Mesh Parameterization I

USTC, 2024 Spring

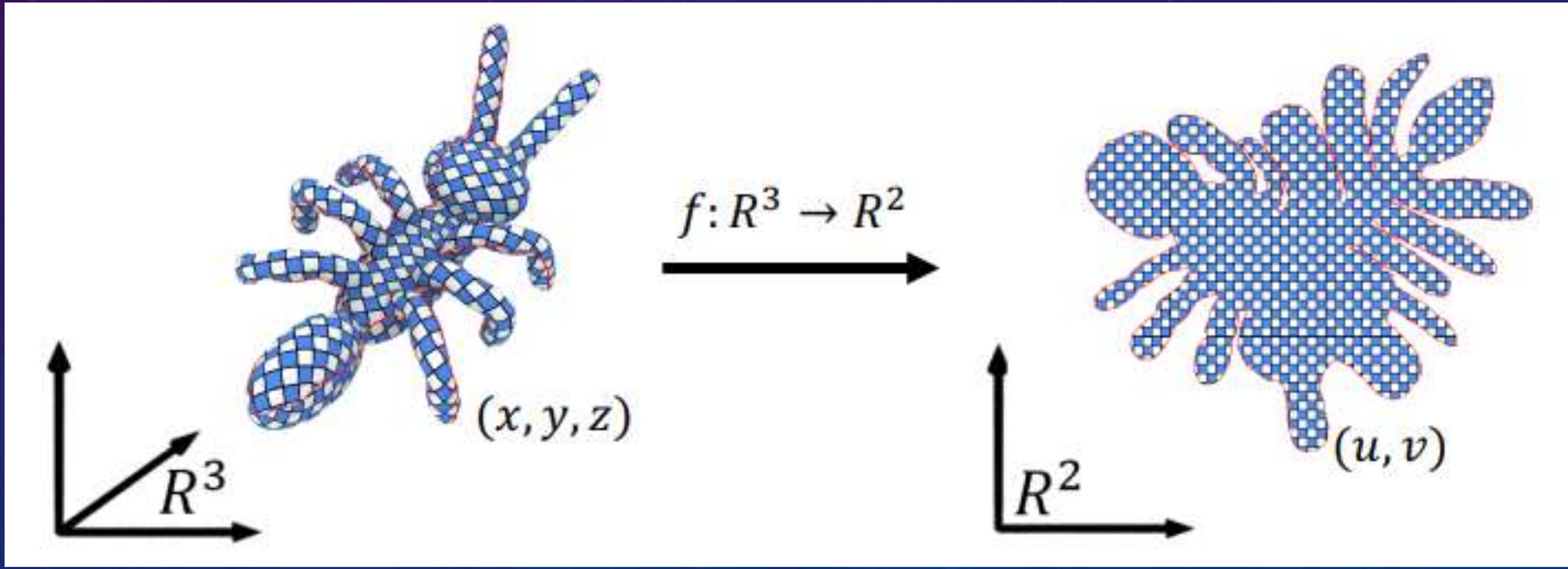
Qing Fang, fq1208@mail.ustc.edu.cn

https://qingfang1208.github.io/

Introduction

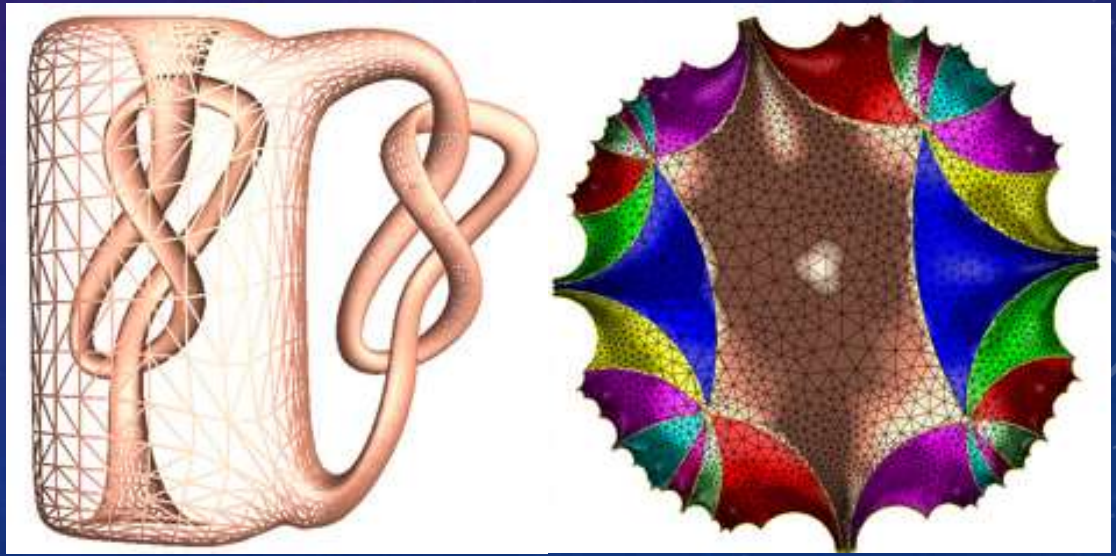
Mesh parameterization

- 3D meshes ⟶ basic domain (plane, sphere, hyperbolic)



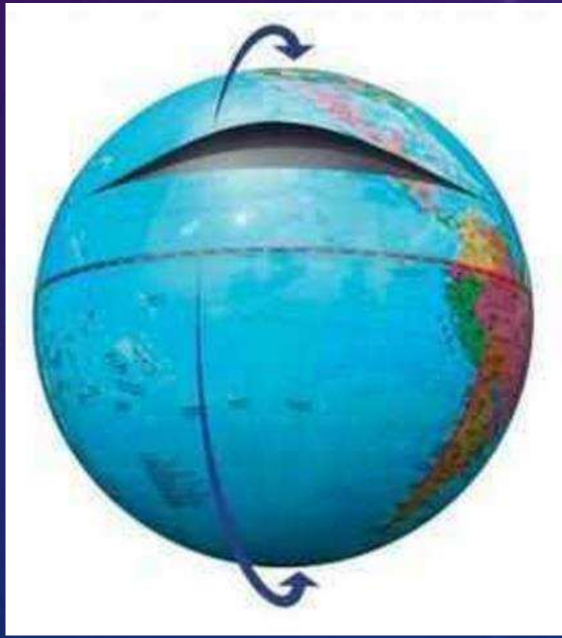
Mesh parameterization

- 3D meshes $\longrightarrow$ basic domain (plane, sphere, hyperbolic)

