Domain with metric

Ω = 2D square

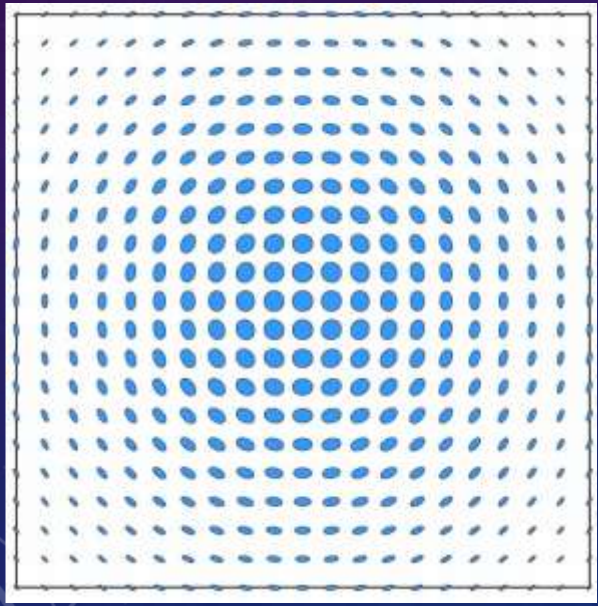
$M(p) : 2 \times 2$

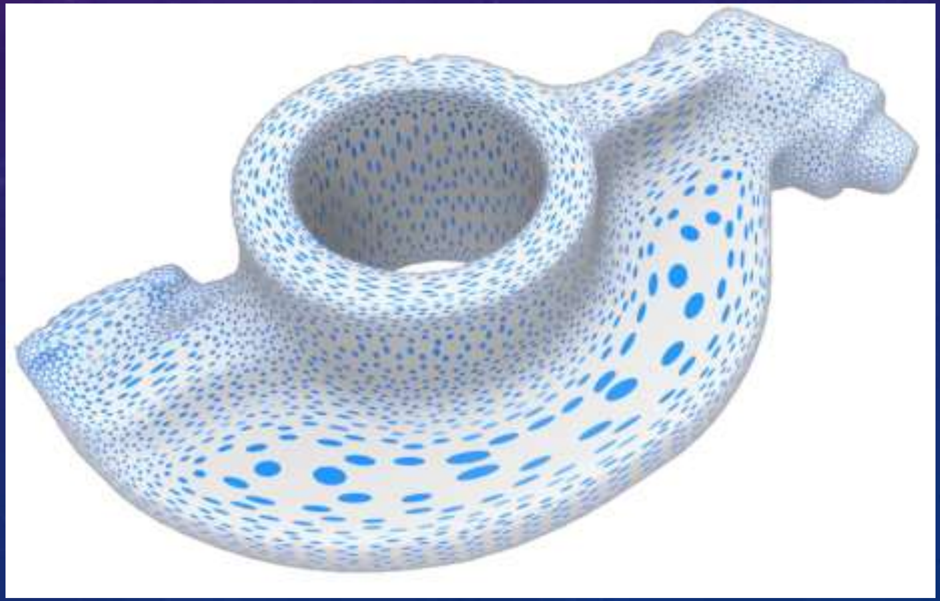
Ω = 3D surface

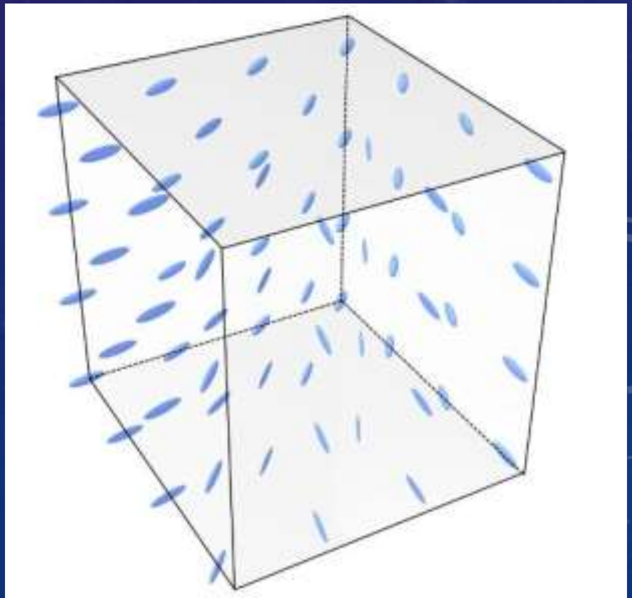
$M(p) : 2 \times 2$

Ω = 3D volume

$M(p) : 3 \times 3$

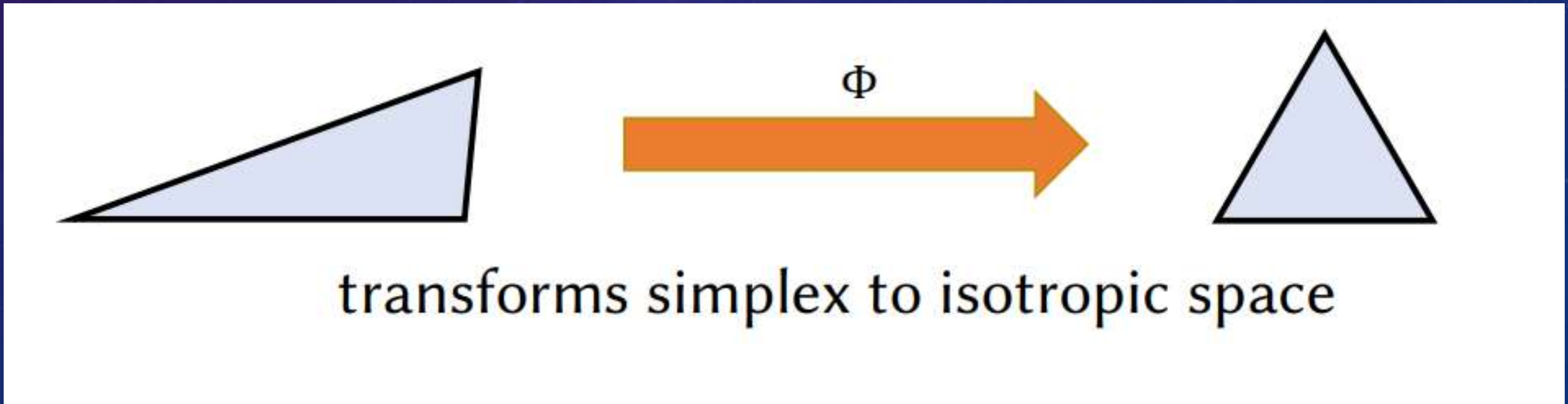






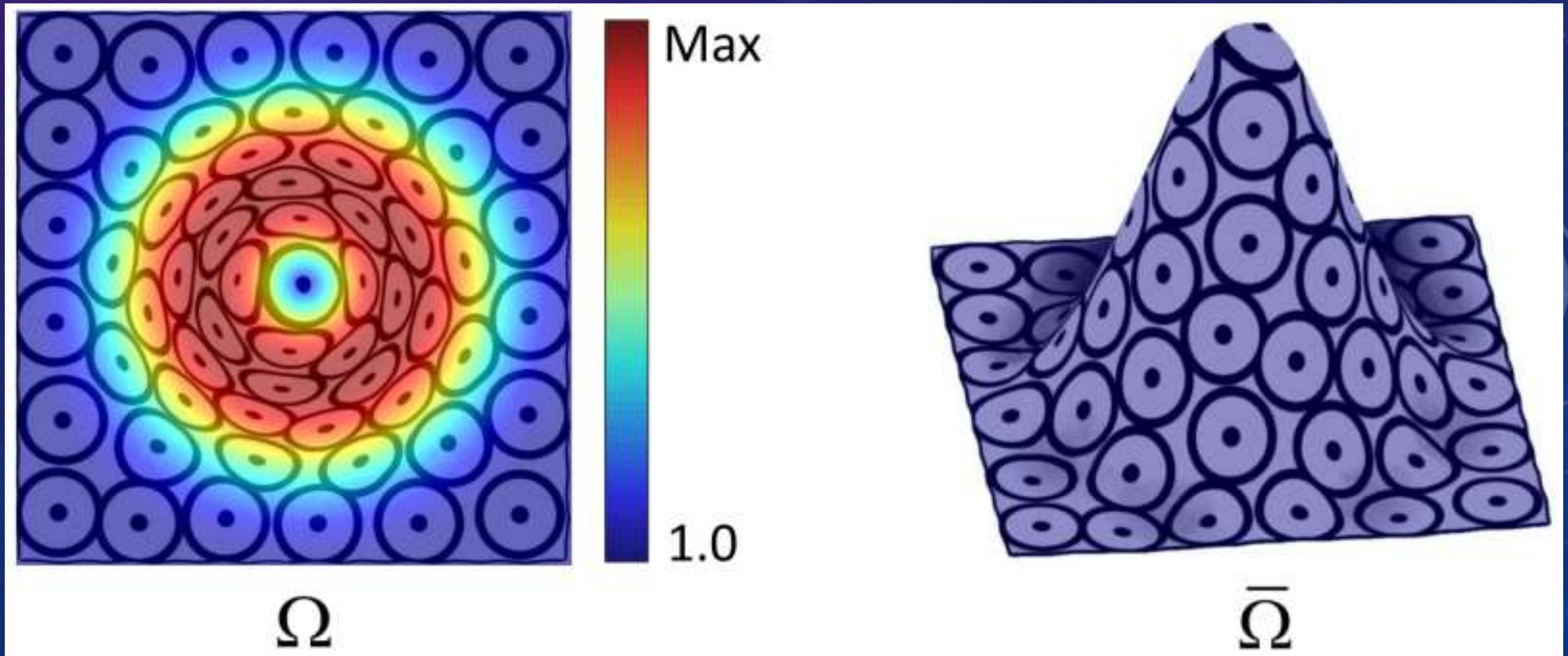
Anisotropic remeshing

- Eigen-decomposition $M(x) = U(x)\Lambda(x)U^T(x)$
- Transformation $\phi = \Lambda^{1/2}(x)U^T(x)$
- Anisotropic remeshing - all edge lengths with metric are as equal as possible



High-dim isometric embedding

- For an arbitrary metric field $M(x)$ defined on the surface or volume $\Omega \subset \mathbb{R}^m$, there exists a high-d space $\mathbb{R}^n$ $(m < n)$ in which Ω can be embedded with Euclidean metric as $\overline{\Omega} \subset \mathbb{R}^n$.



