$$E_{\text{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- How to minimize efficiently? Local-global iterations!

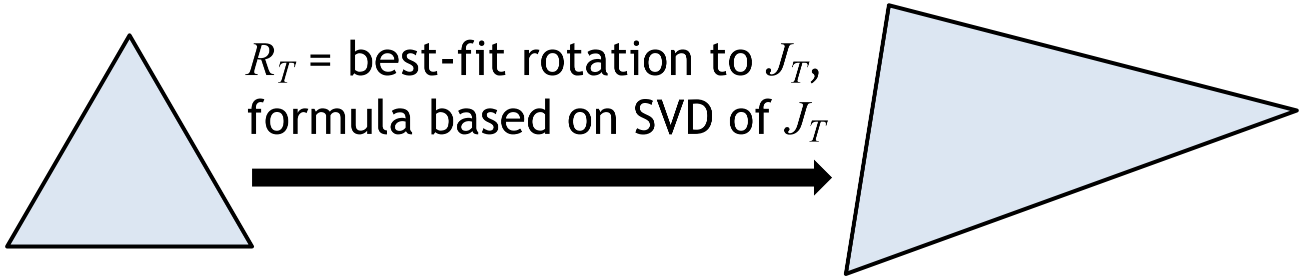
1. Initialize X with LSCM or Tutte or…
2. For each T, compute R_T using the current uv's, i.e. the current value of X
3. Fix all R_T's and solve for new X
4. If ~(stopping criterion) goto 2

ARAP (local-global) algorithm

$$E_{\mathrm{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- How to minimize efficiently? Local-global iterations!

1. Initialize X with LSCM or Tutte or…
2. For each T, compute R_T using the current uv's, i.e. the current value of X



Procrustes Problem

$$R_T = \mathrm{argmin}_{R \in SO(2)} \|R - J_T\|_F$$
$$= \mathrm{argmin}_{R \in SO(2)} \|R - U\Sigma V^T\|_F$$
$$R_T = UV^T$$

ARAP (local-global) algorithm

$$E_{\text{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- How to minimize efficiently? Local-global iterations!

1. Initialize X with LSCM or Tutte or…
2. For each T, compute R_T using the current uv's, i.e. the current value of X
3. Fix all R_T's and solve for new X
4. If ~(stopping criterion) goto 2

ARAP (local-global) algorithm

$$E_{\text{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- $A(T)$: fixed scalar
- J_T: linear in X
- R_T: fixed

➡ Quadratic minimization in X (need to fix at least 1 vertex)

3. Fix all R_T's and solve for new X
4. If ~(stopping criterion) goto 2

ARAP (local-global) algorithm

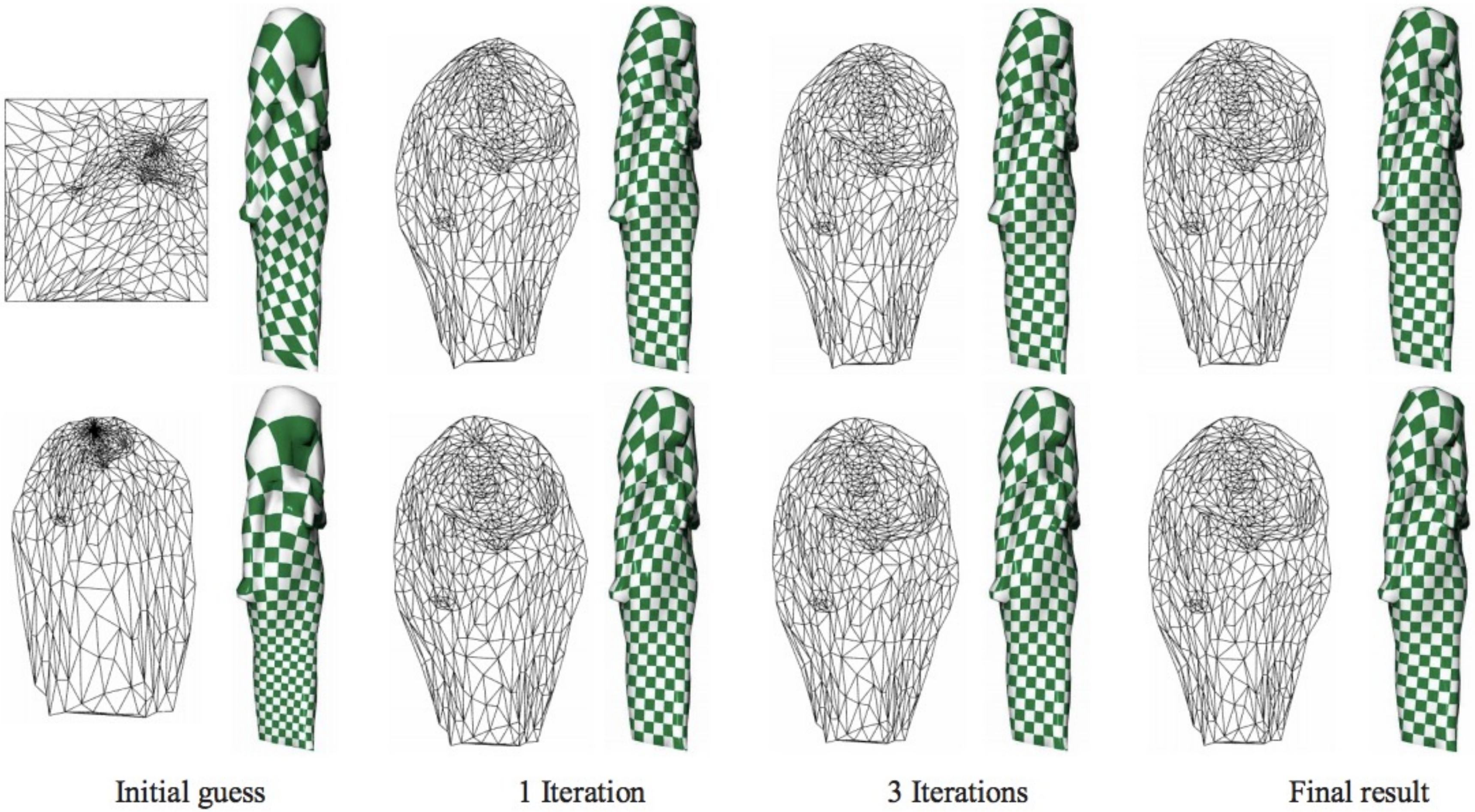
$$E_{\text{ARAP}}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \|J_T - R_T\|_F^2$$

- How to minimize efficiently? Local-global iterations!

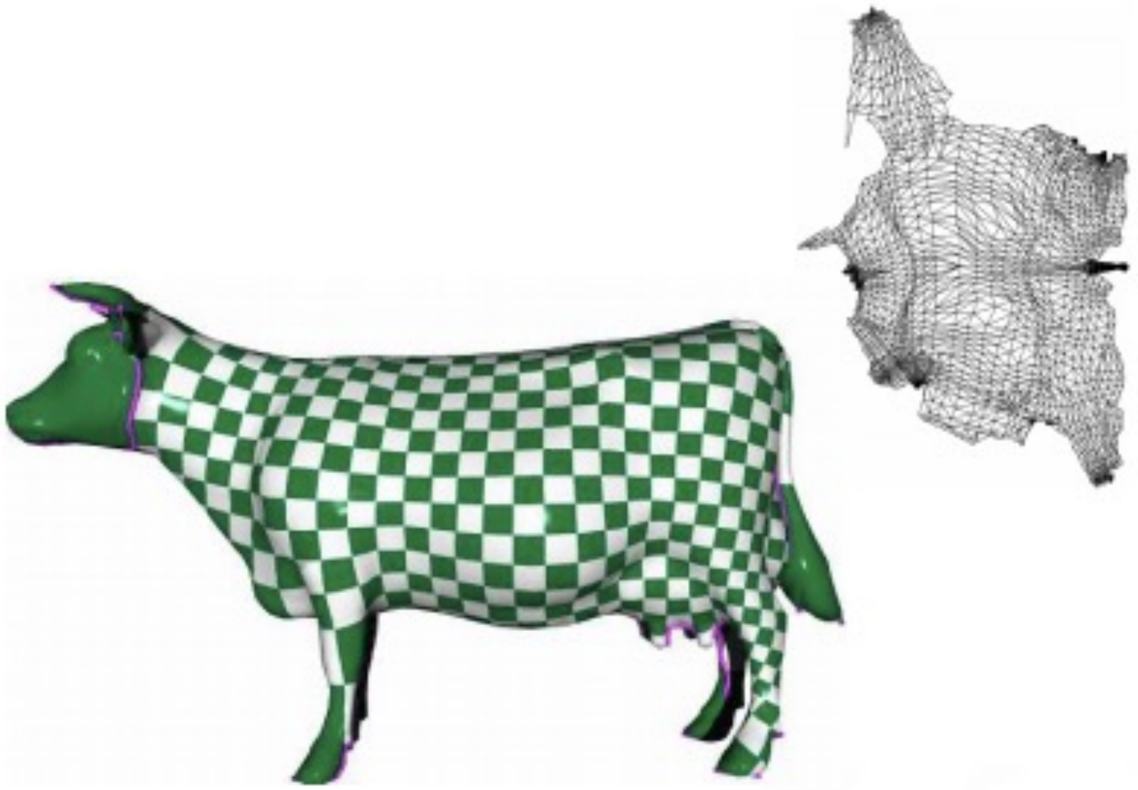
1. Initialize X with LSCM or Tutte or…
2. For each T, compute R_T using the current uv's, i.e. the current value of X
3. Fix all R_T's and solve for new X
4. If ~(stopping criterion) goto 2