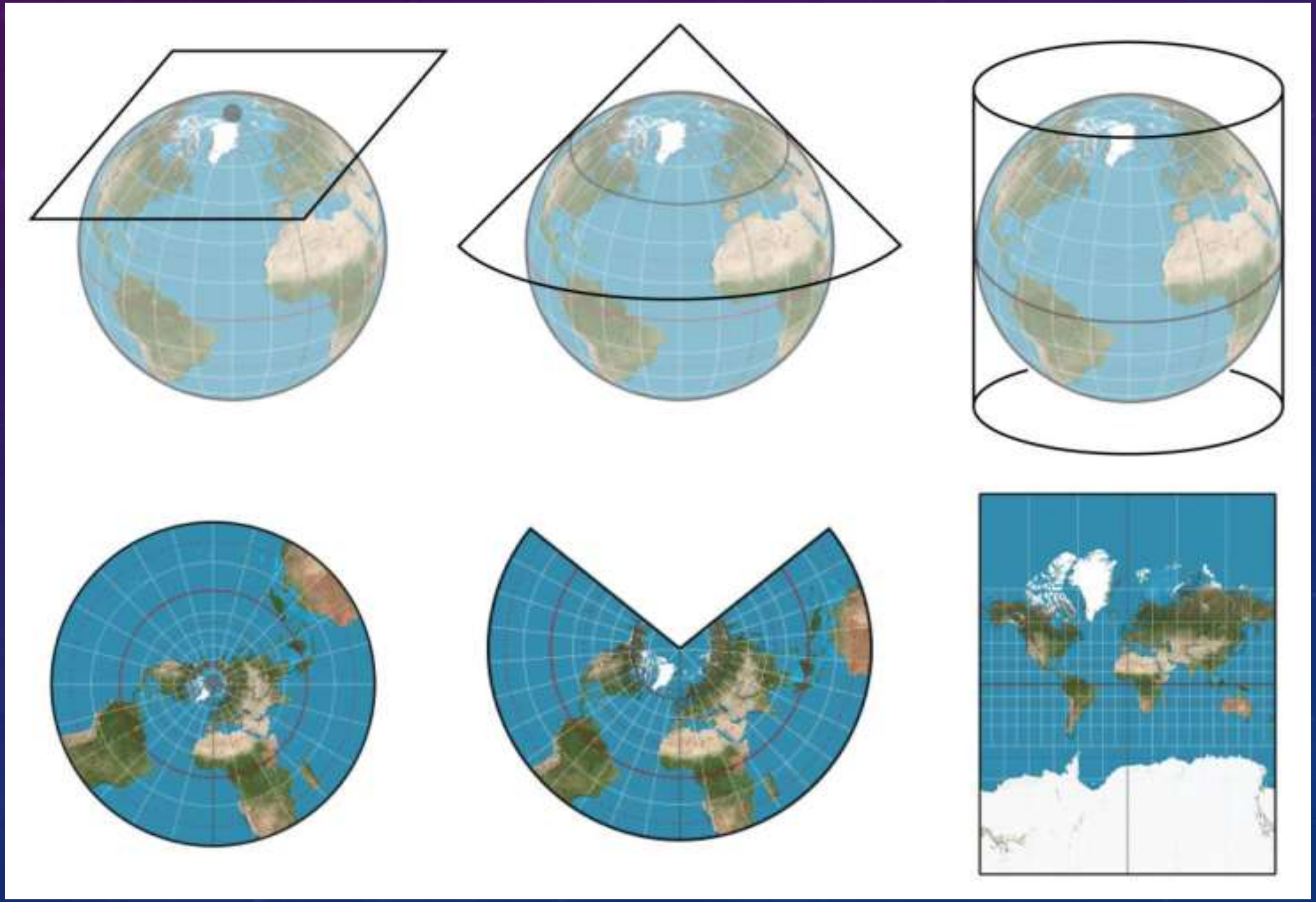


Applications

- Cartography

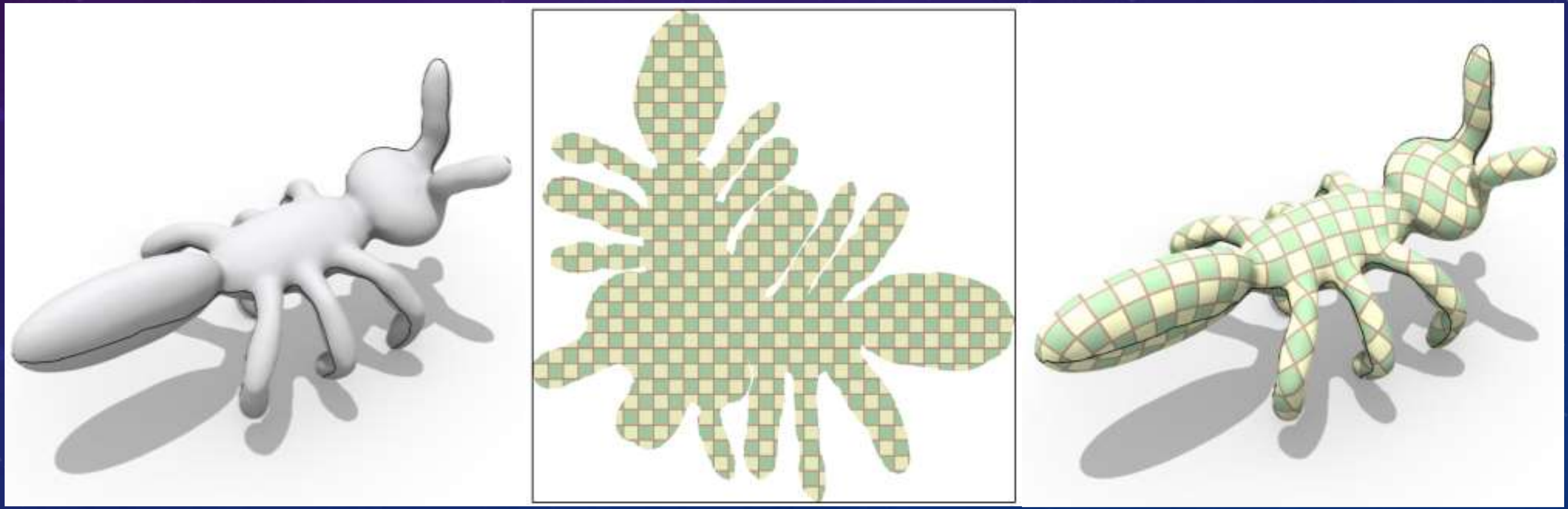


Applications

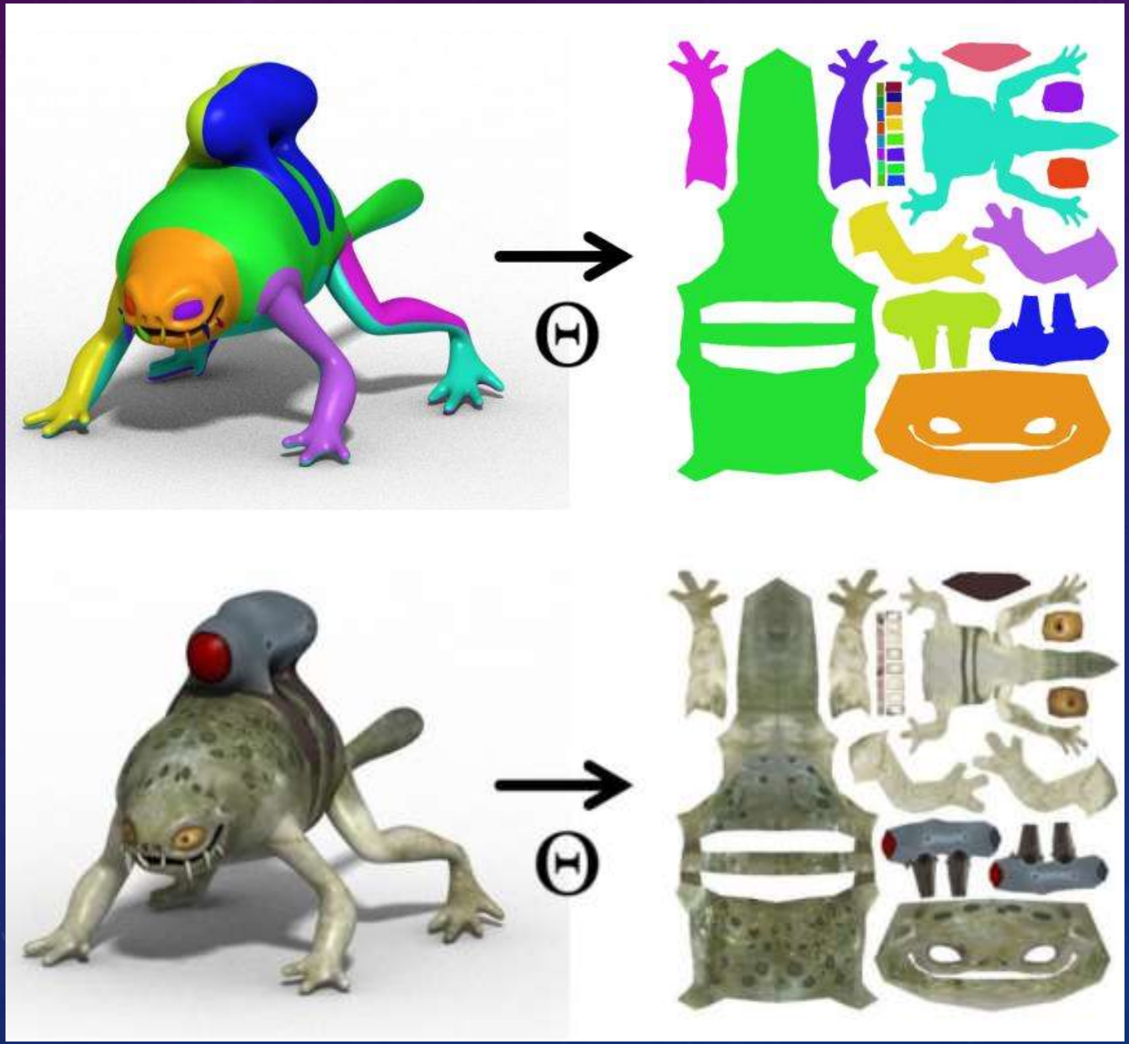


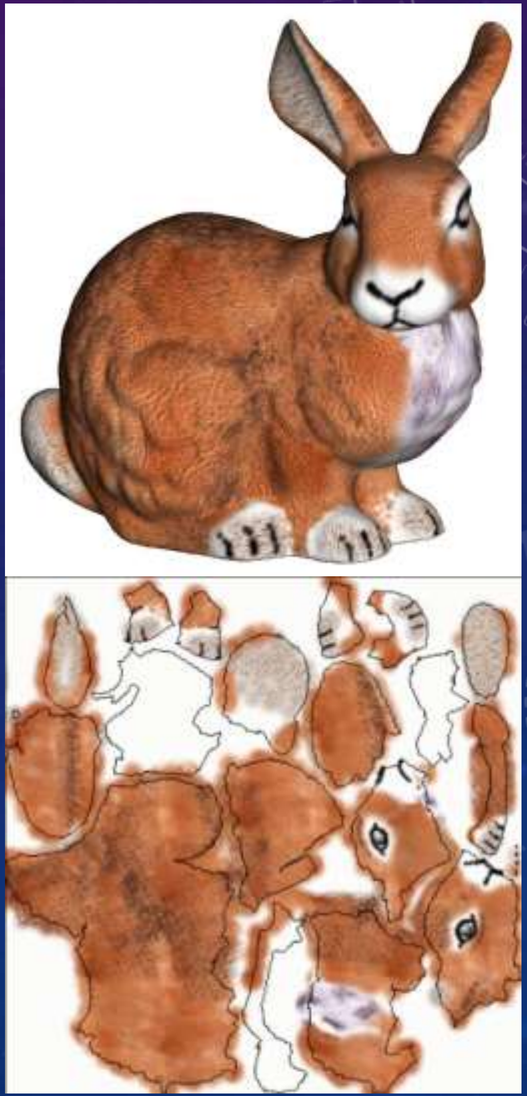
Applications

- Texture (albedo, material, normal, displacement…)



Applications



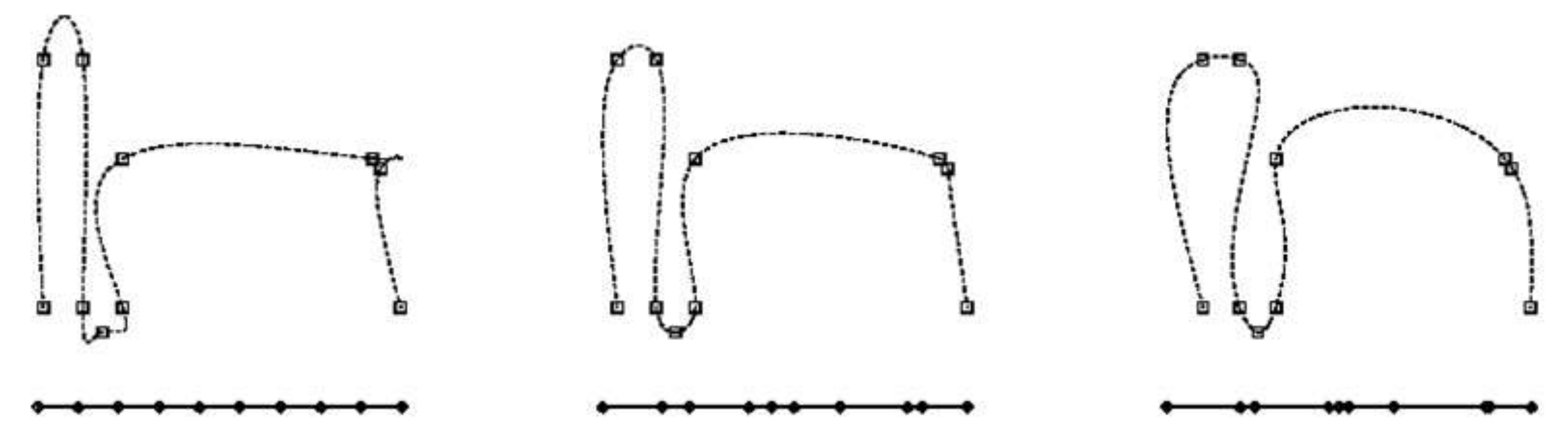


Applications

- Fitting

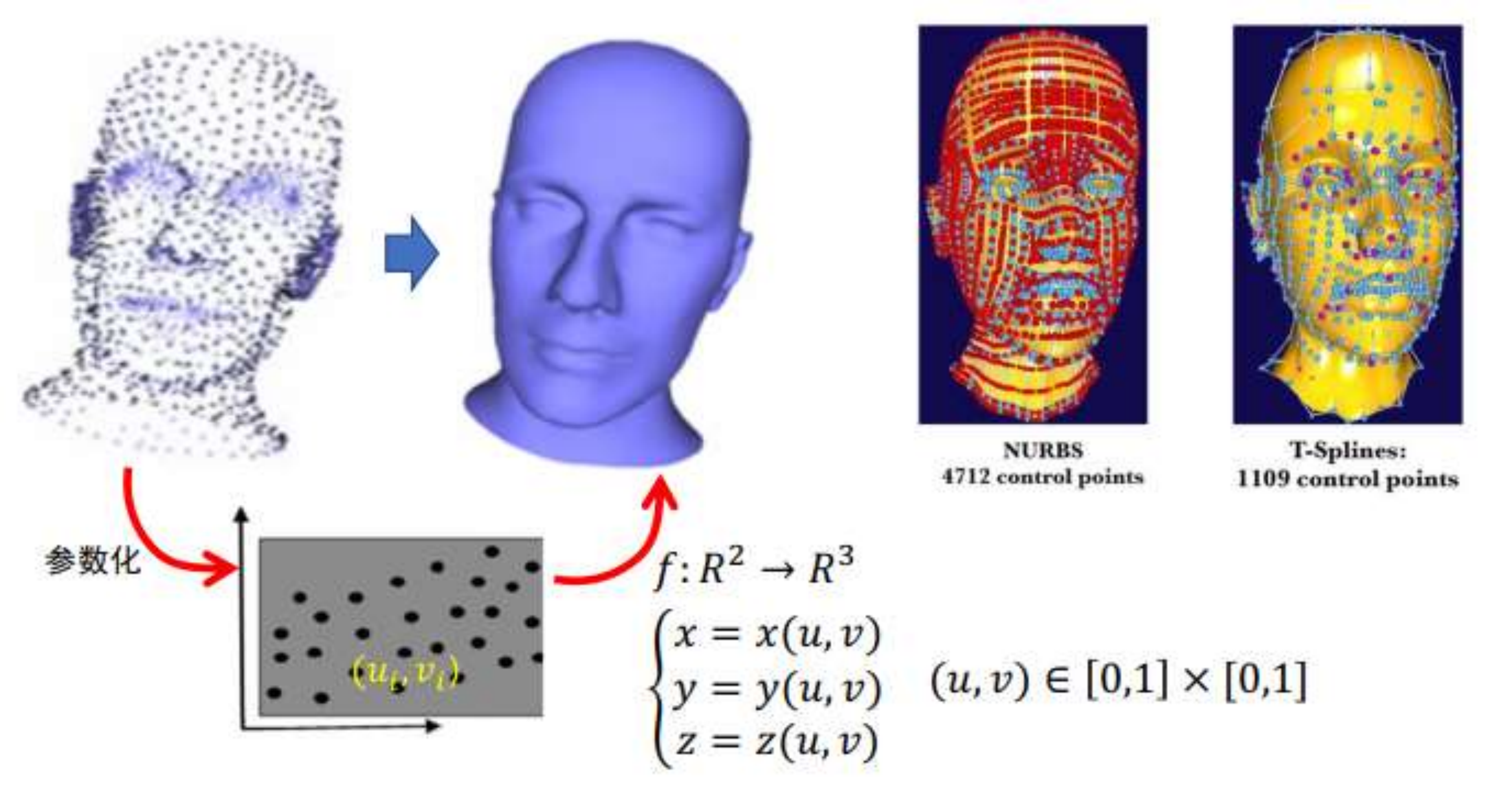
$$f: R^1 \to R^2 \qquad \begin{cases} x = x(t) \\ y = y(t) \end{cases} \qquad t \in [0,1]$$