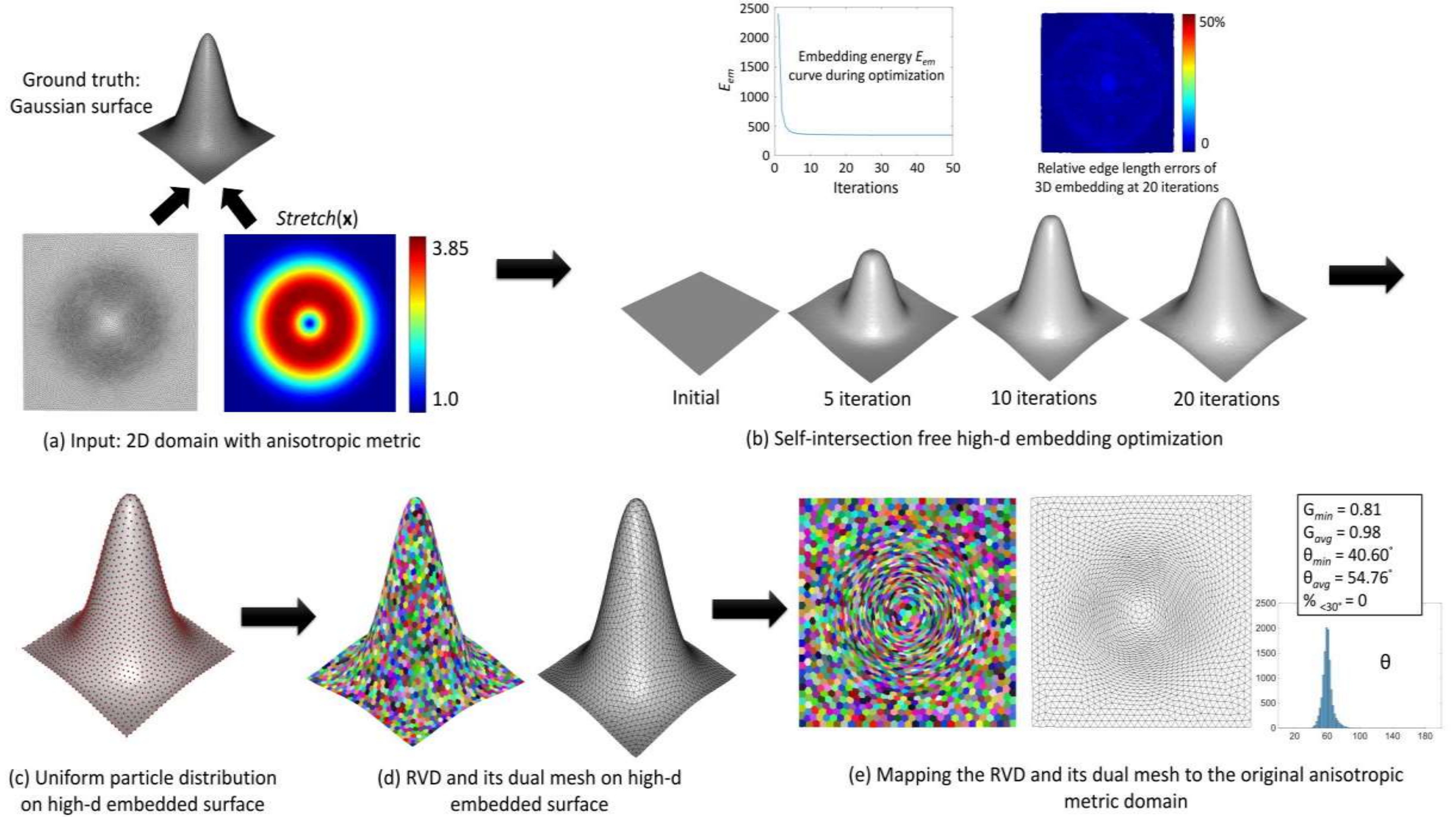
Computing high-dim embedding

local-global solver: $E_{embedding} + \mu E_{smoothing}$

$E_{embedding}$: measure the rigidity, like ARAP

$E_{smoothing}$: measure the smoothness of the embedding

A 3D embedding from a 2D domain with an anisotropic metric



Ground truth:
Gaussian surface
Stretch(x)
3.85
1.0
(a) Input: 2D domain with anisotropic metric
2500
2000
1500
1000
500
0
0 10 20 30 40 50
Iterations
E_{em}
Embedding energy E_{em}
curve during optimization
50%
0
Relative edge length errors of
3D embedding at 20 iterations
Initial
5 iteration
10 iterations
20 iterations
(b) Self-intersection free high-d embedding optimization
(c) Uniform particle distribution
on high-d embedded surface
(d) RVD and its dual mesh on high-d
embedded surface
G_{min} = 0.81
G_{avg} = 0.98
θ_{min} = 40.60°
θ_{avg} = 54.76°
$\%_{<30°}$ = 0
θ
20 60 100 140 180
(e) Mapping the RVD and its dual mesh to the original anisotropic
metric domain

Mesh Deformation I

USTC, 2024 Spring

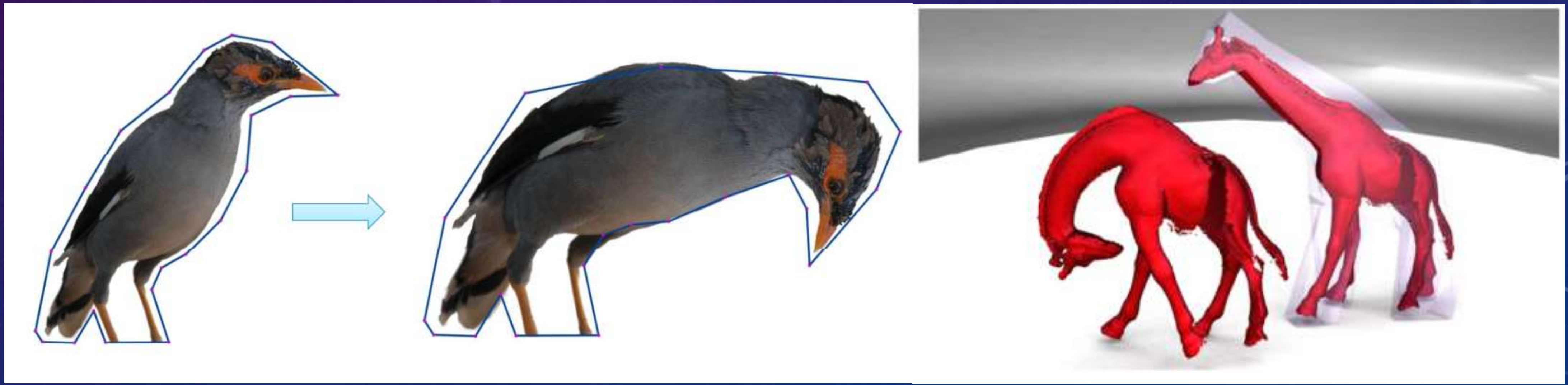
Qing Fang, fq1208@mail.ustc.edu.cn

https://qingfang1208.github.io/

Deformation

Motivation

- Easy modeling – generate new shapes by deforming existing ones

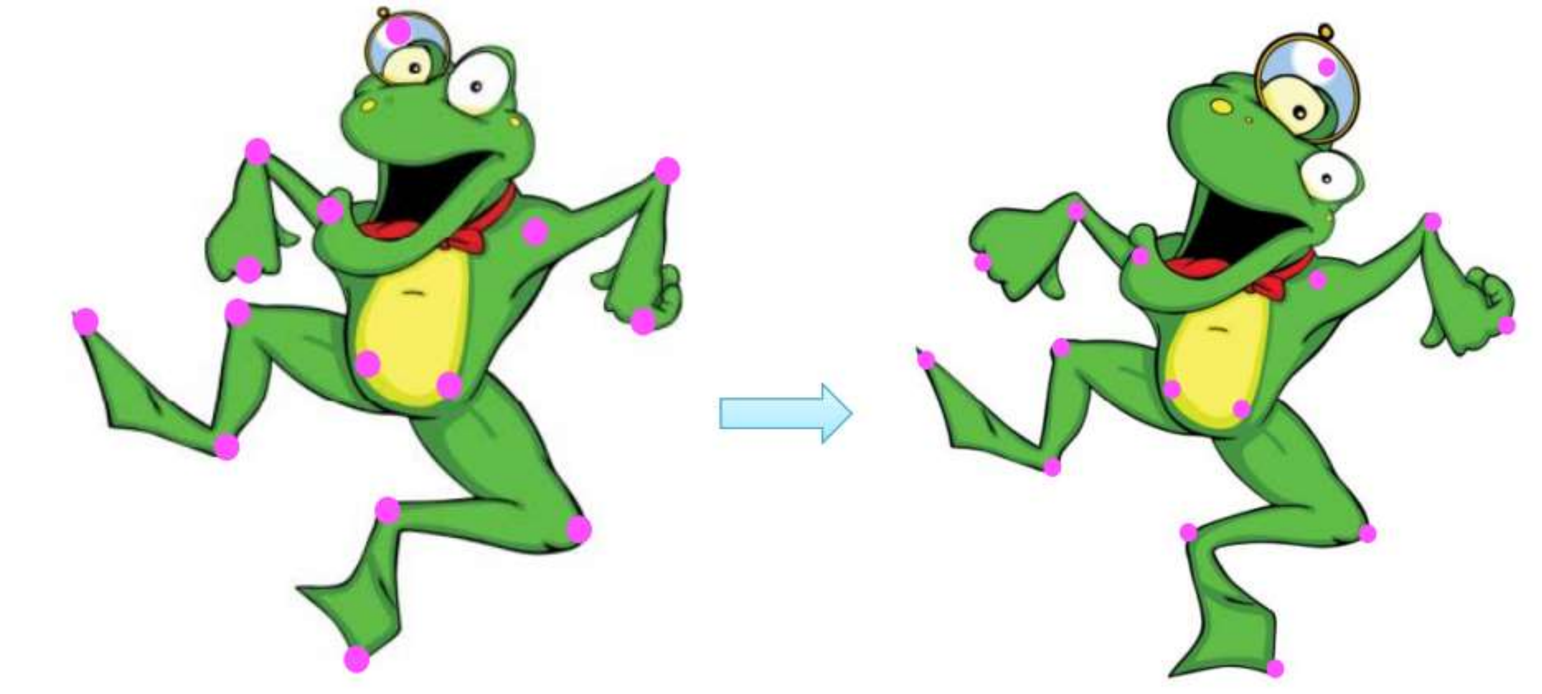


Motivation

- Easy modeling – generate new shapes by deforming existing ones
- Character - posing for animation