More details about rotation fitting and ARAP come later ☺

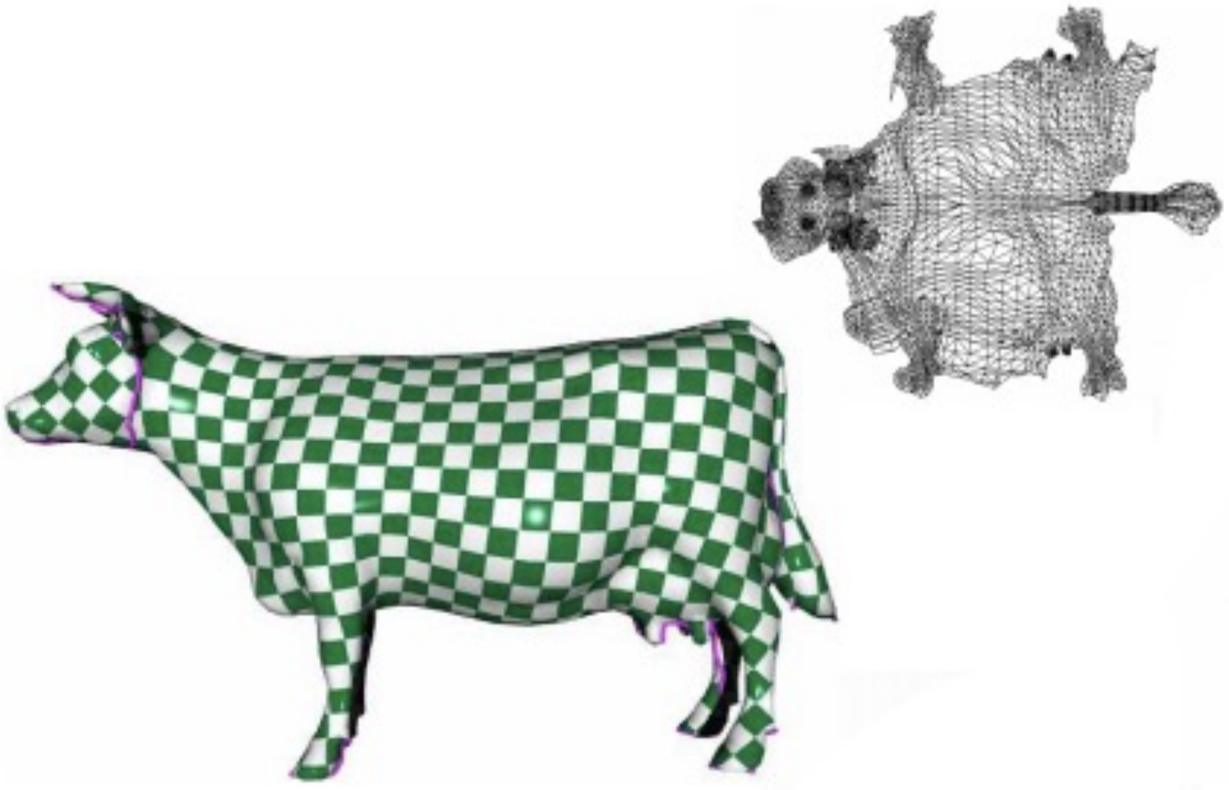
ARAP Param. – Results



ARAP Param. – Results



LSCM



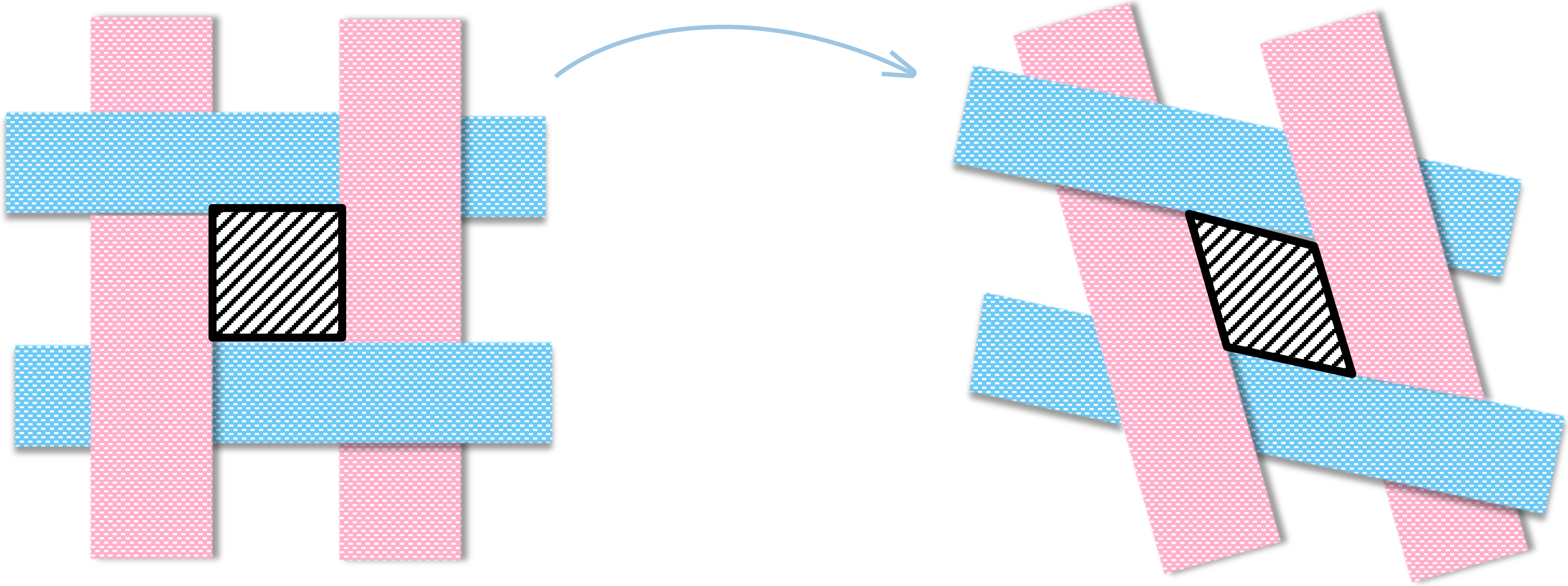
ARAP

Chebyshev Parameterzation:
preserving lengths along the yarn directions

igl

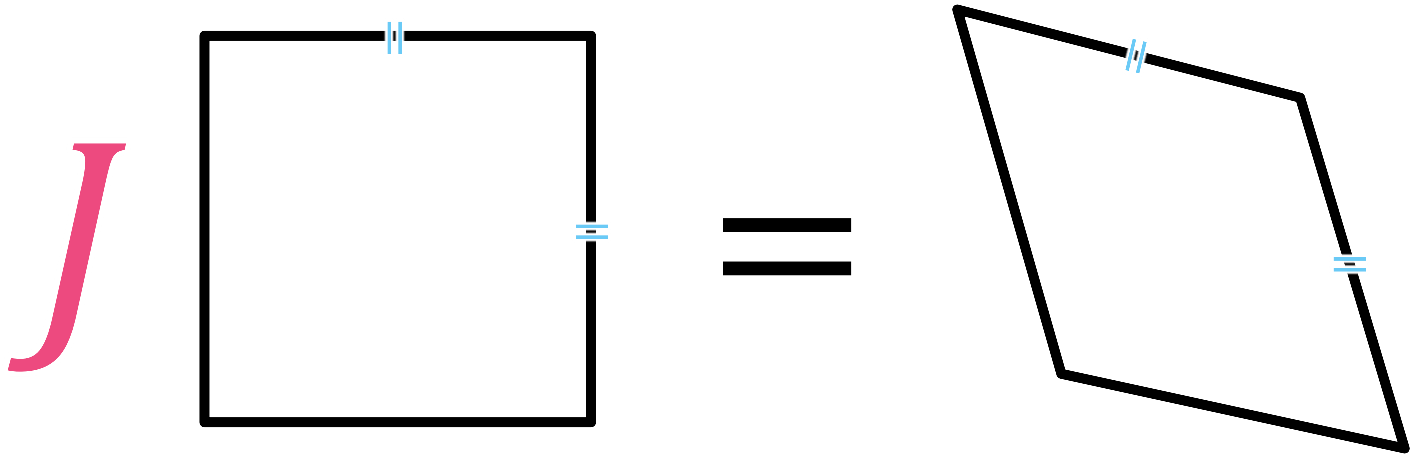
ETH zürich

Chebyshev net

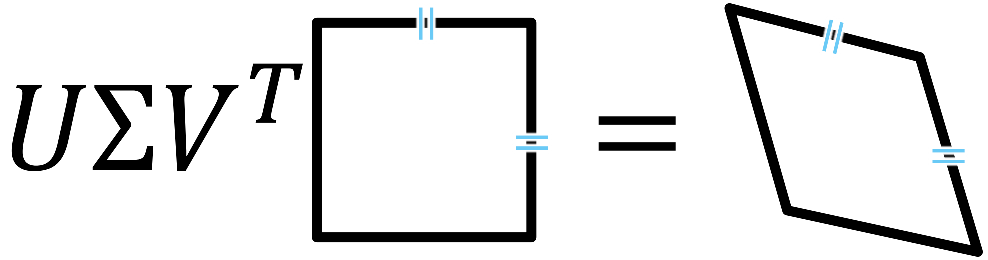


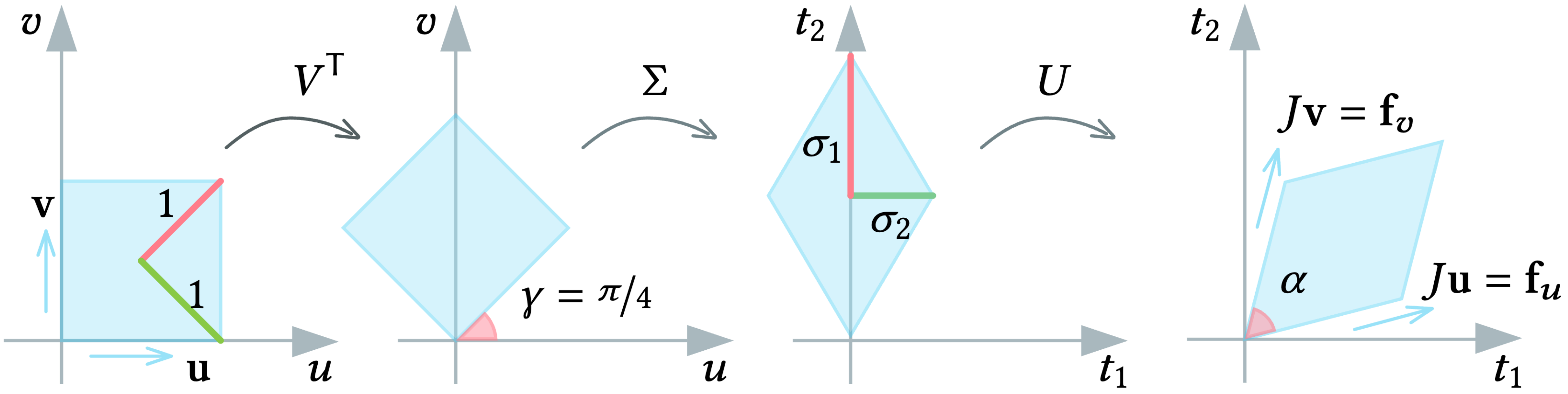
Lengths along the yarn directions are preserved → Squares are mapped to diamonds

SVD of the Jacobian of a Chebyshev net



SVD of the Jacobian of a Chebyshev net