$$\min E = \sum_{i=1}^{n} \| \boldsymbol{p}(t_i) - \boldsymbol{p}_i \|^2$$



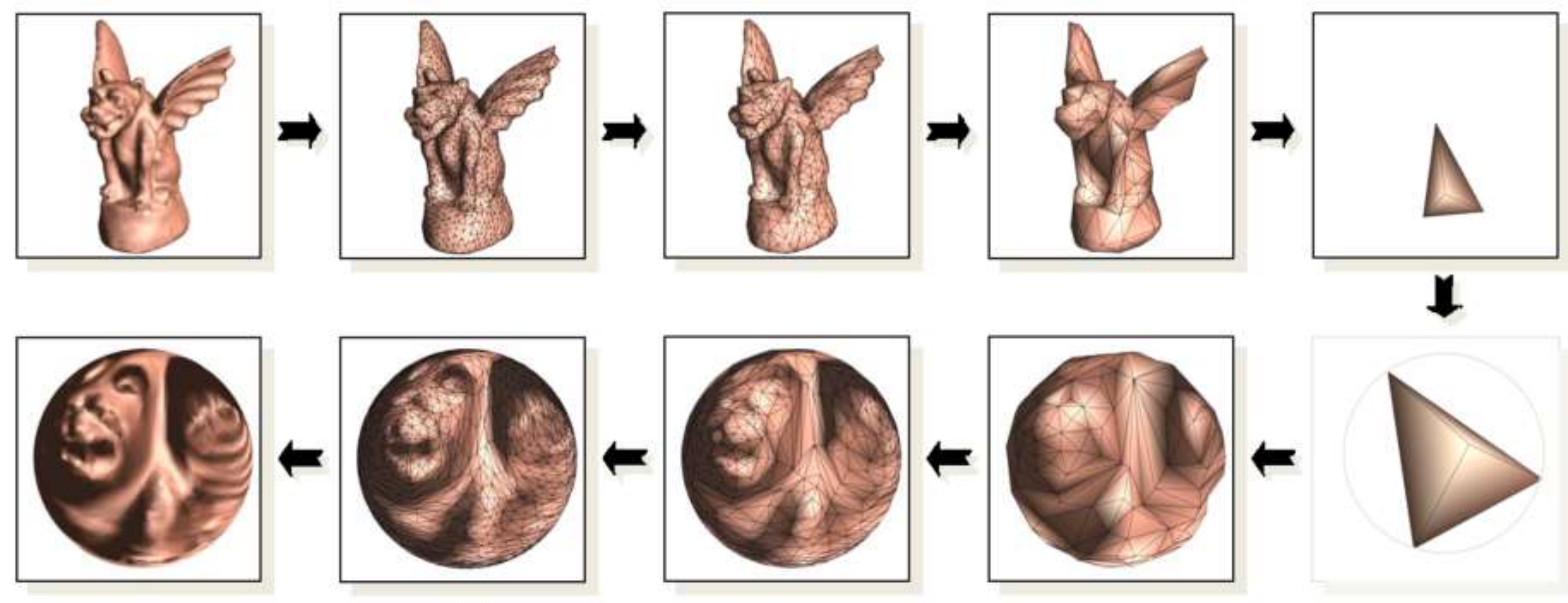
Applications

- Fitting



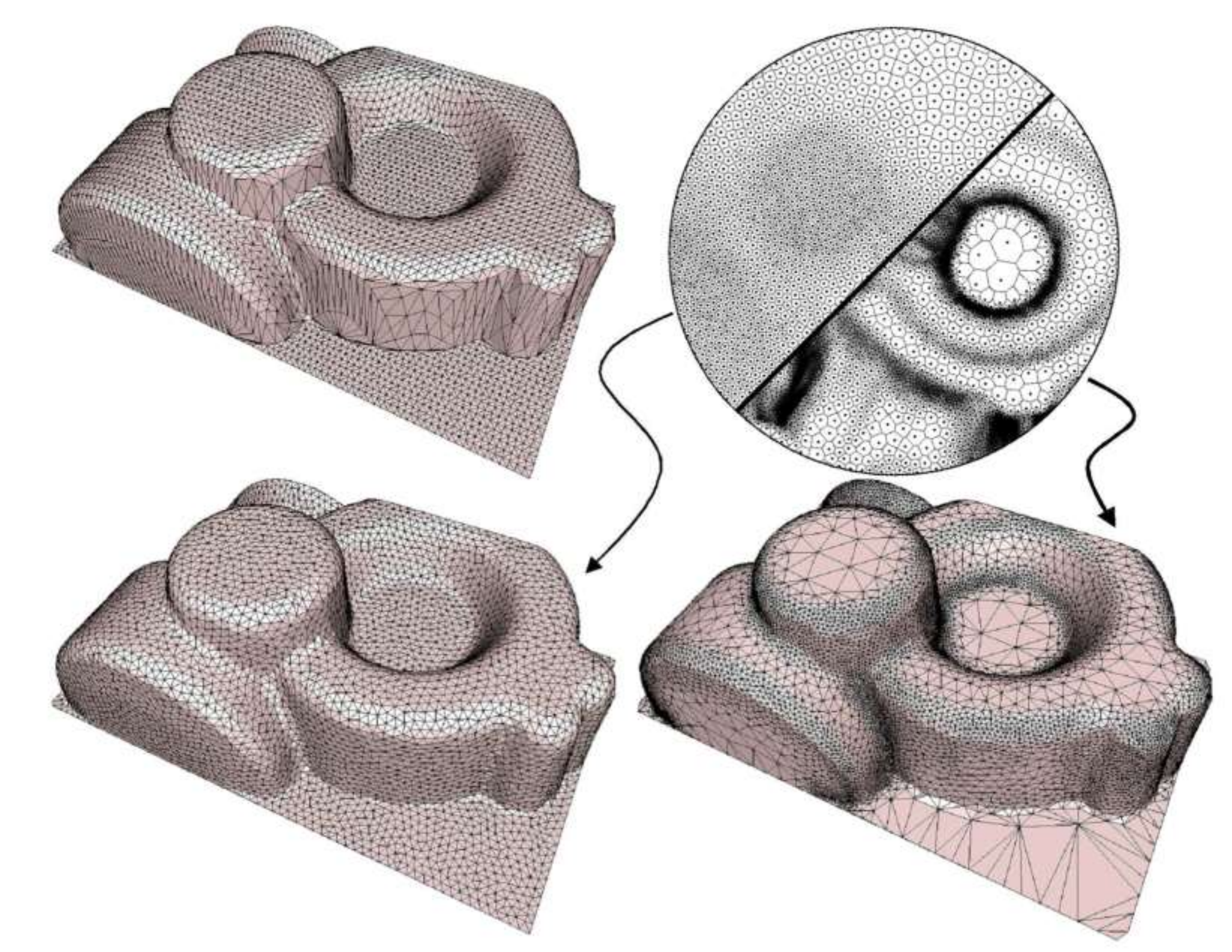
Applications

- Simplification



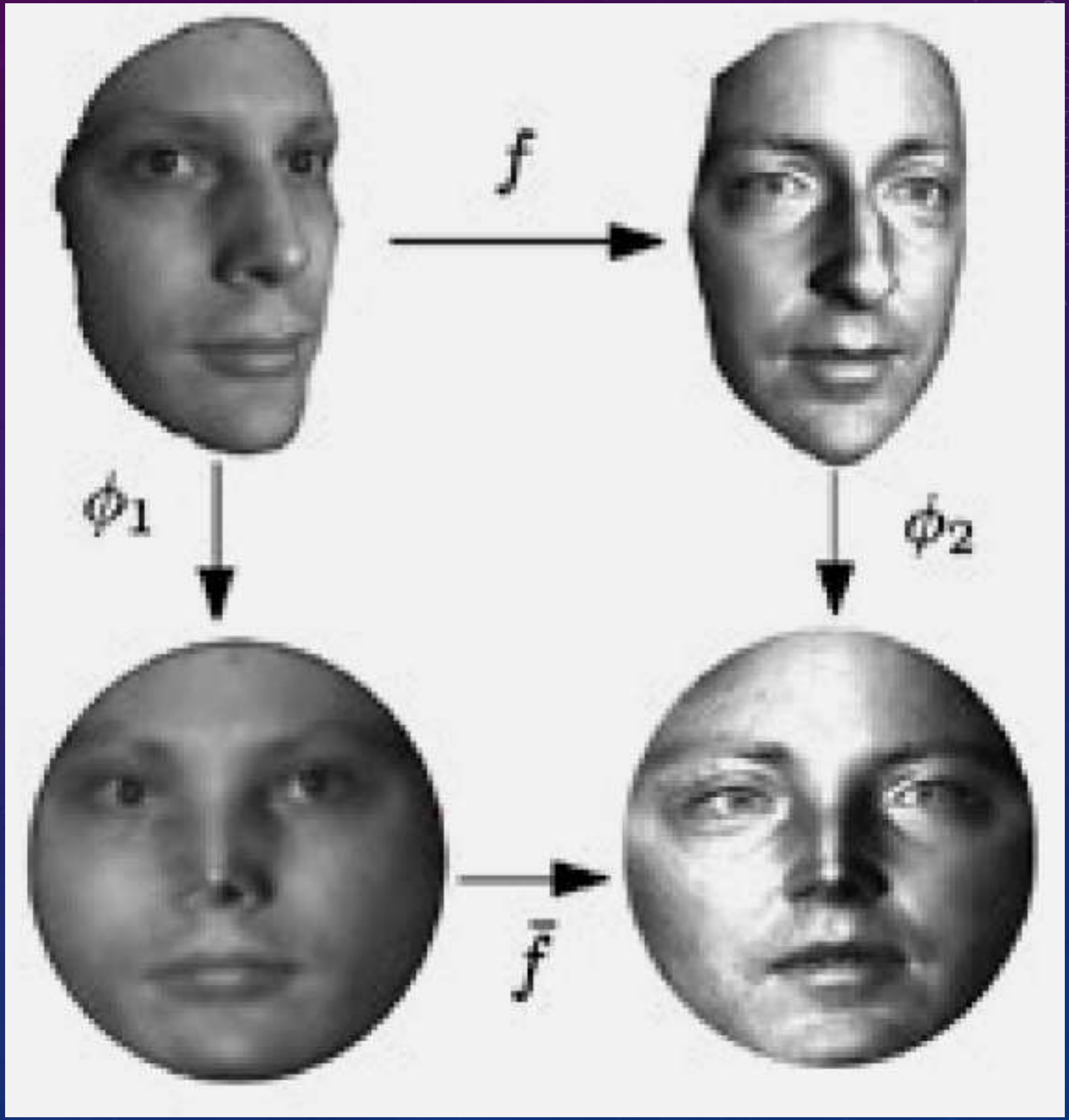
Applications

- Remeshing



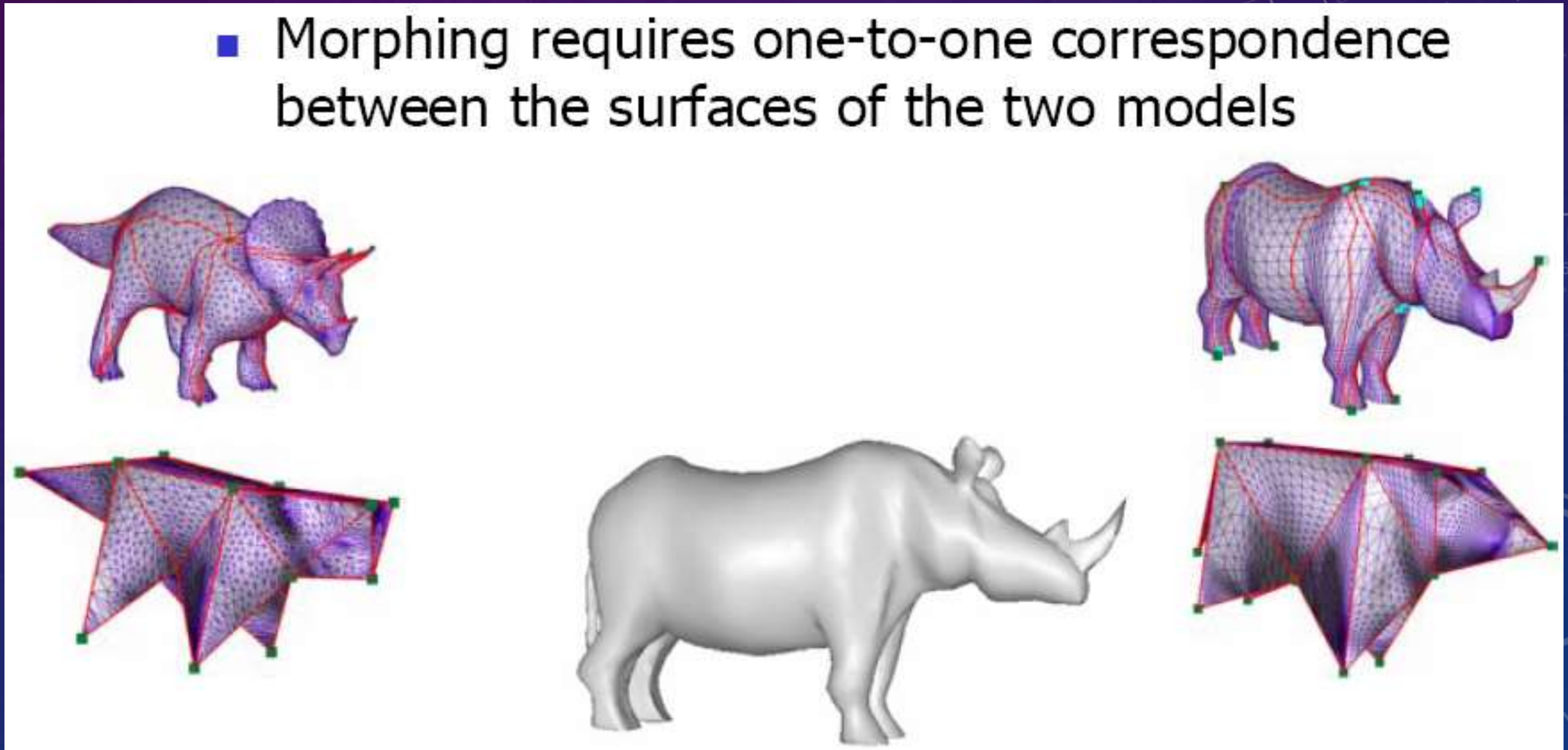
Applications

- Matching



Applications

- Morphing



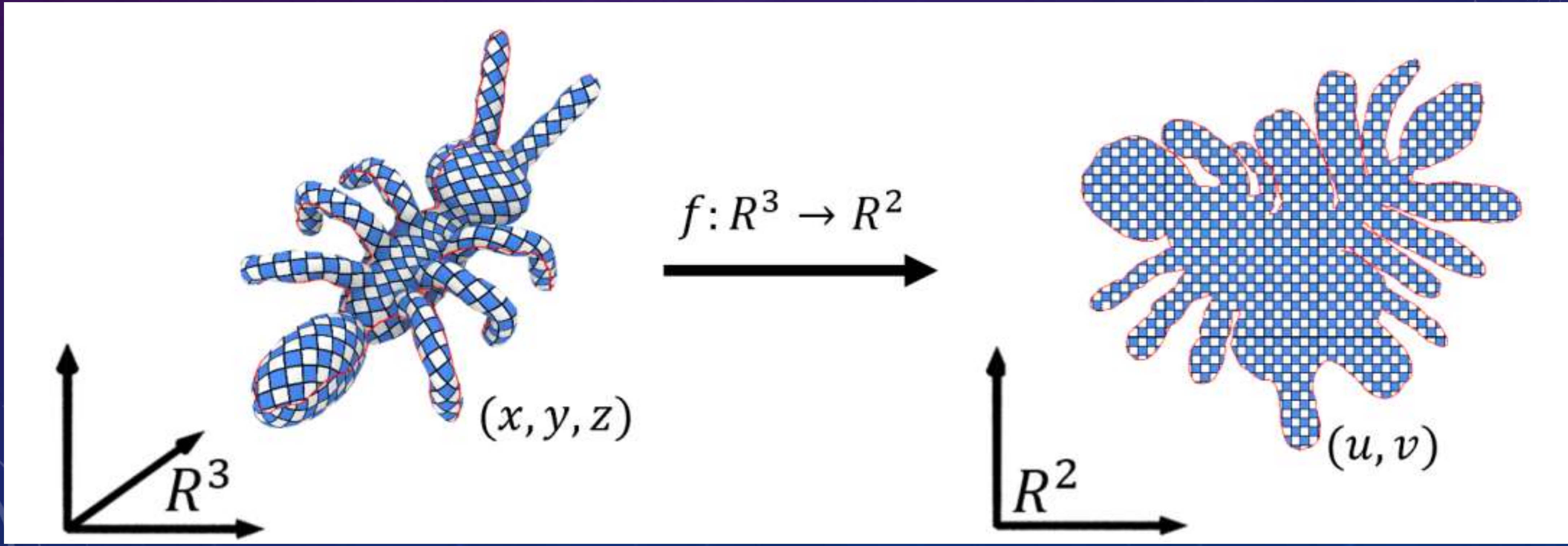
Applications

- Visualization, compression, transmission, reconstruction, repairing, texture synthesis, rendering animation…