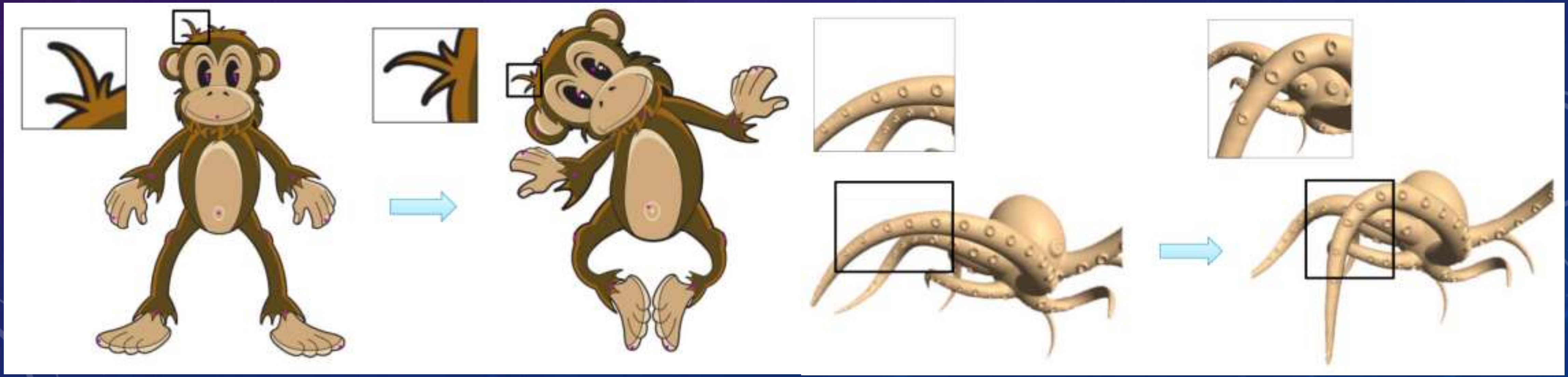
Challenges

- User says as little as possible… (algorithm deduces the rest)



Challenges

- User says as little as possible… (algorithm deduces the rest)
- “Intuitive” deformation (global change + local detail preservation)



Challenges

- User says as little as possible… (algorithm deduces the rest)
- "Intuitive" deformation (global change + local detail preservation)
- Efficient

Definition

- The deformation of a given surface S into the desired surface S'

$$S' = \{p + d(p), p \in S\},\ d(p) \text{ a displacement on each vertex } p \in S$$

- The user controls the deformation by
  - Prescribing displacements for a set of vertices $p_i \in \mathrm{H} \subset S$.
  - Constraining certain parts F stay fixed.
- The main question: determine the displacements for vertices in $S \backslash (H \cup F)$.

Handles

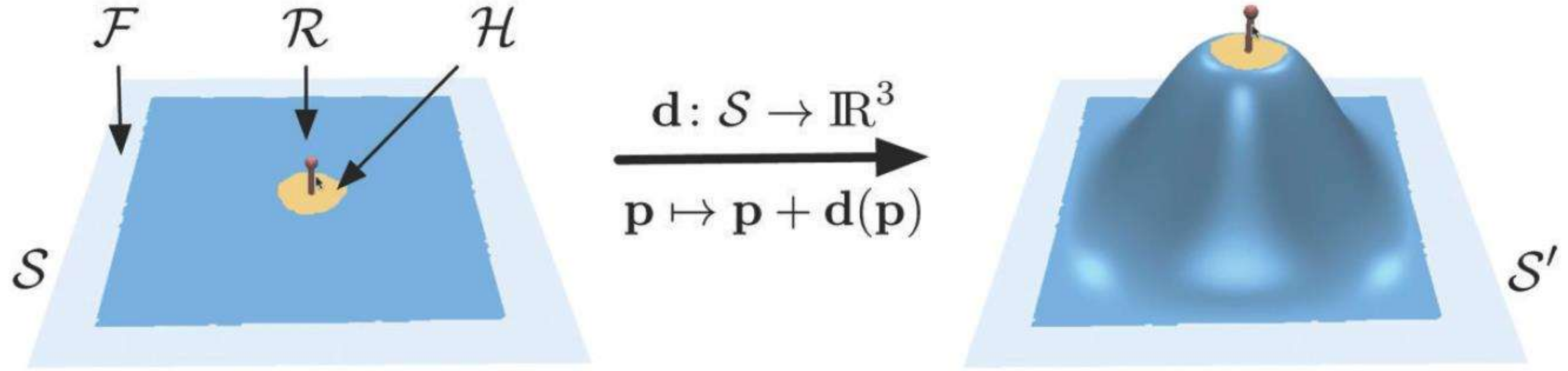
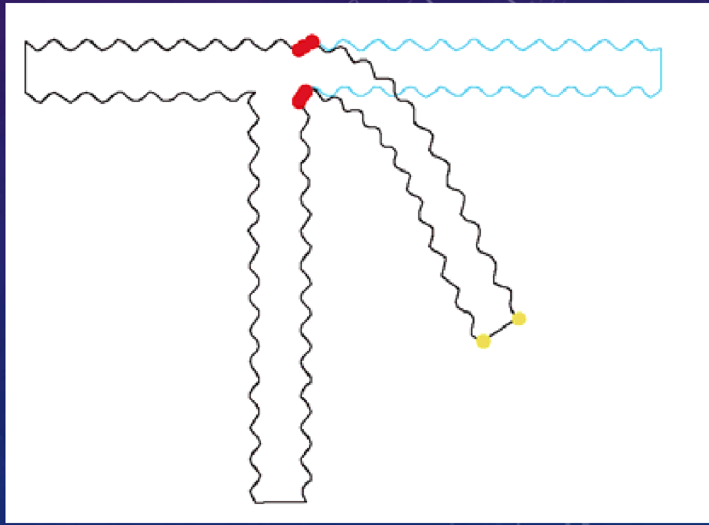


Figure 9.1. A given surface $\mathcal{S}$ is deformed into $\mathcal{S}'$ by a displacement function $\mathbf{d}(\mathbf{p})$. The user controls the deformation by moving a handle region $\mathcal{H}$ (yellow) and keeping the region $\mathcal{F}$ (gray) fixed. The unconstrained deformation region $\mathcal{R}$ (blue) should deform in an intuitive, physically-plausible manner.

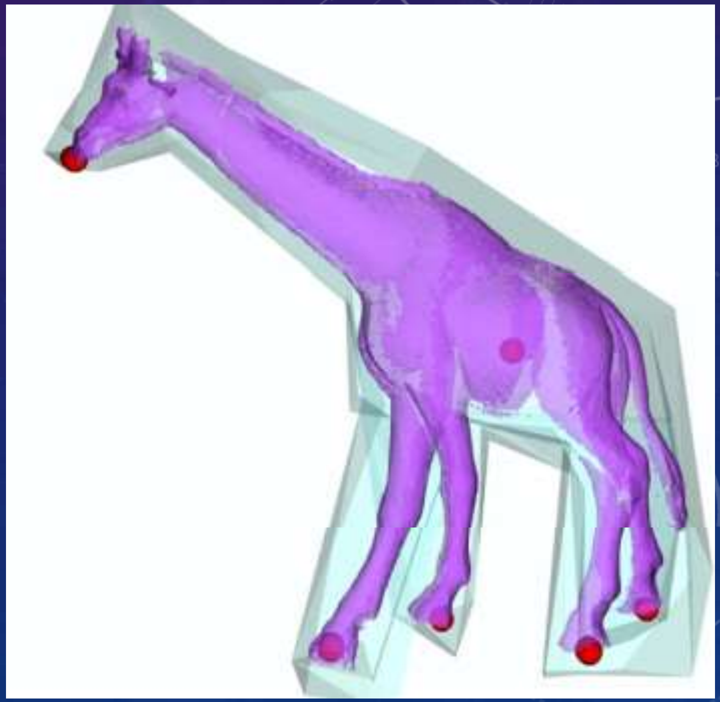
Methods

- Surface-based deformations
  - Shape is empty shell (curve for 2D, surface for 3D)
  - Deformation only defined on shape
  - Deformation coupled with shape representation



Methods

- Surface-based deformations
- Space deformations
  - Shape is volumetric (planar domain for 2D, polyhedral domain for 3D)
  - Deformation defined in neighborhood of shape
  - Can be applied to any shape representation



Surface-based deformations

- Physically-based deformation
- Multi-Scale deformation
- Differential coordinates
- As-Rigid-As-Possible surface deformation

Shell deformation energy

- Stretching
  - Change of local distances
  - Captured by 1st fundamental form
- Bending
  - Change of local curvature
  - Captured by 2nd fundamental form
- 1st and 2nd fundamental forms determine a surface up to rigid motion.

$$\boxed{\int_\Omega k_s \left\| \mathbf{I} - \bar{\mathbf{I}} \right\|^2}$$

$$\mathbf{I} = \begin{bmatrix} \mathbf{x}_u^T \mathbf{x}_u & \mathbf{x}_u^T \mathbf{x}_v \\ \mathbf{x}_v^T \mathbf{x}_u & \mathbf{x}_v^T \mathbf{x}_v \end{bmatrix}$$

$$\boxed{\int_\Omega k_b \left\| \mathbf{II} - \bar{\mathbf{II}} \right\|^2}$$

$$\mathbf{II} = \begin{bmatrix} \mathbf{x}_{uu}^T \mathbf{n} & \mathbf{x}_{uv}^T \mathbf{n} \\ \mathbf{x}_{vu}^T \mathbf{n} & \mathbf{x}_{vv}^T \mathbf{n} \end{bmatrix}$$

Physically-based deformation

- Nonlinear stretching & bending energies

$$\int_{\Omega} k_s \underbrace{\left\|\mathbf{I}-\mathbf{I}'\right\|^2}_{\text{stretching}} + k_b \underbrace{\left\|\mathbf{II}-\mathbf{II}'\right\|^2}_{\text{bending}} \, \mathrm{d}u\mathrm{d}v$$

- Linearize terms → quadratic energy

$$\int_{\Omega} k_s \underbrace{\left(\left\|\mathbf{d}_u\right\|^2 + \left\|\mathbf{d}_v\right\|^2\right)}_{\text{stretching}} + k_b \underbrace{\left(\left\|\mathbf{d}_{uu}\right\|^2 + 2\left\|\mathbf{d}_{uv}\right\|^2 + \left\|\mathbf{d}_{vv}\right\|^2\right)}_{\text{bending}} \mathrm{d}u\mathrm{d}v$$