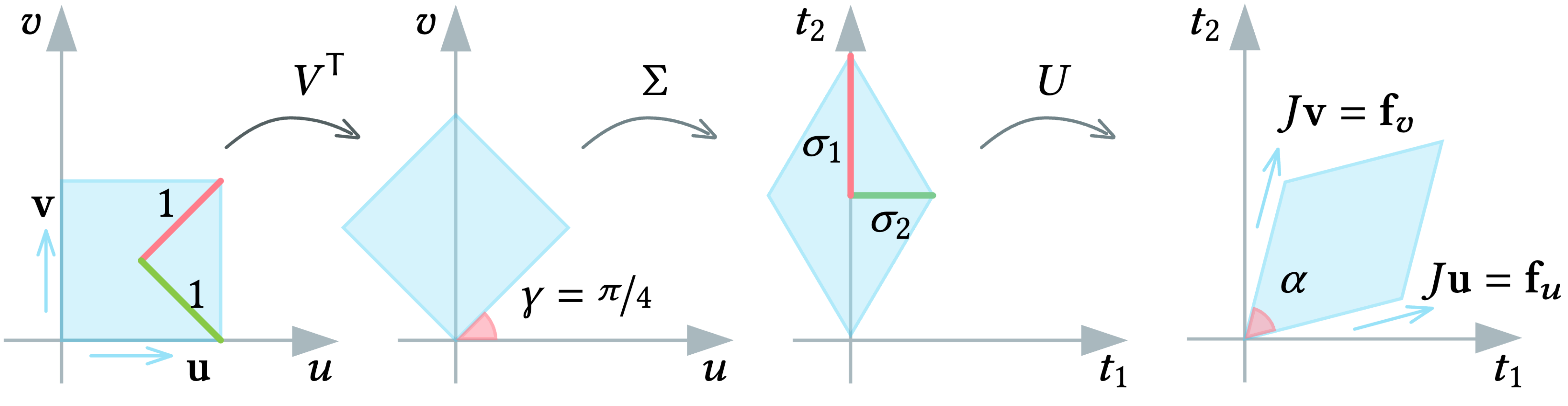




SVD of the Jacobian of a Chebyshev net

$J = U\Sigma V^T$ has a template:

- U is unknown rotation
- Singular values satisfy $\sigma_1^2 + \sigma_2^2 = 2$
- V is a known rotation (by angle $\pi/4$



Chebyshev Parameterization

$$E_{ARAP}(X) = \sum_{\text{triangles } T} A(T) \min_{R_T \in SO(2)} \| J_T - R_T \|_F^2$$

$J = U\Sigma V^T$ has a template:

- U is unknown rotation
- Singular values satisfy $\sigma_1^2 + \sigma_2^2 = 2$
- V is a known rotation (by angle $\pi/4$

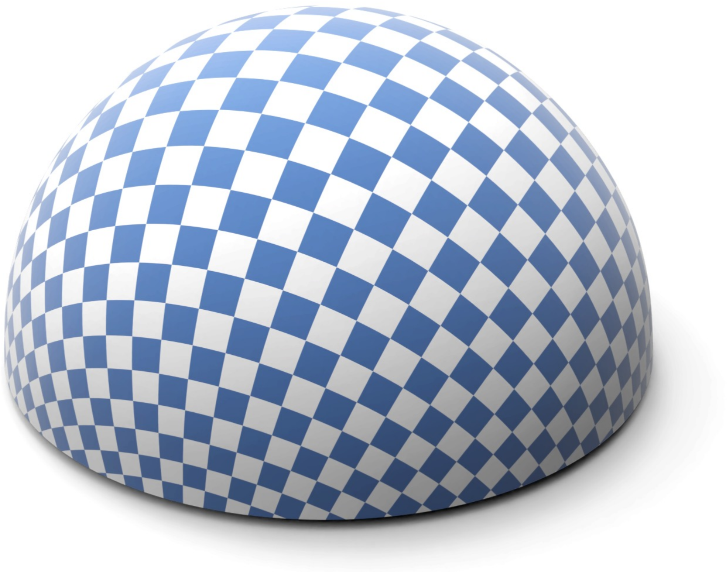
- Local-global iterations!