Representation

Mapping

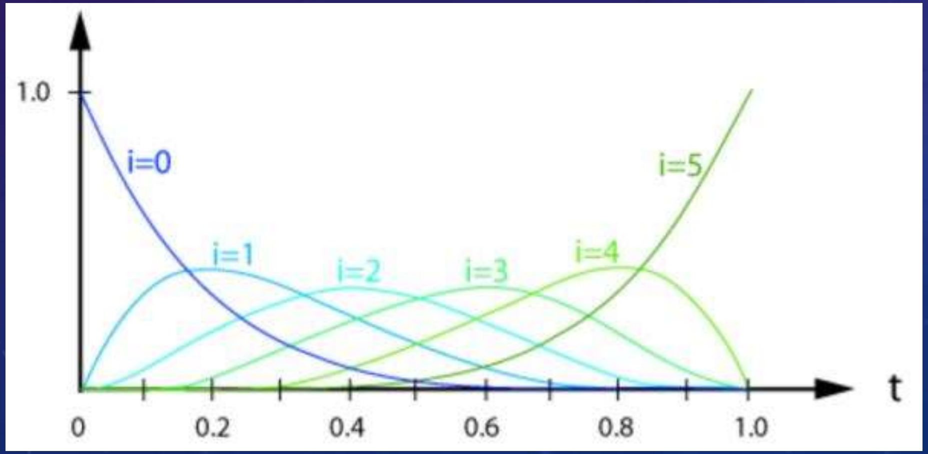
- Functions: $(x_i, y_i, z_i) \rightarrow (u_i, v_i)$

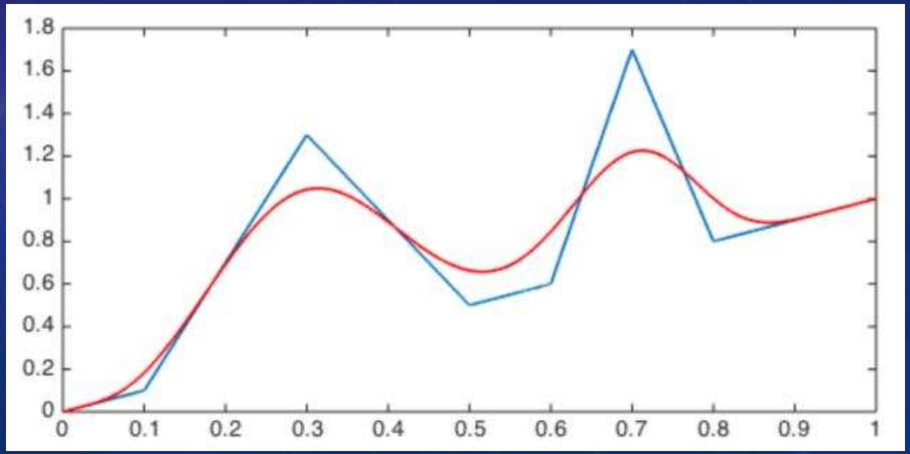


Smooth functions

Basis functions: $f(x) = \begin{pmatrix} u(x) \\ v(x) \end{pmatrix} = \begin{pmatrix} \sum a_i f_i(x) \\ \sum b_i f_i(x) \end{pmatrix}$

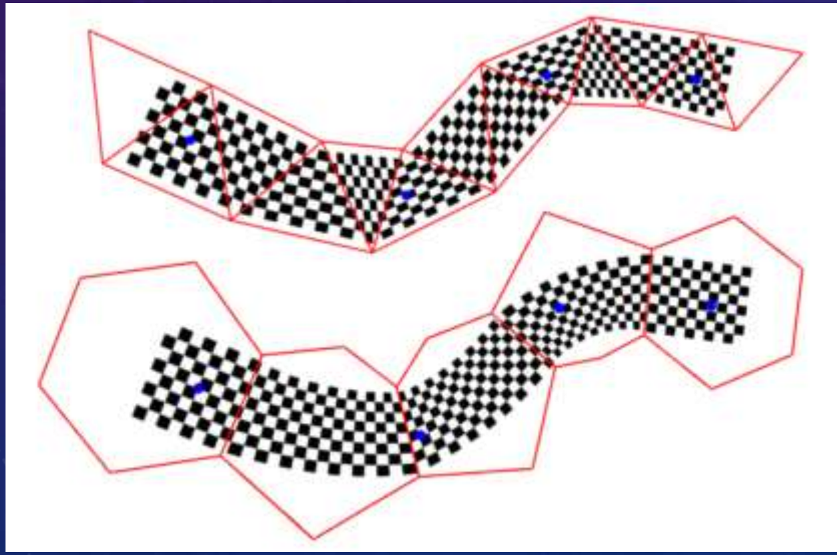
- Bernstein, B-spline, Fourier, wavelet, …
- Barycentric coordinates, harmonic mapping, radial basis function, …