Physically-based deformation

- Minimize linearized bending energy

$$E(\mathbf{d}) = \int_{\mathcal{S}} \|\mathbf{d}_{uu}\|^2 + 2\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \, dudv \rightarrow \min$$

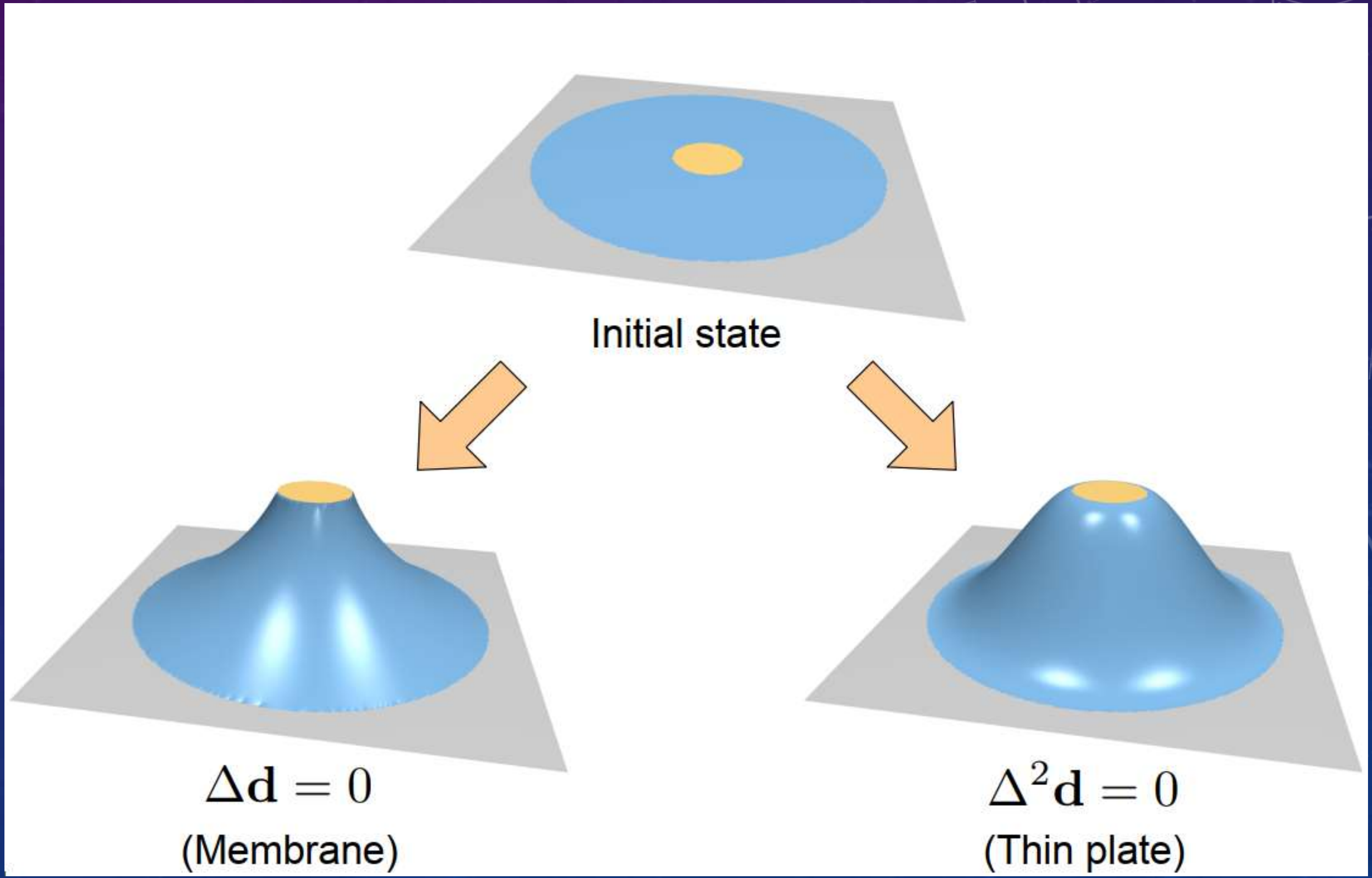
$$\boxed{f(x) \rightarrow \min}$$

- Variational calculus → Euler-Lagrange PDE

$$\Delta^2 \mathbf{d} := \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv} = 0 \qquad \boxed{f'(x) = 0}$$

Deformation energies

"Best" deformation that satisfies constraints



Surface-based deformations

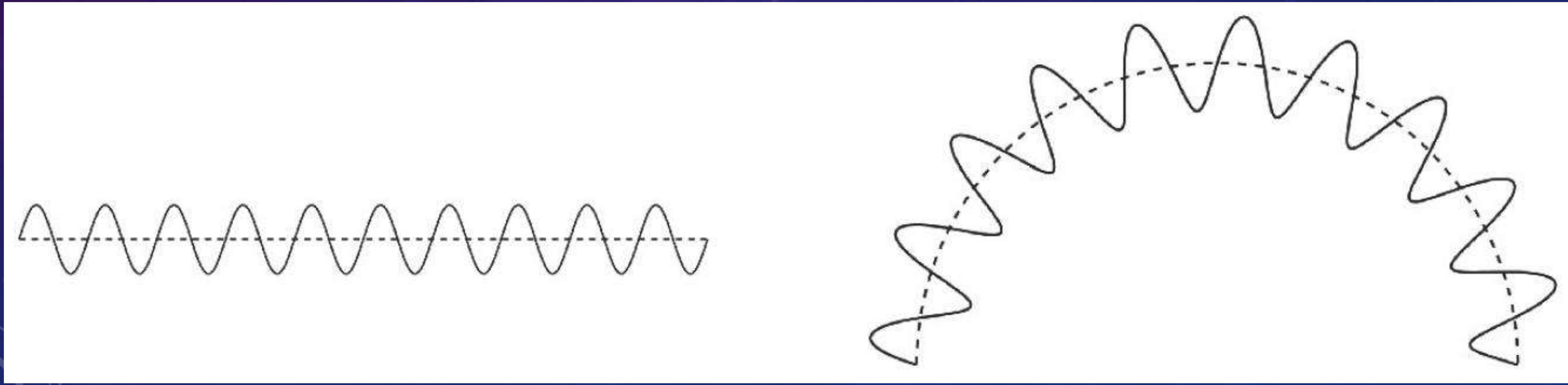
- Physically-based deformation
- Multi-Scale deformation
- Differential coordinates
- As-Rigid-As-Possible surface deformation

Multi-scale deformations

- Main idea: decompose the object into two frequency bands using the smoothing and fairing techniques.
  - The low frequencies correspond to the smooth global shape;
  - The high frequencies correspond to the fine-scale details.

Multi-scale deformations

- Goal: deform the low frequencies (global shape) while preserving the high-frequency details



Framework

- $S = B \oplus D$
- $B \rightarrow B'$
- $S' = B' \oplus D$

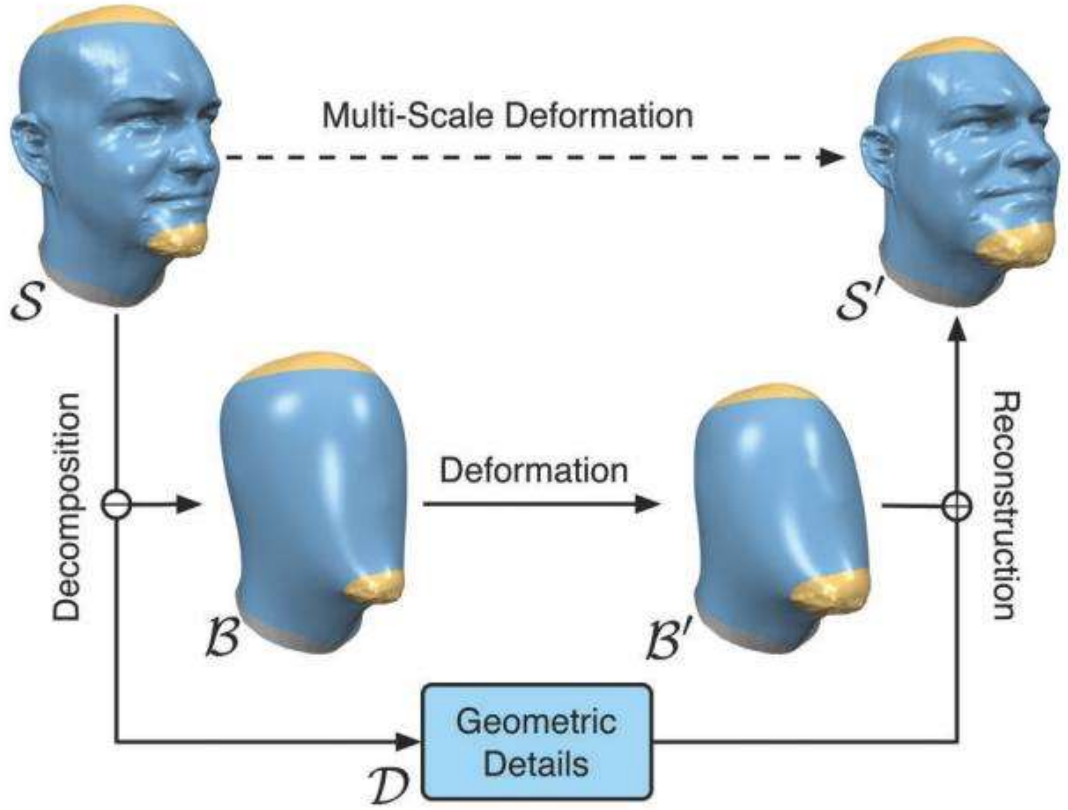
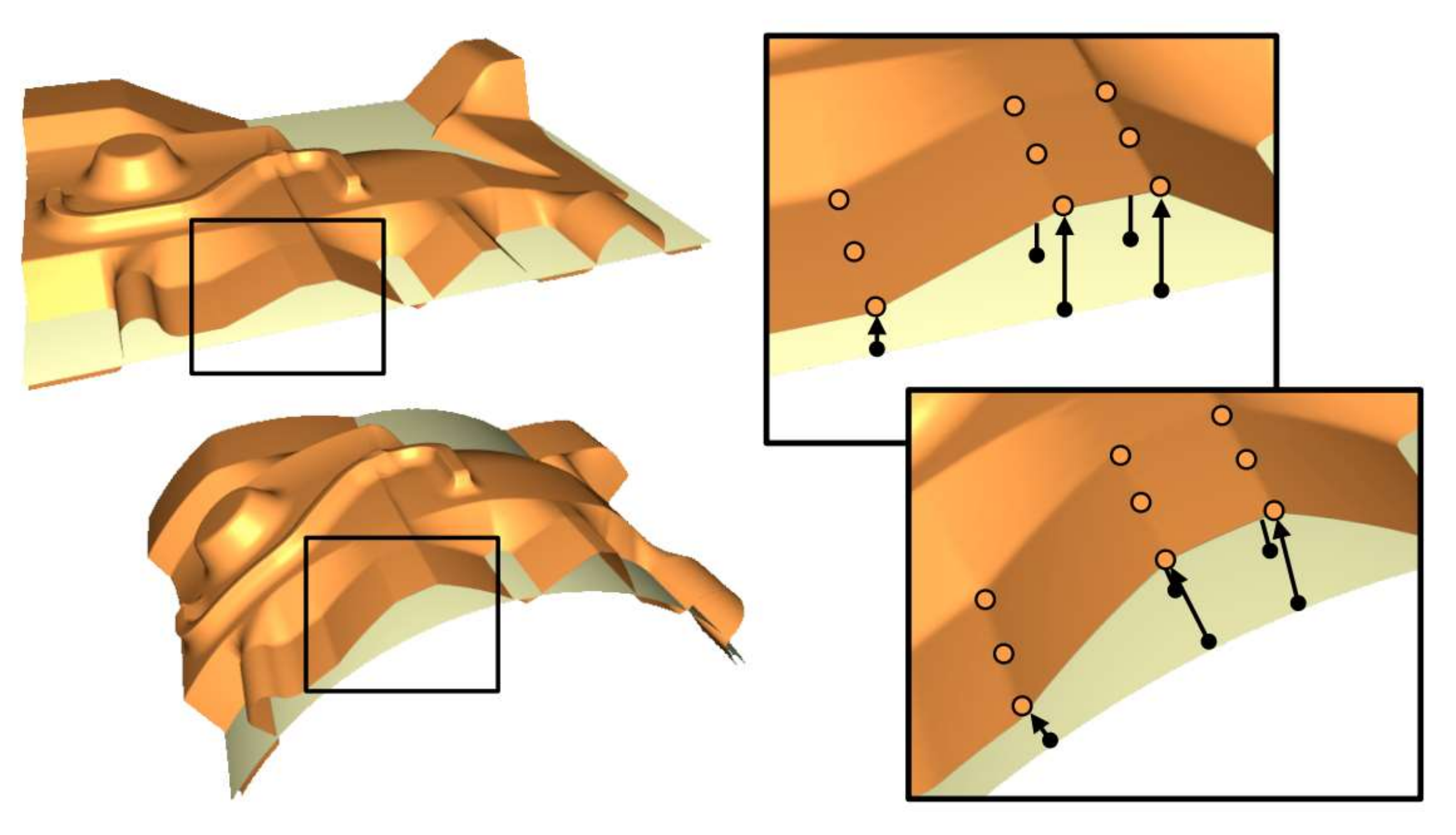