1. Initialize X with LSCM or Tutte or ARAP or smth
2. Local step: For each T, compute rotation and singular values using the current uv's, i.e. the current value of X
3. Global step: Fix all cheby-jacobians and solve for new X
4. If ~(stopping criterion) goto 2

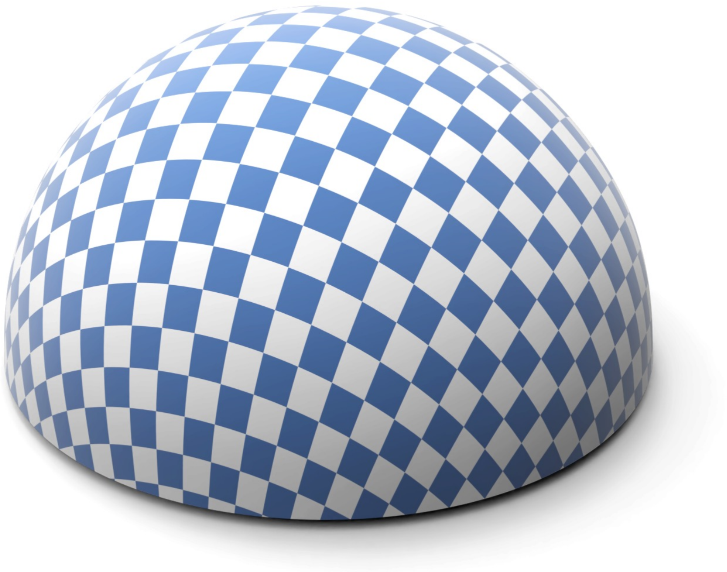
Note: the template here is derived from 2D to 3D; for parameterization we need to consider its inverse. See the paper for more details

LSCM, ARAP, vs. Chebyshev parameterization

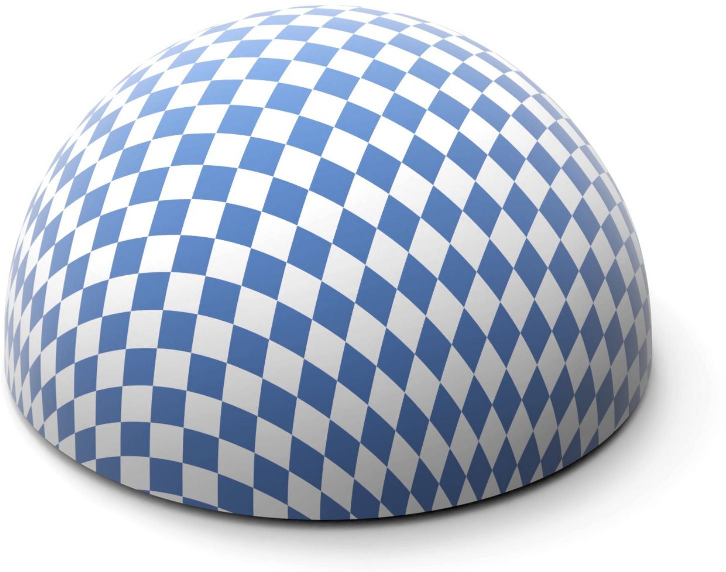
LSCM



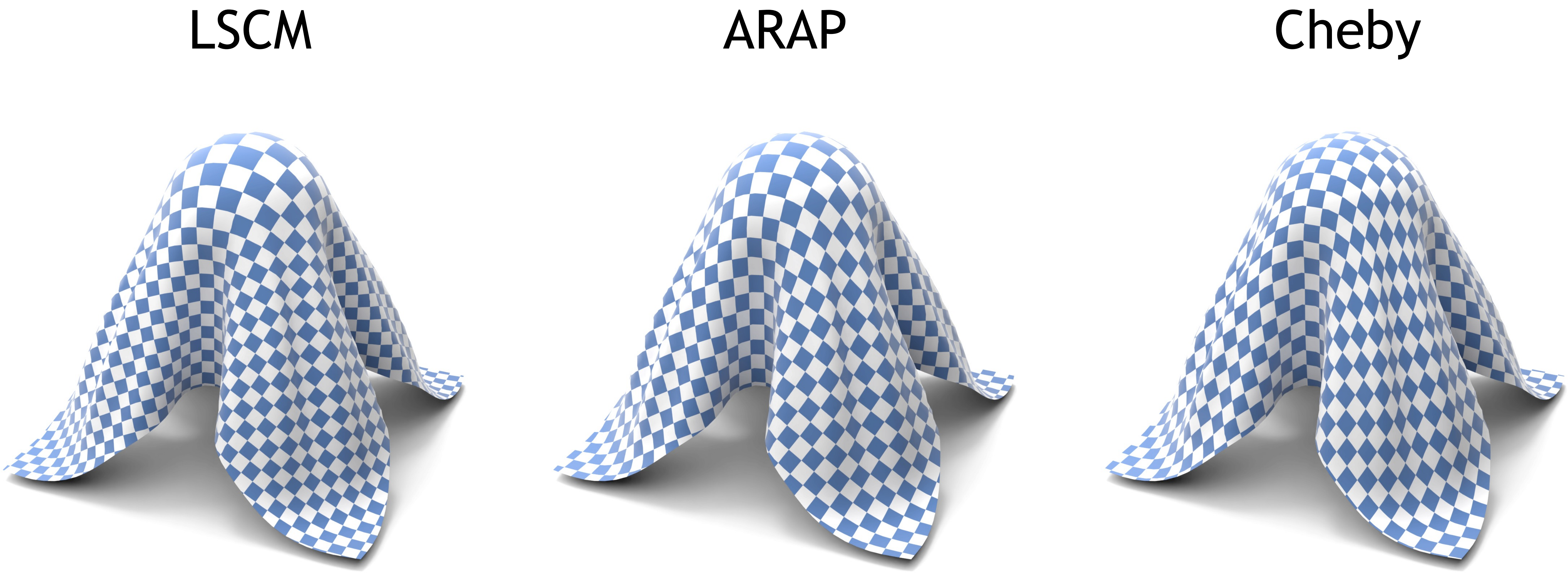
ARAP



Cheby



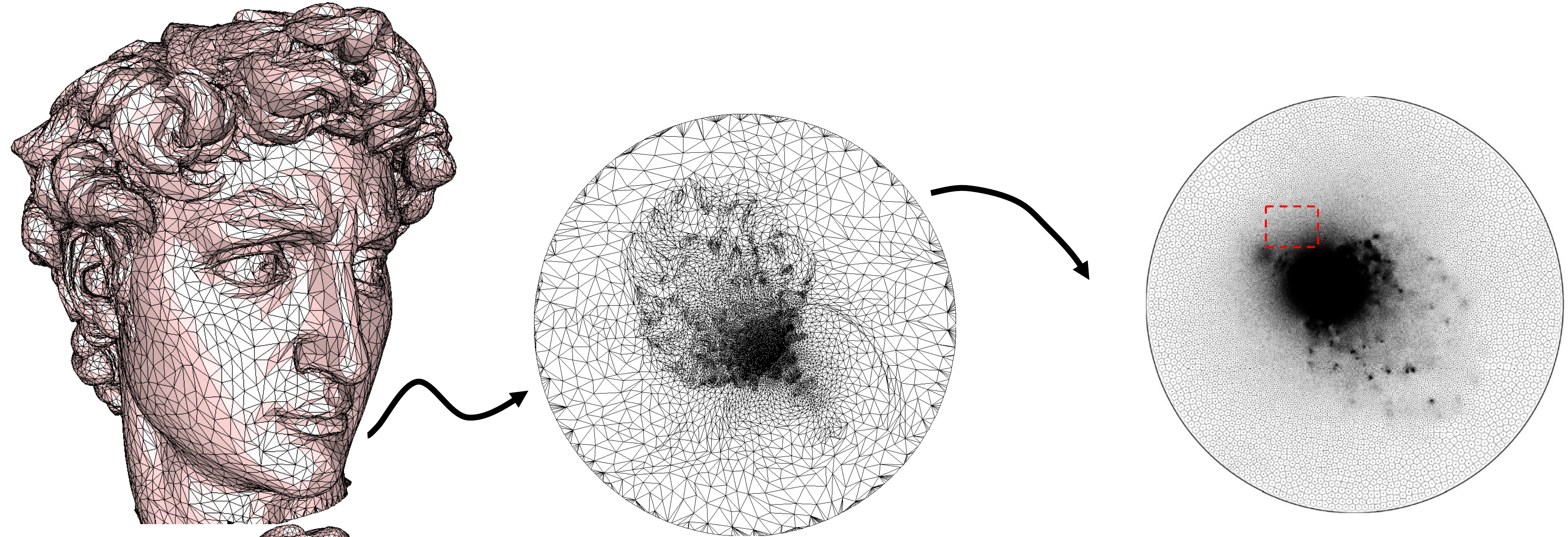
LSCM, ARAP, vs. Chebyshev parameterization