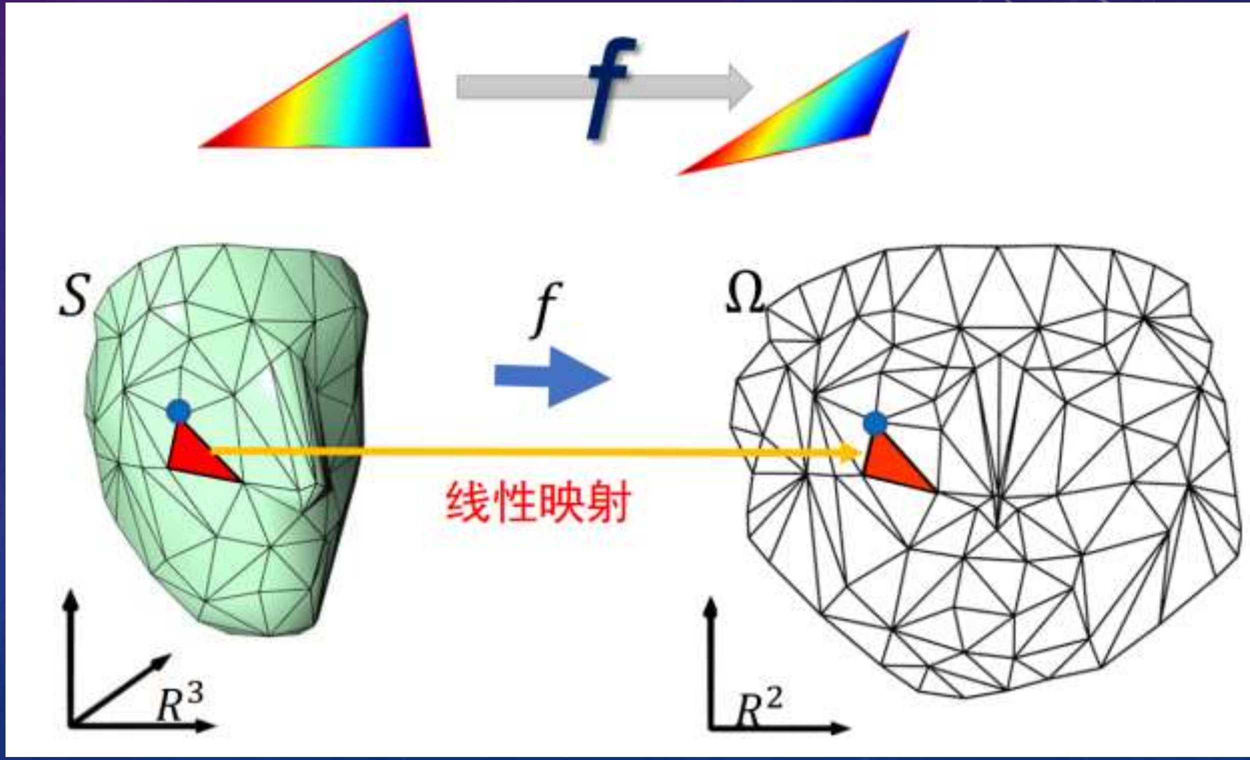




Piecewise functions

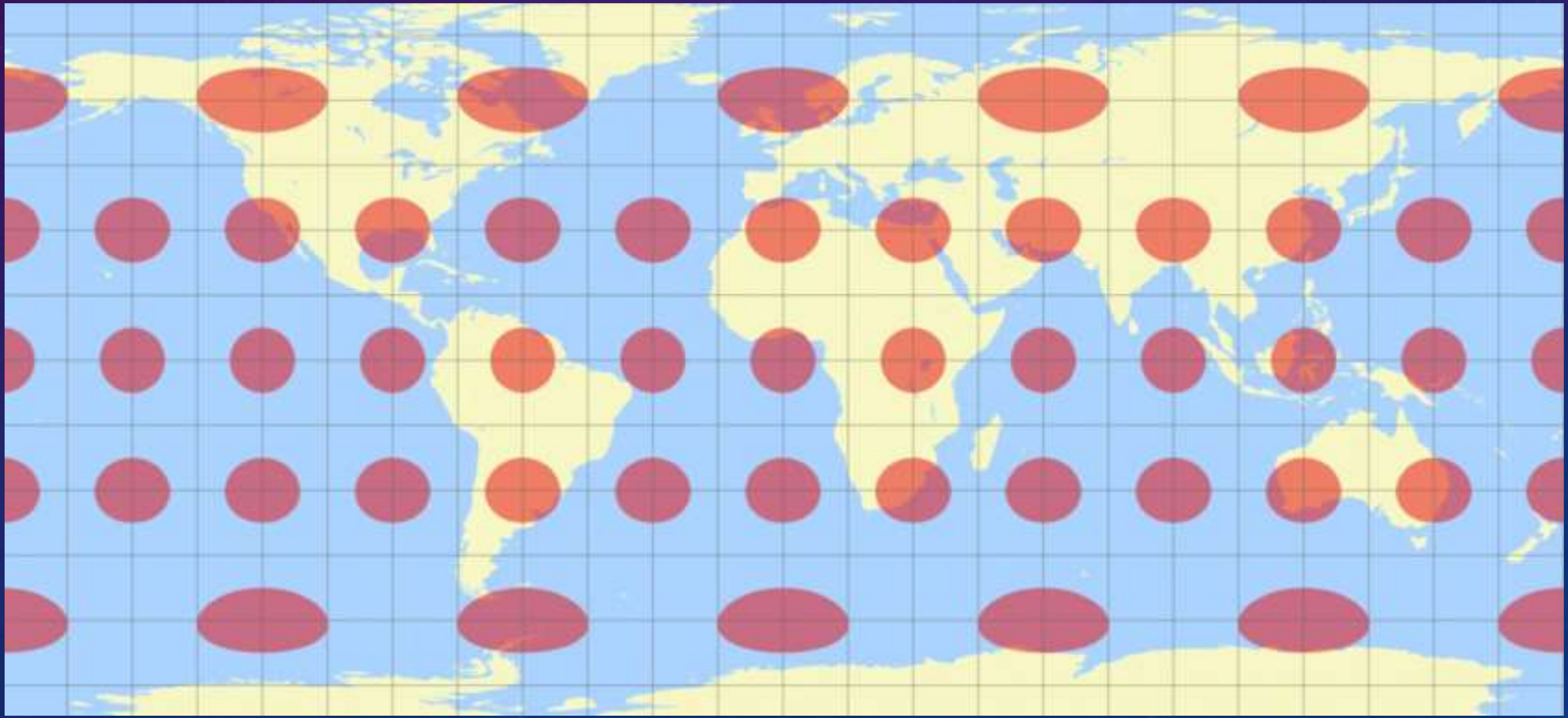
- Piecewise linear
- Piecewise high order





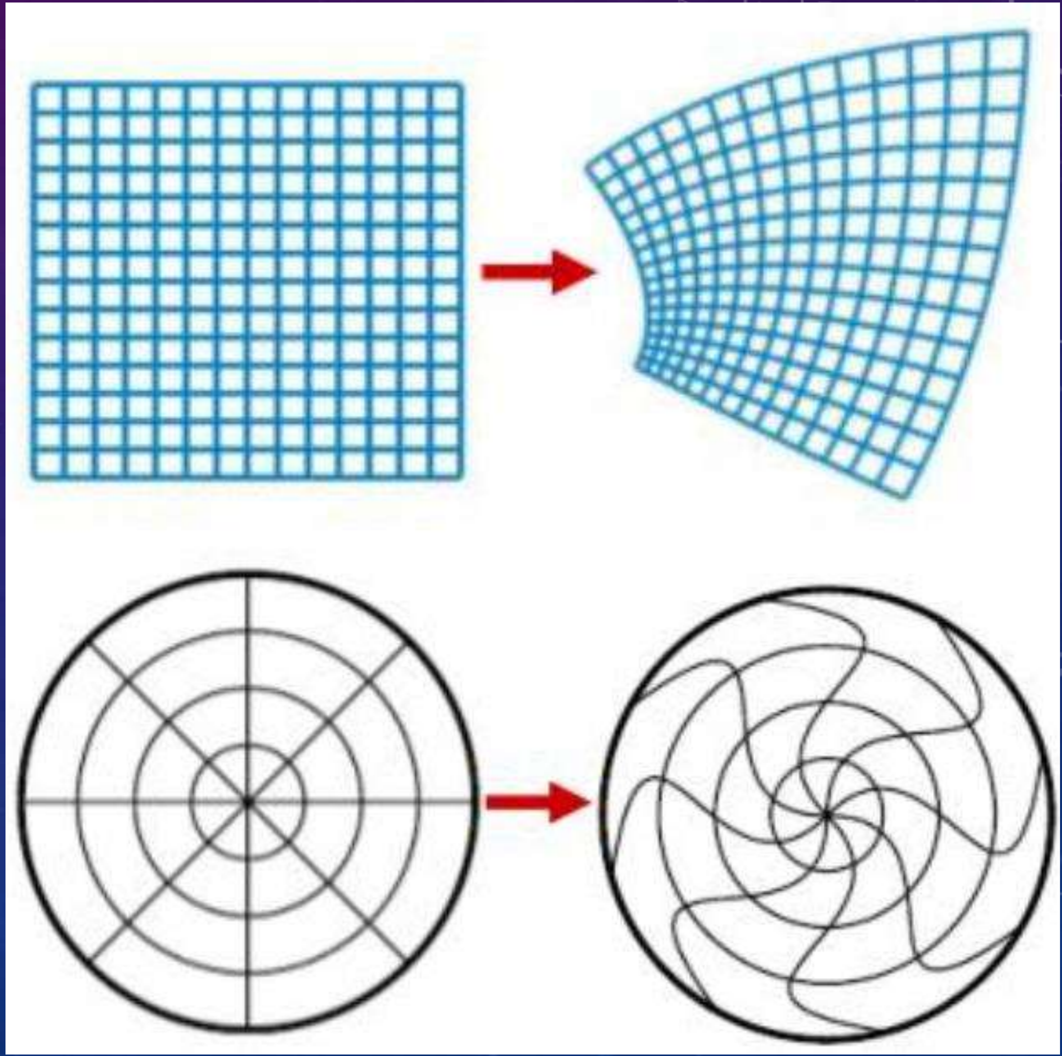
Optimizing parameterization

- Metric/distortion



Optimizing parameterization

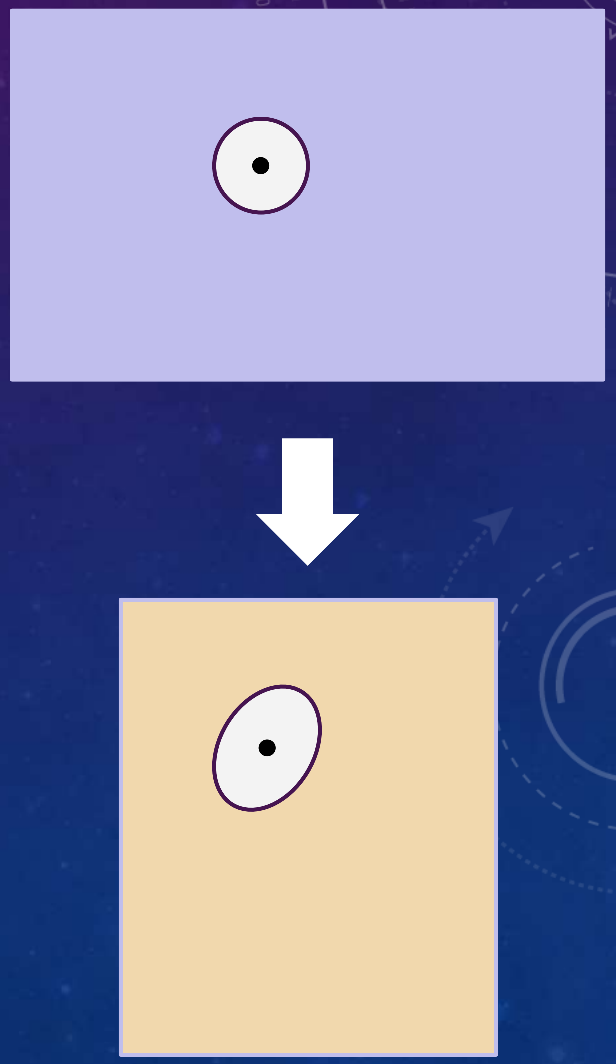
- Angle-preserving $\Leftrightarrow$ conformal
- Area-preserving $\Leftrightarrow$ authalic
- Conformal + authalic $\Leftrightarrow$ isometric



Jacobian

- Jacobian of mapping : $f(u,v) \rightarrow \mathcal{J}f = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}$
- $\mathcal{J}f$ measure the stretch ratio of the local neighborhood

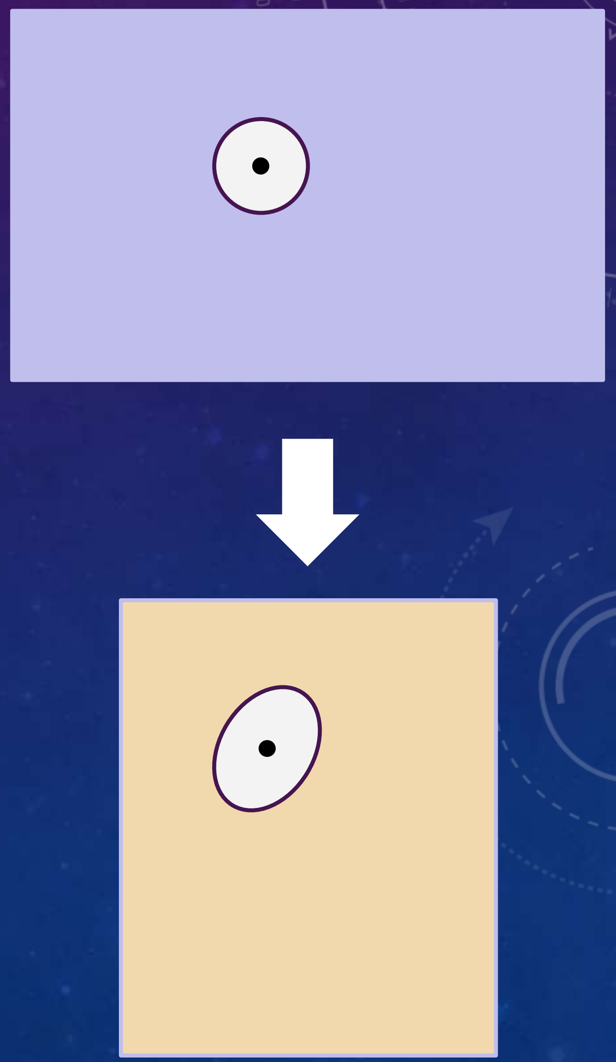
$$\mathcal{J}f = \mathrm{U}\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}\mathrm{V}^{\mathrm{T}}$$



Jacobian

$$\mathcal{J}f = \mathrm{U}\begin{pmatrix}\sigma_1 & 0 \\ 0 & \sigma_2\end{pmatrix}\mathrm{V}^{\mathrm{T}}$$

- Conformal $\sigma_1 = \sigma_2$
- Authalic $\sigma_1 \sigma_2 = 1$
- Isometric $\sigma_1 = \sigma_2 = 1$

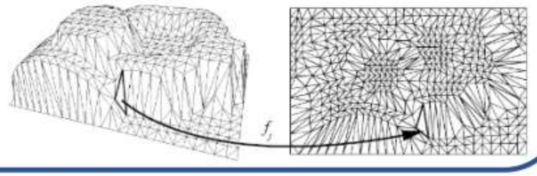


Distortion metric

- **Conformal**
  [Degener et al. 2003]
  $$\frac{\sigma_2}{\sigma_1}$$

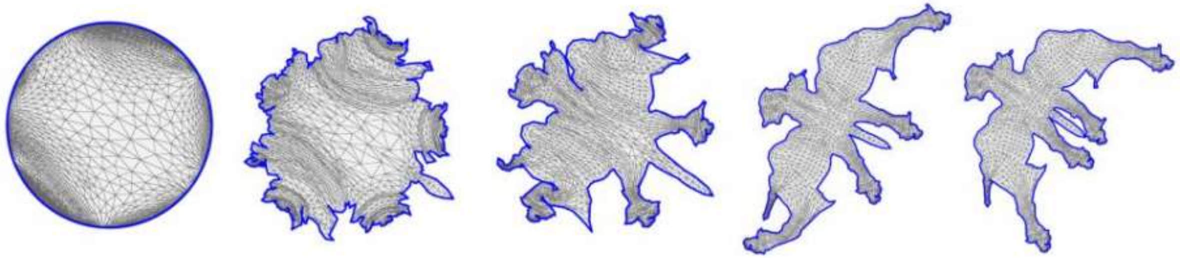
- **Maximal Isometric Distortion**
  [Sorkine et al. 2002]
  $$\max(\sigma_2, \frac{1}{\sigma_1})$$

- **MIPS**
  [Hormann and Greiner 2000]
  $$\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1}$$



- **Isometric**
  [Aigermann et al. 2014]
  $$\sqrt{\sigma_2^2 + \frac{1}{\sigma_1^2}}$$

- **Symmetric Dirichlet energy**
  [Smith and Schaefer 2015]
  $$\sigma_1^2 + \frac{1}{\sigma_1^2} + \sigma_2^2 + \frac{1}{\sigma_2^2}$$



Methods

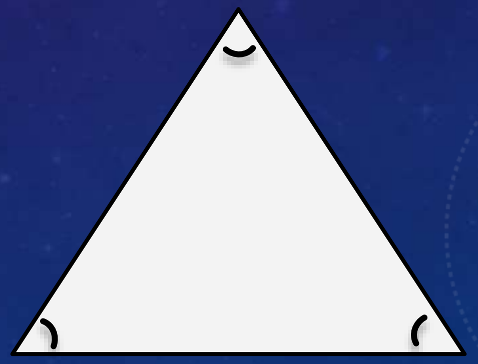
Only low distortions

Angle-based flattening (ABF)

- Sheffer A, de Sturler E. Parameterization of faceted surfaces for meshing using angle-based flattening[J]. Engineering with computers, 2001, 17(3): 326-337.
- Sheffer A, Lévy B, Mogilnitsky M, et al. ABF++: fast and robust angle based flattening[J]. ACM Transactions on Graphics (TOG), 2005, 24(2): 311-330.
- Zayer R, Lévy B, Seidel H P. Linear angle based parameterization[C]//Fifth Eurographics Symposium on Geometry Processing-SGP 2007. Eurographics Association, 2007: 135-141

Angle-Based Flattening (ABF)

- Key observation: the parameterized triangles are uniquely defined by all the angles at the corners of the triangles
  - Calculate angles of each triangle.
  - Use angles to reconstruct (u_i, v_i) coordinates



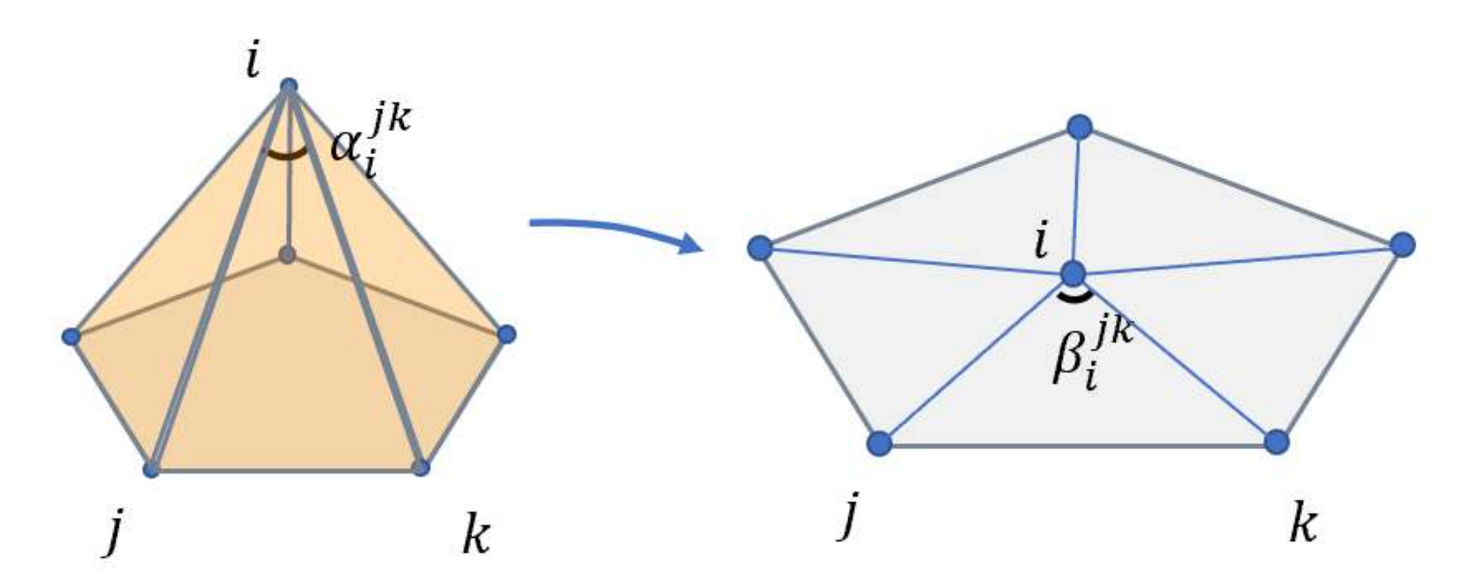
Angle-Based Flattening (ABF)

- Angel preservation
  - Interior vertex:

$$\beta_i^{jk} = \frac{\alpha_i^{jk} \cdot 2\pi}{\sum_i \alpha_i^{jk}}$$

  - Boundary vertex:

$$\beta_i^{jk} = \alpha_i^{jk}$$

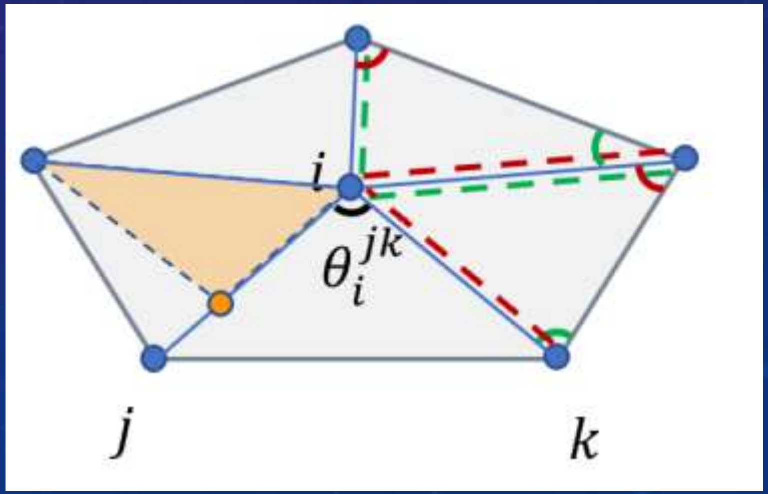


Angle-Based Flattening (ABF)

- Optimization goal:

$$\min_{\theta>0} \sum\nolimits_{ijk} w_{ijk}\{\left(\beta_i^{jk} - \theta_i^{jk}\right)^2 + \left(\beta_j^{ki} - \theta_j^{ki}\right)^2 + \left(\beta_k^{ij} - \theta_k^{kj}\right)^2\}$$

$$\text{s.t.} \begin{cases} \sum_{t_{ijk}\in St(i)} \theta_i^{jk} = 2\pi, \ \forall\ i \text{ interior vertex} \\ \theta_i^{jk} + \theta_j^{ki} + \theta_k^{kj} = \pi, \ \forall\ t_{ijk} \\ \prod_{t_{ijk}\in St(i)} \frac{sin\theta_j^{ki}}{sin\theta_k^{ij}} = 1, \forall\ i \text{ interior vertex} \end{cases}$$



$$\frac{sin\theta_j^{ki}}{sin\theta_k^{ij}} = \frac{l_{ki}}{l_{ij}}$$

Linear ABF

- Reconstruction constraints are nonlinear and hard to solve.
- Initial estimation + estimation error

Let $\theta_i^{jk} = \beta_i^{jk} + \phi_i^{jk}$, then $\log\sin\theta_i^{jk} = \log\sin(\beta_i^{jk} + \phi_i^{jk}) \approx \log\sin\beta_i^{jk} + \phi_i^{jk}\cot\beta_i^{jk}$

$$\prod_{t_{ijk}\in St(i)} sin\theta_j^{ki} = \prod_{t_{ijk}\in St(i)} sin\theta_k^{ij} \Leftrightarrow \log\prod_{t_{ijk}\in St(i)} sin\theta_j^{ki} = \log\prod_{t_{ijk}\in St(i)} sin\theta_k^{ij}$$

$\sum_{t_{ijk}\in St(i)} \log\sin\beta_j^{ki} + \phi_j^{ki}\cot\beta_j^{ki} \approx \sum_{t_{ijk}\in St(i)} \log\sin\beta_k^{ij} + \phi_k^{ij}\cot\beta_k^{ij}$ linear

Linear ABF

- New problem

$$\min_{\theta>0} \sum_{ijk} w_{ijk} \{ \left(\phi_i^{jk}\right)^2 + \left(\phi_j^{ki}\right)^2 + \left(\phi_k^{ij}\right)^2 \}, \qquad \mathrm{s.t.}\ \mathrm{A}\phi = b$$

$$\Rightarrow \begin{pmatrix} D & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \phi \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix}$$

$$\Rightarrow \phi = D^{-1} A^T \left(A D^{-1} A^T\right)^{-1} b$$

Reconstruct parameterization

- Greedy method
  - Construct the triangles one by one using a depth-first traversal
  - Key: for each triangle, given the position of two vertices and the angles, the position of the third vertex can be uniquely derived
- Least squares method
  - An angle-based least squares formulation - Solving a set of linear equations relating angles to coordinates

Greedy method

- Initialize: choose a mesh edge $e_1 = (v_a^1, v_b^1)$ and project v_a^1 to $(0,0)$ and v_b^1 to $(\|e_1\|, 0)$. Push e_1 on the stack S.
- While stack S not empty, pop an edge $e = (v_a, v_b)$.
- For each face $f_i = (v_a, v_b, v_c)$ containing e :
  - If f_i is marked as set, continue.
  - If v_c is not projected, compute its position based on v_a, v_b and the face angles of f_i.
  - Mark f_i as set, push edge $(v_b, v_c)$and (v_c, v_a) on the stack.