Figure 9.7. A general multi-scale editing framework consists of three main operators: the decomposition operator, which separates the low and high frequencies; the editing operator, which deforms the low frequency components; and the reconstruction operator, which adds the details back onto the modified base surface. Since the lower part of this scheme is hidden in the multi-scale kernel, only the multi-scale edit in the top row is visible to the designer. (Image taken from [Botsch and Sorkine 08]. ©2008 IEEE. Model courtesy of Cyberware.)

Representation for the geometric detail

- The straightforward representation: a vector-valued displacement function
  - Associates a displacement vector to each point on the base surface.
  - Per-vertex displacement vectors
  - $p_i = b_i + h_i, \;\; p_i \in S, \;\; b_i \in B, \;\; h_i \in \mathbb{R}^3$
- Encoded in local frame $h_i = \alpha_i n_i + \beta_i t_{i,1} + \gamma_i t_{i,2}$, where n_i: normal, $t_{i,1}, t_{i,2}$: two tangent vectors

Normal displacements



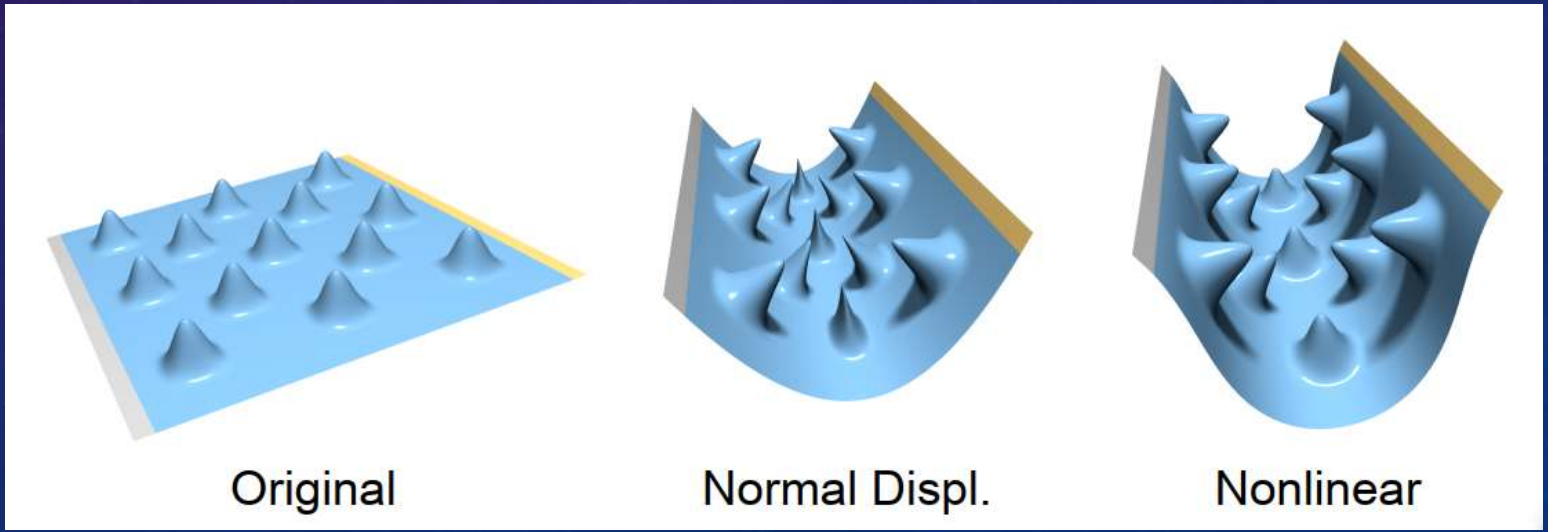
Encoded in local frame

$$h_i' = \alpha_i n_i' + \beta_i t_{i,1}' + \gamma_i t_{i,2}', \qquad p_i' = b_i' + h_i'$$

Figure 9.8. Representing the displacements with regard to the global coordinate system does not lead to the desired result (left). The geometrically intuitive solution is achieved by storing the details with regard to local frames that rotate according to the local tangent plane's rotation of $\mathcal{B}$ (right). (Image taken from [Botsch 05].)

Limitations

- Neighboring displacements are not coupled
  - Surface bending changes their angle
  - Leads to volume changes or self-intersections



Limitations

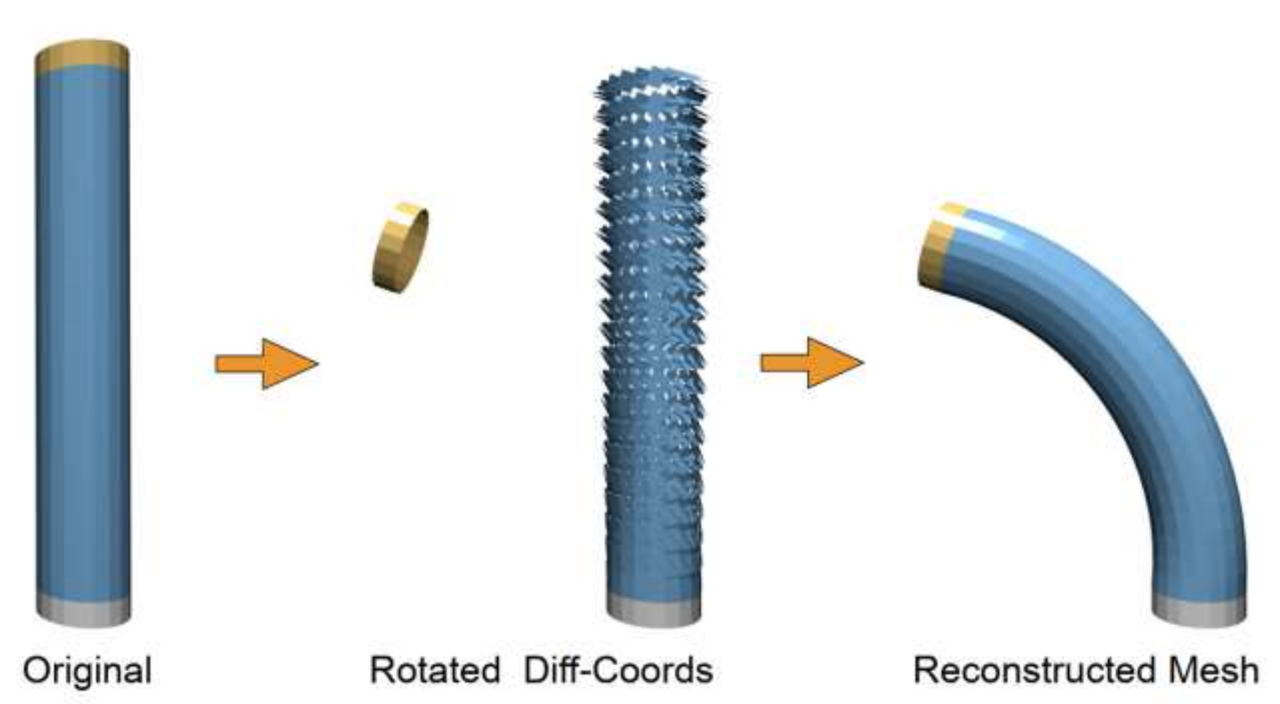
- Neighboring displacements are not coupled
  - Surface bending changes their angle
  - Leads to volume changes or self-intersections
- Multiresolution hierarchy difficult to compute
  - Complex topology
  - Complex geometry
  - Might require more hierarchy levels

Surface-based deformations

- Physically-based deformation
- Multi-Scale deformation
- Differential coordinates
- As-Rigid-As-Possible surface deformation

Differential coordinates

- Manipulate differential coordinates instead of spatial coordinates
  - Gradients, Laplacians, local frames
  - Close connection to surface normal



Differential coordinates

- Which differential coordinate δ_i?
  - Gradients
  - Laplacians
- How to get local transformations $T_i(\delta_i)$?
  - Smooth propagation
  - Implicit optimization

Gradient-based deformation

- Manipulate gradient of a function: $g = \nabla f,\ g \mapsto g' = T(g)$
- Find function f' whose gradient is (close to) g'

$$f' = \underset{f}{\operatorname{argmin}} \int_{\Omega} \|\nabla f - T(g)\|^2 dA$$

- Variational calculus → Euler-Lagrange PDE

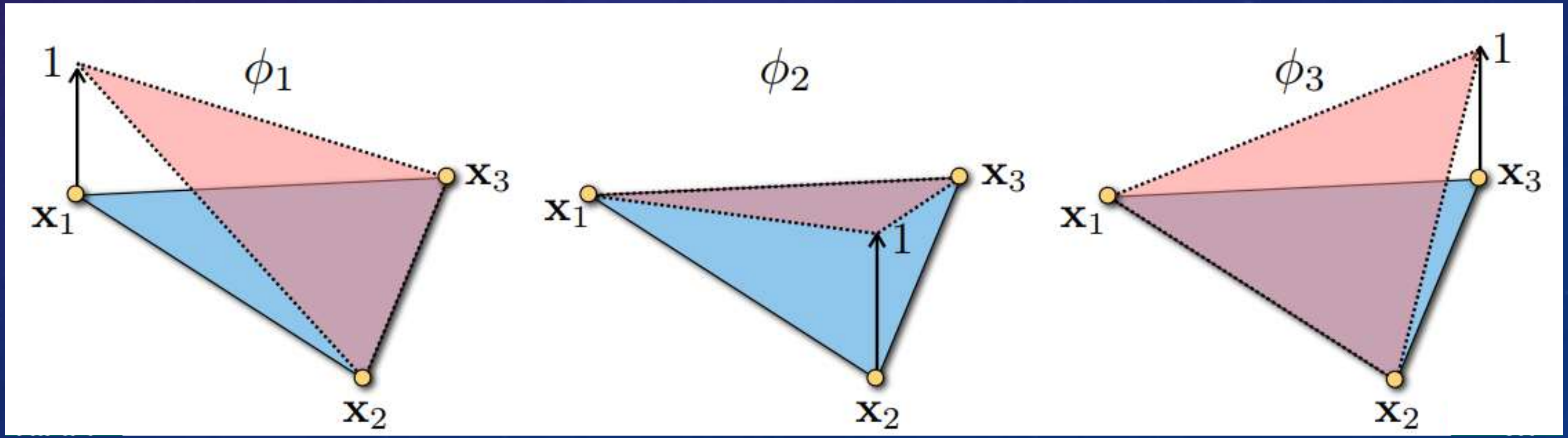
$$\Delta f' = \operatorname{div} T(g)$$

Gradient-based deformation

- Consider piecewise linear coordinate function

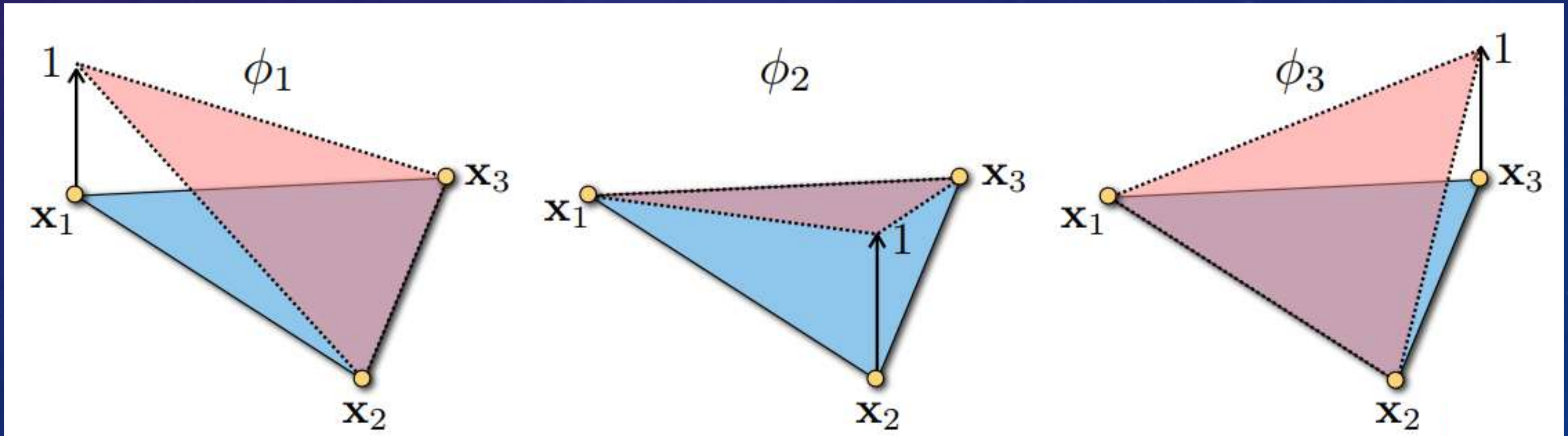
$$p(u,v) = \sum_i p_i \phi_i(u,v)$$

- Its gradient is $\nabla p(u,v) = \sum_i p_i \nabla \phi_i(u,v)$



Gradient-based deformation

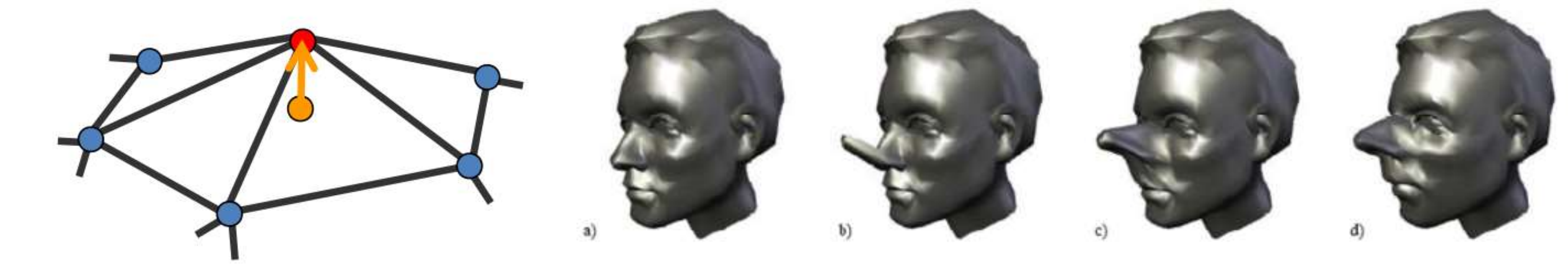
- Its gradient is $\nabla p(u,v) = \sum_i p_i \nabla \phi_i(u,v)$
- It is constant per triangle $\nabla p|_{ijk} = g_{ijk} \in \mathbb{R}^{3\times 3}$
- $Lp' = \operatorname{div} T_{ijk}(g_{ijk})$ per vertex



Laplacian coordinates

- $\delta = Lp$ - approximation to normals - unique up to translation
- Reconstruct by solving $Lp' = \delta$ for p' (with some constraints)

$$p' = \arg\min_{p'} \sum_i \|T_i \delta_i - (Lp')_i\|^2$$

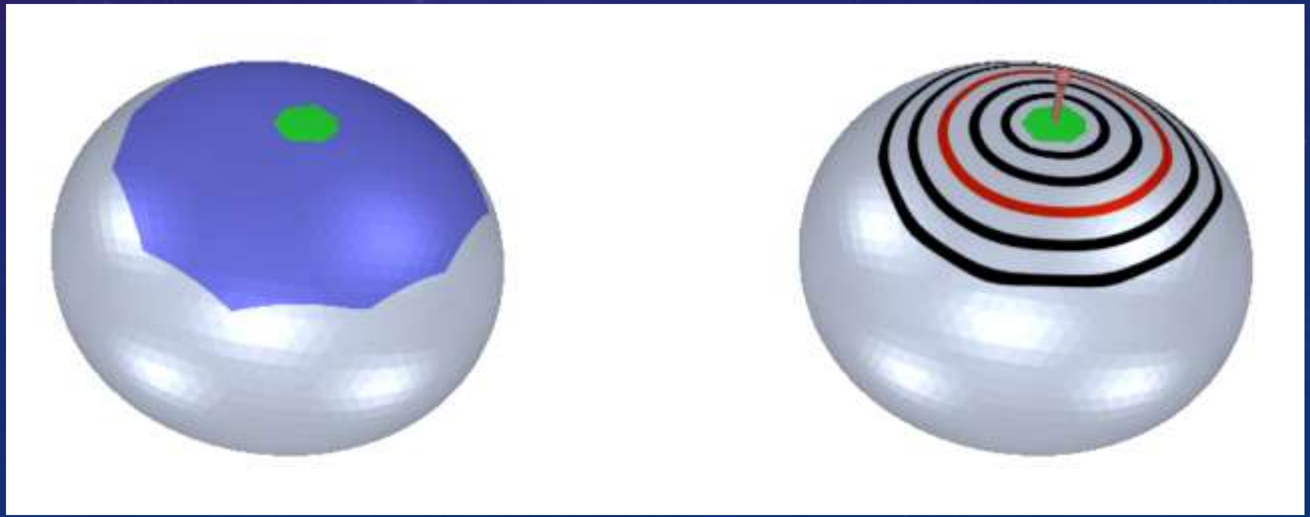


Differential coordinates

- Which differential coordinate δ_i?
  - Gradients
  - Laplacians
- How to get local transformations $T_i(\delta_i)$?
  - Smooth propagation
  - Implicit optimization

Smooth propagation

- Compute handle's deformation gradient
- Extract rotation and scale/shear components
- Propagate damped rotations over ROI



Smooth propagation

- Controlled by a scalar field:
  - 1 is at the handle;
  - 0 is at the fixed region;
  - smoothly blends between 1 and 0 within the support region.

$d_F(p)$ - distance from p to the fixed region, $d_H(p)$ - distance from p to the handle

$$s(p) = \frac{d_F(p)}{d_F(P) + d_H(p)}$$

Harmonic field

- Solving equation

$$\begin{cases} \Delta s = 0, in\ S \backslash (H \cup F) \\ s = 1, \qquad in\ H \\ s = 0, \qquad in\ F \end{cases}$$

- Easy to implement

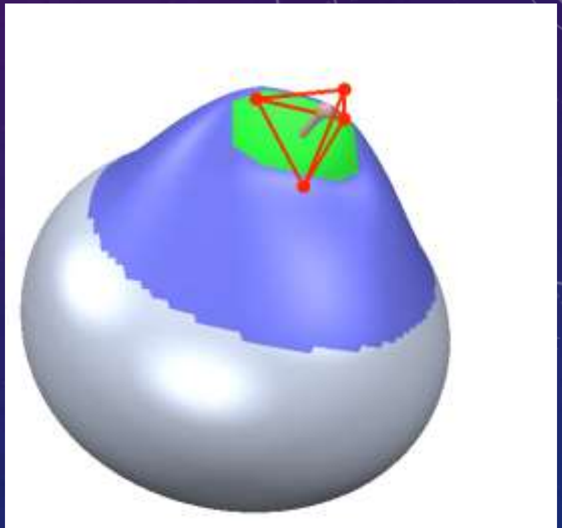
Deformation gradient

- Handle has been transformed affinely

$$T(x) = Ax + t$$

- Deformation gradient is $\nabla T(x) = A$
- Extract rotation R and scale/shear S

$$A = U\Sigma V^T \Longrightarrow R = UV^T, S = V\Sigma V^T$$

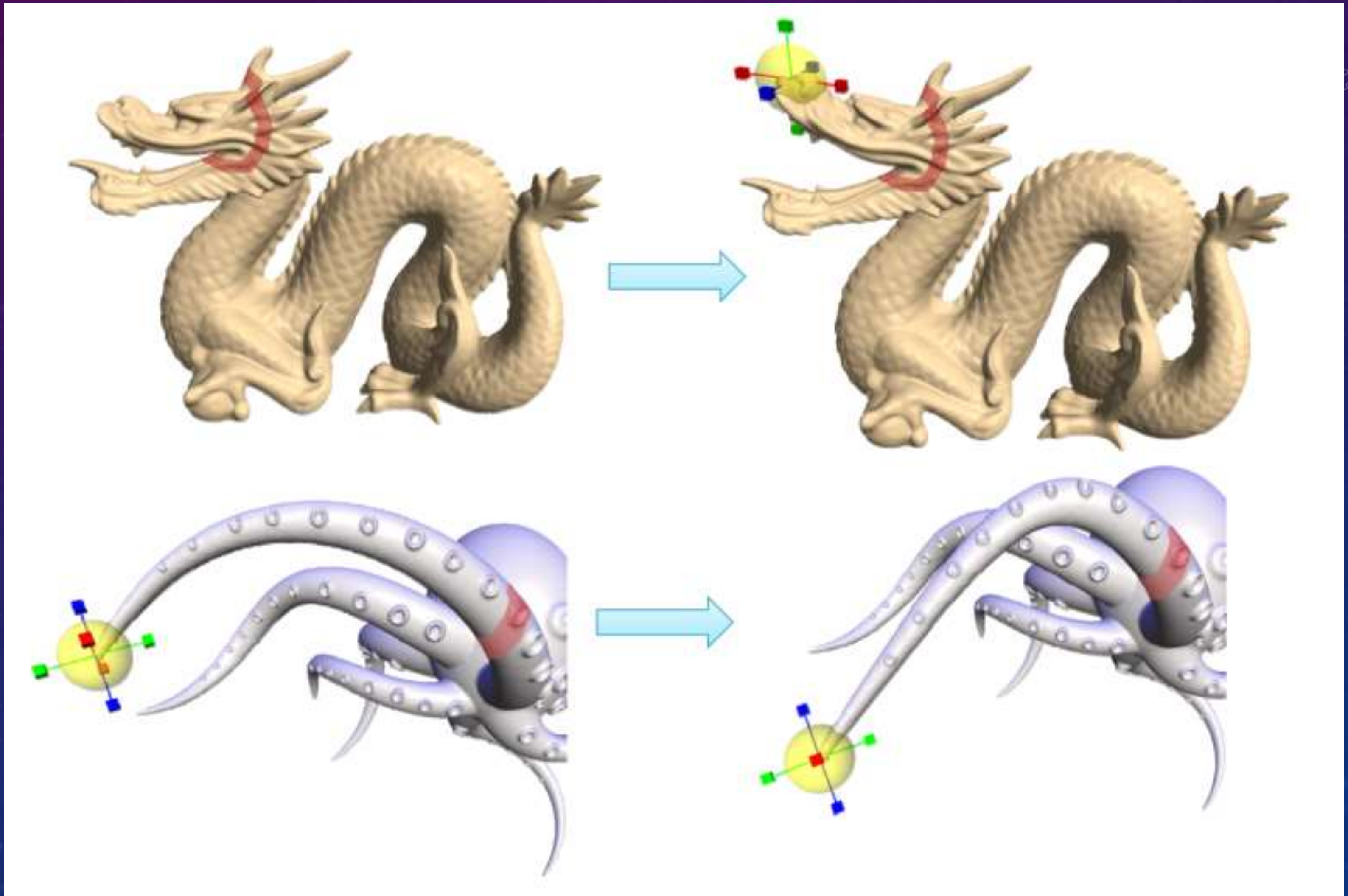


Linearize rotation/scale

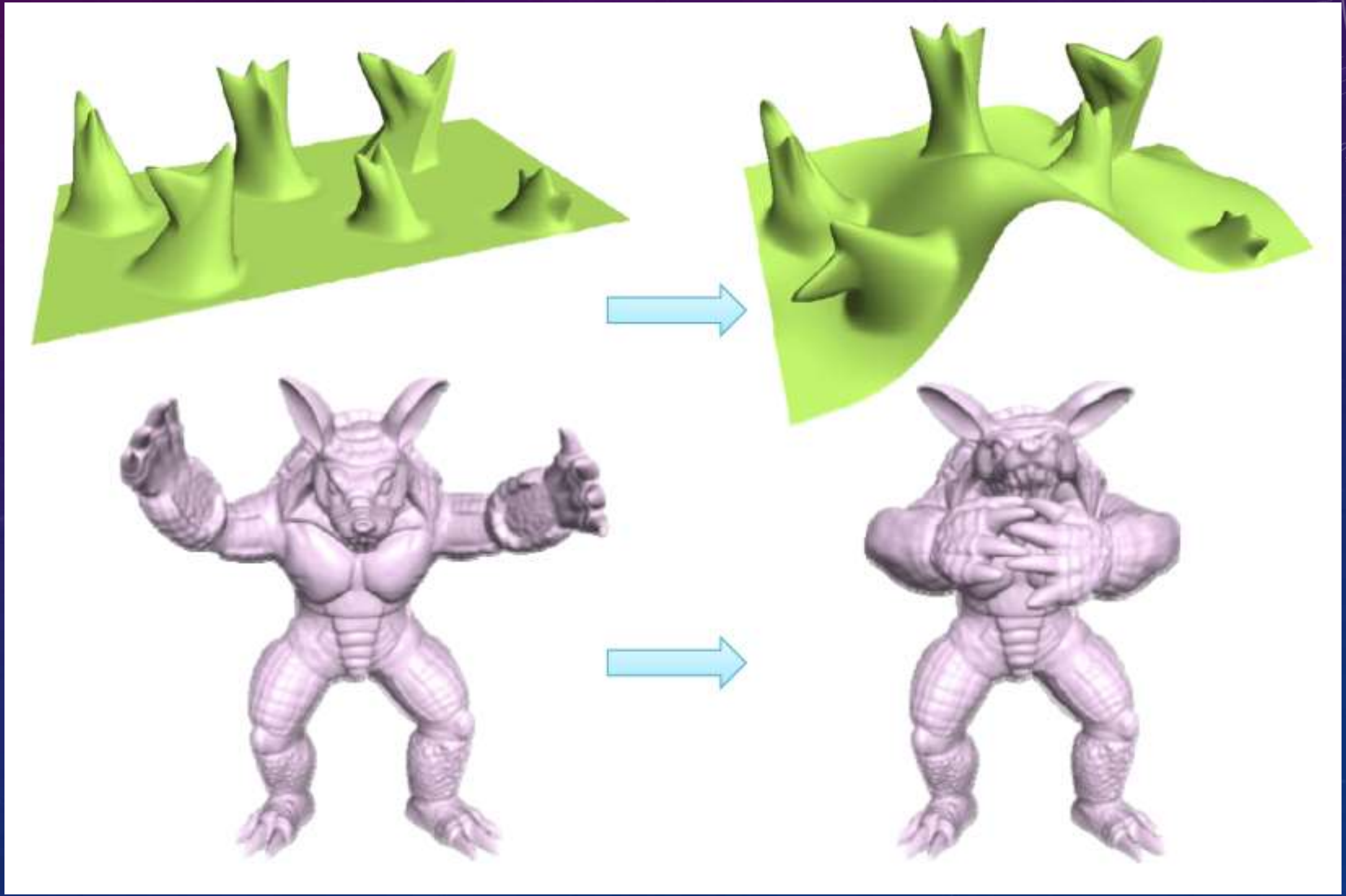
- 2D case: $T_{ijk} = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$
- 3D case: T_i skew-symmetric

$$T_i = \begin{pmatrix} s & -h_3 & h_2 \\ h_3 & s & -h_1 \\ -h_2 & h_1 & s \end{pmatrix} \Longrightarrow T_i x = sx + h \times x$$

Results



Results



Connection to shells

- Neglect local transformations T_i

$$p' = \arg\min_{p'} \sum_i \|(Lp)_i - (Lp')_i\|^2 \Longrightarrow \arg\min_{d} \sum_i \|(Ld)_i\|^2$$

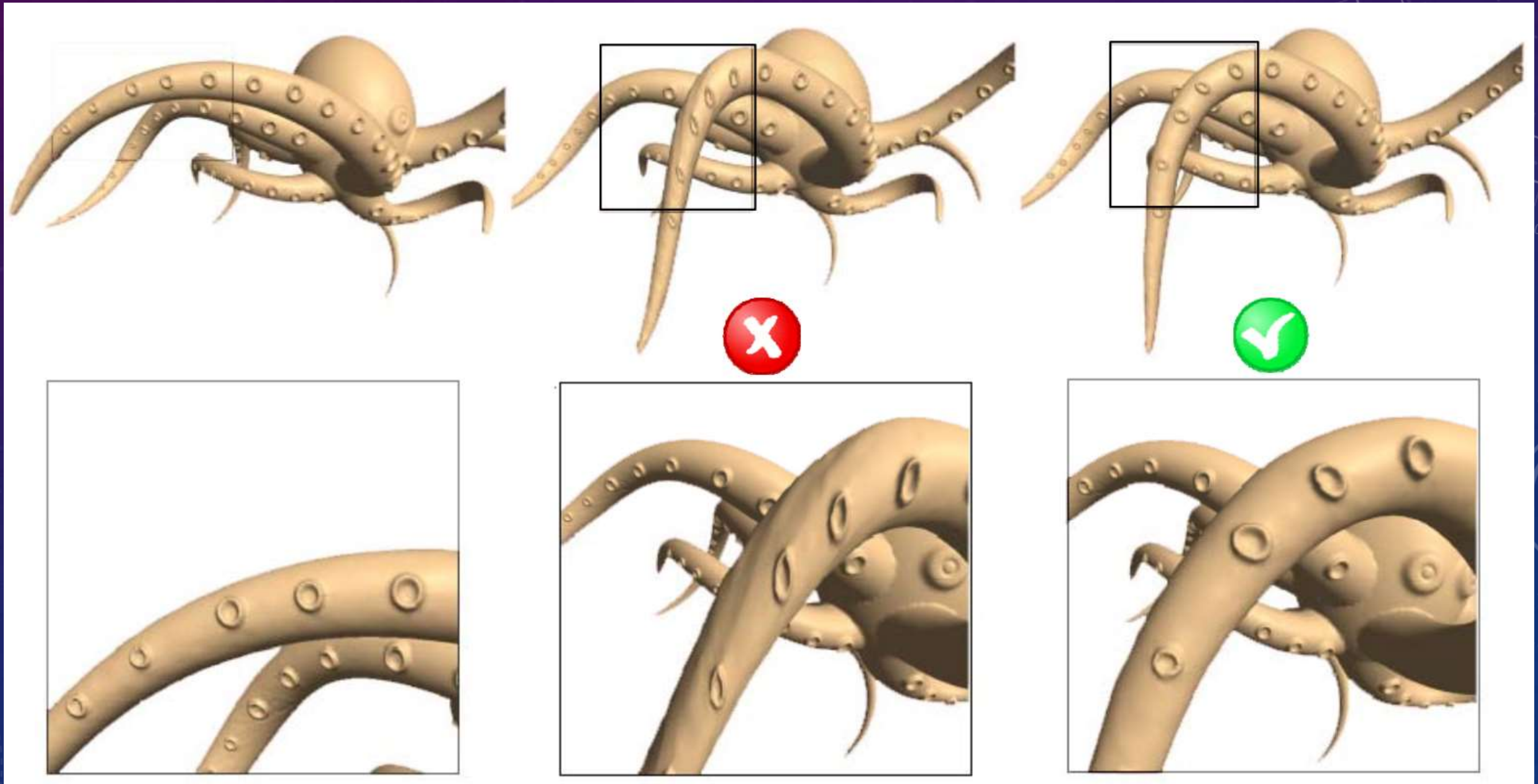
$$E(\mathbf{d}) = \int_S \|\mathbf{d}_{uu}\|^2 + 2\,\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \,\mathrm{d}u\mathrm{d}v \rightarrow \min$$

$$\boxed{f(x) \rightarrow \min}$$

$$\Delta^2\mathbf{d} := \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv} = 0$$

$$\boxed{f'(x) = 0}$$

Connection to shells



Surface-based deformations

- Physically-based deformation
- Multi-Scale deformation
- Differential coordinates
- As-Rigid-As-Possible surface deformation

As-Rigid-As-Possible surface deformation

- Goal: preserve shape meaning that an object is only rotated or translated, but not scaled or sheared – local parts should change as rigidly as possible
- Energy:

$$E(R, p') = \sum_i w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

w_i : local average area, w_{ij} : cot weight

Local step

$$E(R,p') = \sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

- Fix p_i', compute R_i

$$E_i(R_i) = \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

Let $e_{ij}' = p_i' - p_j'$ and $e_{ij} = p_i - p_j$

$$E_i(R_i) = \sum_{j\in\Omega(i)} w_{ij} \left(e_{ij}' - R_i e_{ij}\right)^T (e_{ij}' - R_i e_{ij}) = -\sum_{j\in\Omega(i)} w_{ij} {e_{ij}'}^T R_i e_{ij} + C$$

Local step

$$E(R,p') = \sum_i w_i \sum_{j\in\Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i(p_i - p_j) \right\|^2$$

$$\underset{R_i}{\operatorname{argmax}} \sum_{j\in\Omega(i)} w_{ij} {e'_{ij}}^T R_i e_{ij} = \underset{R_i}{\operatorname{argmax}} \sum_{j\in\Omega(i)} tr\left(w_{ij} R_i e_{ij} {e'_{ij}}^T\right)$$

$$= \underset{R_i}{\operatorname{argmax}}\, tr\left(R_i \sum_{j\in\Omega(i)} w_{ij} e_{ij} {e'_{ij}}^T\right)$$

Let $M_i = \sum_{j\in\Omega(i)} w_{ij} e_{ij} {e'_{ij}}^T$ and $M_i = U_i S_i V_i$ (signed SVD), then $R_i = V_i U_i^T$

Global step

$$E(R, p') = \sum_i w_i \sum_{j \in \Omega(i)} w_{ij} \left\| (p_i' - p_j') - R_i (p_i - p_j) \right\|^2$$

- Fix R_i, compute p_i'
- Linear least square, easy to solve
- Initial deformation
  - Previous frame (for interactive manipulation)
  - Naive Laplacian editing
  - …

Results

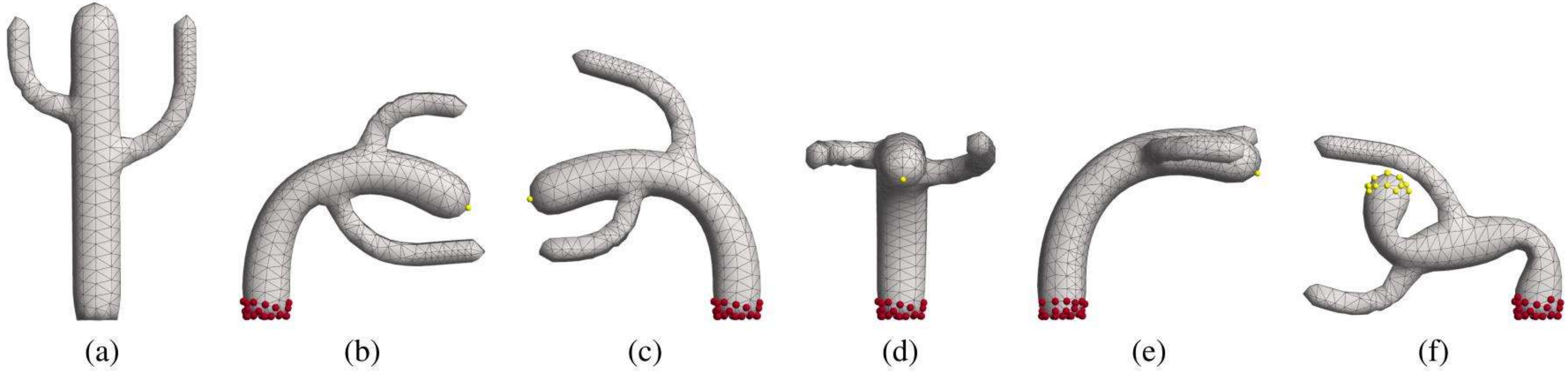
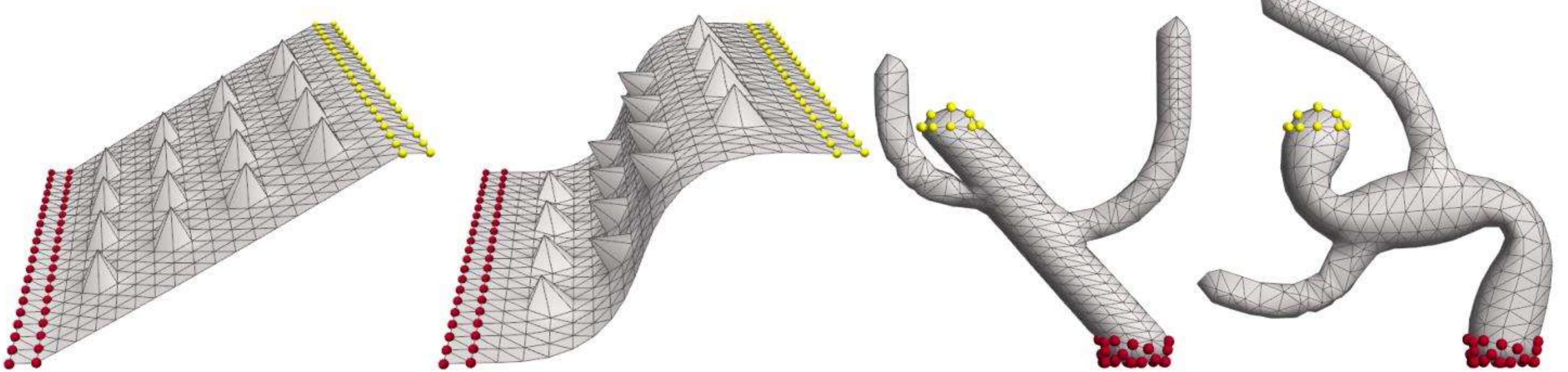


Figure 7: *Bending the Cactus. (a) is the original model; yellow handles are translated to yield the results (b-f). (d) and (e) show side and front views of forward bending, respectively. Note that in (b-e) a* single vertex *at the tip of the Cactus serves as the handle, and the bending is the result of translating that vertex, no rotation constraints are given.*



Poisson　　our method　　Poisson　　our method

Figure 5: *Comparison with Poisson mesh editing. The original models appear in Figures 2 and 7. The yellow handle was only translated; this poses a problem for rotation-propagation methods such as [YZX* 04, ZRKS05, LSLCO05].*

QA

- How to implement ARAP tetrahedral deformation?