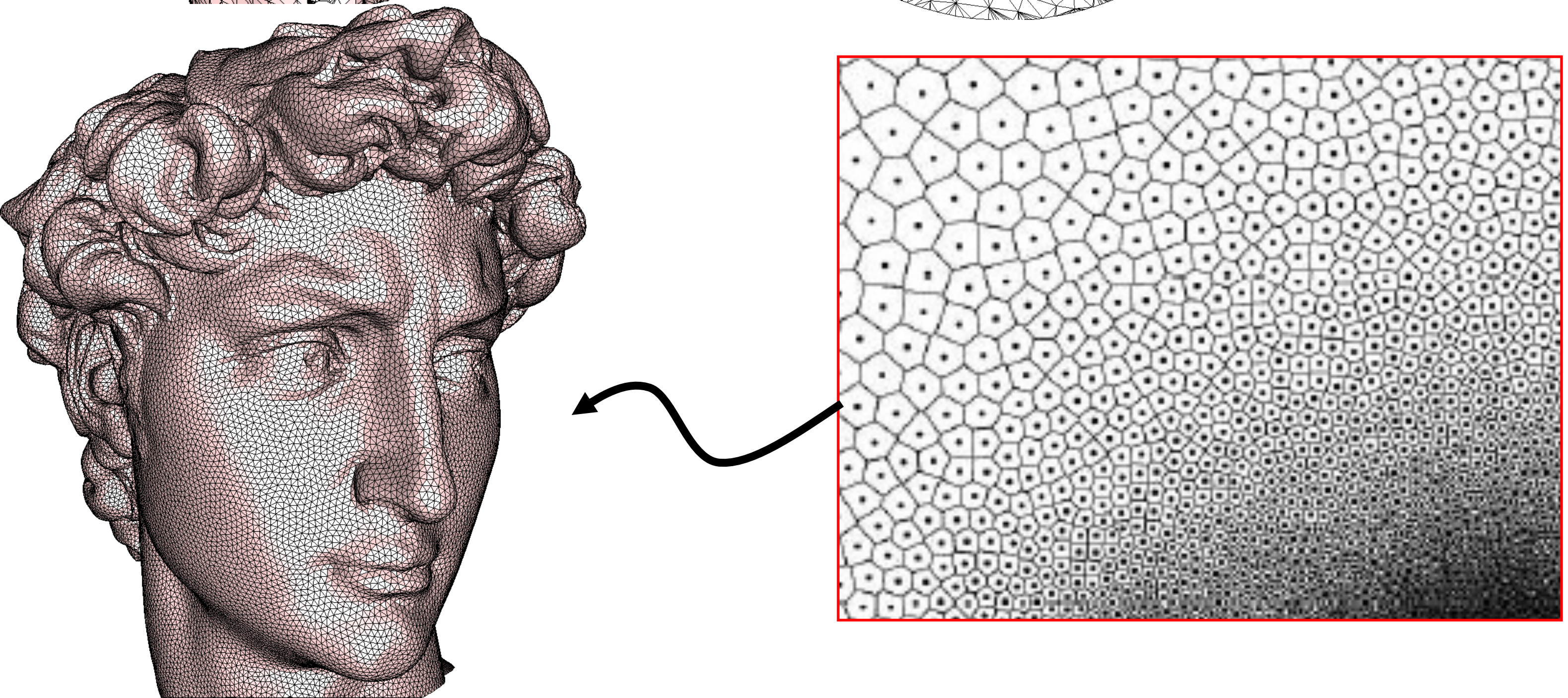


Remeshing

INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



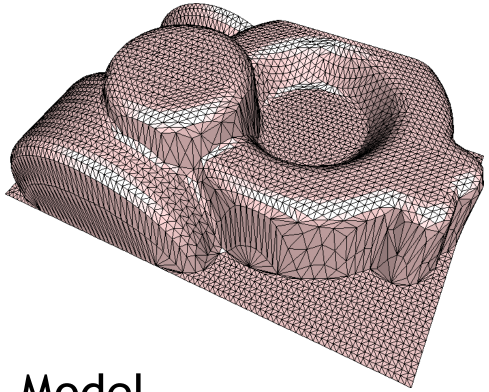


Remeshing

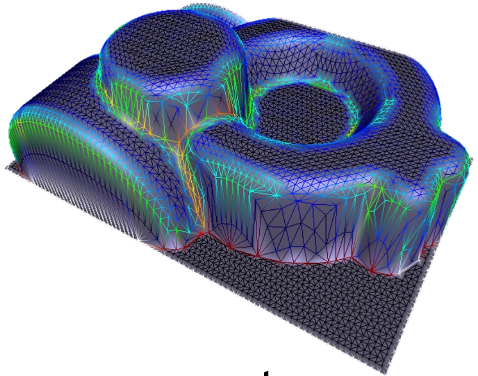
- Run parameterization
- Resample in the uv domain
  - w.r.t area distortion
- Centroidal Voronoi tessellation on the samples
- Map back to 3D surface

Interactive Geometry Remeshing

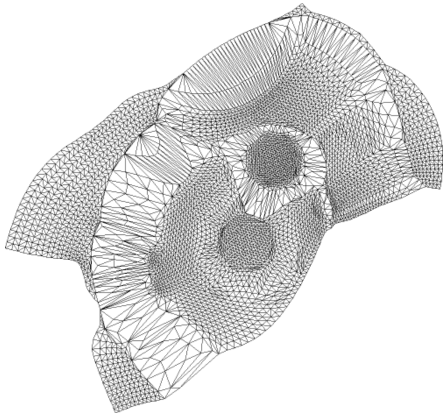
[Alliez et al., SIGGRAPH 2002]