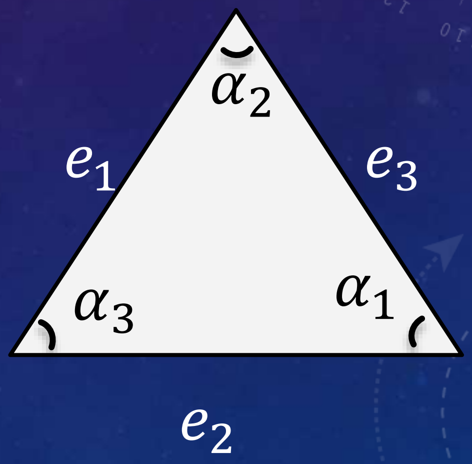
Accumulate numerical error

Least squares method

- The ratio of triangle edge lengths e_1 and e_2 is

$$\frac{\|e_1\|}{\|e_2\|} = \frac{\sin\alpha_1}{\sin\alpha_2}$$

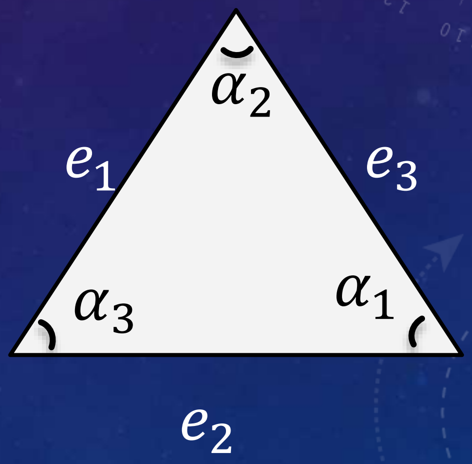
$$\Longrightarrow \vec{e}_1 = -\frac{\sin\alpha_1}{\sin\alpha_2}\begin{pmatrix}\cos\alpha_3 & -\sin\alpha_3 \\ \sin\alpha_3 & \cos\alpha_3\end{pmatrix}\vec{e}_2$$



Least squares method

- For any triangle

$$\vec{e}_1 = -\frac{\sin\alpha_1}{\sin\alpha_2}\begin{pmatrix}\cos\alpha_3 & -\sin\alpha_3 \\ \sin\alpha_3 & \cos\alpha_3\end{pmatrix}\vec{e}_2$$

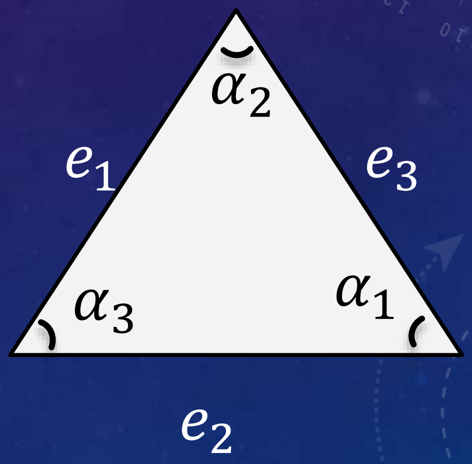


- Two equations per triangle for the u and v coordinates of the vertices.
- The angles of a planar triangulation define it uniquely up to rigid transformation and global scaling.
- Introduce four constraints which eliminate these degrees of freedom
- Fix two vertices sharing a common edge

Least squares method

- Choose a mesh edge $e_1 = \left(v_a^1, v_b^1\right)$ and project v_a^1 to $(0,0)$ and v_b^1 to $(\|e_1\|, 0)$
- Solve following energy to compute positions of other vertices:

$$E = \sum_{ijk} \left\| e_{ij} - M_{ijk} e_{jk} \right\|^2, \qquad e_{ij} = \begin{pmatrix} u_j \\ v_j \end{pmatrix} - \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$



Results

Least-Square conformal mapping (LSCM)

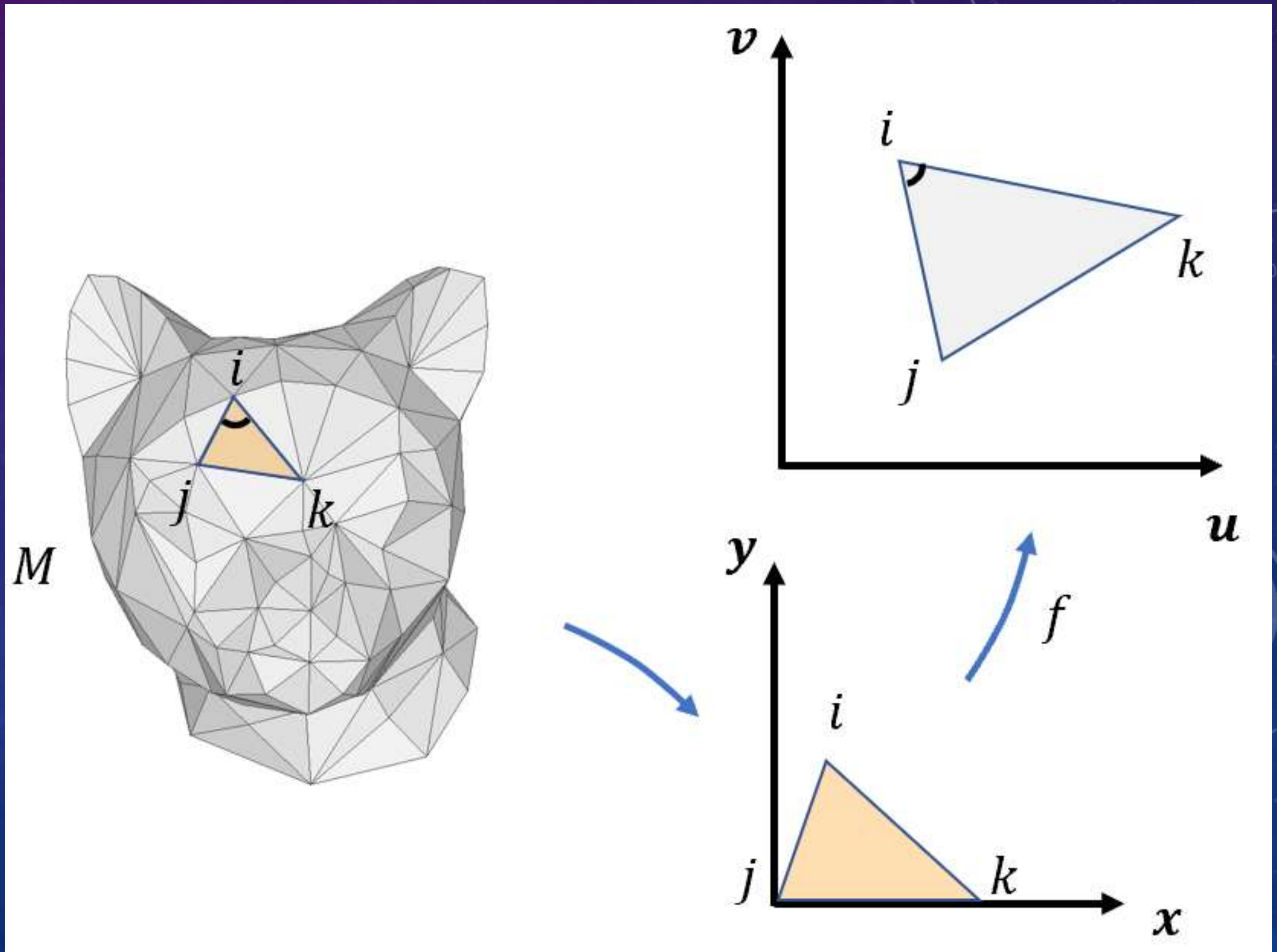
- Lévy B, Petitjean S, Ray N, et al. Least squares conformal maps for automatic texture atlas generation[J]. ACM transactions on graphics (TOG), 2002, 21(3): 362-371.

Least-Square conformal mapping (LSCM)

- Edges of a triangle $\{l_1, l_2, l_3\}$

For triangle t_{ijk}:

$$\begin{pmatrix} du \\ dv \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix}$$

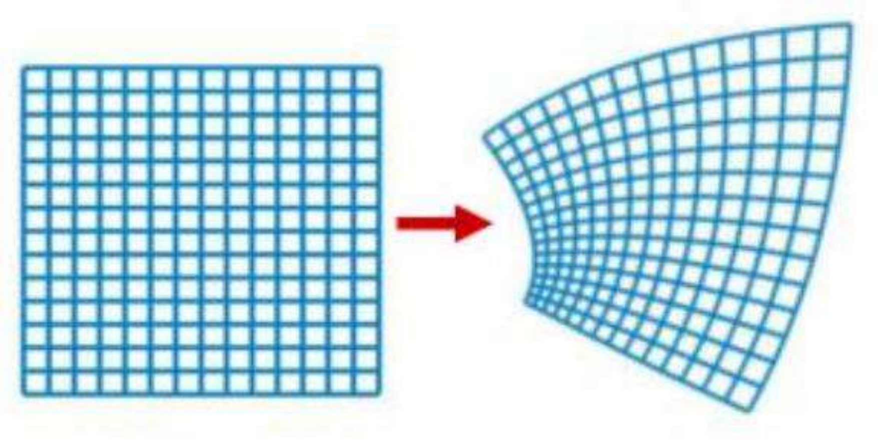


Similar transforms

- For $J_t = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}$ and conformal mapping

$$J_t = s\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

$$\Longrightarrow \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \qquad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}$$



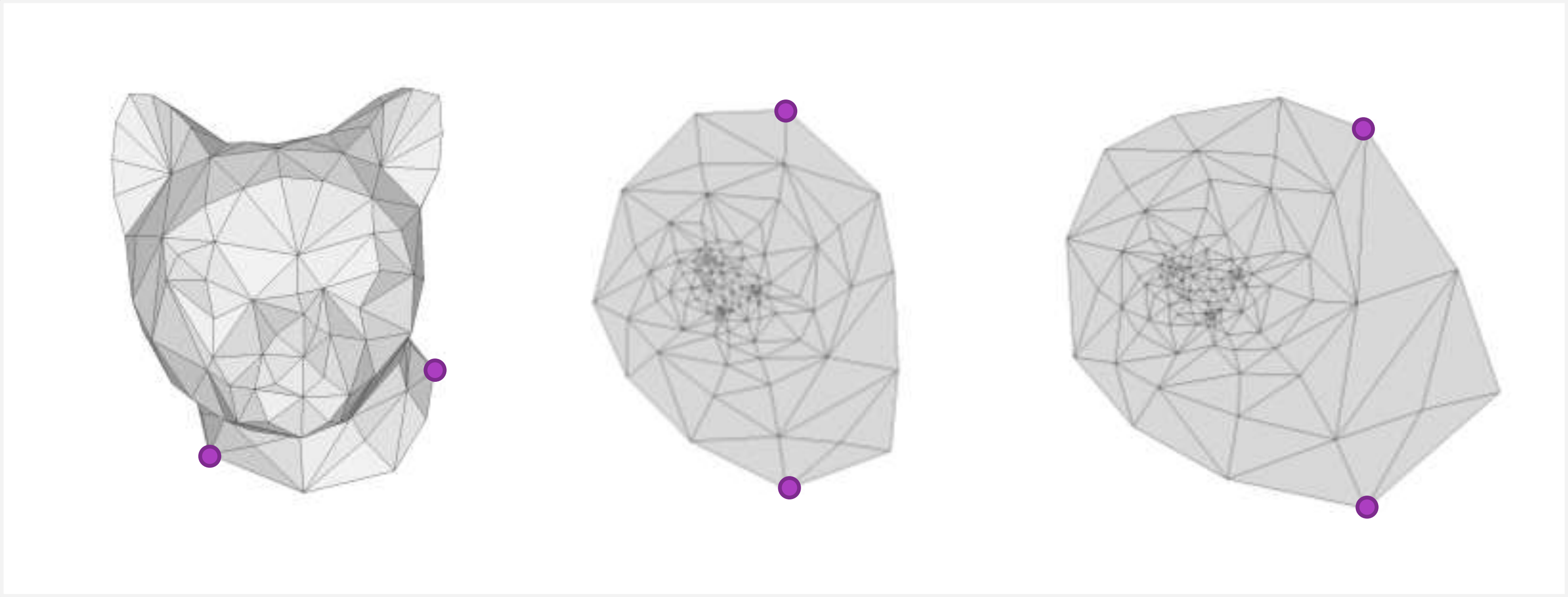
Least-Square conformal mapping (LSCM)

- Least square optimization:

$$E_{LSCM} = \sum_{ijk} A_{ijk} \left(\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 \right)$$

  - Measuring non-conformality
  - It is invariant with respect to arbitrary translations and rotations.
  - E_{LSCM} does not have a unique minimizer.
  - Fixing at least two vertices. Significantly affect the results

Least-Square conformal mapping (LSCM)

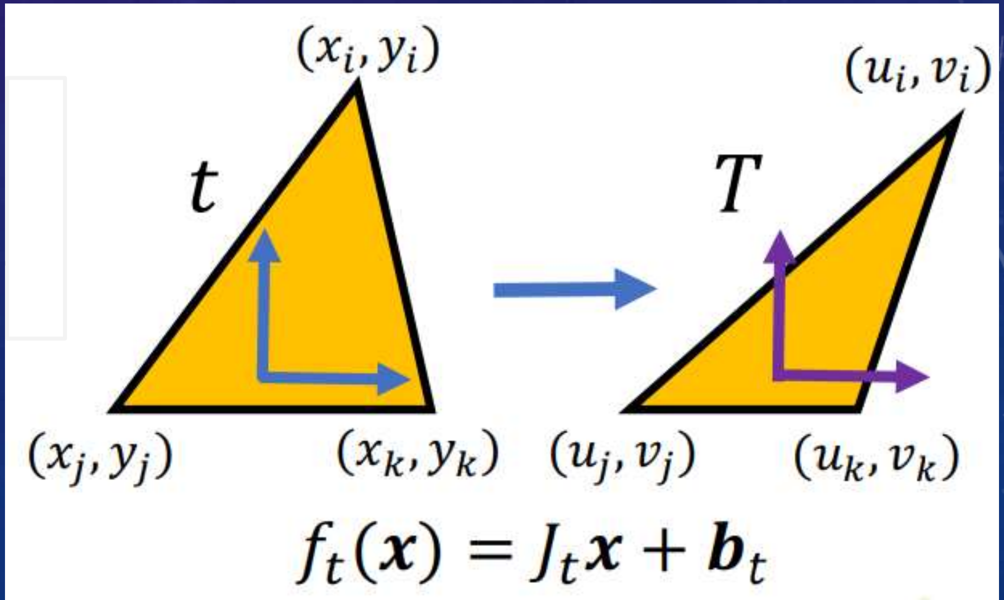


As-rigid-as-possible (ARAP)

- Liu L, Zhang L, Xu Y, et al. A local/global approach to mesh parameterization [C]//Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 2008, 27(5): 1495-1504
- Homework #3, ARAP + ASAP, Deadline 2024.03.31

Formulation

- Variables: parameterization coordinate (u_i, v_i) and target transformations L_t
- Energy: $E(u, v, L) = \sum_t A_t \|J_t(u, v) - L_t\|_F^2$
- Target transformation L_t
  - Isometric mapping: rotation matrix
  - Conformal mapping: similar matrix



Local-global solver

$$E(u, v, L) = \sum_t A_t \|J_t(u, v) - L_t\|_F^2$$

Alternatively optimization

- Local step: fix (u, v), optimize L_t.
- Global step: fix L_t, optimize (u, v)

Local-global solver

$$E(u,v,L) = \sum_t A_t \|J_t(u,v) - L_t\|_F^2$$

Alternatively optimization

- Local step: fix (u,v), optimize L_t (SVD).
- Global step: fix L_t, optimize (u,v) (quadratic energy)

Global solver

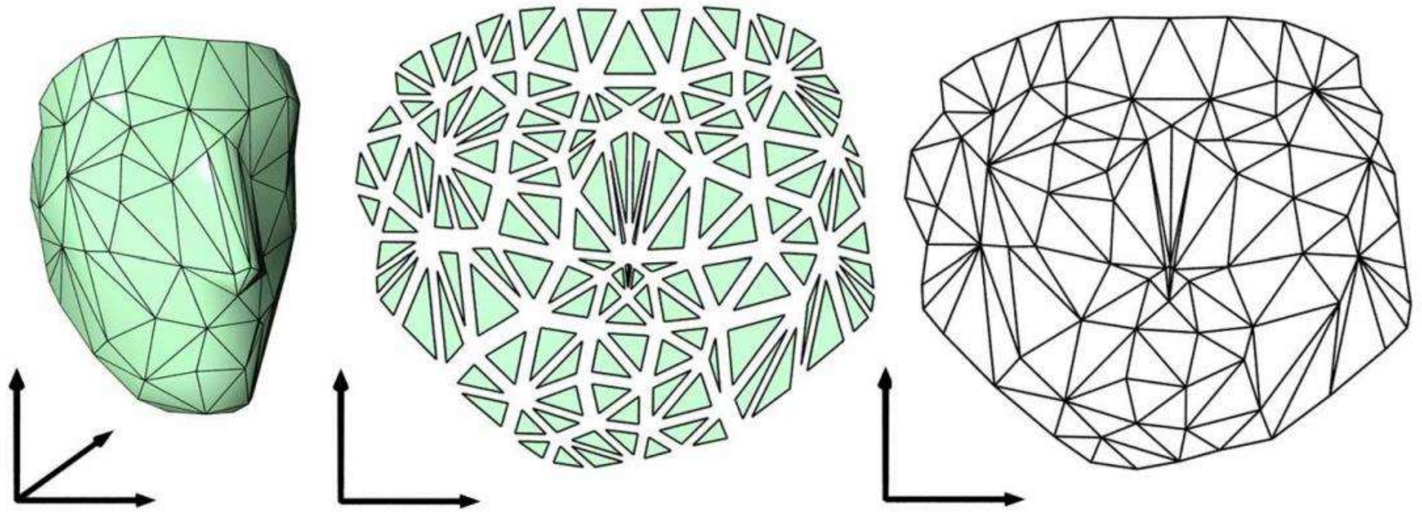
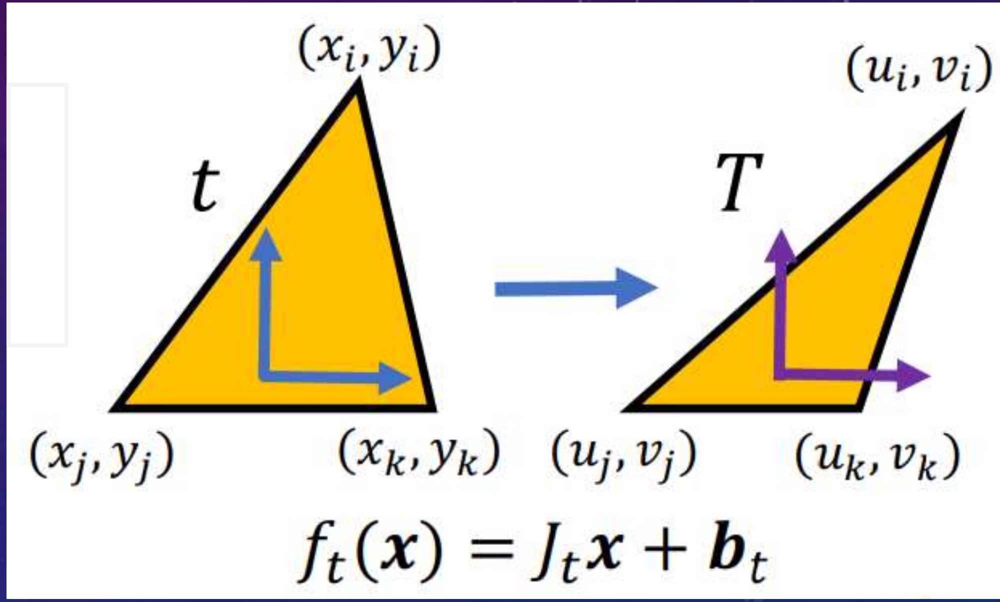


Figure 2: *Parameterizing a mesh by aligning locally flattened triangles. (Left) Original 3D mesh; (middle) flattened triangles; (right) 2D parameterization.*

Directly optimizing

$$\begin{pmatrix} u_j - u_i & u_k - u_j \\ v_j - v_i & v_k - v_j \end{pmatrix} = J_t \begin{pmatrix} x_j - x_i & x_k - x_j \\ y_j - y_i & y_k - y_j \end{pmatrix}$$

$$\Longrightarrow J_{ijk} = \begin{pmatrix} u_j - u_i & u_k - u_j \\ v_j - v_i & v_k - v_j \end{pmatrix} \begin{pmatrix} x_j - x_i & x_k - x_j \\ y_j - y_i & y_k - y_j \end{pmatrix}^{-1}$$



Jacobian (linear function of uv)

$$J_{ijk} = \begin{pmatrix} u_j - u_i & u_k - u_j \\ v_j - v_i & v_k - v_j \end{pmatrix} \begin{pmatrix} x_j - x_i & x_k - x_j \\ y_j - y_i & y_k - y_j \end{pmatrix}^{-1}$$

$$\Rightarrow J_{ijk}^T = \begin{pmatrix} x_j - x_i & y_j - y_i \\ x_k - x_j & y_k - y_j \end{pmatrix}^{-1} \begin{pmatrix} u_j - u_i & v_j - v_i \\ u_k - u_j & v_k - v_j \end{pmatrix}$$

$$\Rightarrow J_{ijk}^T = \frac{1}{2A_{ijk}} \begin{pmatrix} y_k - y_j & y_i - y_j \\ x_j - x_k & x_j - x_i \end{pmatrix} \begin{pmatrix} u_j - u_i & v_j - v_i \\ u_k - u_j & v_k - v_j \end{pmatrix}$$

Jacobian (linear function of uv)

$$J_{ijk}^T = \frac{1}{2A_{ijk}} \begin{pmatrix} y_k - y_j & y_i - y_j \\ x_j - x_k & x_j - x_i \end{pmatrix} \begin{pmatrix} u_j - u_i & v_j - v_i \\ u_k - u_j & v_k - v_j \end{pmatrix}$$

$$vec(J_{ijk}^T) = \frac{1}{2A_{ijk}} \begin{pmatrix} y_j - y_k & y_k - y_i & y_i - y_j & & & \\ x_k - x_j & x_i - x_k & x_j - x_i & & & \\ & & & y_j - y_k & y_k - y_i & y_i - y_j \\ & & & x_k - x_j & x_i - x_k & x_j - x_i \end{pmatrix} \begin{pmatrix} u_i \\ u_j \\ u_k \\ v_i \\ v_j \\ v_k \end{pmatrix}$$

Singular value

$$J_{ijk}^T J_{ijk} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a^2 + c^2 & ab + cd \\ ab + cd & b^2 + d^2 \end{pmatrix}$$

Let $a' = \frac{a+d}{2}, c' = \frac{a-d}{2}, b' = \frac{c-b}{2}, d' = \frac{c+b}{2}$, then

$$\sigma = \sqrt{a'^2 + b'^2} + \sqrt{c'^2 + d'^2}, \qquad \tau = \left| \sqrt{a'^2 + b'^2} - \sqrt{c'^2 + d'^2} \right|$$

Distortion measure

Dirichlet: $f_D(u,v) = \sum_{ijk} A_{ijk} \left\| J_{ijk} \right\|_F^2 = \sum_{ijk} A_{ijk} \left(\sigma_{ijk}^2 + \tau_{ijk}^2\right)$

ARAP: $f_A(u,v) = \sum_{ijk} A_{ijk} \left\| J_{ijk} - R_{ijk} \right\|_F^2 = \sum_{ijk} A_{ijk} \left(\left(\sigma_{ijk} - 1\right)^2 + \left(\tau_{ijk} - 1\right)^2 \right)$

Symmetric Dirichlet: $f_{SD}(u,v) = \sum_{ijk} A_{ijk} \left(\left\| J_{ijk} \right\|_F^2 + \left\| J_{ijk} \right\|_F^{-2} \right)$

$$= \sum_{ijk} A_{ijk} \left(\sigma_{ijk}^2 + \sigma_{ijk}^{-2} + \tau_{ijk}^2 + \tau_{ijk}^{-2}\right)$$

Optimization problem

$$\min_{u,v} \sum\nolimits_{ijk} A_{ijk}\, \mathcal{D}(J_{ijk}(u,v)) \triangleq \min_{u,v} E(u,v)$$

1. Initial point (u_0, v_0), iter $n = 0$
2. Descent direction $\langle p, \nabla E \rangle < 0$
3. Step size $\min_{\alpha} E((u_n, v_n) + \alpha p)$
4. Update $(u_{n+1}, v_{n+1}) = (u_n, v_n) + \alpha p, n = n + 1$

Method

$$\min_{\delta} E(uv + \delta)$$

Taylor expansion: $E(uv + \delta) = E(uv) + \nabla E(uv)^T \delta + \frac{1}{2} \delta^T H \delta + \cdots$

- First order: descent steps by preconditioning the gradient
- Second order: Newton-type methods uses the energy Hessian

First-order method (slower convergence)

- Kovalsky SZ, Galun M, Lipman Y. Accelerated quadratic proxy for geometric optimization. ACM Transactions on Graphics (TOG). 2016 Jul 11;35(4):1-1.
- Rabinovich M, Poranne R, Panozzo D, Sorkine-Hornung O. Scalable locally injective mappings. ACM Transactions on Graphics (TOG). 2017 Apr 14;36(4):1.
- Zhu Y, Bridson R, Kaufman DM. Blended cured quasi-newton for distortion optimization. ACM Transactions on Graphics (TOG). 2018 Jul 30;37(4):1-4.

Second-order method (high computation)

- Shtengel A, Poranne R, Sorkine-Hornung O, Kovalsky SZ, Lipman Y. Geometric optimization via composite majorization. ACM Trans. Graph.. 2017 Jul 1;36(4):38-1.
- Golla B, Seidel HP, Chen R. Piecewise linear mapping optimization based on the complex view. InComputer Graphics Forum 2018 Oct (Vol. 37, No. 7, pp. 233-243).
- Liu L, Ye C, Ni R, Fu XM. Progressive parameterizations. ACM Trans. Graph.. 2018 Jul 30;37(4):41.