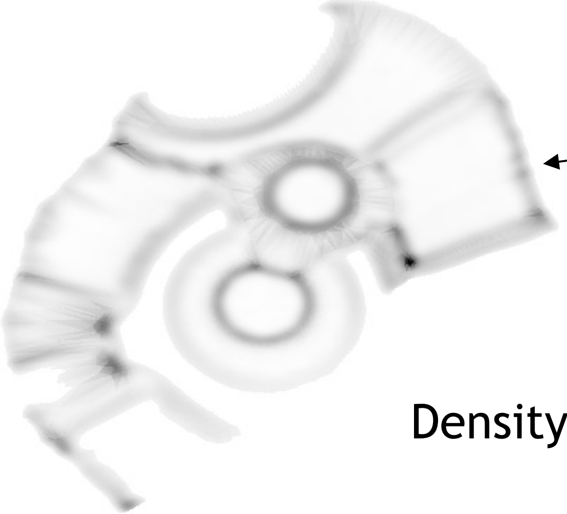
Model



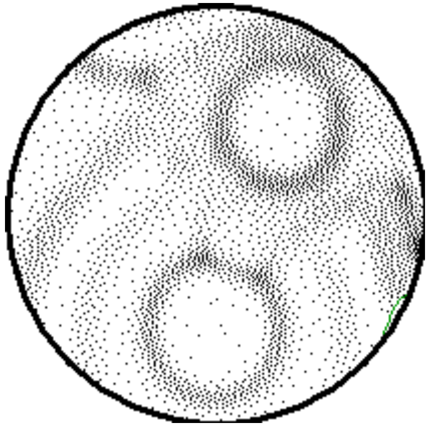
Measure curvature



Flatten it **conformally**

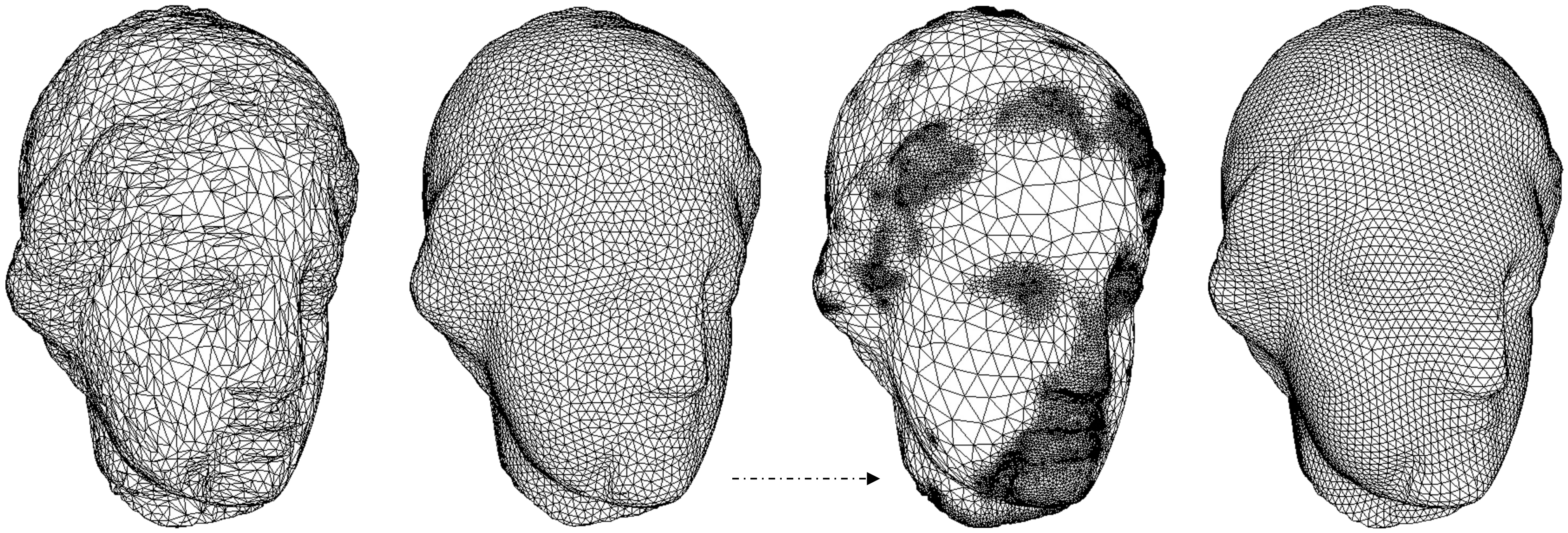


Density function in parameter space



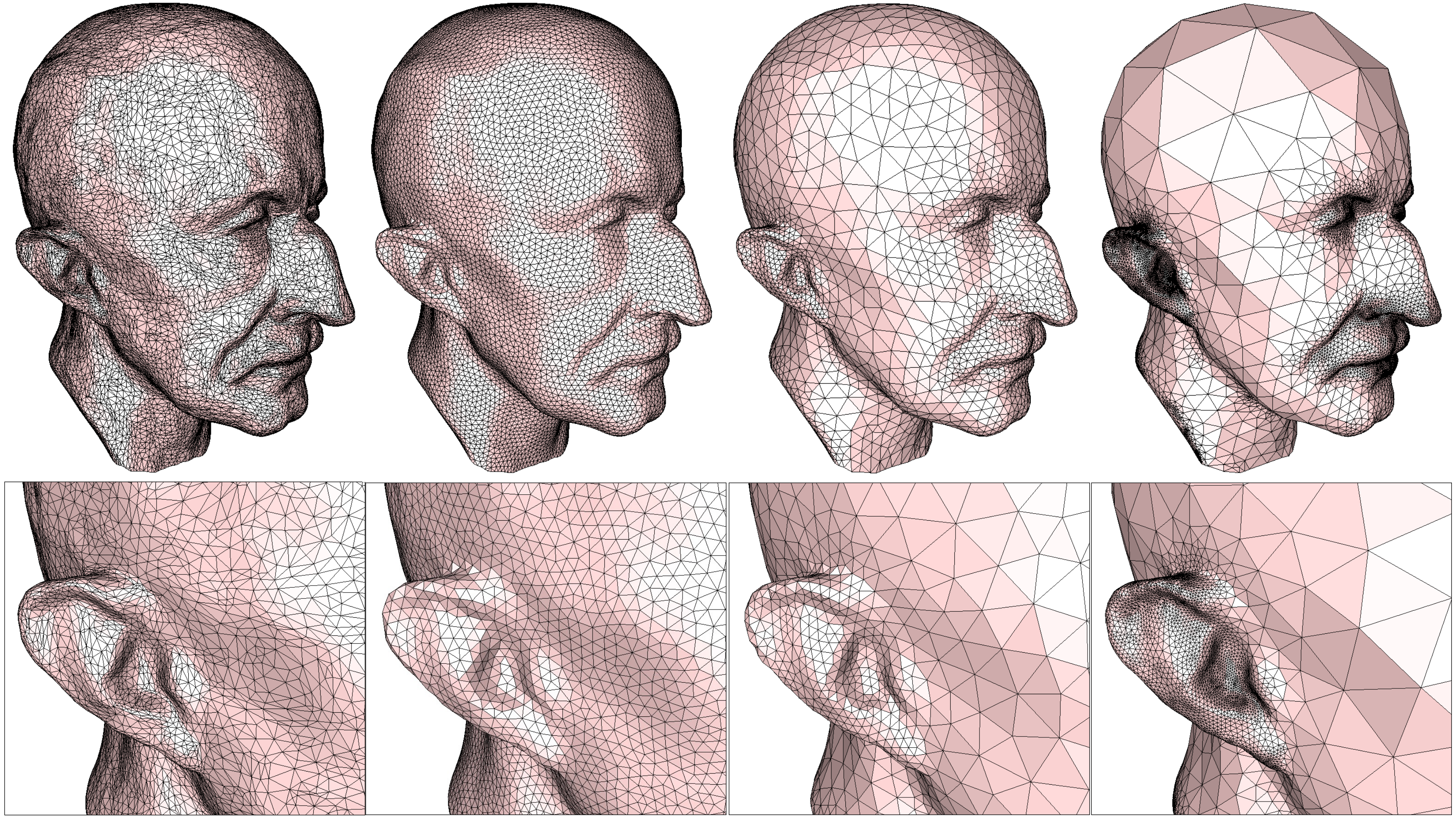
Remeshing

- Remeshing type can be adjusted to the application's needs



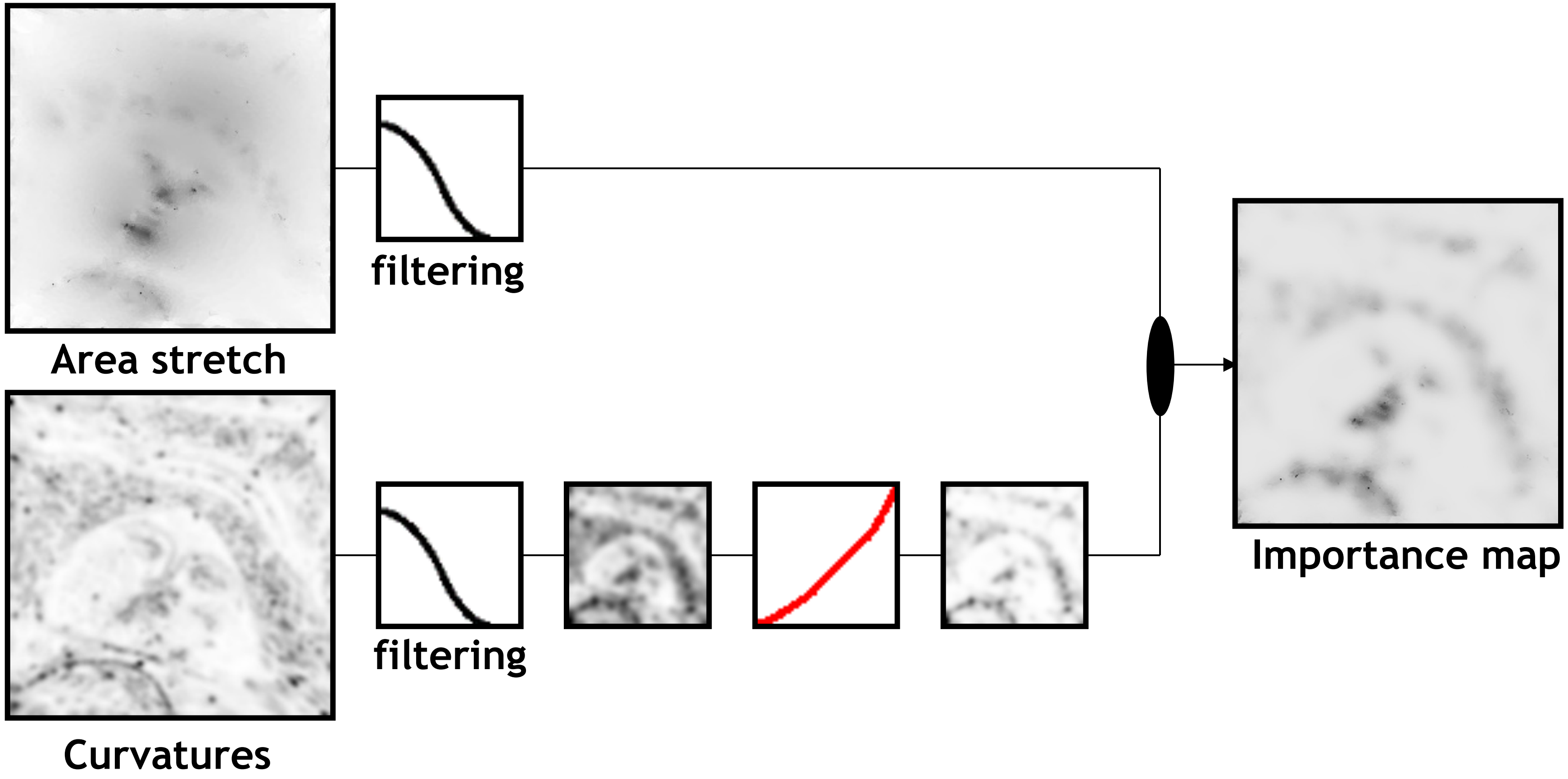
original uniform adapted semi-regular

Remeshing Examples



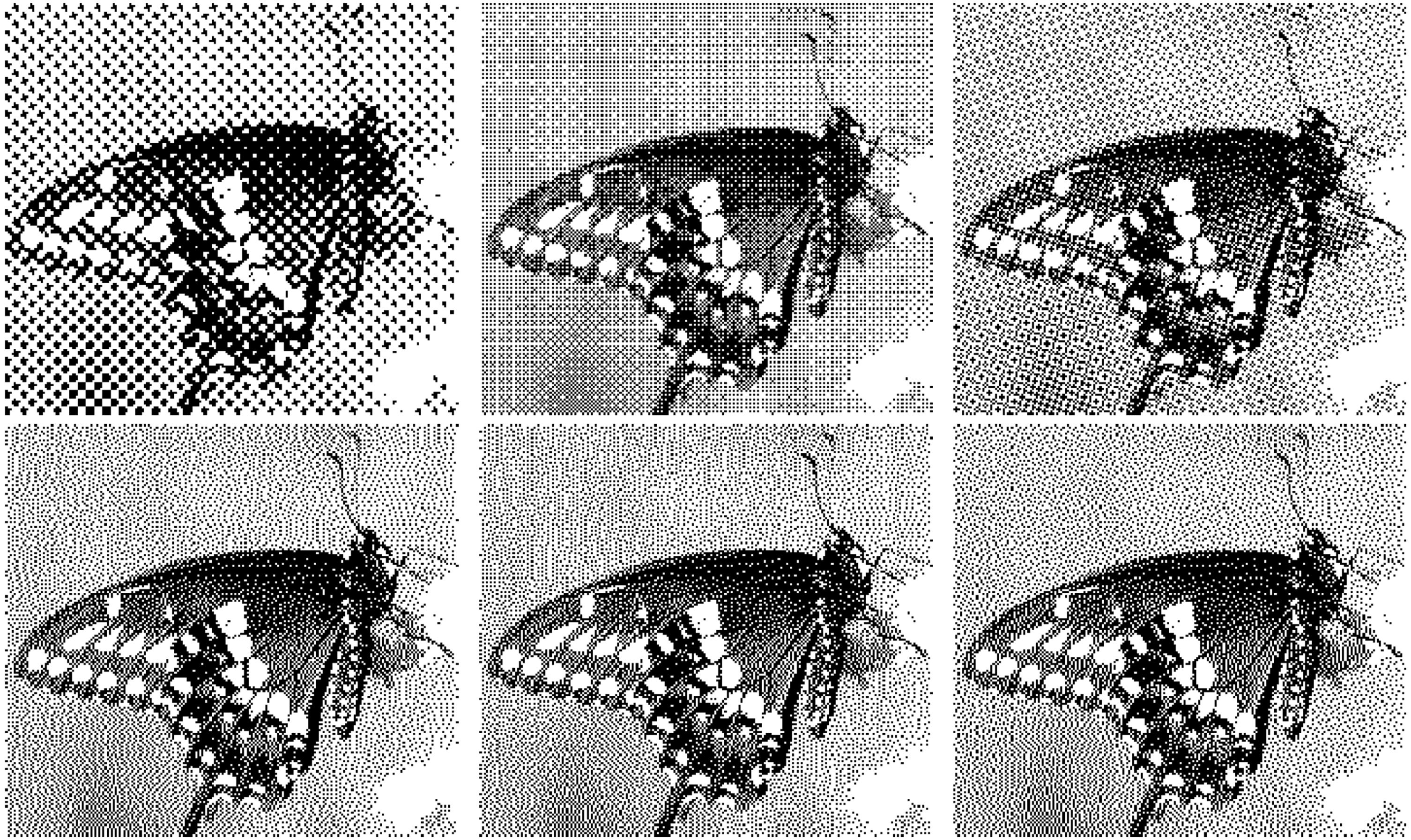
Interactive Geometry Remeshing

- Importance map created according to application needs



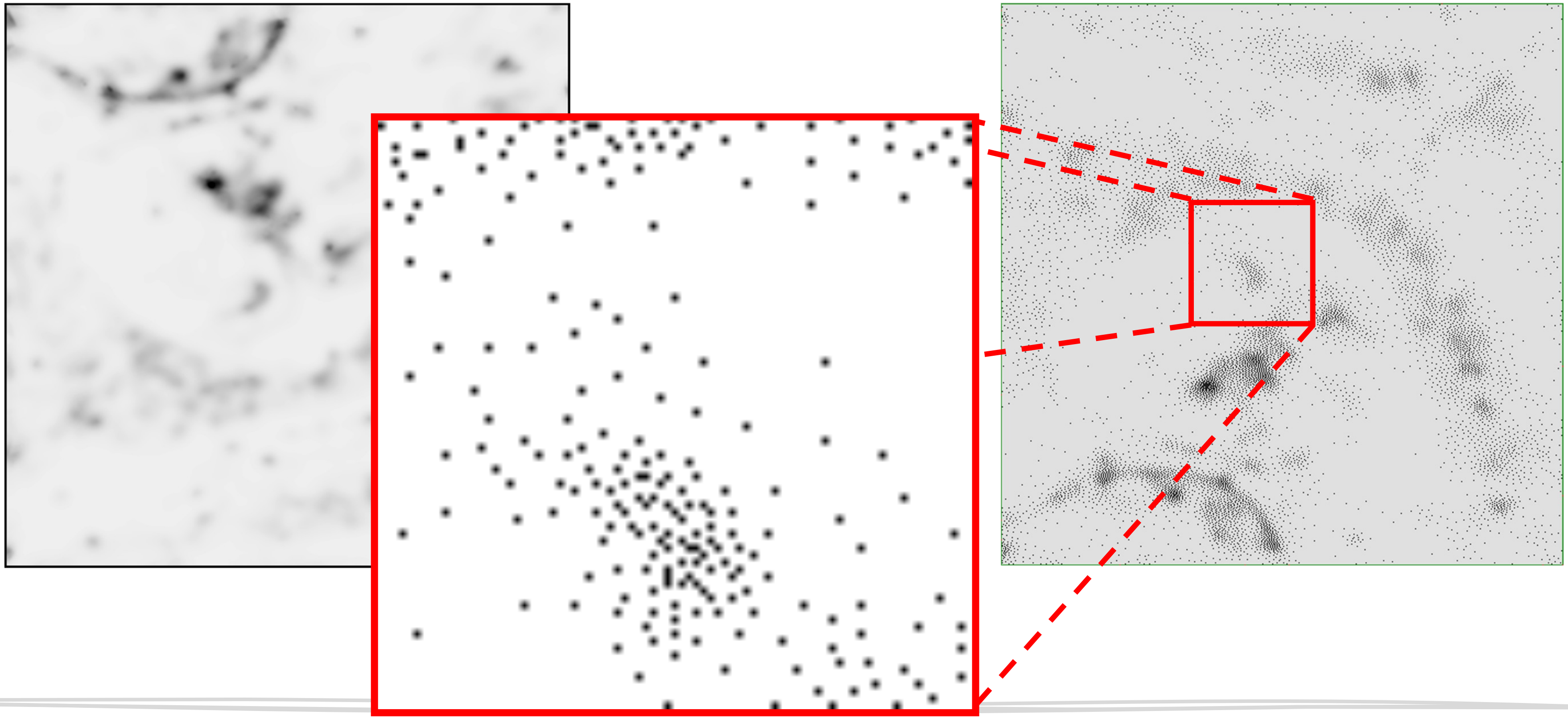
Interactive Geometry Remeshing

- Importance map is sampled by points – as in halftoning



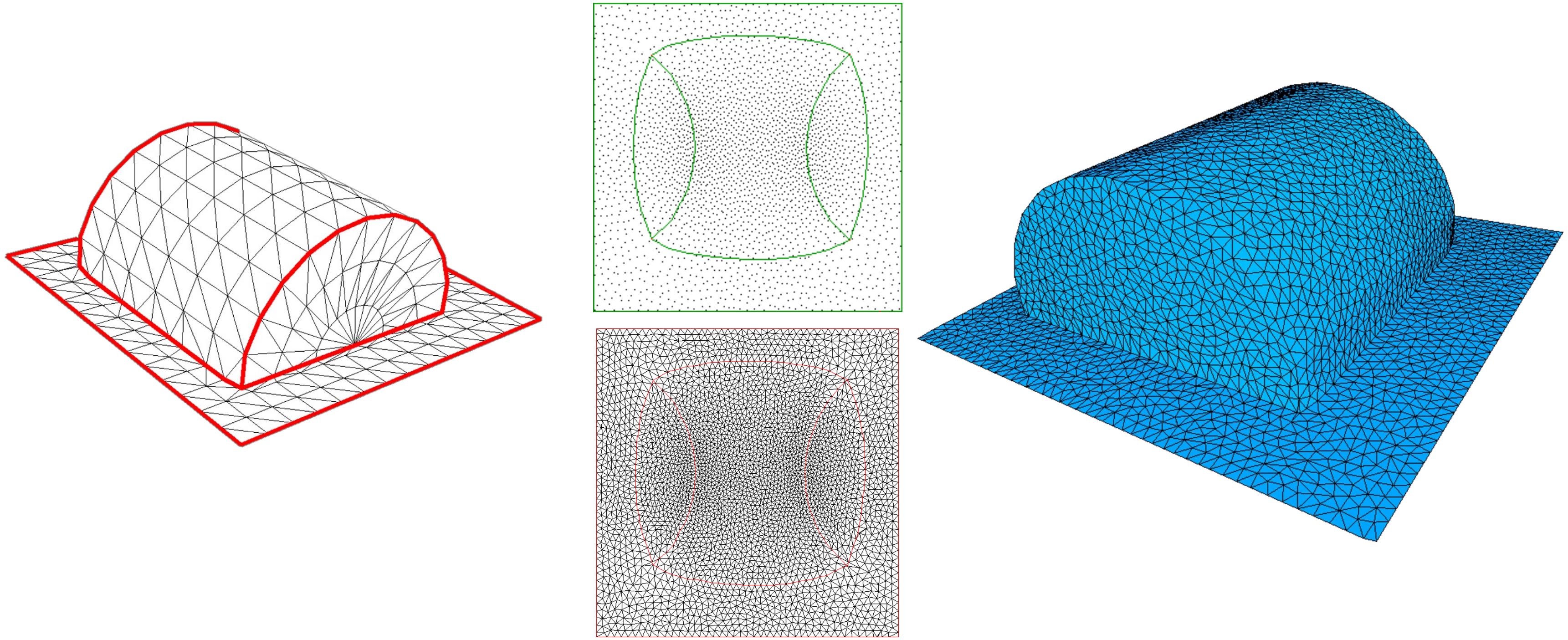
Interactive Geometry Remeshing

- Importance map is sampled by points – as in halftoning (error diffusion process)

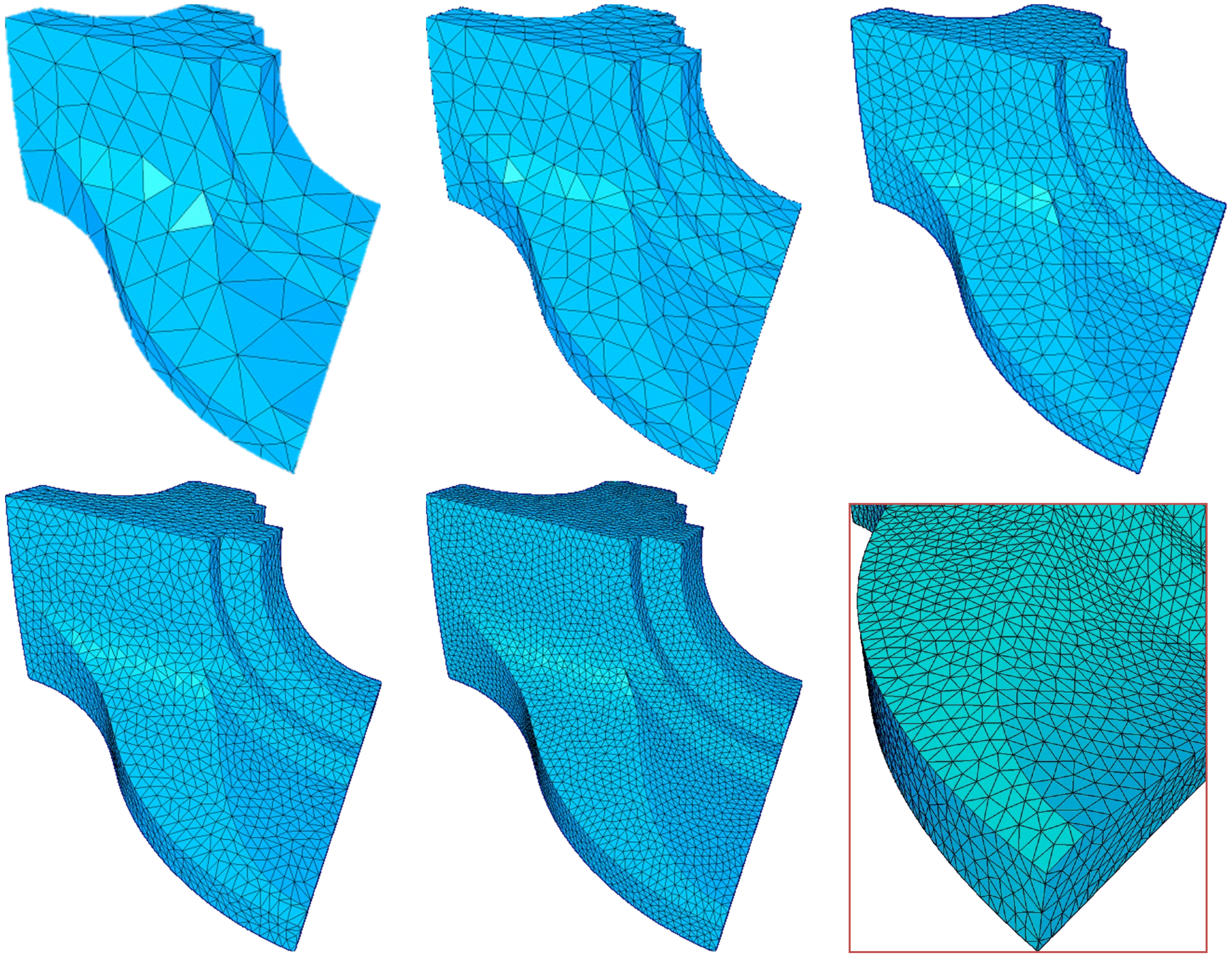


Interactive Geometry Remeshing

- Sampled points are triangulated using Delaunay
- Using the parameterization, the 2D points are lifted back into 3D

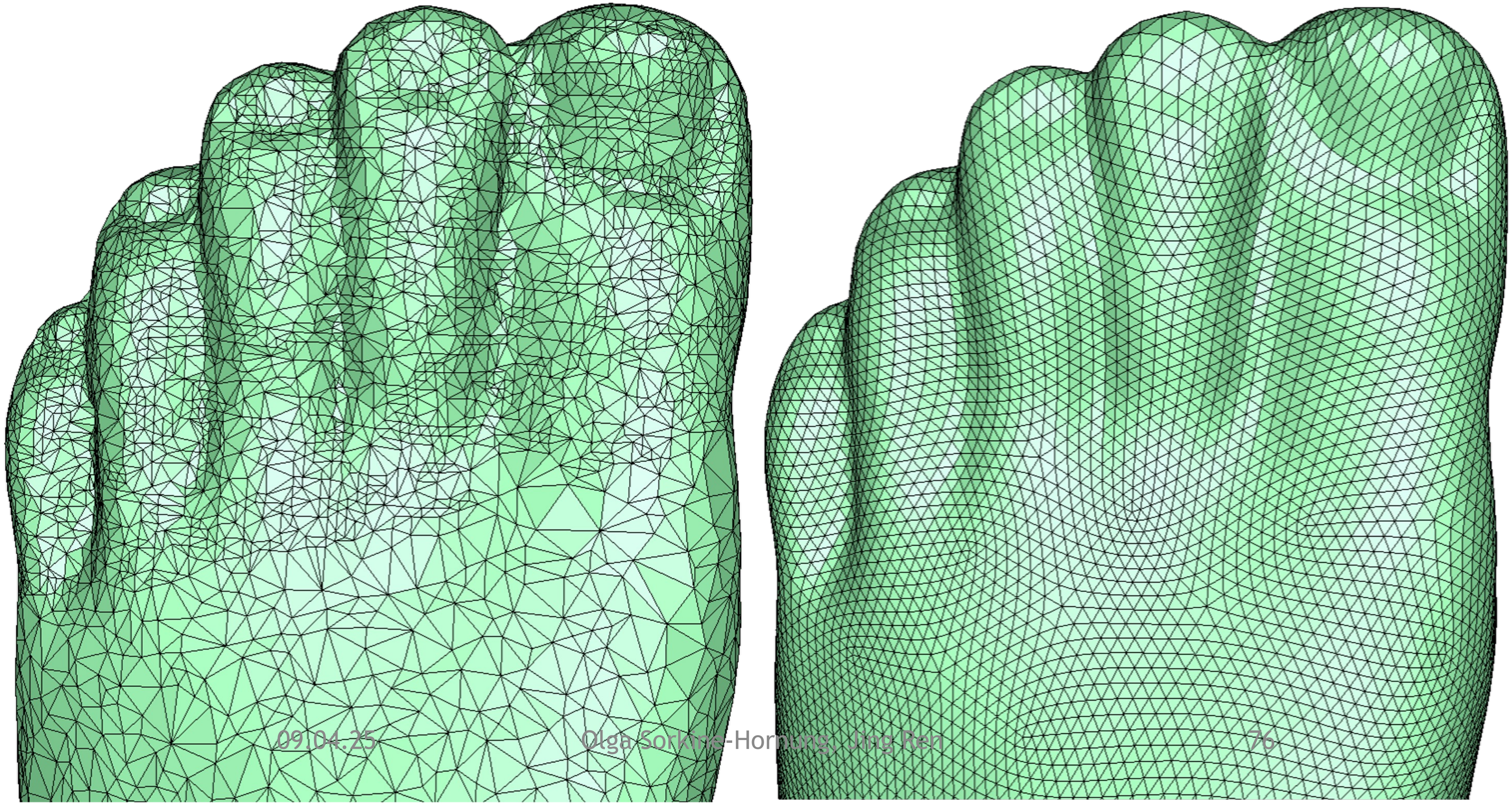


Interactive Geometry Remeshing



Interactive Geometry Remeshing

- More results



Thank You!

INTERACTIVE GEOMETRY LAB

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich