SGP 2012 Grad School:
Shape Deformation

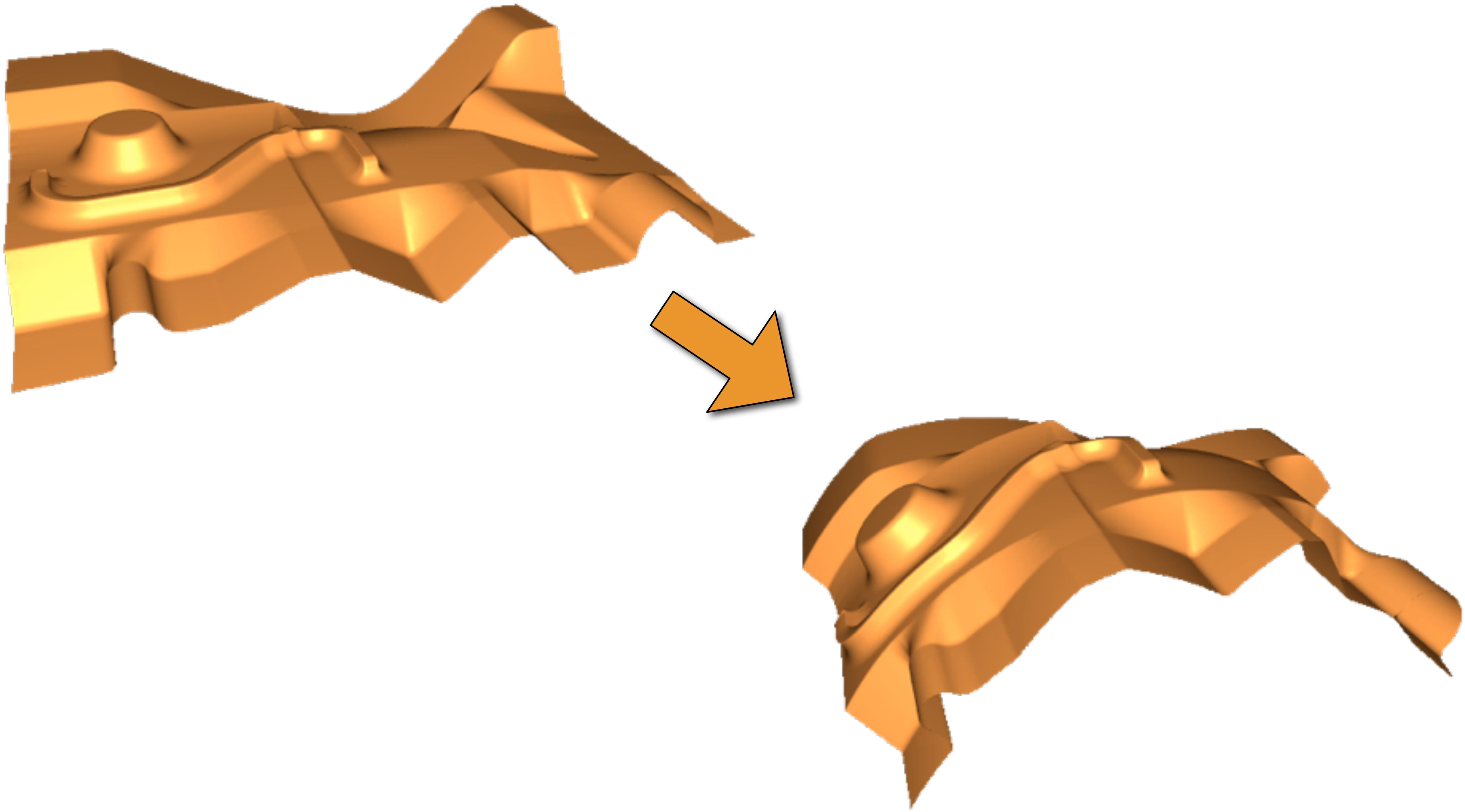
Mario Botsch

Computer Graphics & Geometry Processing
Bielefeld University

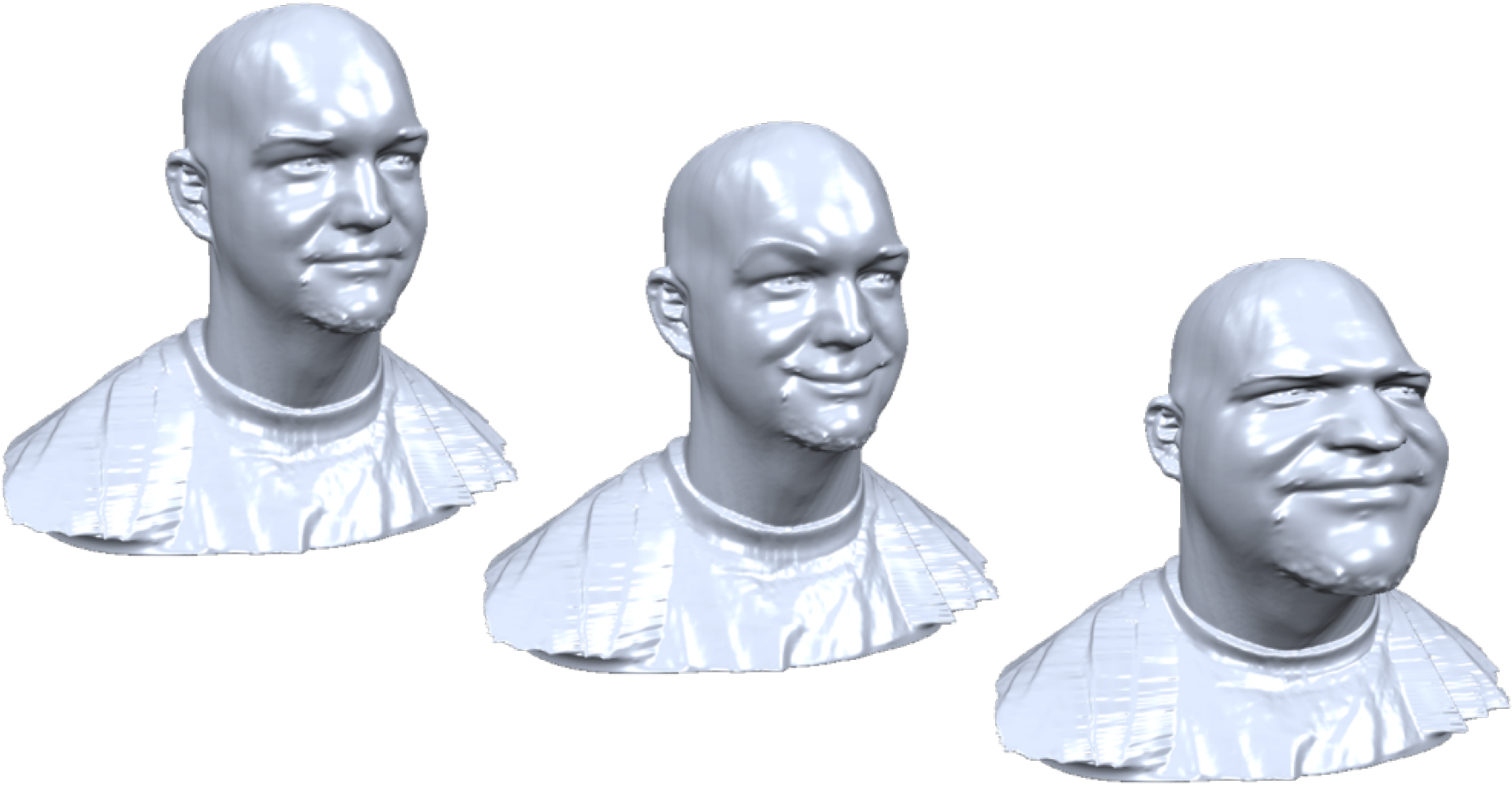
Universität Bielefeld

CITEC

Detail Preservation



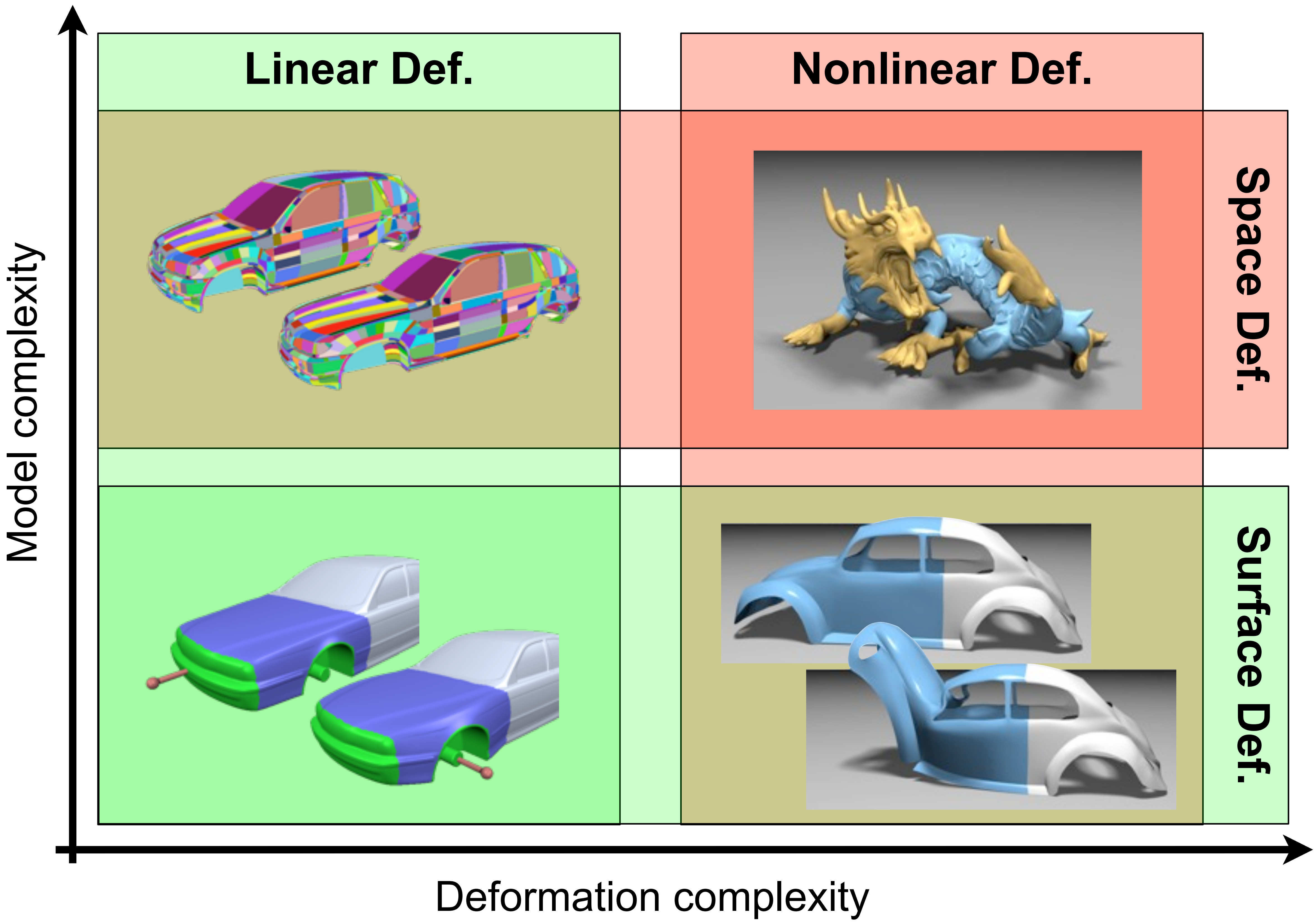
Local & Global



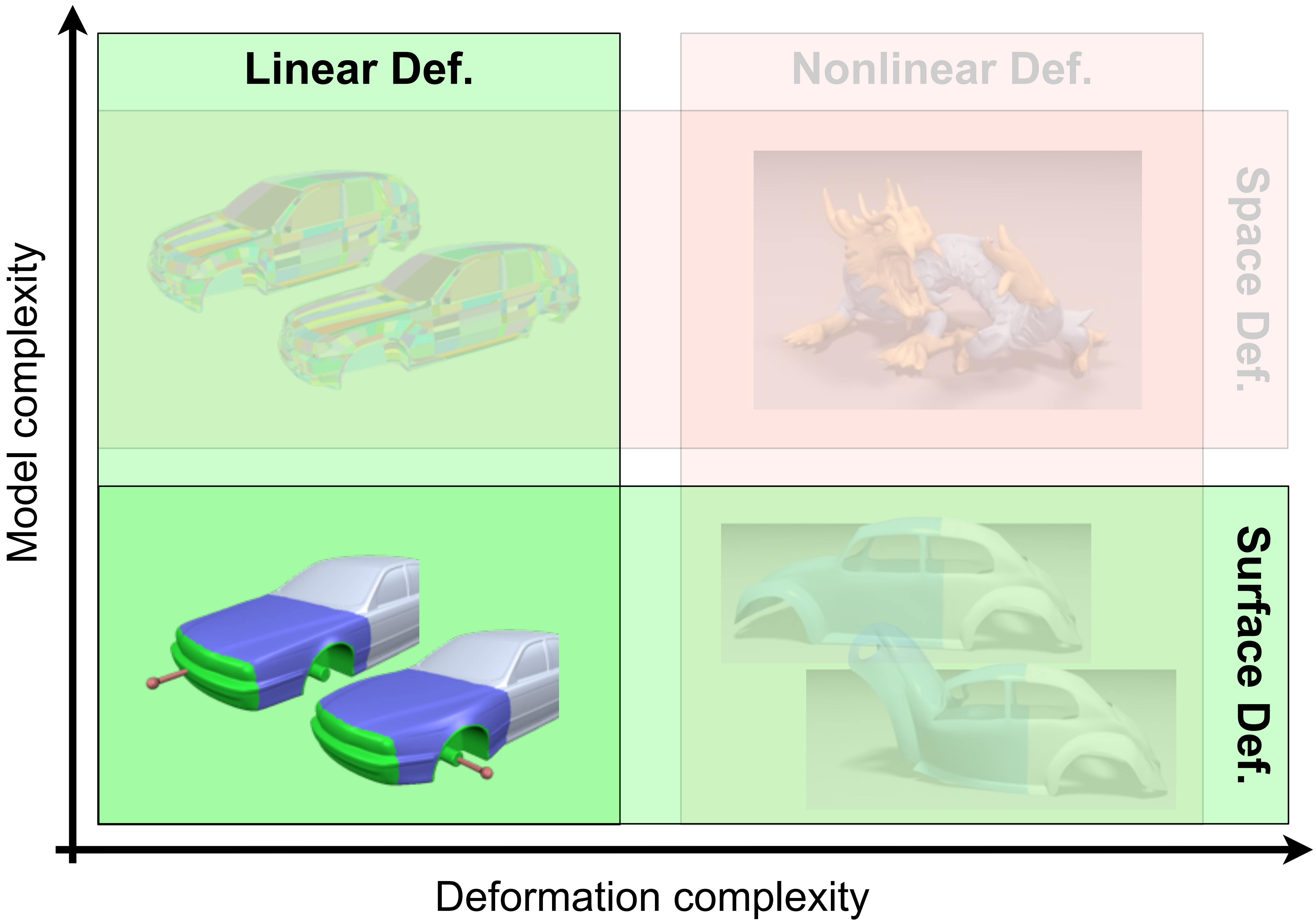
Non-Manifold Models

Complex Models

Shape Deformation



Shape Deformation

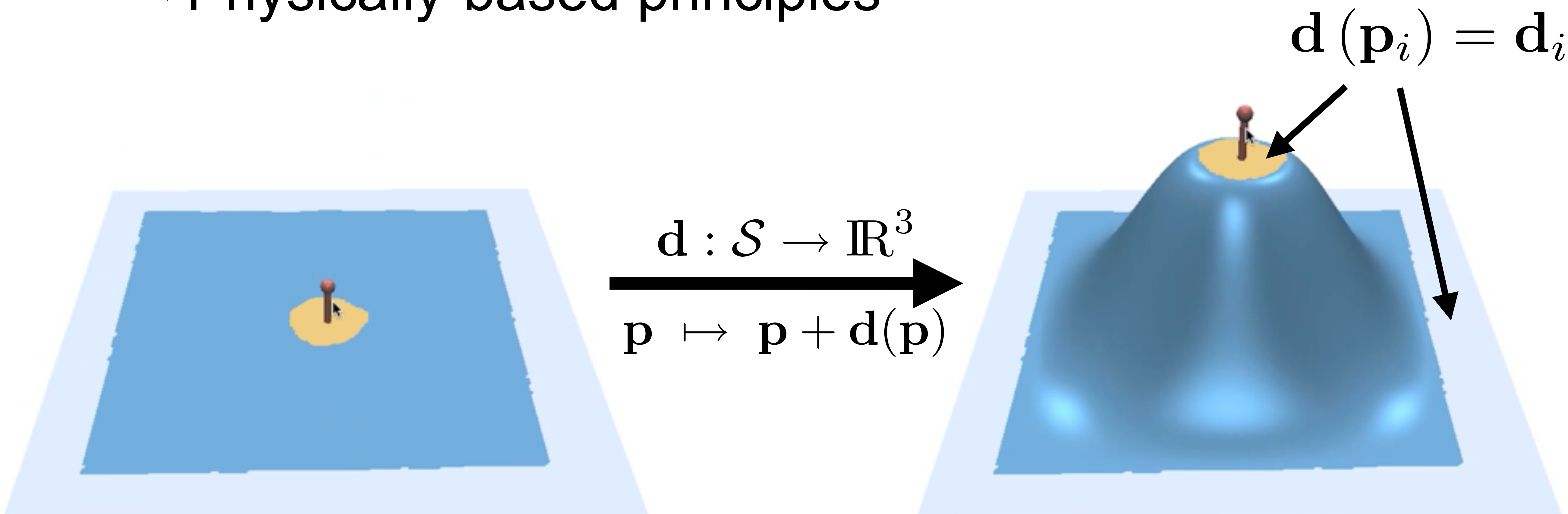


Linear Surface-Based Deformation

- **Shell-Based Deformation**
- Multi-Scale Deformation
- Differential Coordinates

Surface Deformation

- Mesh deformation by displacement function **d**
  - – Interpolate prescribed constraints
  - – Smooth, intuitive deformation
  - ➡Physically-based principles



Shell Deformation

- **Stretching**
  - Change of local distances
  - Captured by 1st fundamental form

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

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

- **Bending**
  - Change of local curvature
  - Captured by 2nd fundamental form

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

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

- **Stretching & bending is sufficient**
  - 1st and 2nd fundamental forms determine a surface up to rigid motion.

Shell 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

$$\int_\Omega k_s \underbrace{\left(\left\|\frac{\partial \mathbf{d}}{\partial u}\right\|^2 + \left\|\frac{\partial \mathbf{d}}{\partial v}\right\|^2\right)}_{\text{stretching}} + k_b \underbrace{\left(\left\|\frac{\partial^2 \mathbf{d}}{\partial u^2}\right\|^2 + 2\left\|\frac{\partial^2 \mathbf{d}}{\partial u \partial v}\right\|^2 + \left\|\frac{\partial^2 \mathbf{d}}{\partial v^2}\right\|^2\right)}_{\text{bending}} \mathrm{d}u\mathrm{d}v$$

Shell Deformation

- Minimize linearized shell energy

$$\int_\Omega k_s \left(\left\| \frac{\partial \mathbf{d}}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{d}}{\partial v} \right\|^2 \right) + k_b \left(\left\| \frac{\partial^2 \mathbf{d}}{\partial u^2} \right\|^2 + 2 \left\| \frac{\partial^2 \mathbf{d}}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 \mathbf{d}}{\partial v^2} \right\|^2 \right) \mathrm{d}u\mathrm{d}v$$

$$f(x) \rightarrow \min$$

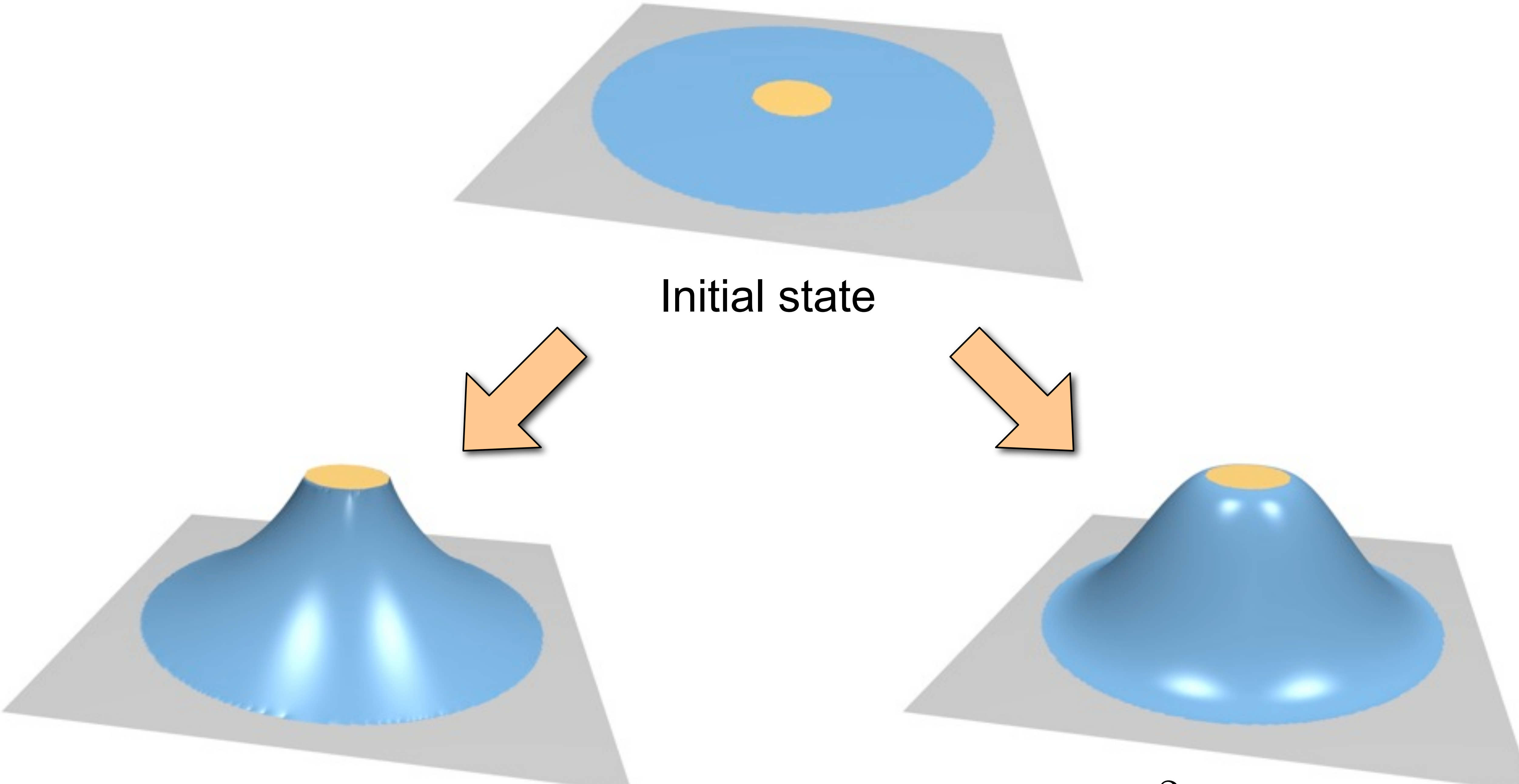
- Variational calculus → Euler-Lagrange PDE

$$-k_s \Delta \mathbf{d} + k_b \Delta^2 \mathbf{d} = 0$$

$$f'(x) = 0$$

➡ "Best" deformation that satisfies constraints

Stretching & Bending



$$\Delta \mathbf{d} = 0$$

(Membrane)

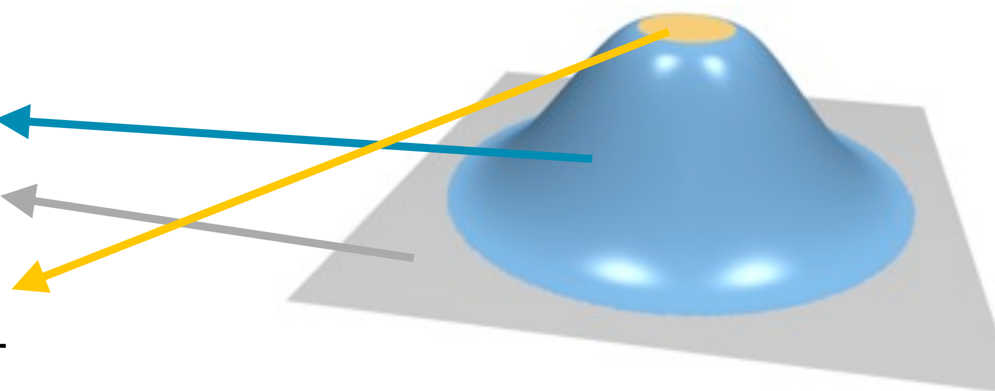
$$\Delta^2 \mathbf{d} = 0$$

(Thin plate)

PDE Discretization

- Euler-Lagrange PDE

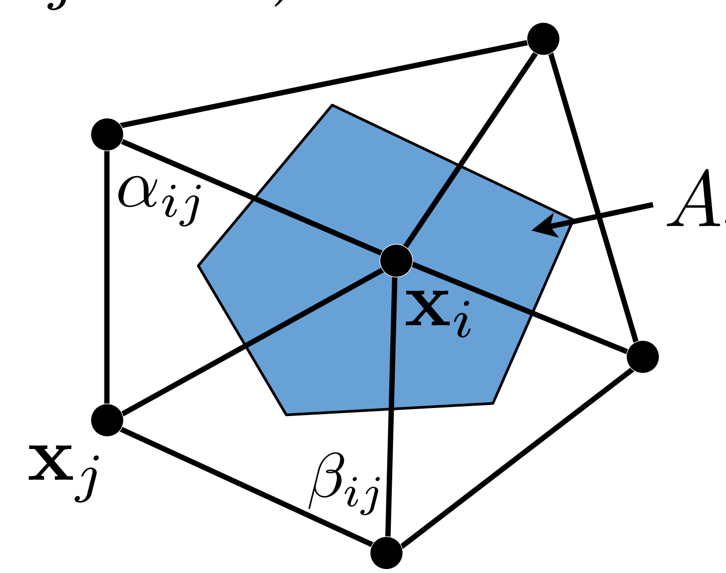
$$\Delta^2 \mathbf{d} = \mathbf{0}$$
$$\mathbf{d} = \mathbf{0}$$
$$\mathbf{d} = \delta \mathbf{h}$$



- Laplace discretization

$$\Delta \mathbf{d}_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}_i} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{d}_j - \mathbf{d}_i)$$

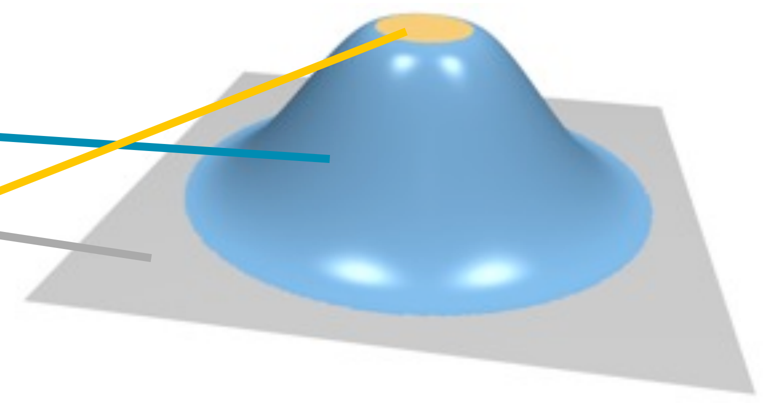
$$\Delta^2 \mathbf{d}_i = \Delta(\Delta \mathbf{d}_i)$$



Linear System

- Sparse linear system (19 nz/row)

$$\begin{pmatrix} & \Delta^2 & \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \vdots \\ \mathbf{d}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \delta\mathbf{h}_i \end{pmatrix}$$



  - Turn into symmetric system

- Solve this system *each frame*
  - Only right hand side changes
  - Symmetric positive definite matrix
  - Use efficient linear solvers !!!

Sparse SPD Solvers

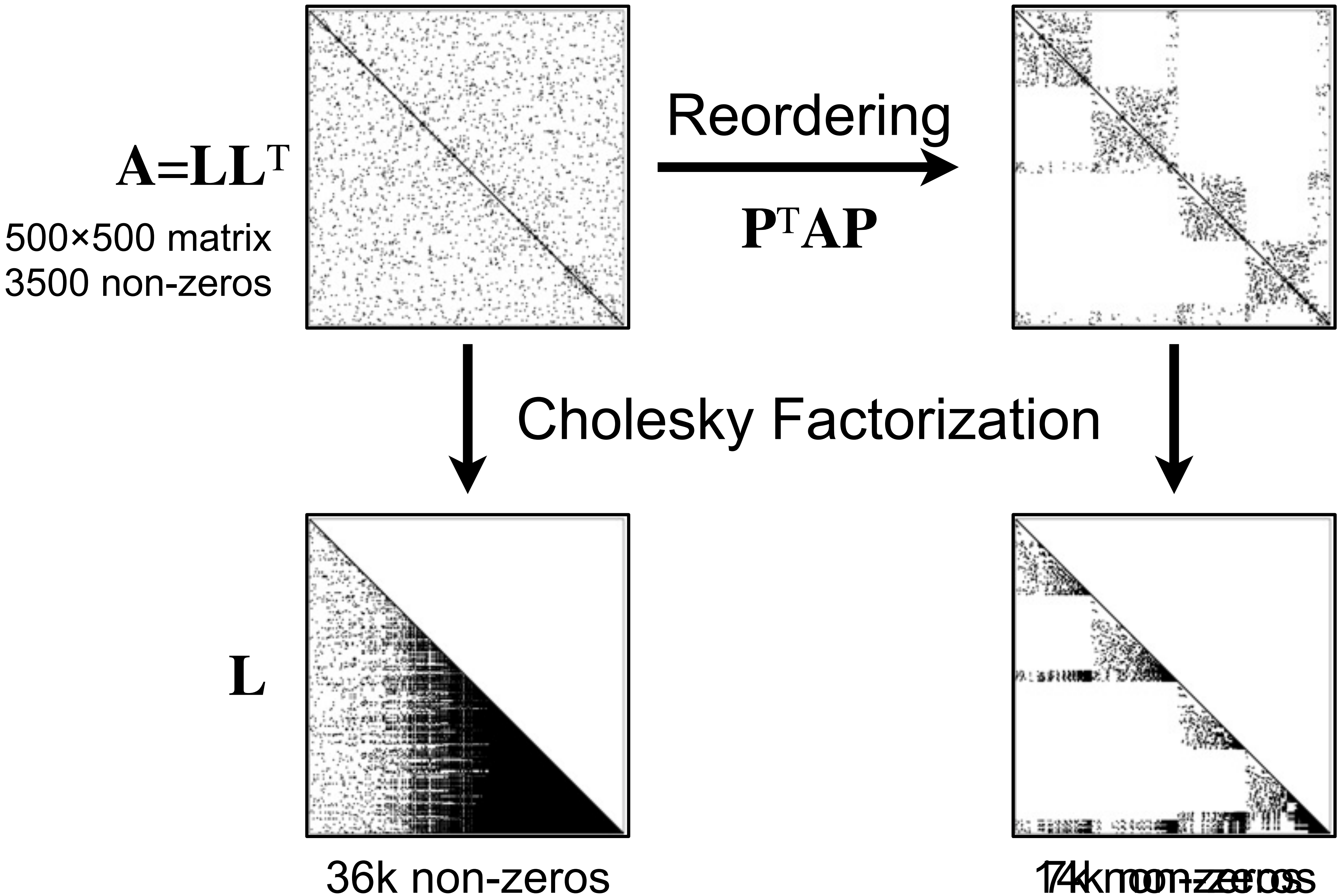
- Cholesky factorization
  - Cubic complexity
  - High memory consumption (doesn't exploit sparsity)
- Iterative conjugate gradients
  - Quadratic complexity
  - Need sophisticated preconditioning
- Multigrid solvers
  - Linear complexity
  - But rather complicated to develop (and to use)
- Sparse Cholesky factorization!

Dense Cholesky Solver

Solve $\mathbf{A}\mathbf{x} = \mathbf{b}$

1. Cholesky factorization $\mathbf{A} = \mathbf{L}\mathbf{L}^T$
2. Solve system $\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}, \quad \mathbf{x} = \mathbf{L}^{-T}\mathbf{y}$

Sparse Cholesky Factorization



Sparse Cholesky Solver

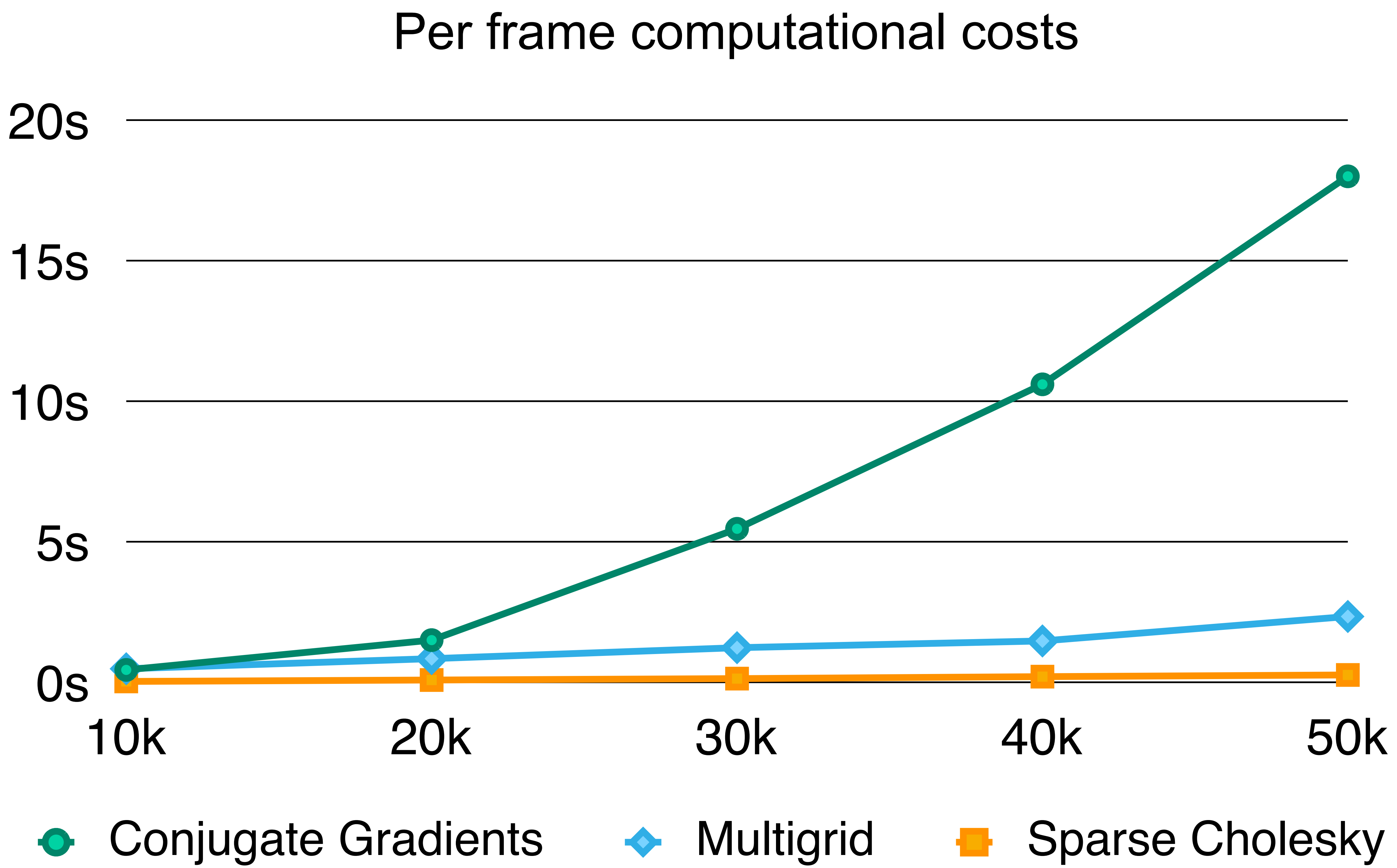
Solve $\mathbf{Ax} = \mathbf{b}$

Pre-computation

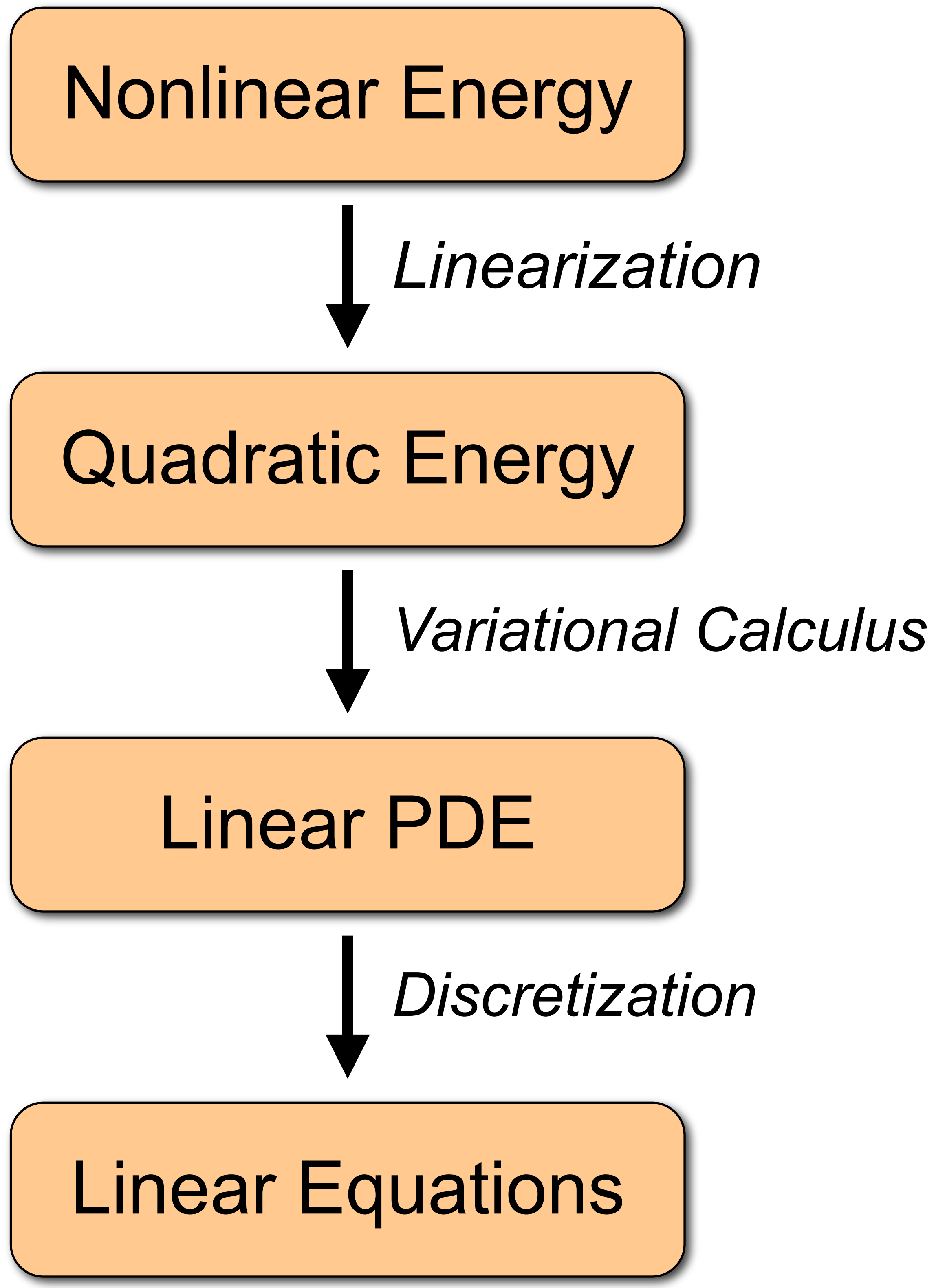
1. Matrix re-ordering $\tilde{\mathbf{A}} = \mathbf{P}^T\mathbf{AP}$
2. Cholesky factorization $\tilde{\mathbf{A}} = \mathbf{LL}^T$
3. Solve system $\mathbf{y} = \mathbf{L}^{-1}\mathbf{P}^T\mathbf{b}, \quad \mathbf{x} = \mathbf{PL}^{-T}\mathbf{y}$

Per-frame computation

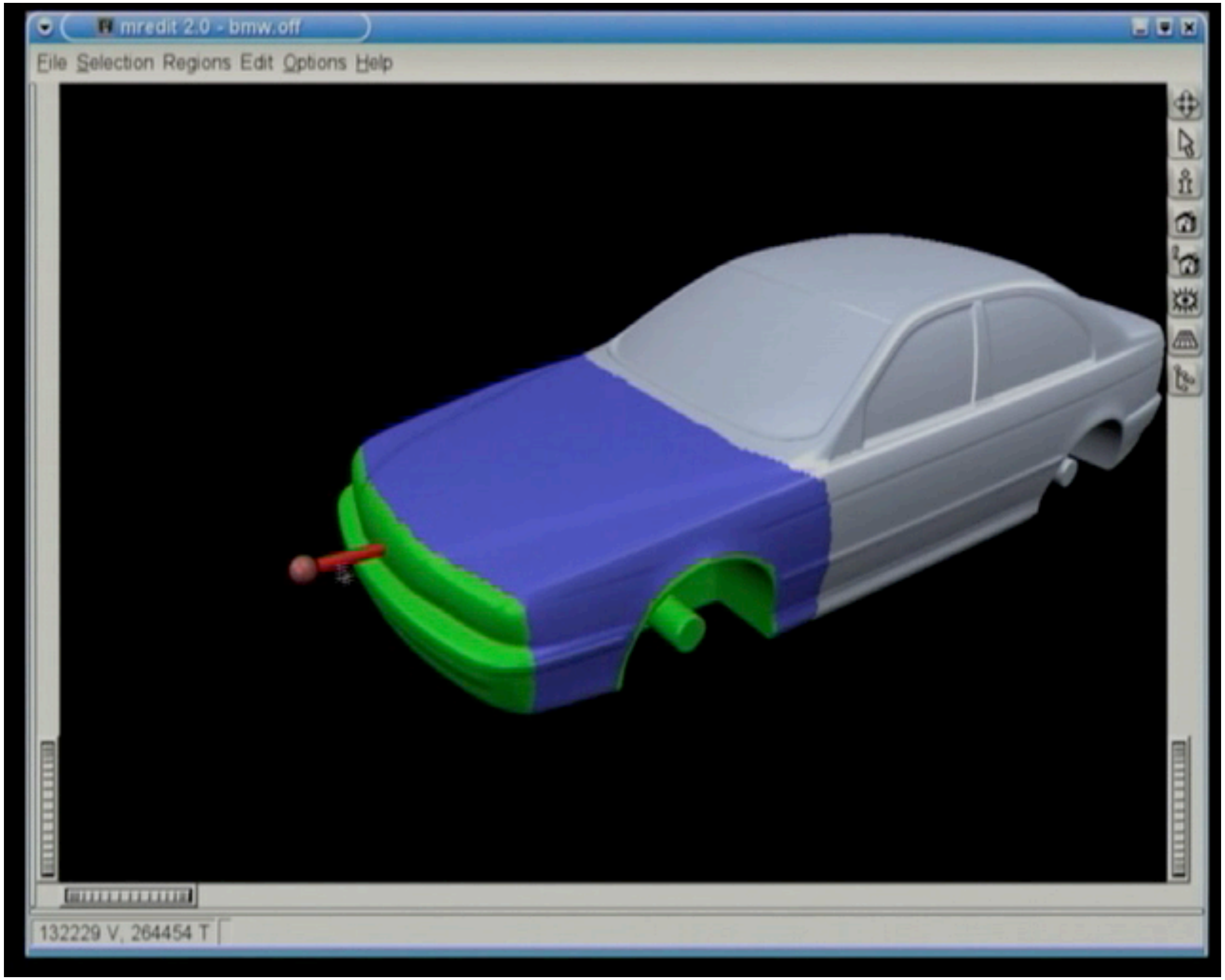
Linear System Solver



Derivation Steps

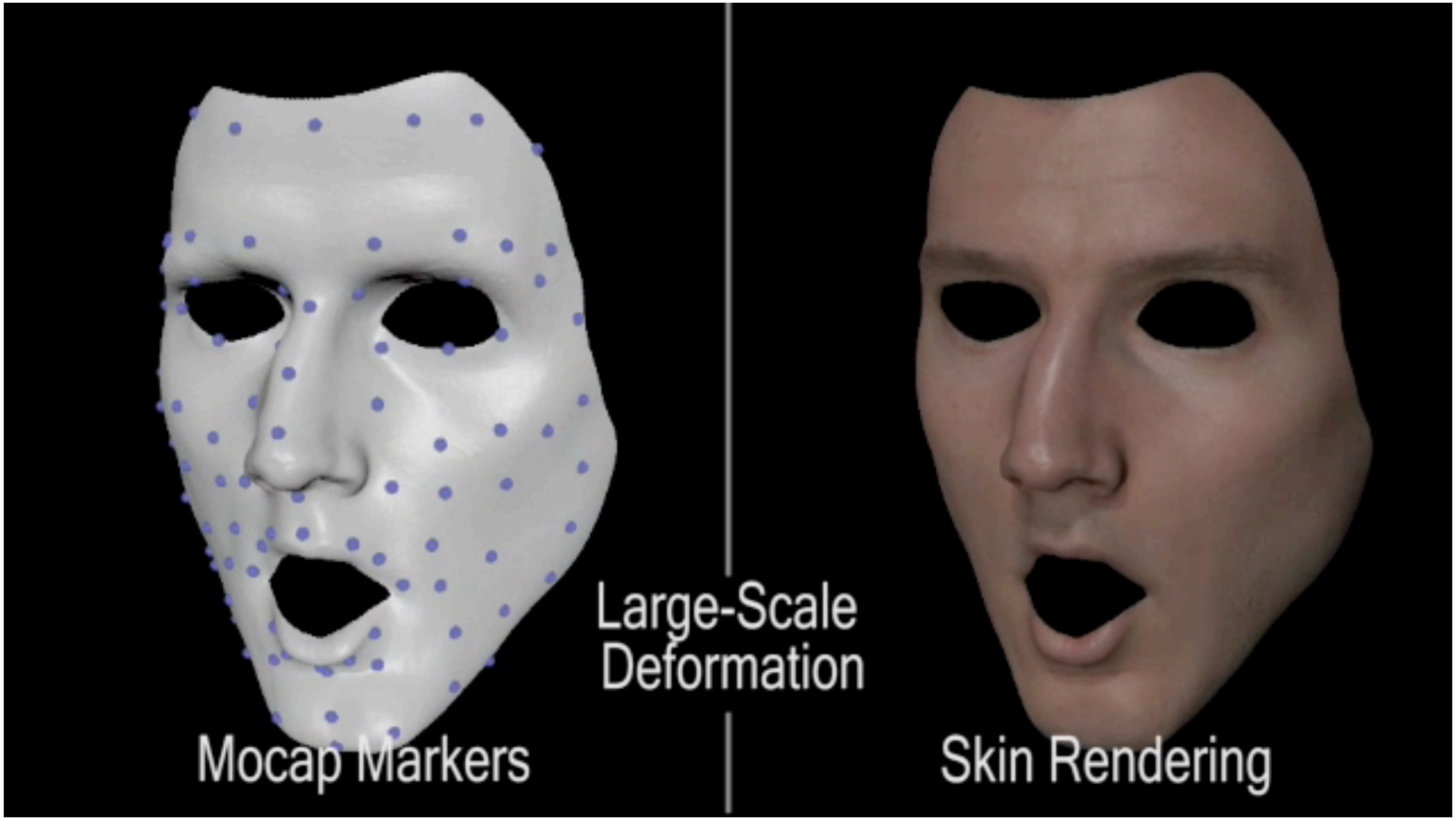


CAD-Like Deformation



[Botsch & Kobbelt, SIGGRAPH 04]

Face Animation



[Bickel et al, SIGGRAPH 07]

Literature

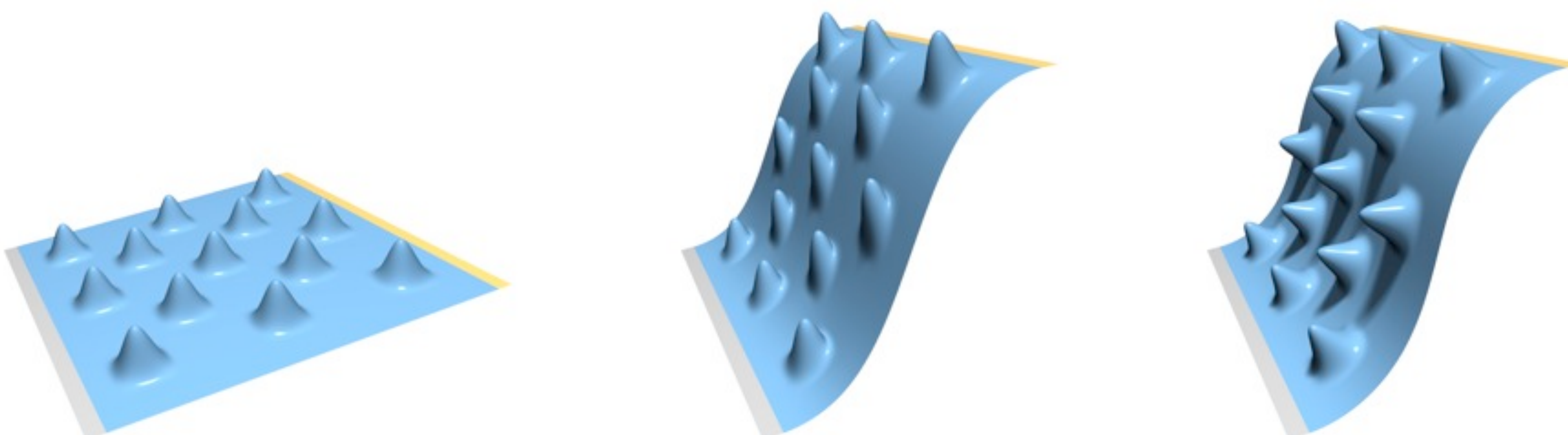
- Kobbelt et al, *Interactive multi-resolution modeling on arbitrary meshes*, SIGGRAPH 1998
- Botsch & Kobbelt, *An intuitive framework for real-time freeform modeling*, SIGGRAPH 2004

Linear Surface-Based Deformation

- Shell-Based Deformation
- **Multi-Scale Deformation**
- Differential Coordinates

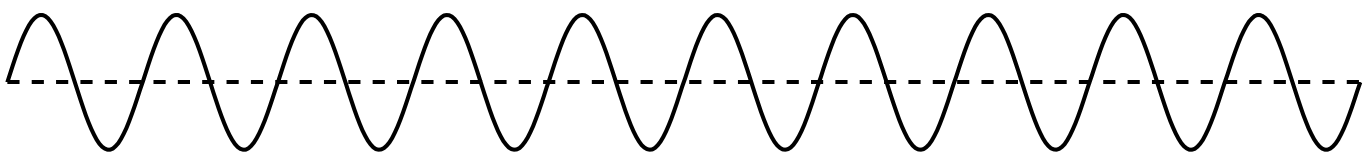
Multi-Scale Modeling

- Even pure translations induce local rotations!
  - ➡ Inherently non-linear coupling
- Alternative approach
  - – Linear deformation + multi-scale decomposition...



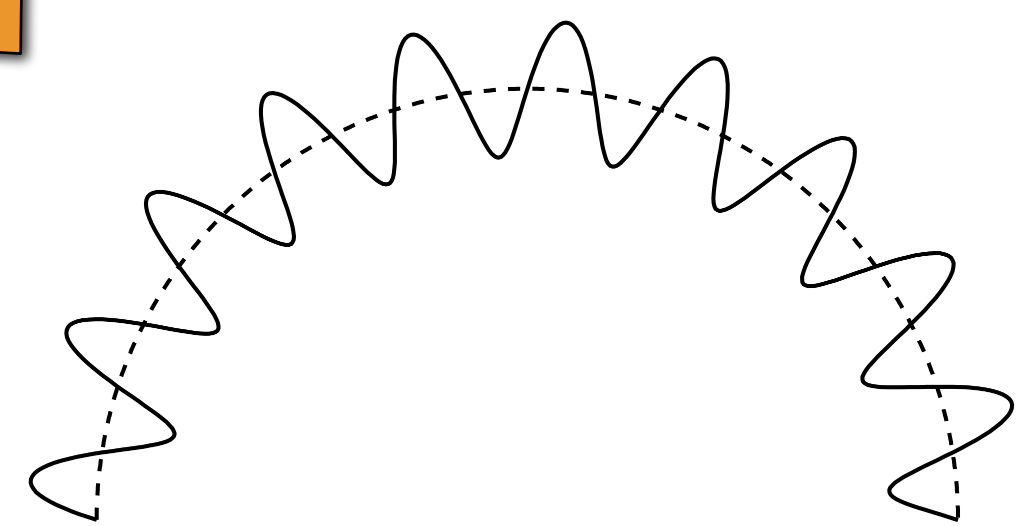
Original Linear Nonlinear

Multi-Scale Editing



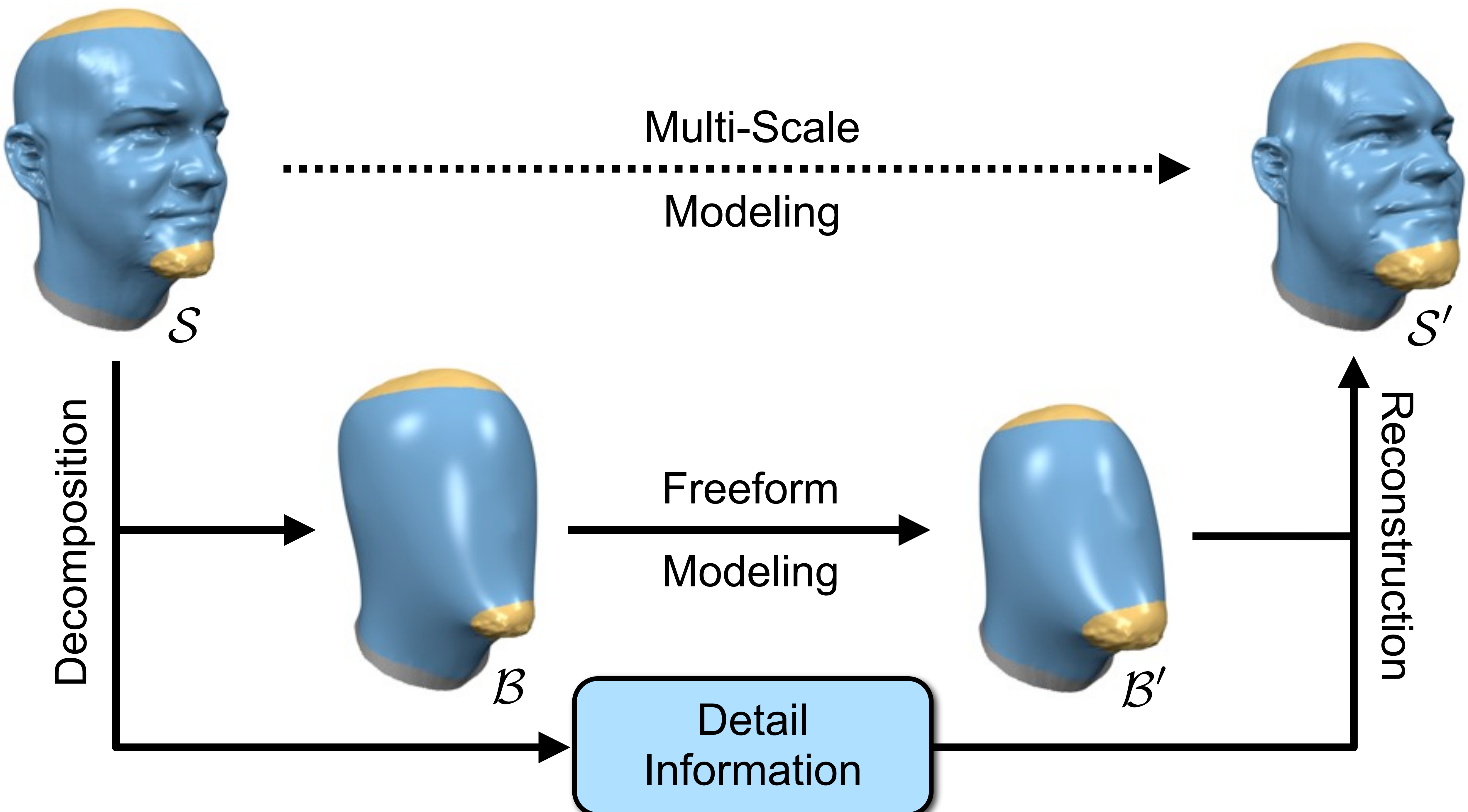
Frequency decomposition

Change low frequencies



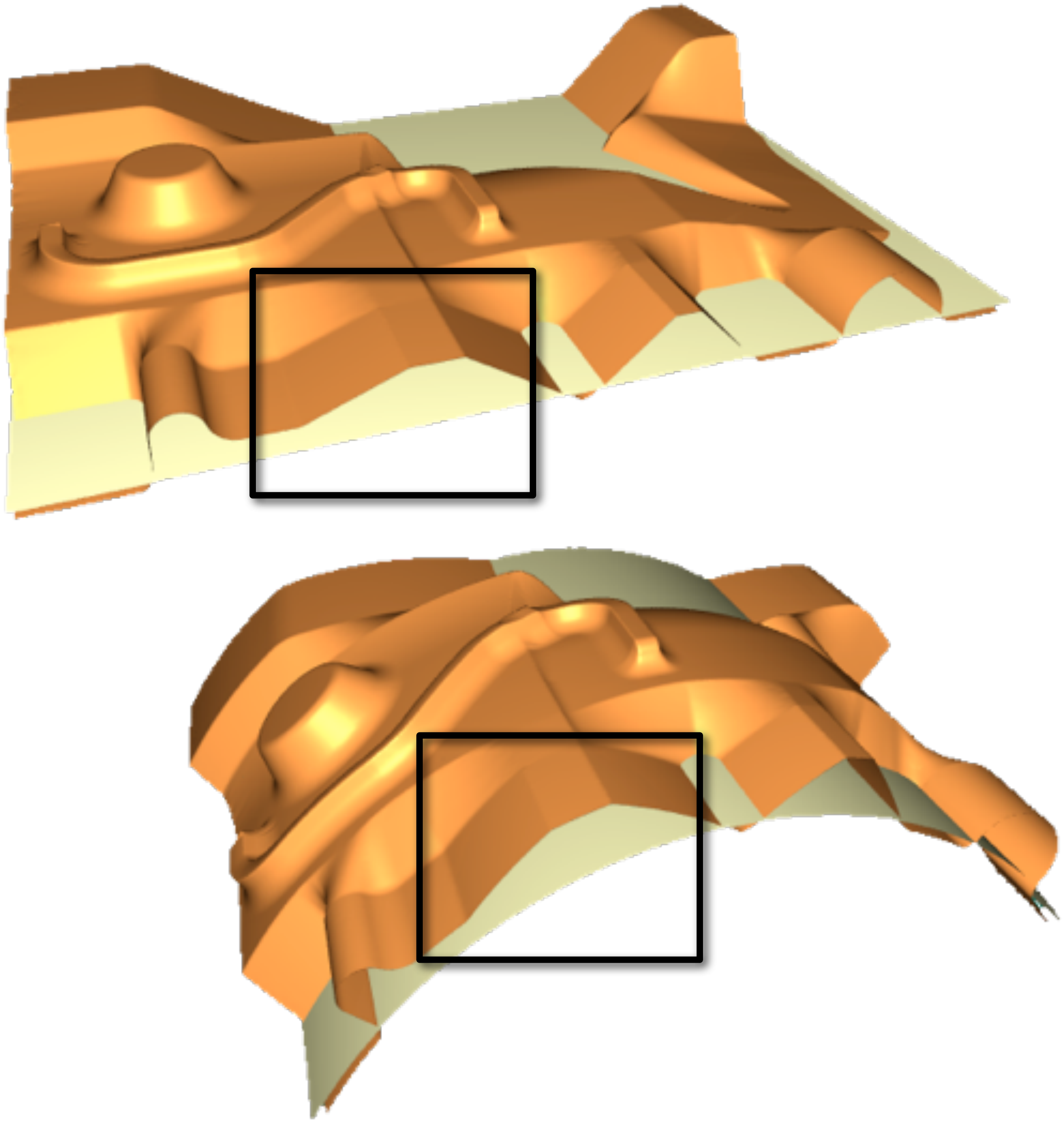
Add high frequency details, stored in local frames

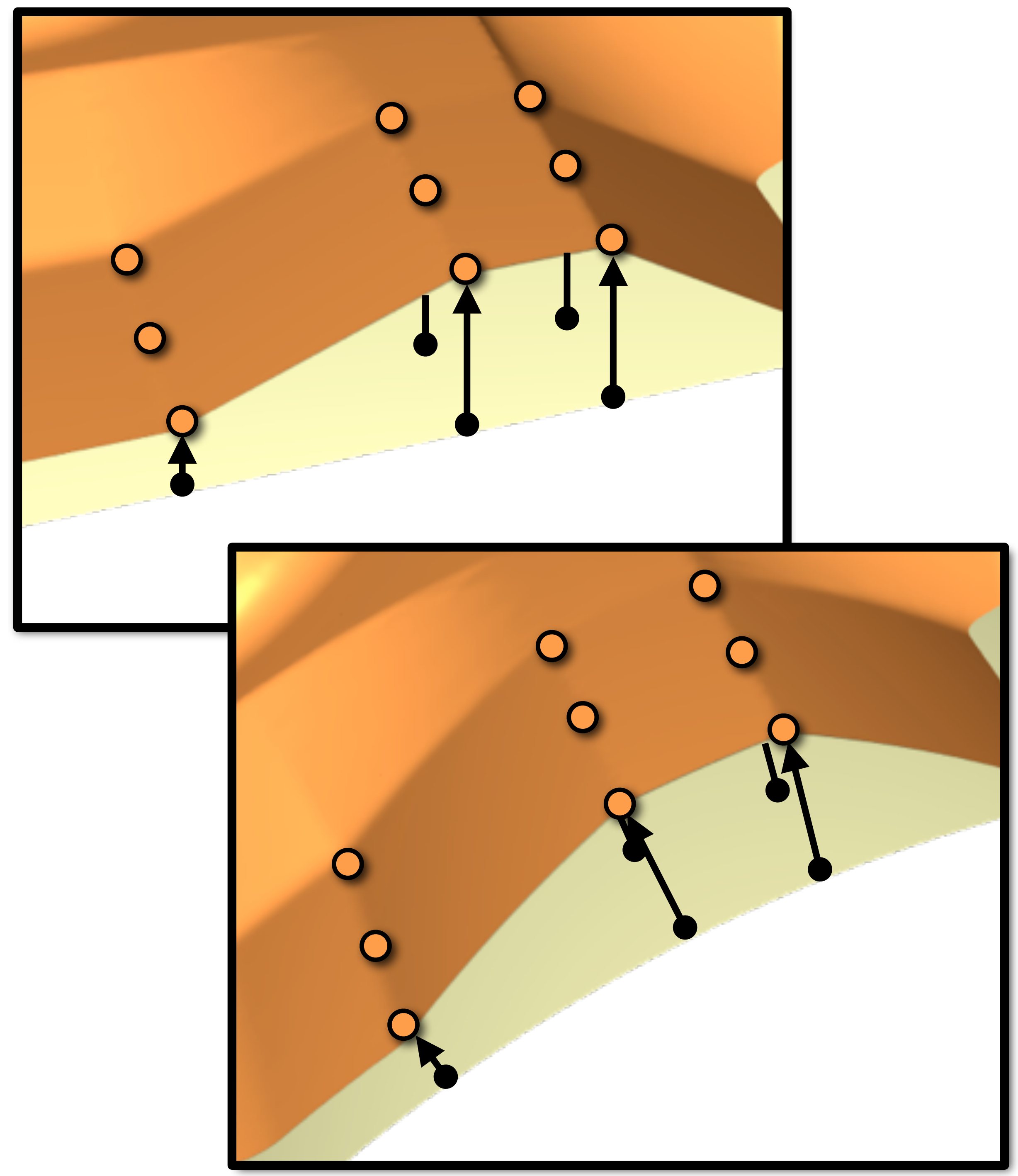
Multi-Scale Editing



[Kobbelt et al, SIGGRAPH 98]

Normal Displacements

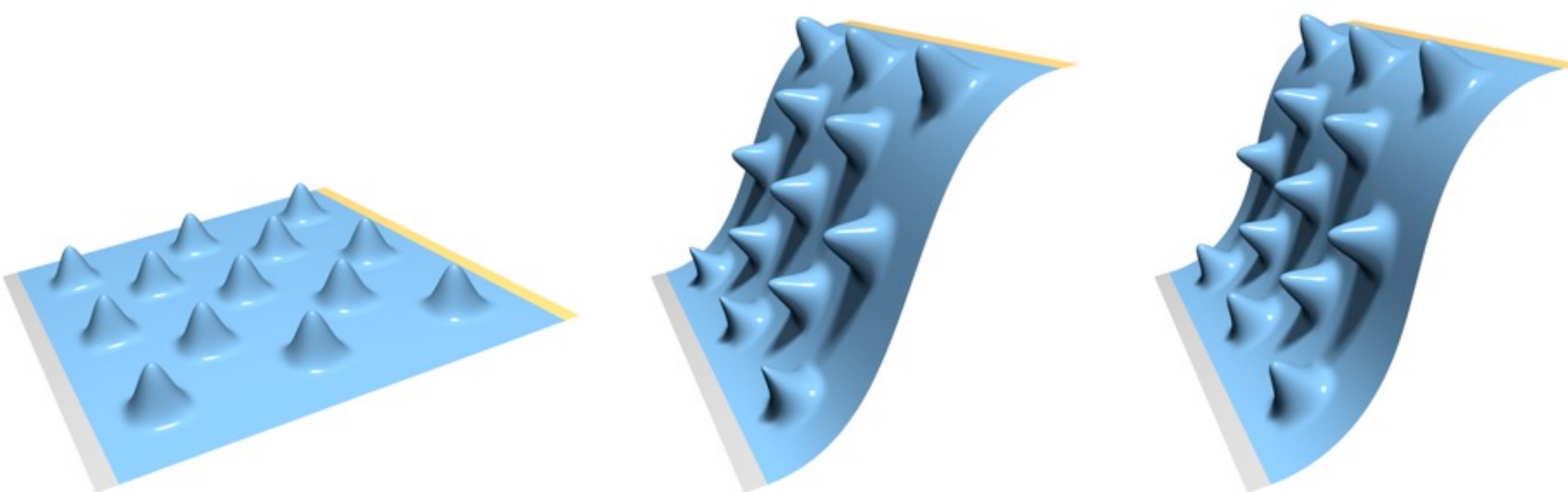




[Kobbelt et al, SIGGRAPH 98]

Limitations

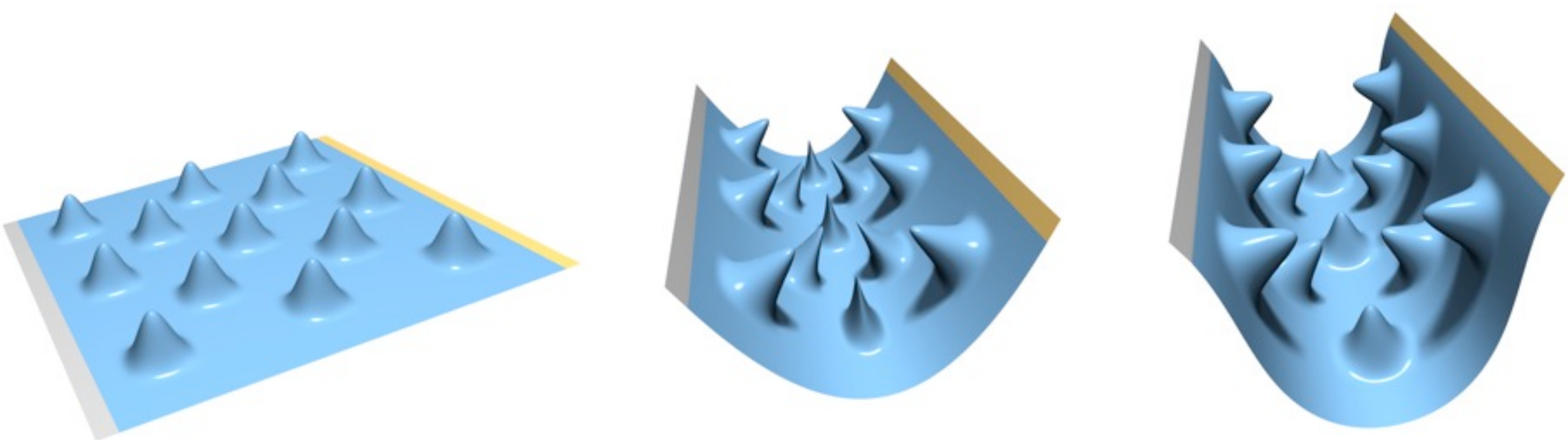
- Neighboring displacements are not coupled
  - Surface bending changes their angle
  - Leads to volume changes or self-intersections



[Botsch et al, EG 03, VMV 06]

Limitations

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



Original　　　Normal Displ.　　　Nonlinear

[Botsch et al, EG 03, VMV 06]

Literature

- Kobbelt et al, *Interactive multi-resolution modeling on arbitrary meshes*, SIGGRAPH 1998
- Kobbelt et al, *Multiresolution hierarchies on unstructured triangle meshes*, Comp. Geo. 1999
- Botsch & Kobbelt, *Multiresolution surface representation based on displacement volumes,* Eurographics 2003
- Botsch et al, *Deformation transfer for detail-preserving surface editing*, VMV 2006

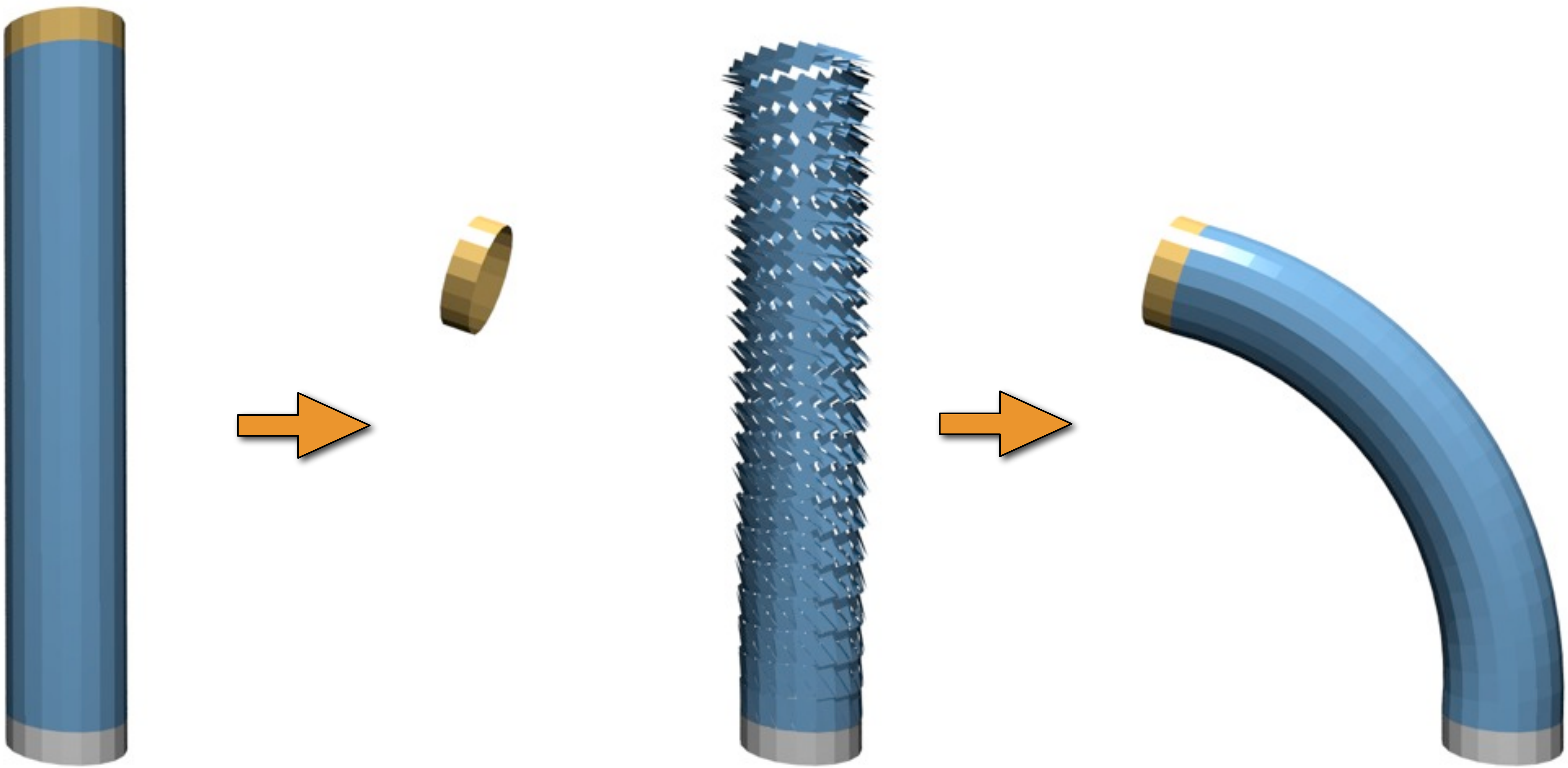
Linear Surface-Based Deformation

- Shell-Based Deformation
- Multi-Scale Deformation
- **Differential Coordinates**

Differential Coordinates

1. **Manipulate differential coordinates**
   - Gradients, Laplacians, local frames
   - Intuition: Close connection to surface normal
2. **Find mesh with these differential coords**
   - Cannot be solved exactly
   - Formulate as variational minimization

Differential Coordinates



Original Rotated Diff-Coords Reconstructed Mesh

Differential Coordinates

- **Which differential coordinate $\boldsymbol{\delta}_i$?**
  - Gradients
  - Laplacians
  - ...
- How to get local transformations $T_i(\boldsymbol{\delta}_i)$?
  - Smooth propagation
  - Implicit optimization
  - ...

Gradient-Based Editing

- Manipulate gradient of a function (e.g. a surface)

$$\mathbf{g} = \nabla \mathbf{f} \qquad \mathbf{g} \mapsto \mathbf{T}(\mathbf{g})$$

- Find function **f**' whose gradient is closest to **g**'

$$\mathbf{f}' = \underset{\mathbf{f}}{\operatorname{argmin}} \int_{\Omega} \left\| \nabla \mathbf{f} - \mathbf{T}(\mathbf{g}) \right\|^2 \mathrm{d}u \mathrm{d}v$$

- Variational calculus → Euler-Lagrange PDE

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

[Yu et al, SIGGRAPH 04]

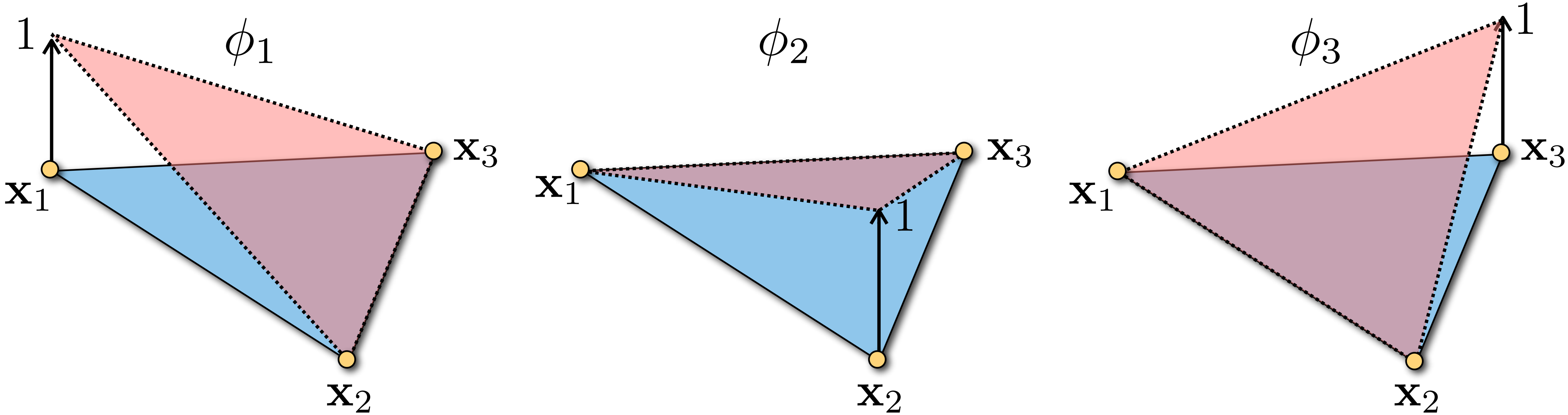
Gradient-Based Editing

- Consider piecewise linear ***coordinate function***

$$\mathbf{p}(u,v) = \sum_{v_i} \mathbf{p}_i \cdot \phi_i(u,v)$$

- Its gradient is

$$\nabla \mathbf{p}(u,v) = \sum_{v_i} \mathbf{p}_i \cdot \nabla \phi_i(u,v)$$



Gradient-Based Editing

- Consider piecewise linear ***coordinate function***

$$\mathbf{p}(u,v) = \sum_{v_i} \mathbf{p}_i \cdot \phi_i(u,v)$$

- Its gradient is

$$\nabla\mathbf{p}(u,v) = \sum_{v_i} \mathbf{p}_i \cdot \nabla\phi_i(u,v)$$

- It is constant per triangle

$$\nabla\mathbf{p}\big|_{f_j} =: \mathbf{g}_j \in \mathbb{R}^{3\times 3}$$

Gradient-Based Editing

- Gradient of coordinate function $\mathbf{p}$

$$\begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_F \end{pmatrix} = \underbrace{\mathbf{G}}_{(3F \times V)} \begin{pmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_V^T \end{pmatrix}$$

- Manipulate per-face gradients

$$\mathbf{g}_j \;\mapsto\; \mathbf{T}_j(\mathbf{g}_j)$$

Gradient-Based Editing

- Reconstruct mesh from new gradients
  - Overdetermined $(3F \times V)$ system
  - Weighted least squares system
  - ➡Linear Poisson system $\Delta \mathbf{p}' = \operatorname{div} \mathbf{T}(\mathbf{g})$

$$\underbrace{\mathbf{G}^T\mathbf{D}\mathbf{G}}_{\operatorname{div}\nabla = \Delta} \cdot \begin{pmatrix} {\mathbf{p}'_1}^T \\ \vdots \\ {\mathbf{p}'_V}^T \end{pmatrix} = \underbrace{\mathbf{G}^T\mathbf{D}}_{\operatorname{div}} \cdot \begin{pmatrix} \mathbf{T}_1(\mathbf{g}_1) \\ \vdots \\ \mathbf{T}_F(\mathbf{g}_F) \end{pmatrix}$$

Laplacian-Based Editing

- Manipulate Laplacians field of a surface

$$\mathbf{l} = \Delta(\mathbf{p}) \ , \quad \mathbf{l} \mapsto \mathbf{T}(\mathbf{l})$$

- Find surface whose Laplacian is closest to $\boldsymbol{\delta}$'

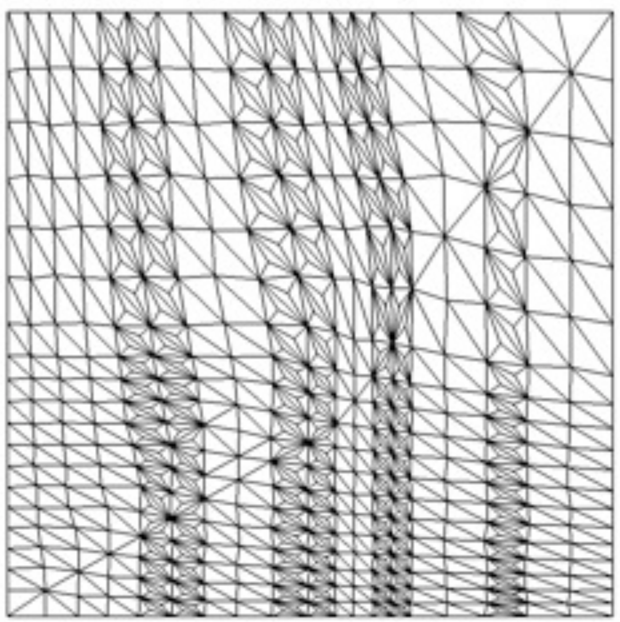
$$\mathbf{p}' = \underset{\mathbf{p}}{\operatorname{argmin}} \int_{\Omega} \left\| \Delta\mathbf{p} - \mathbf{T}(\mathbf{l}) \right\|^2 \mathrm{d}u \mathrm{d}v$$

- Variational calculus yields Euler-Lagrange PDE

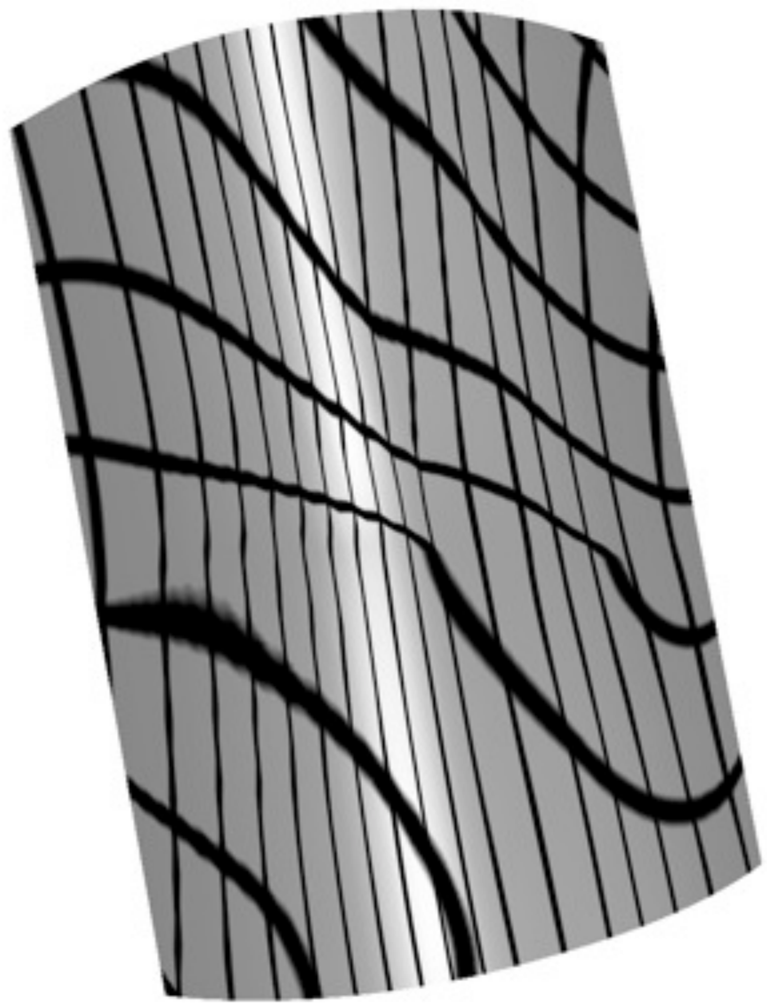
$$\Delta^2 \mathbf{p}' = \Delta \mathbf{T}(\mathbf{l})$$

[Sorkine et al, SGP 04]

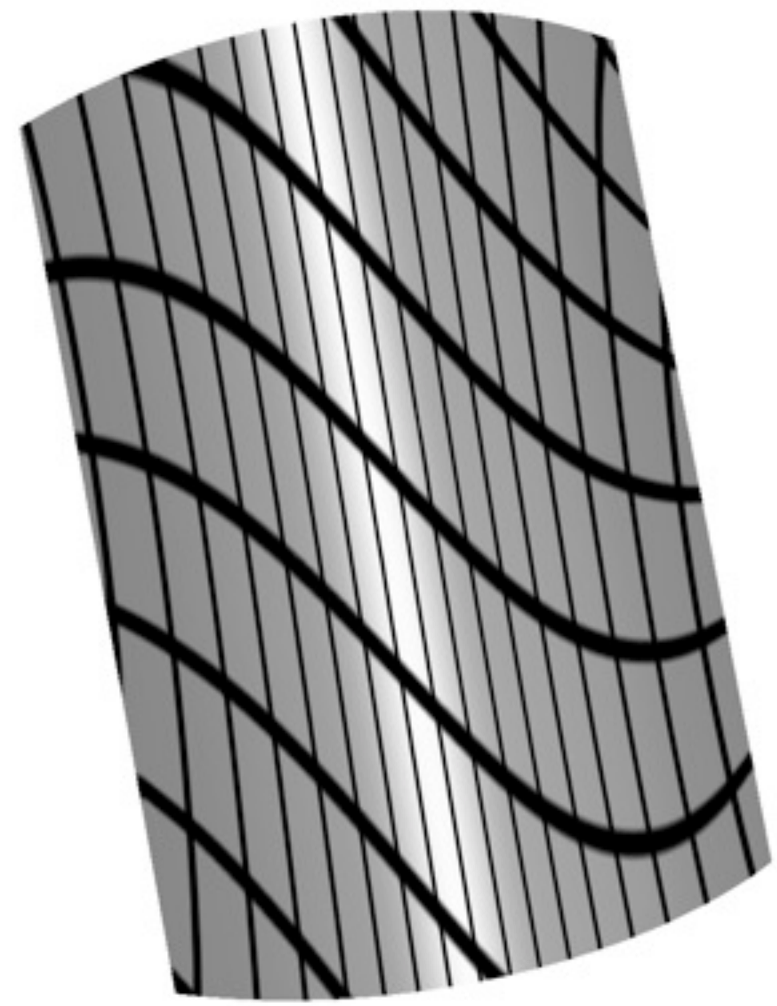
Careful Discretization!



Irregular mesh



$$\mathbf{L}^T\mathbf{L}\mathbf{p}' = \mathbf{L}^T\boldsymbol{\delta}'$$



$$\mathbf{L}^2\mathbf{p}' = \mathbf{L}\boldsymbol{\delta}'$$

[Botsch & Sorkine, TVCG 08]

Differential Coordinates

- Which differential coordinate $\boldsymbol{\delta}_i$?
  - Gradients
  - Laplacians
  - ...

- **How to get local transformations $\mathbf{T}_i(\boldsymbol{\delta}_i)$?**
  - Smooth propagation
  - Implicit optimization
  - ...

Smooth Propagation

1. Compute handle’s deformation gradient
2. Extract rotation and scale/shear components
3. Propagate damped rotations over ROI

Deformation Gradient

- Handle has been transformed *affinely*

$$\mathbf{T}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$

- Deformation gradient is

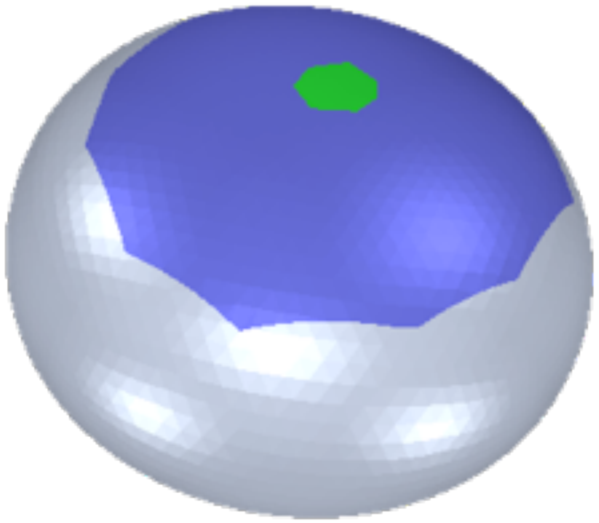
$$\nabla \mathbf{T}(\mathbf{x}) = \mathbf{A}$$

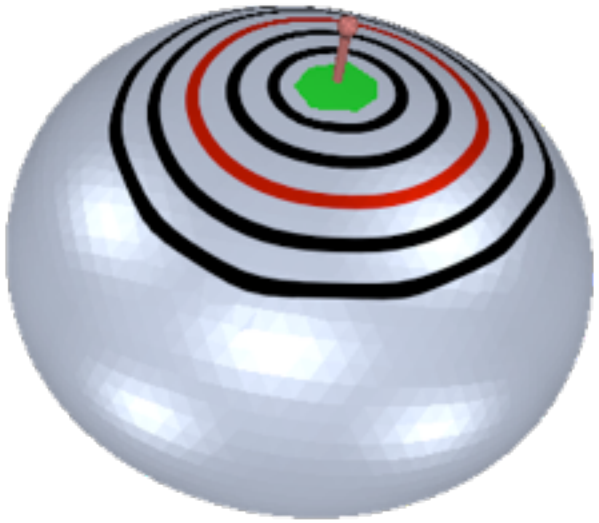
- Extract rotation $\mathbf{R}$ and scale/shear $\mathbf{S}$ by *polar decomposition*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \Rightarrow \quad \mathbf{R} = \mathbf{U}\mathbf{V}^T,\ \mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$$

Smooth Propagation

- Construct smooth scalar field [0,1]
  - $s(\mathbf{x})=1$: Full deformation (handle)
  - $s(\mathbf{x})=0$: No deformation (fixed part)
  - $s(\mathbf{x})\in(0,1)$: Damp handle transformation in between

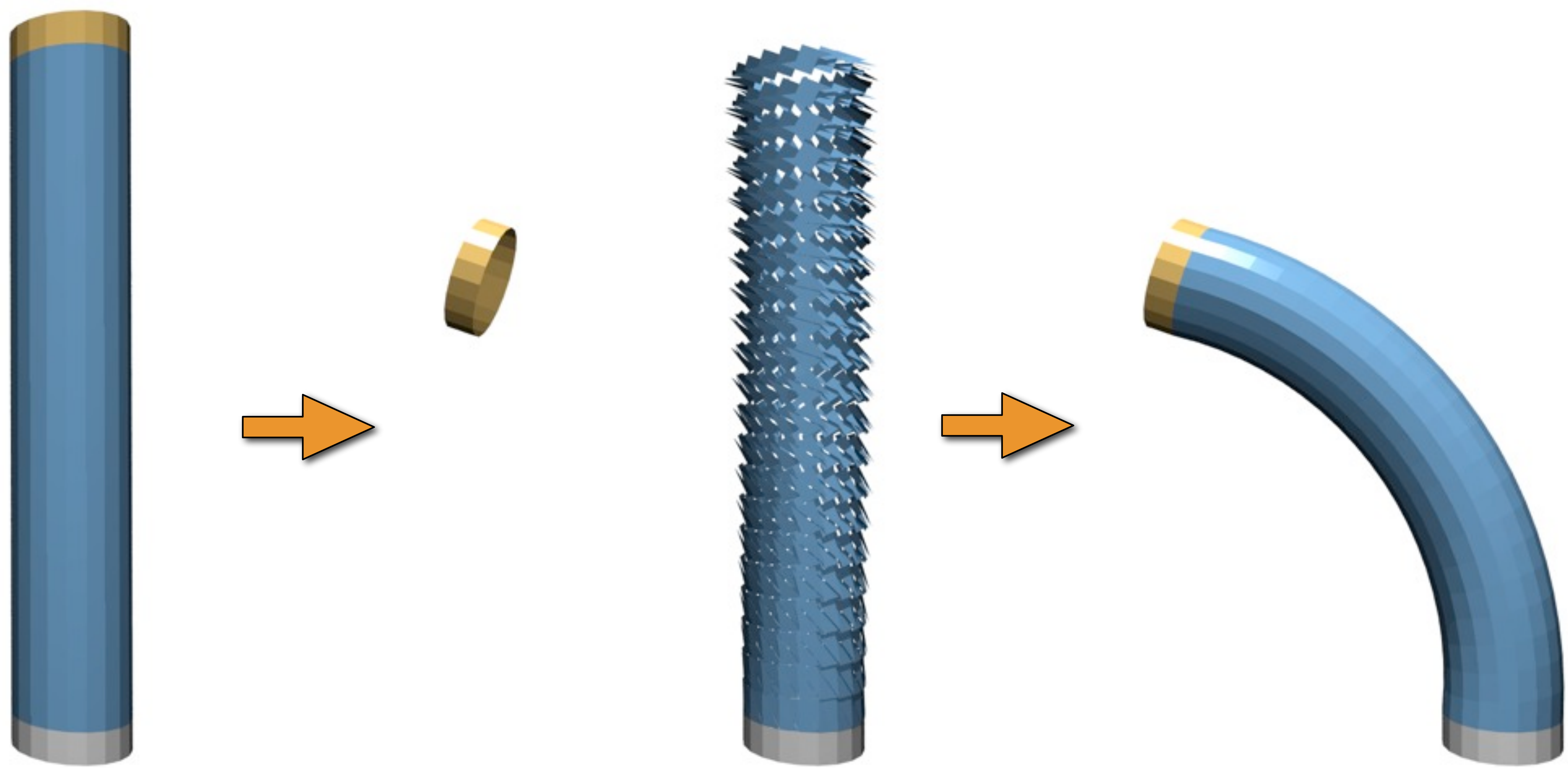




Damp Handle Transformation

- Full handle transformation
  - Rotation: $R(\mathbf{c}, \mathbf{a}, \alpha)$
  - Scaling: $S(s)$
- Damped by scalar λ
  - Rotation: $R(\mathbf{c}, \mathbf{a}, \lambda \cdot \alpha)$
  - Scaling: $S(\lambda \cdot s + (1-\lambda) \cdot 1)$

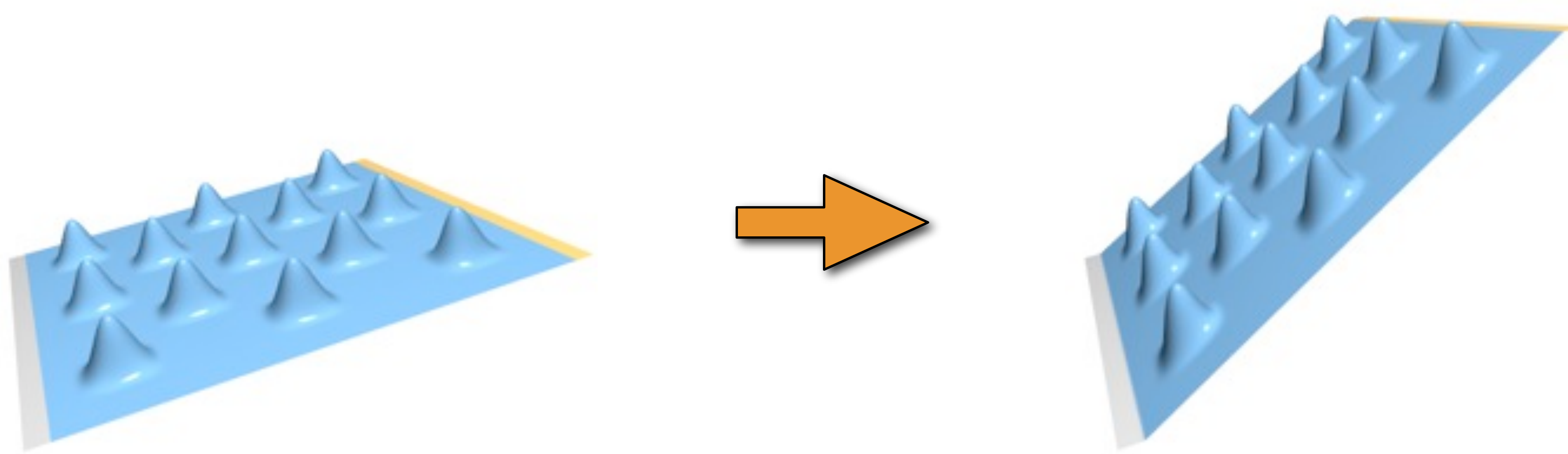
Differential Coordinates



Original Rotated Diff-Coords Reconstructed Mesh

Limitations

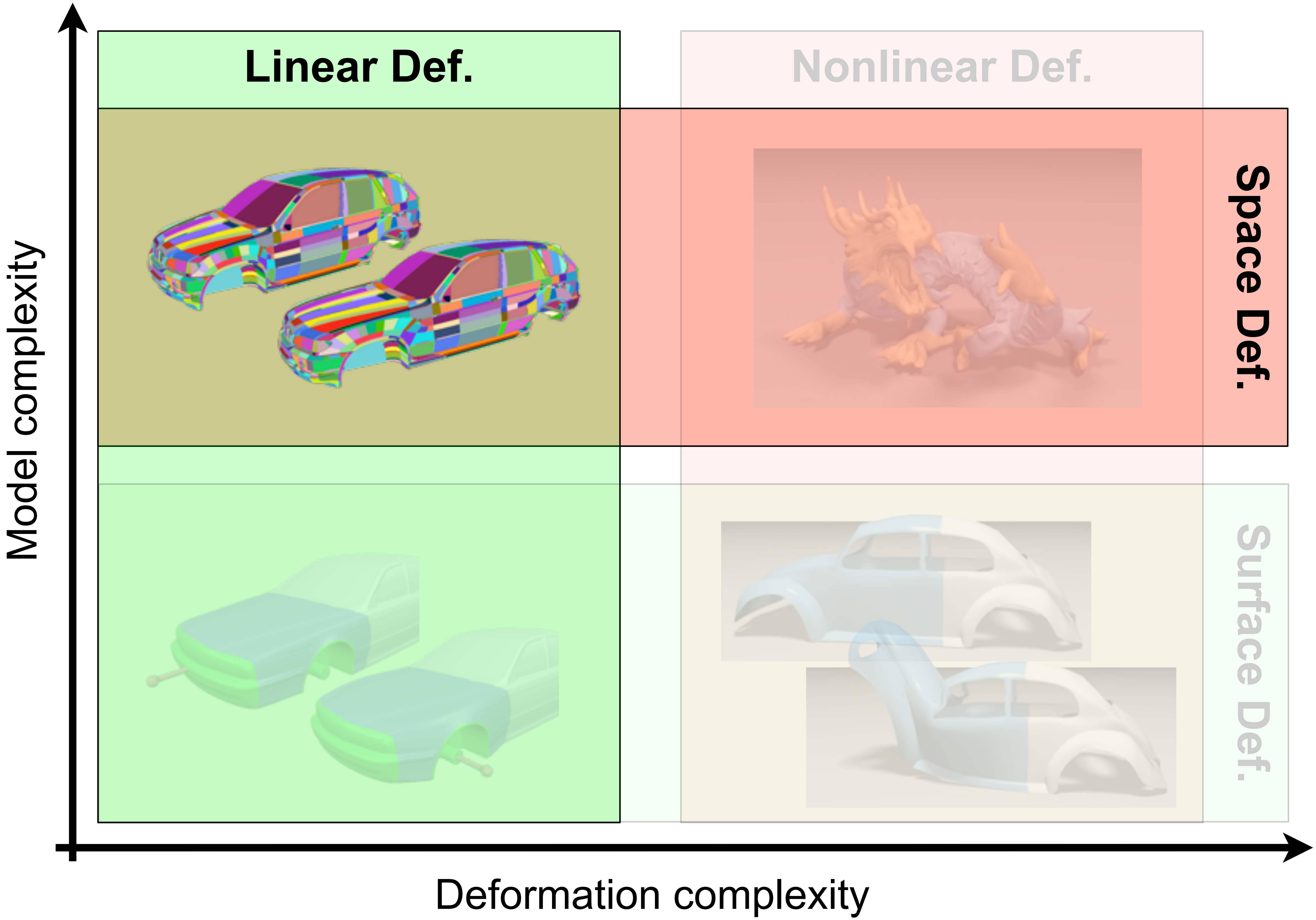
- Differential coordinates work well for **rotations**
  - Represented by deformation gradient
- **Translations** don't change deformation gradient
  - Translations don't change differential coordinates
  - *"Translation insensitivity"*



Literature

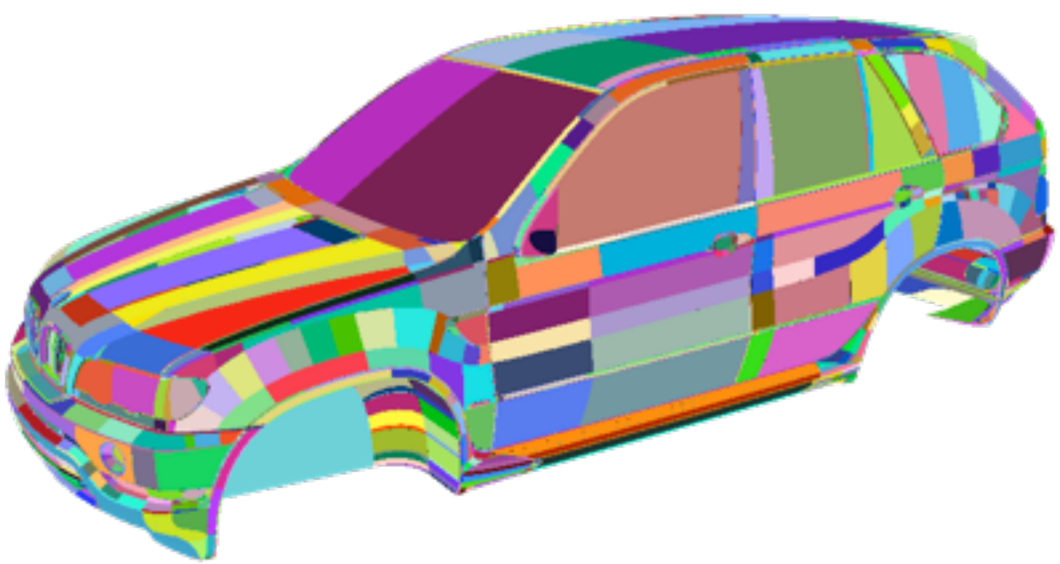
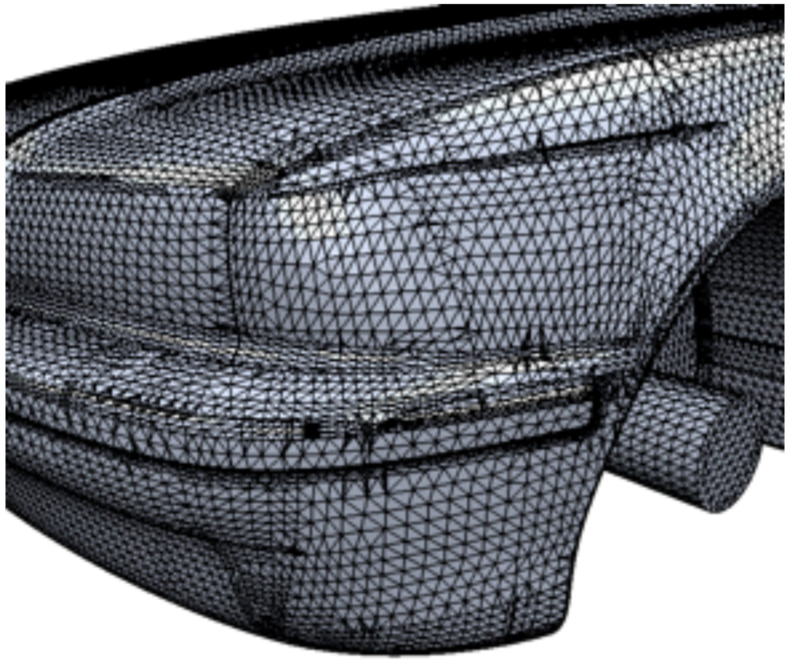
- Yu et al, *Mesh editing with Poisson-based gradient field manipulation*, SIGGRAPH 2004
- Sorkine et al, *Laplacian surface editing*, SGP 2004

Shape Deformation



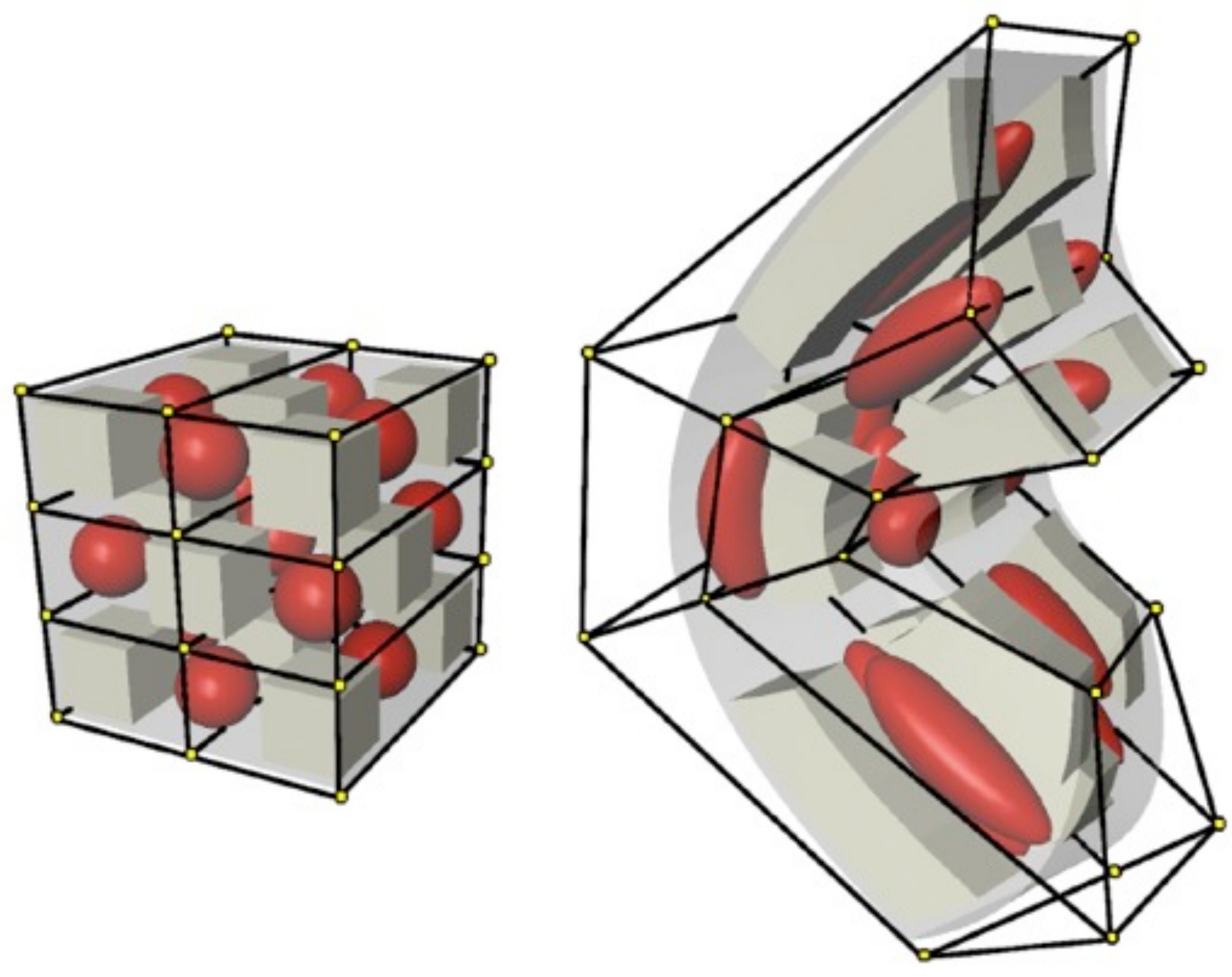
Surface-Based Deformation

- Problems with
  - Highly complex models
  - Topological inconsistencies
  - Geometric degeneracies

Freeform Deformation

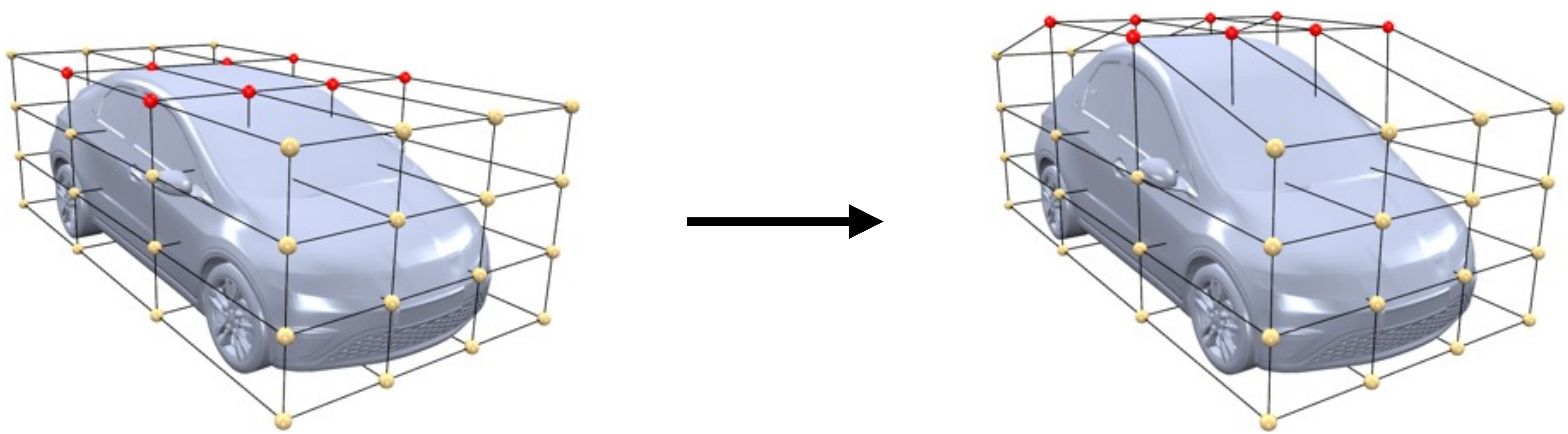
- Deform object's bounding box
  - Implicitly deforms embedded objects



Freeform Deformation (FFD)

- Trivariate tensor-product spline

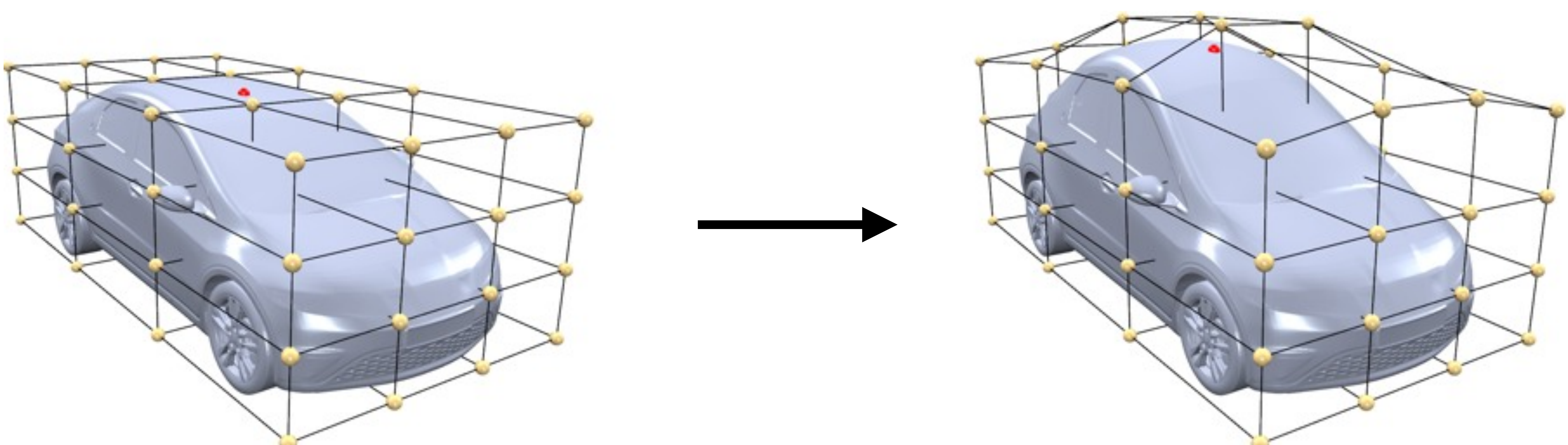
$$\mathbf{d}(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \mathbf{d}_{ijk} N_i(u)\, N_j(v)\, N_k(w)$$



[Sederberg & Perry, SIGGRAPH 87]

Direct Manipulation FFD

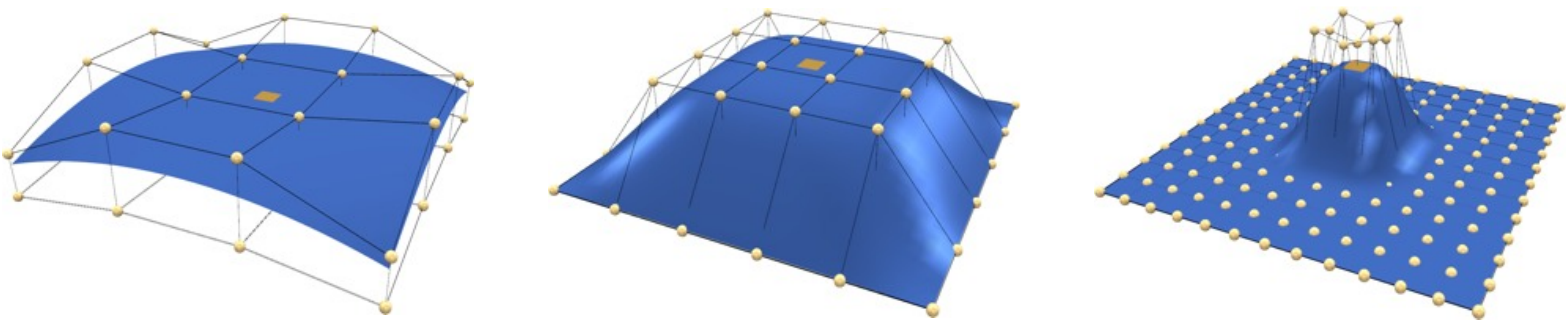
- How to prescribe displacement constraints?
  - Solve linear system for control points
  - Can be over- or under-determined
  - Pseudo-inverse: least squares, least norm



[Hsu et al, SIGGRAPH 92]

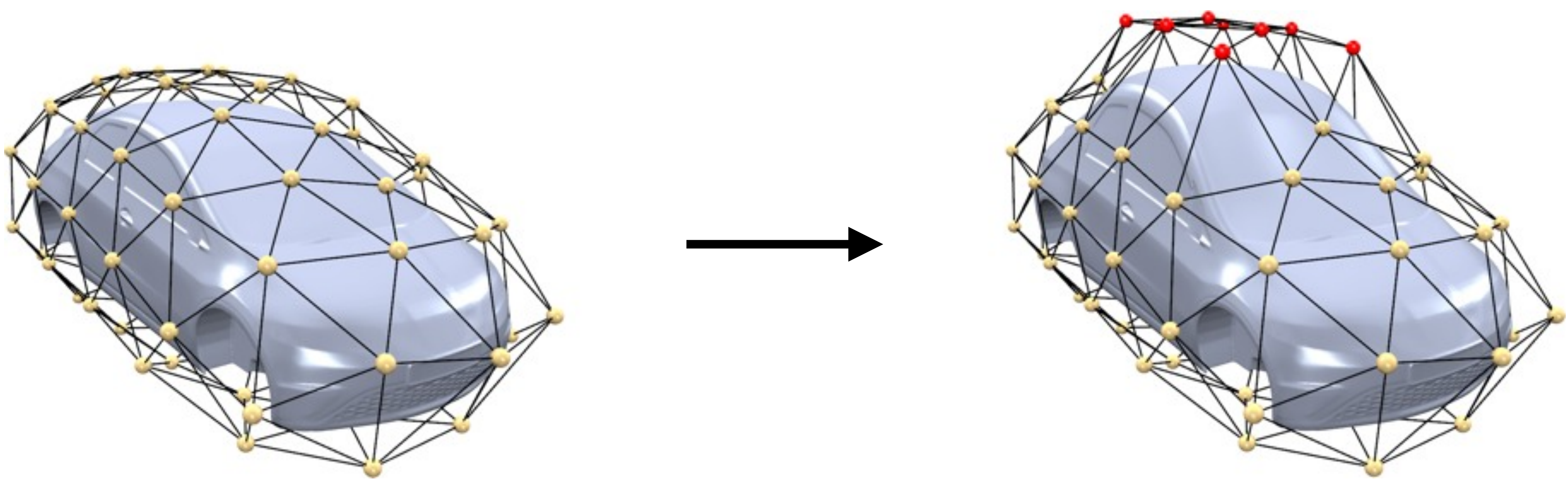
Direct Manipulation

- Depends a lot on grid resolution
  - Minimize control point movement ≠ minimize physical energies!



Cage Deformation

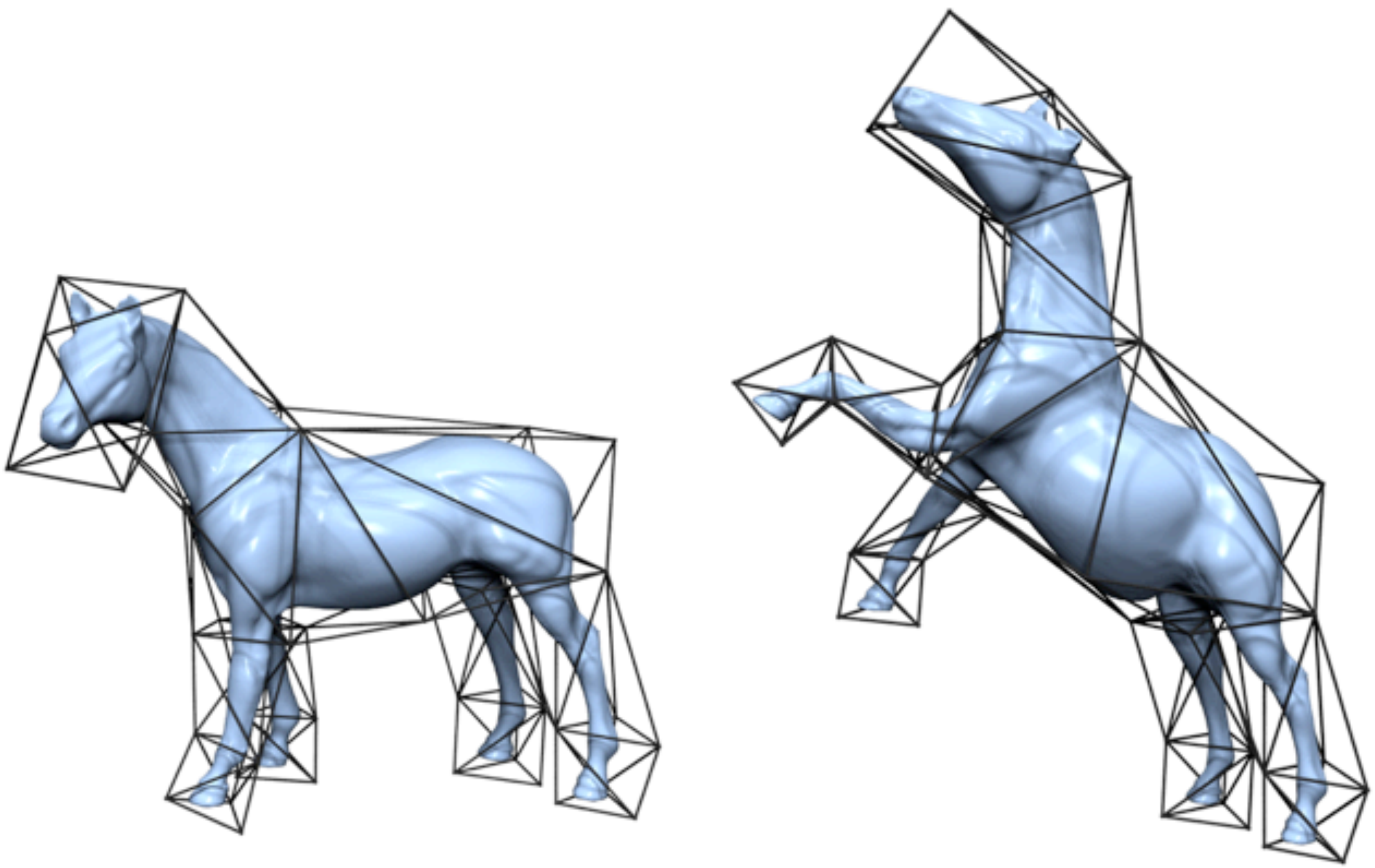
- Deform object through *control cage*
  - Spline control points → cage vertices
  - Spline basis → generalized barycentric coordinates



[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]

Cage Deformation

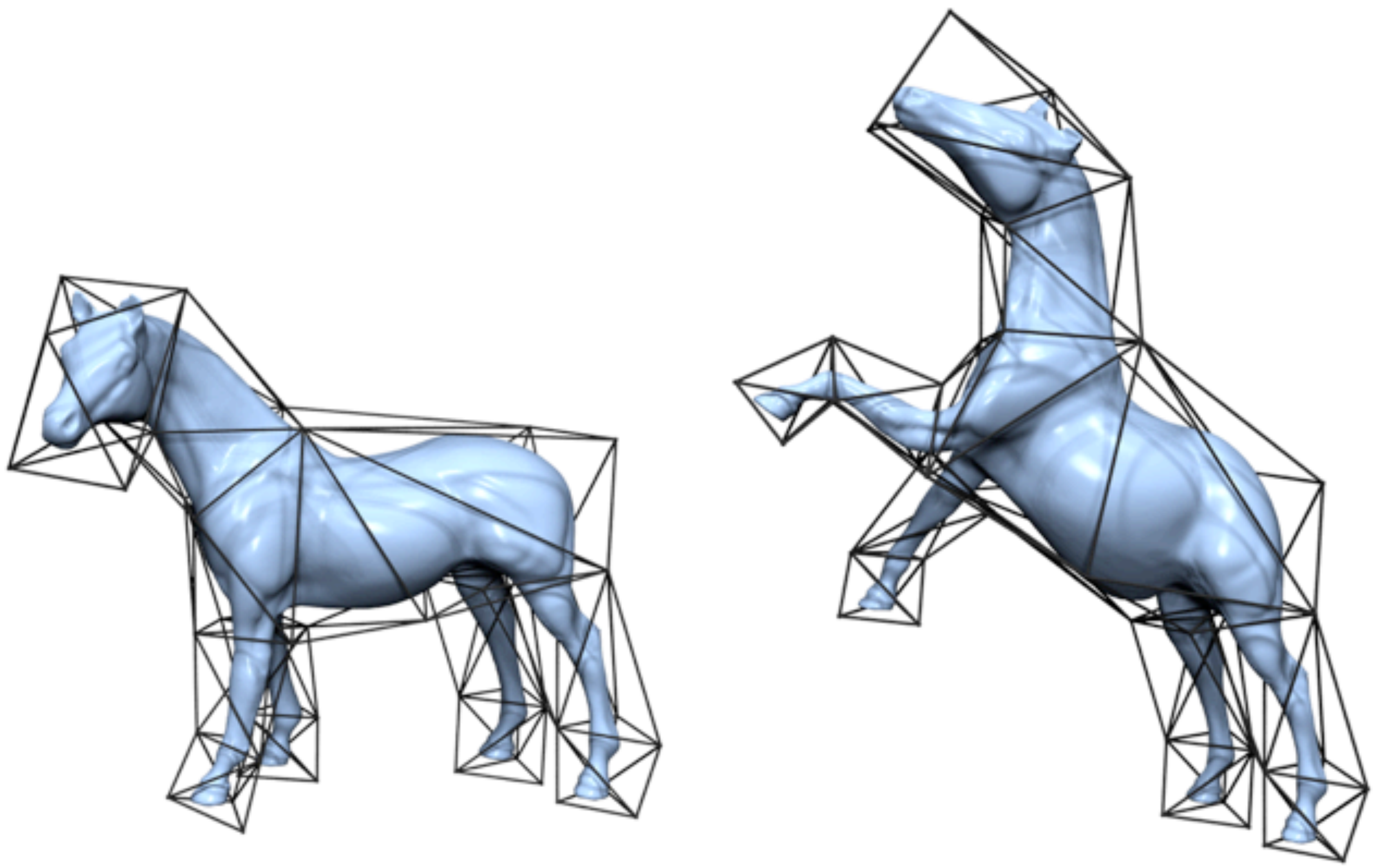
- Deform object through *control cage*
  - Spline control points → cage vertices
  - Spline basis → generalized barycentric coordinates



[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]

Cage Deformation

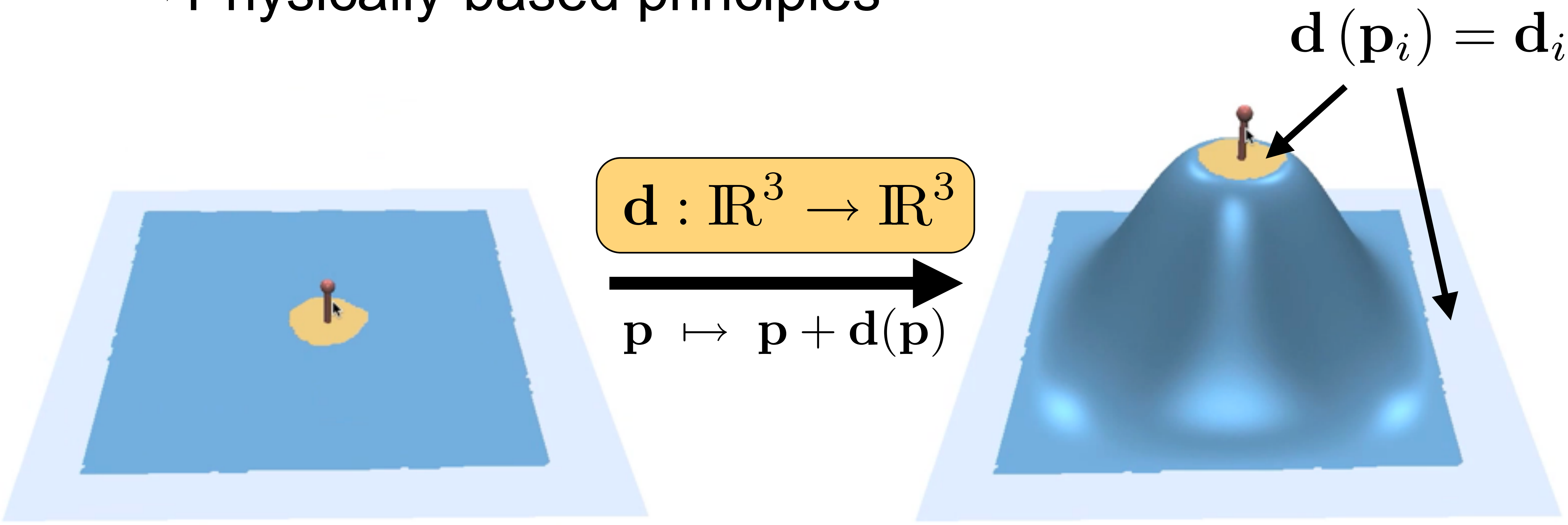
- Deform object through *control cage*
  - More flexible than spline control grids
  - Same limitation for direct manipulation



[Ju et al, SIGGRAPH 05], [Joshi et al, SIGGRAPH 07], [Lipman, SIGGRAPH 08]

Space Deformation

- Mesh deformation by displacement function **d**
  - Interpolate prescribed constraints
  - Smooth, intuitive deformation
  - ➡ Physically-based principles



Volumetric Energy Minimization

- Minimize similar energies to surface case

$$\int_{\mathbb{R}^3} \|\mathbf{d}_{uu}\|^2 + \|\mathbf{d}_{uv}\|^2 + \ldots + \|\mathbf{d}_{ww}\|^2 \;\mathrm{d}V \;\rightarrow\; \min$$

- But displacement function lives in 3D...
  – Need a volumetric space tessellation?
  – No, same functionality provided by RBFs

Radial Basis Functions

- Represent deformation by RBFs

$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi(\|\mathbf{c}_j - \mathbf{x}\|) + \mathbf{p}(\mathbf{x})$$

- Choose basis function $\varphi(r) = r^3$
  - Function $\mathbf{d}$ is triharmonic $\Delta^3 \mathbf{d} = 0$
  - Minimizes fairness energy

$$\int_{\mathbb{R}^3} \|\mathbf{d}_{uuu}\|^2 + \|\mathbf{d}_{vuu}\|^2 + \ldots + \|\mathbf{d}_{www}\|^2 \, \mathrm{d}u \, \mathrm{d}v \, \mathrm{d}w \;\rightarrow\; \min$$

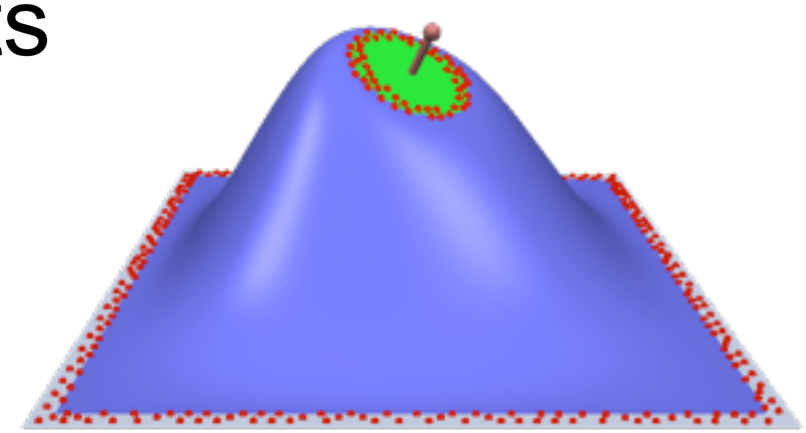
[Botsch & Kobbelt, EG 05]

RBF Deformation

- Represent deformation by RBFs

$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi(\|\mathbf{c}_j - \mathbf{x}\|) + \mathbf{p}(\mathbf{x})$$

1. **RBF fitting**
   - Interpolate displacement constraints
   - Solve linear system for $\mathbf{w}_j$ and $\mathbf{p}$



[Botsch & Kobbelt, EG 05]

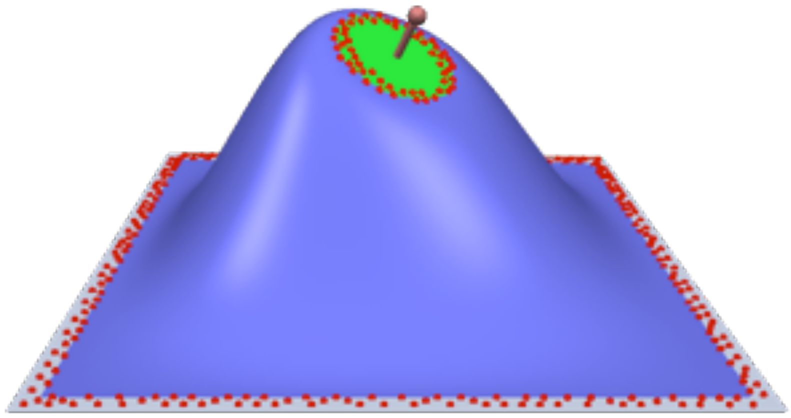
RBF Deformation

- Represent deformation by RBFs

$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi(\|\mathbf{c}_j - \mathbf{x}\|) + \mathbf{p}(\mathbf{x})$$

2. RBF evaluation

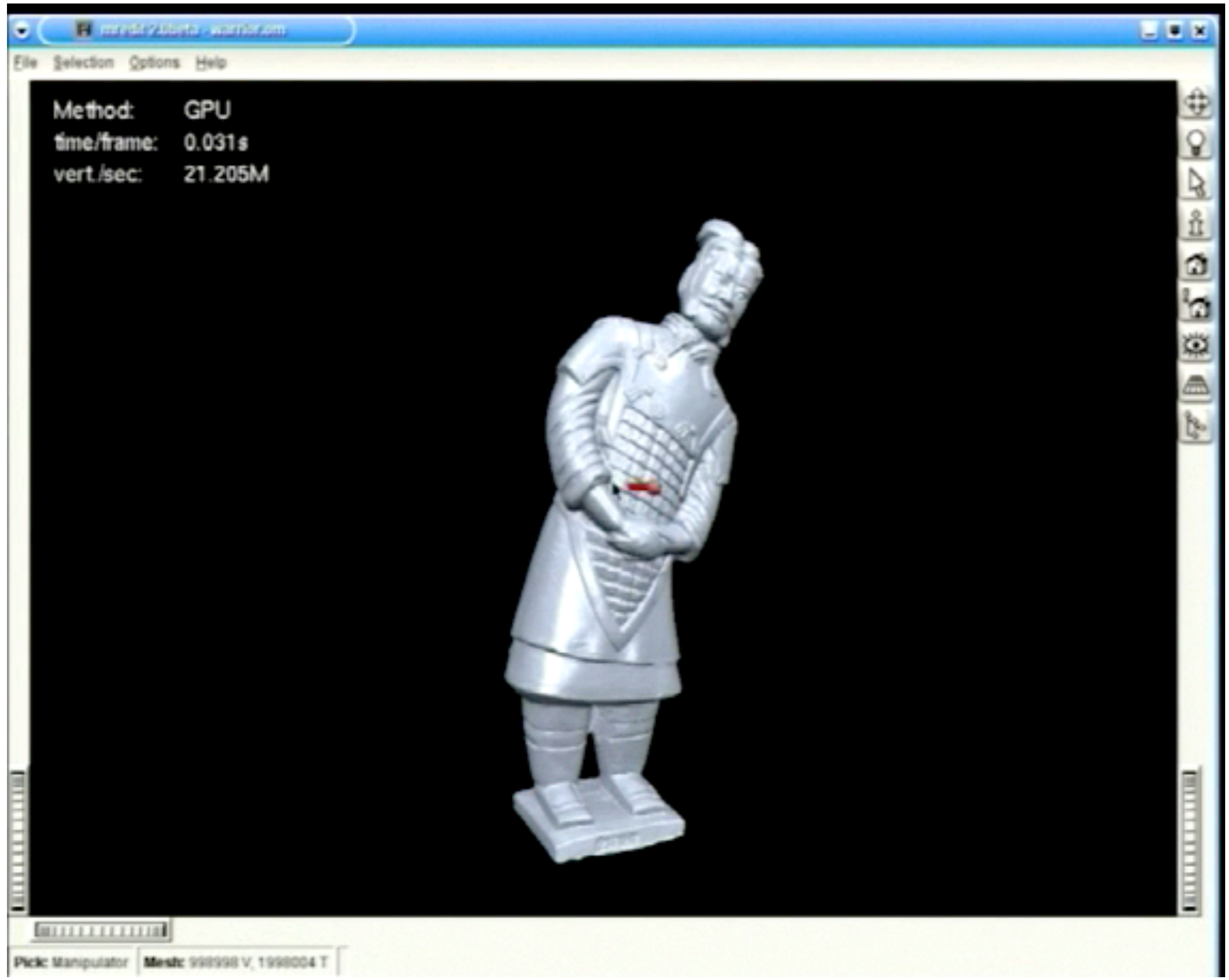
- Function $\mathbf{d}$ transforms points
- Jacobian $(\nabla\mathbf{d})^{-T}$ transforms normals
- Evaluate on the GPU!



[Botsch & Kobbelt, EG 05]

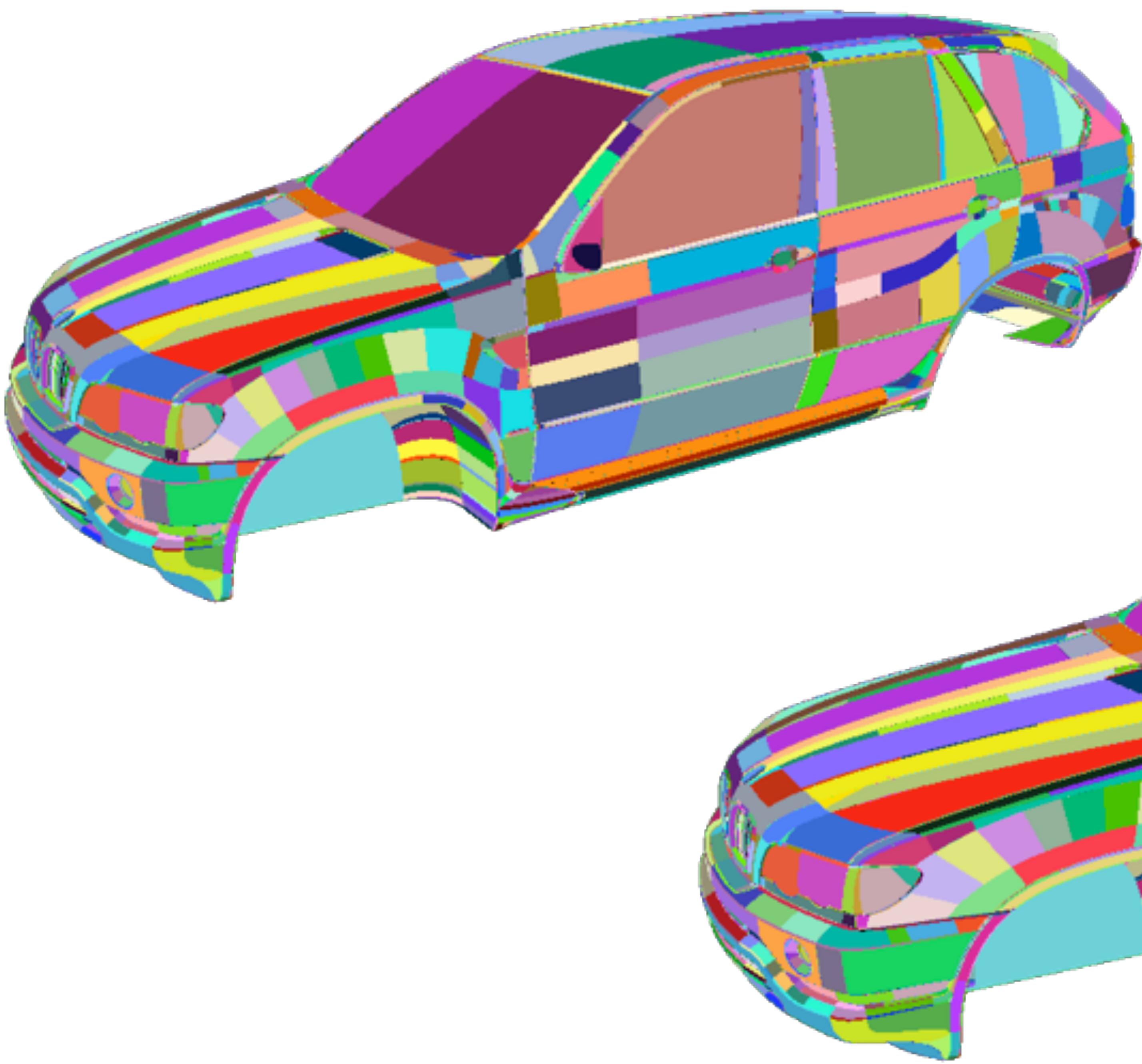
RBF Deformation

1M vertices



[Botsch & Kobbelt, EG 05]

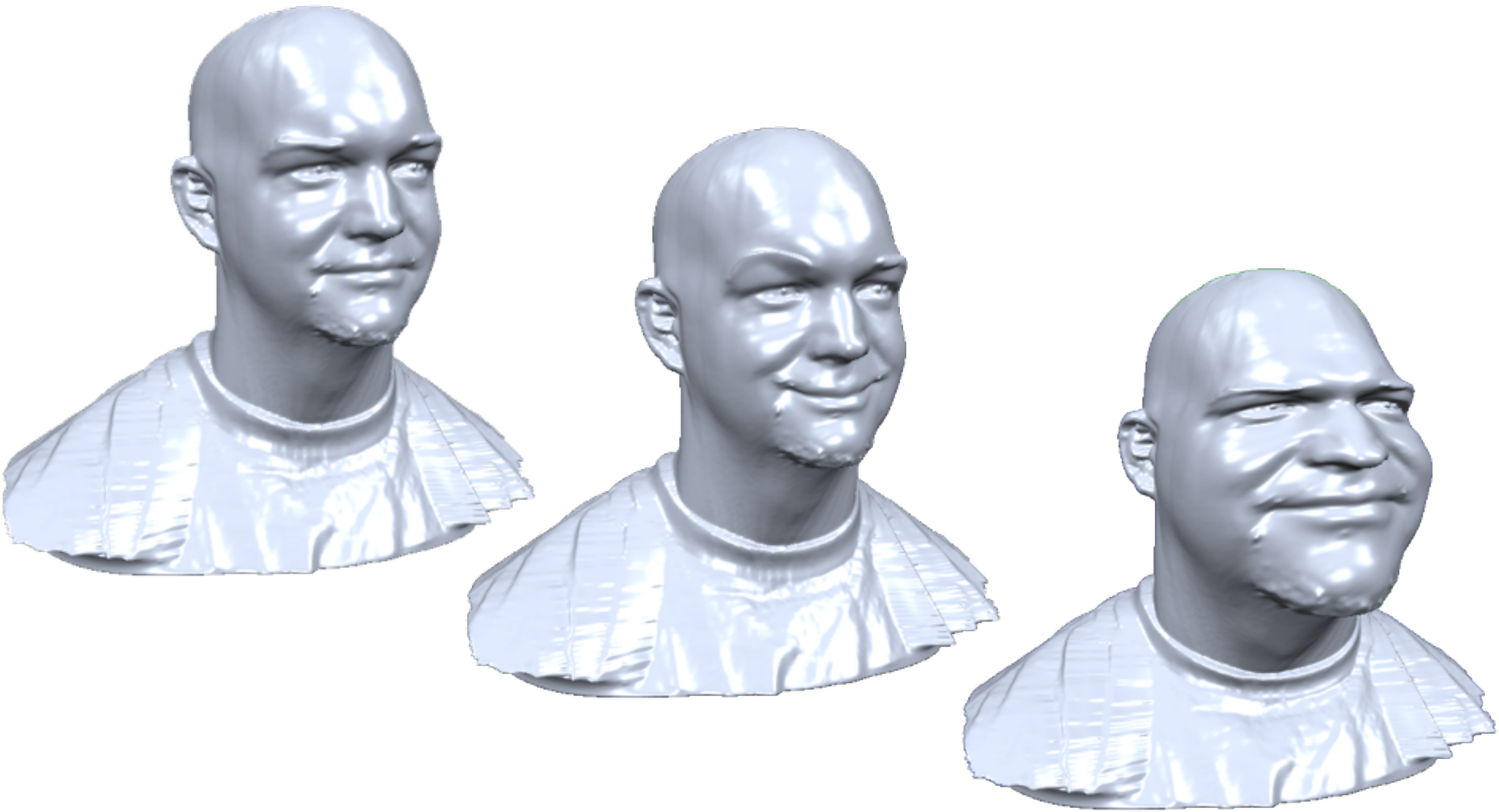
“Bad Meshes”



- 3M triangles
- 10k components
- Not oriented
- Not manifold

[Botsch & Kobbelt, EG 05]

Local & Global Deformations

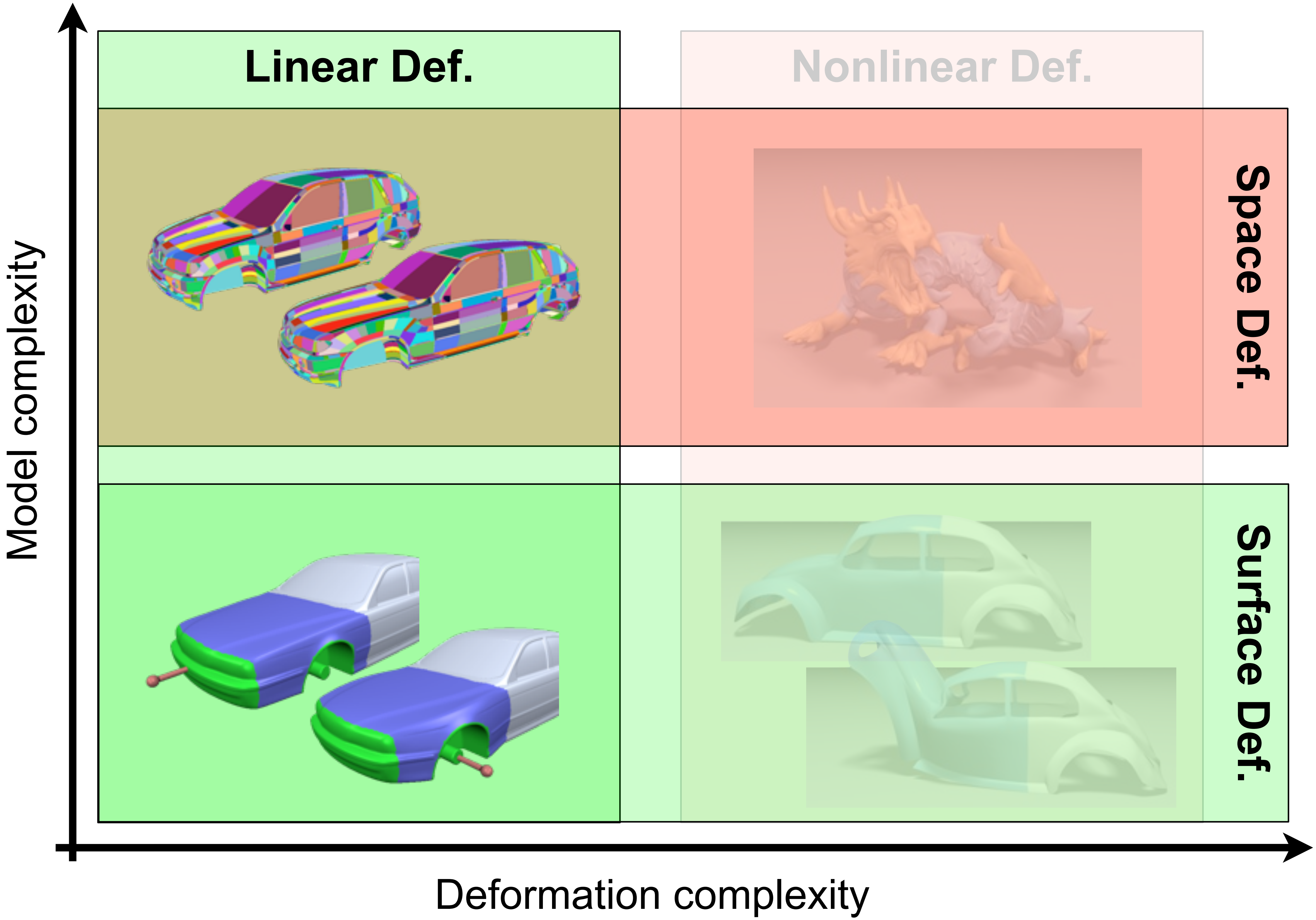


[Botsch & Kobbelt, EG 05]

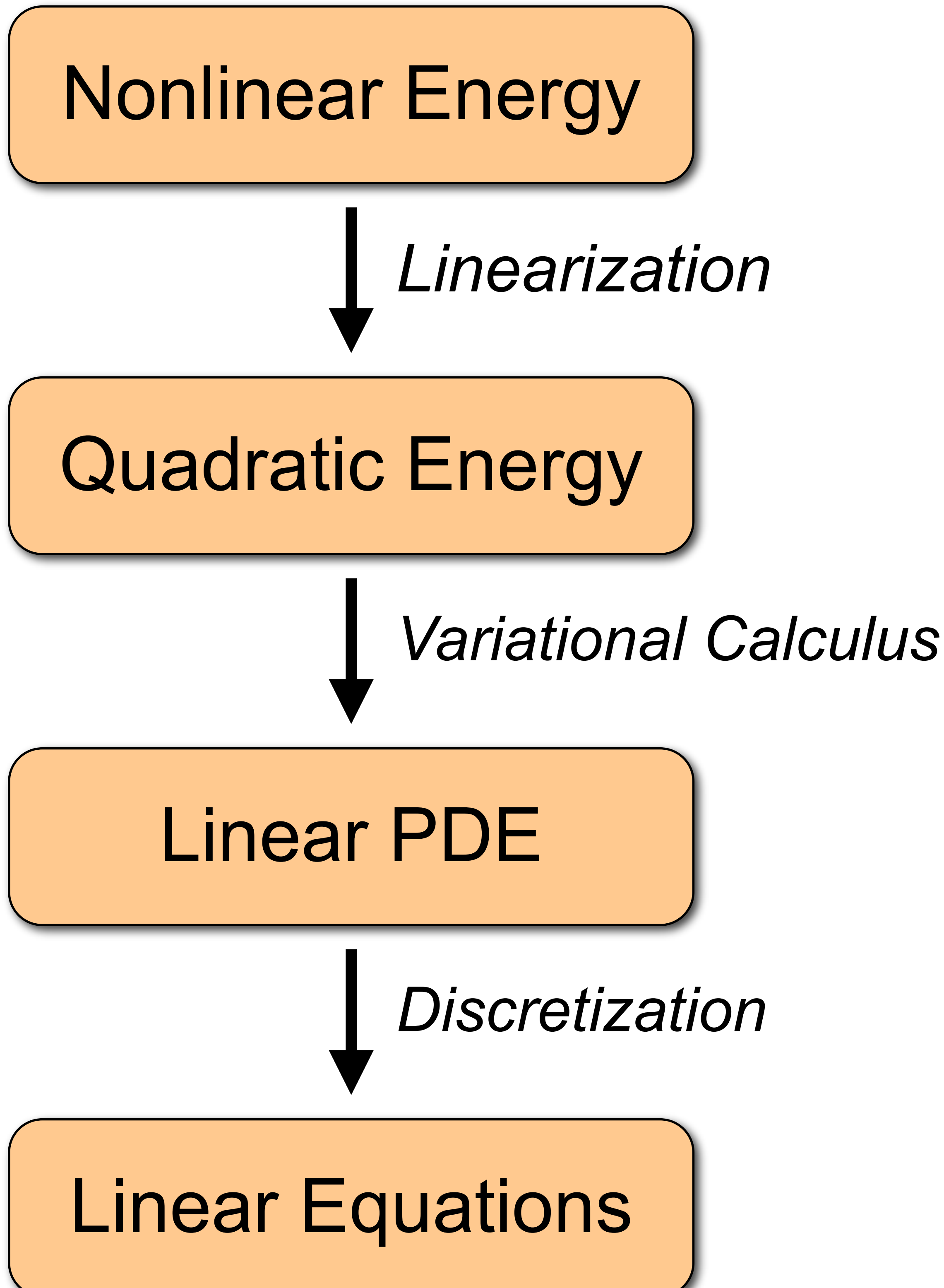
Literature

- Sederberg & Parry, *Free-Form Deformation of Solid Geometric Models*, SIGGRAPH 1986
- Botsch & Kobbelt, *Real-time shape editing using radial basis functions*, Eurographics 2005
- Ju et al, *Mean value coordinates for closed triangular meshes*, SIGGRAPH 2005
- Joshi et al, *Harmonic coordinates for character animation*, SIGGRAPH 2007
- Lipman et al, *Green coordinates*, SIGGRAPH 2008

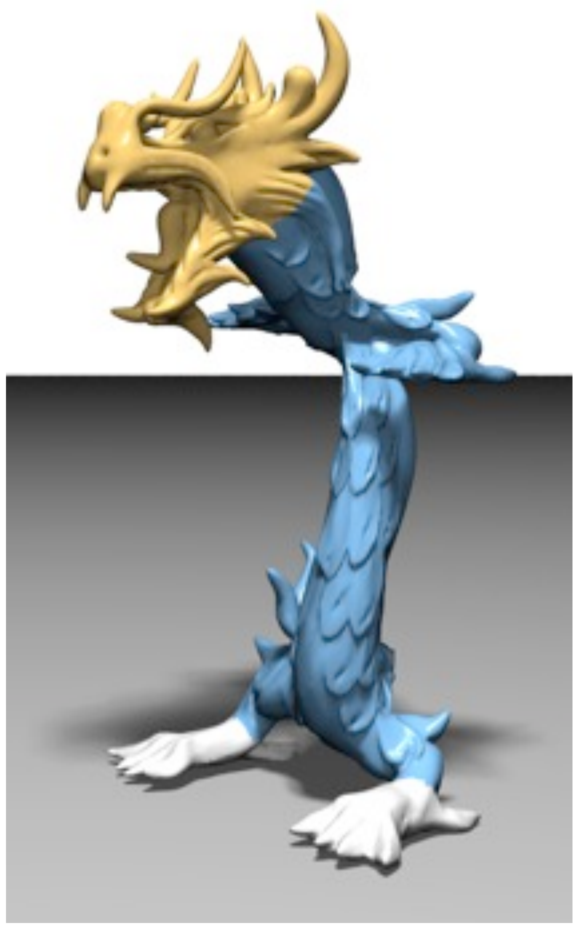
Shape Deformation

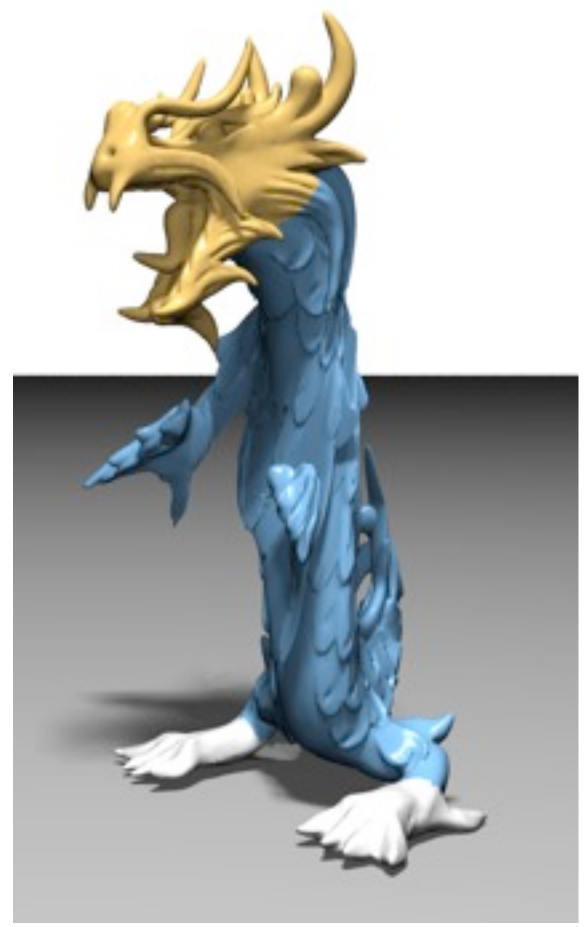


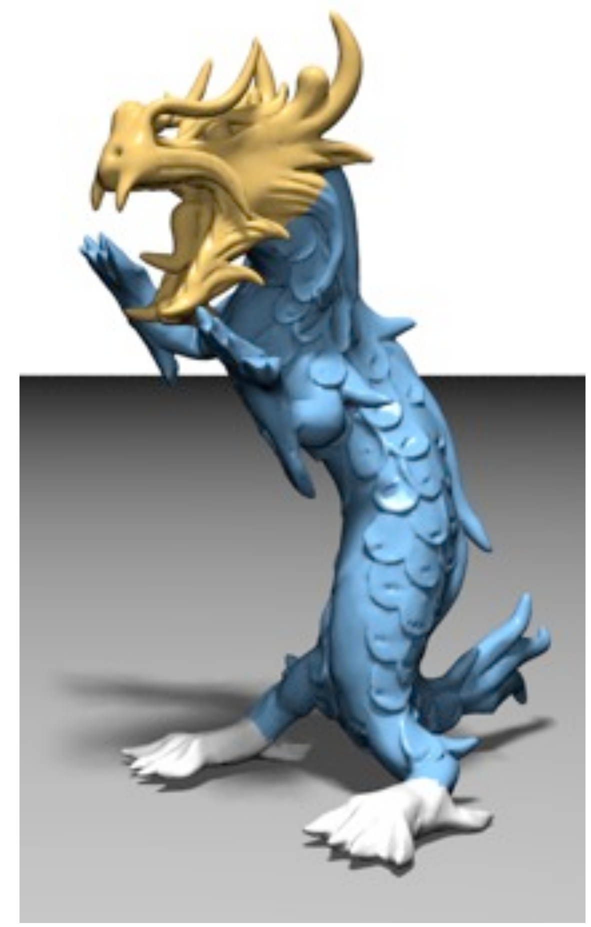
Derivation Steps



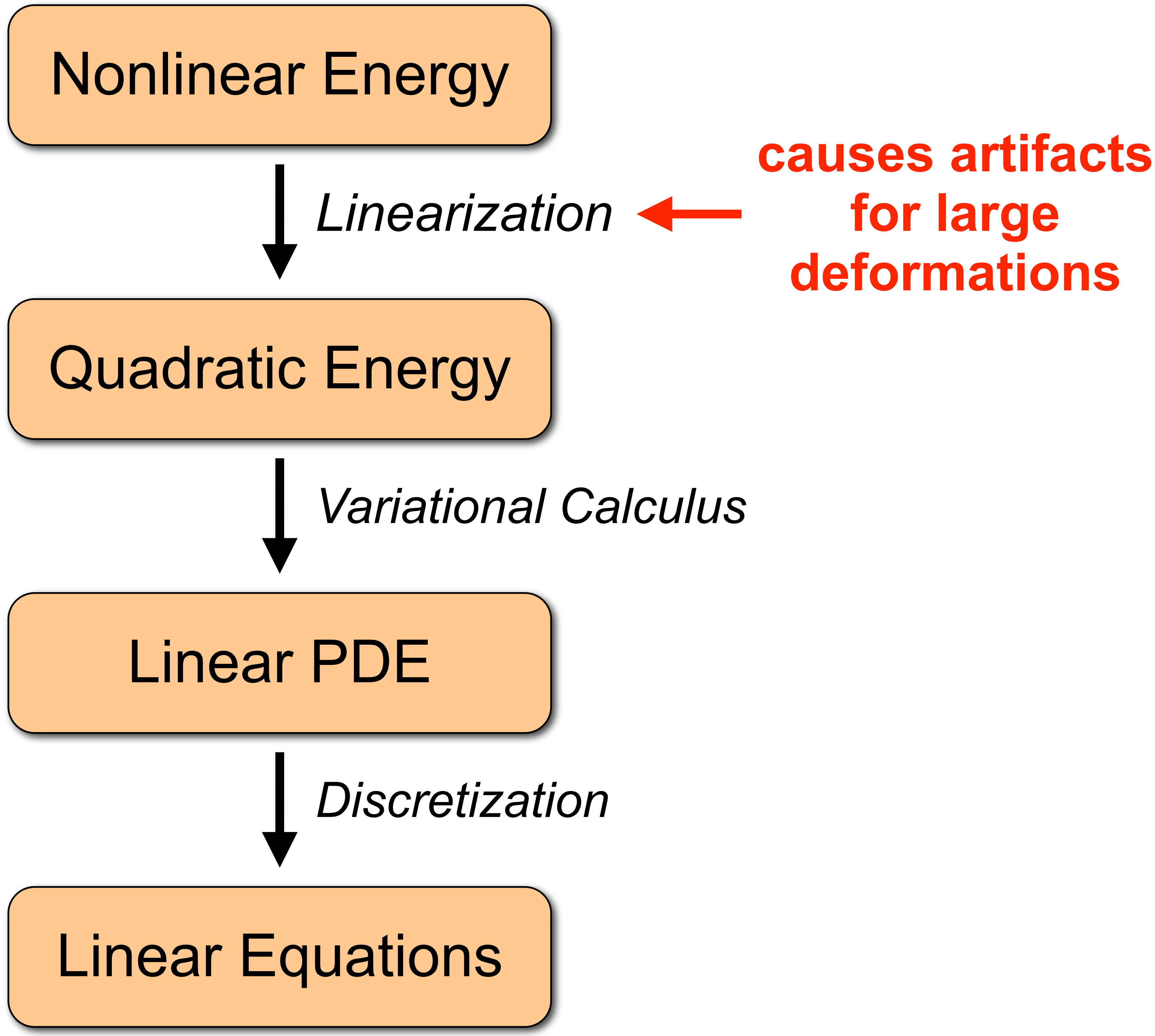
Linear vs. Nonlinear


Surface-Based


RBF


Nonlinear

Linear Approaches



Linearizations / Simplifications

- **Shell-based deformation**

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

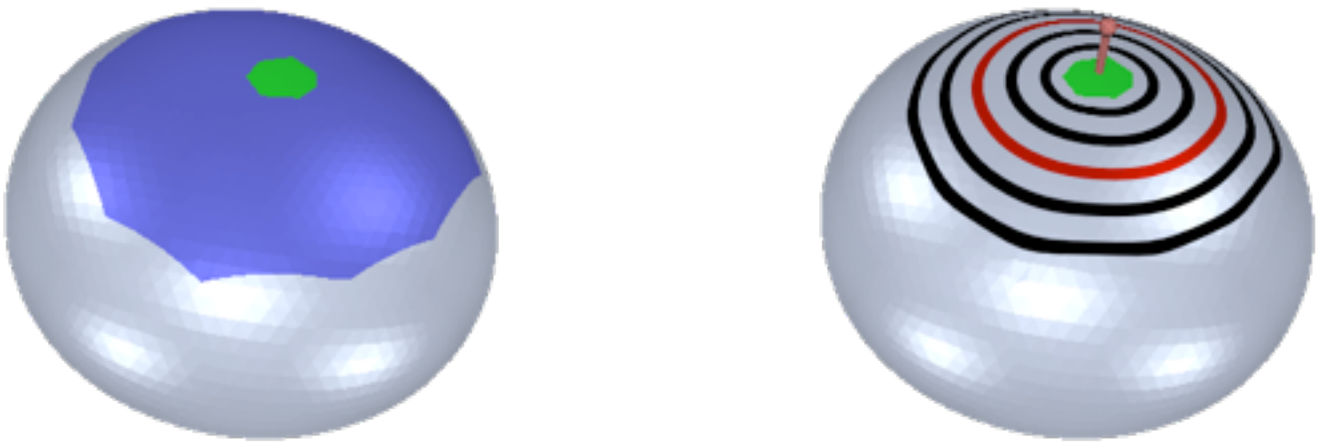
$$\Downarrow$$

$$\int_\Omega k_s \left(\left\| \frac{\partial \mathbf{d}}{\partial u} \right\|^2 + \left\| \frac{\partial \mathbf{d}}{\partial v} \right\|^2 \right) \; + \; k_b \left(\left\| \frac{\partial^2 \mathbf{d}}{\partial u^2} \right\|^2 + 2 \left\| \frac{\partial^2 \mathbf{d}}{\partial u \partial v} \right\|^2 + \left\| \frac{\partial^2 \mathbf{d}}{\partial v^2} \right\|^2 \right) \mathrm{d}u\mathrm{d}v$$

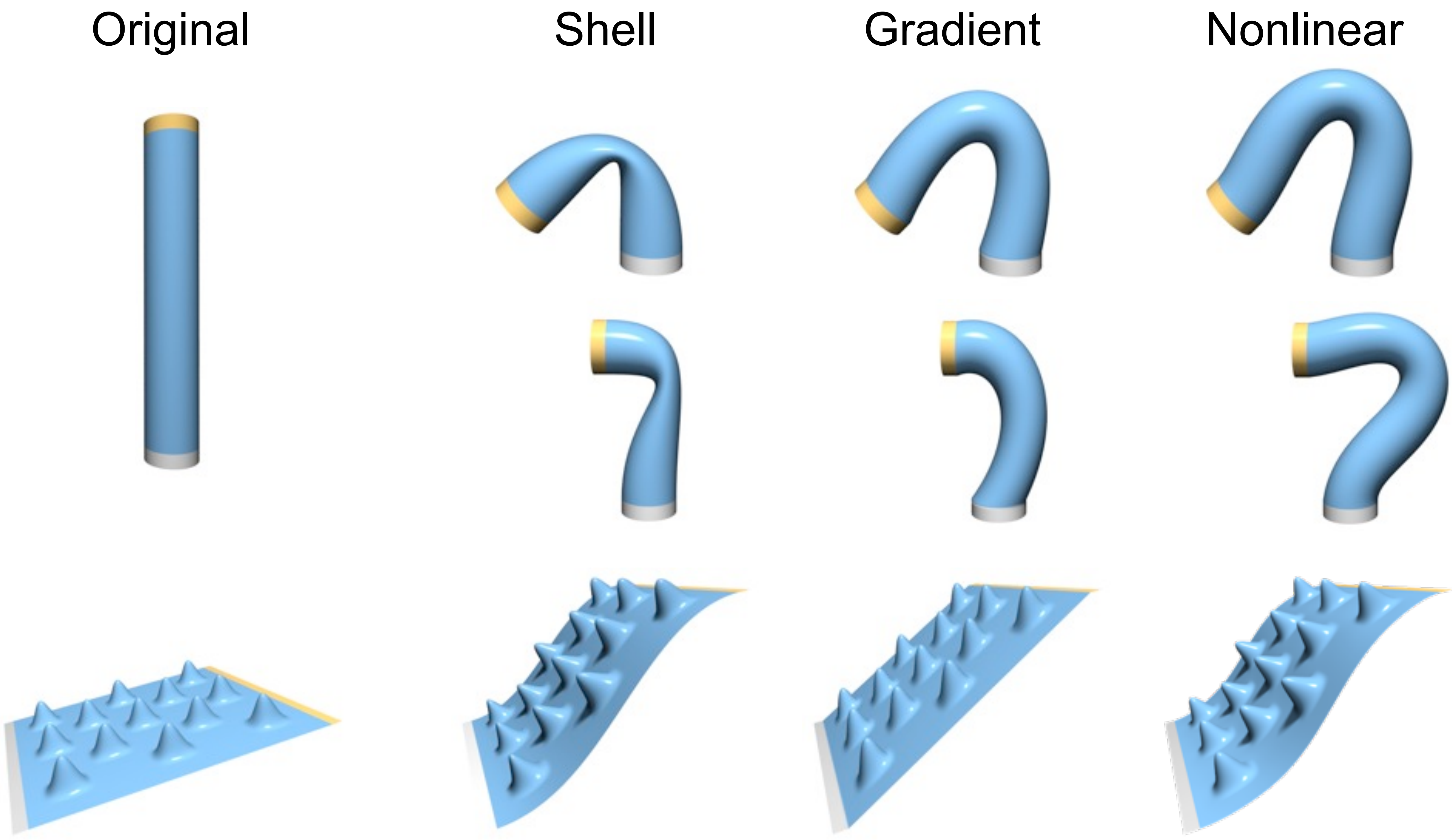
Linearizations / Simplifications

- **Gradient-based editing**

$$\nabla \mathbf{T}(\mathbf{x}) = \mathbf{A}$$

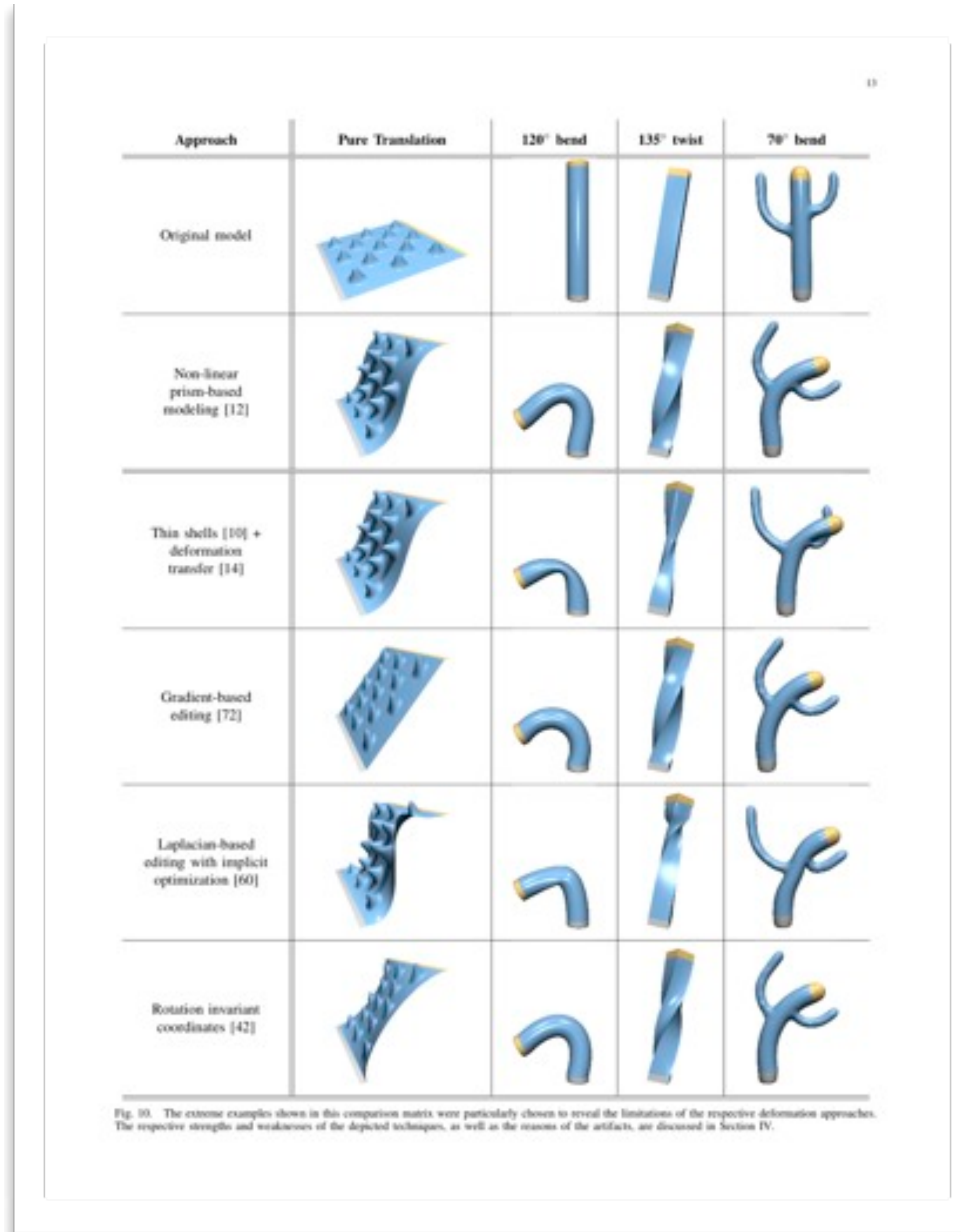


Linear vs. Nonlinear



Linear vs. Nonlinear

- Analyze existing methods
  - Some work for translations
  - Some work for rotations
  - No method works for both

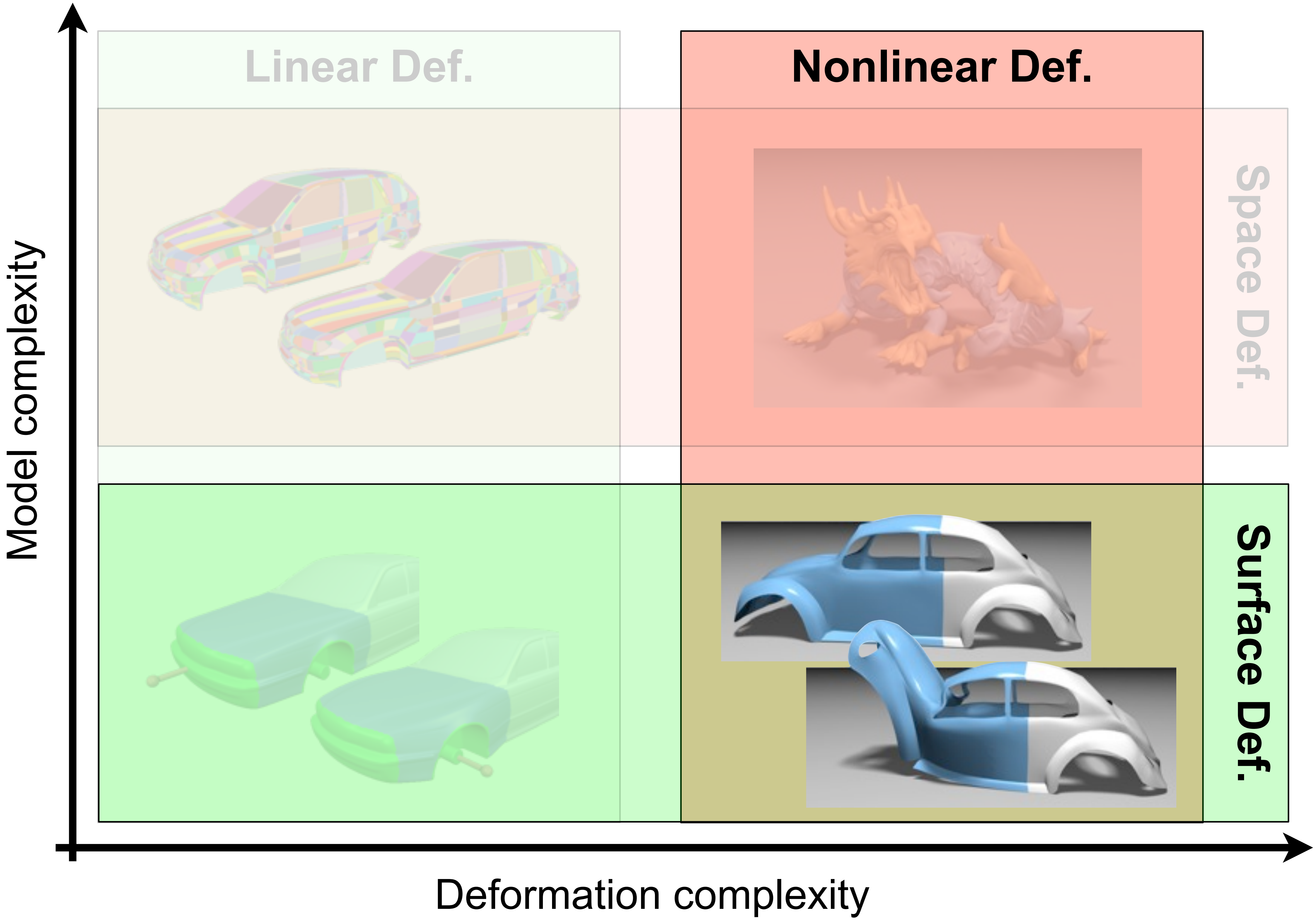


[Botsch & Sorkine, TVCG 08]

Literature

- Botsch et al, *PriMo: Coupled prisms for intuitive surface modeling*, SGP 2006
- Botsch & Sorkine, *On linear variational surface deformation methods*, TVCG 2008

Shape Deformation



Nonlinear Deformation?

- Sounds easy: “Just don’t linearize.”
- Not so easy though...
  - Solve nonlinear problems (Newton, Gauss-Newton)
  - No convergence guarantees
  - Robustness issues
  - Considerably slower

Nonlinear Surface Deformation

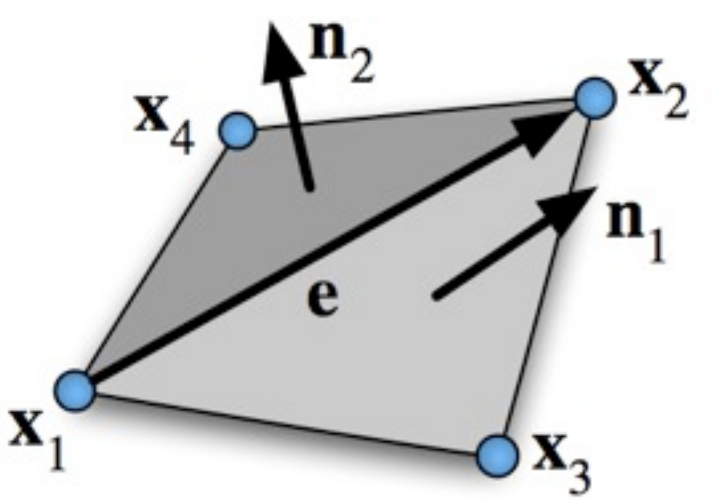
- **Shell-Based Deformation**
- Rigid Cells
- As-rigid-as-possible deformation

Nonlinear Discrete Shells

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m) \;=\; \lambda \sum_e w_{s,e} \left(l_e - L_e\right)^2 \;+\; \mu \sum_e w_{b,e} \left(\theta_e - \Theta_e\right)^2$$

Stretching:
change of
edge length

Bending:
change of
dihedral angle



[Grinspun et al, SCA 2003]

Gauss-Newton Minimization

- Residual function

$$\mathbf{f} : \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \end{bmatrix} \mapsto \begin{bmatrix} \sqrt{\lambda\, w_{s,1}}\ (l_1 - L_1) \\ \vdots \\ \sqrt{\lambda\, w_{s,m}}\ (l_m - L_m) \\ \sqrt{\mu\, w_{b,1}}\ (\theta_1 - \Theta_1) \\ \vdots \\ \sqrt{\mu\, w_{b,m}}\ (\theta_m - \Theta_m) \end{bmatrix}, \qquad E(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T\, \mathbf{f}(\mathbf{x}) \rightarrow \min$$

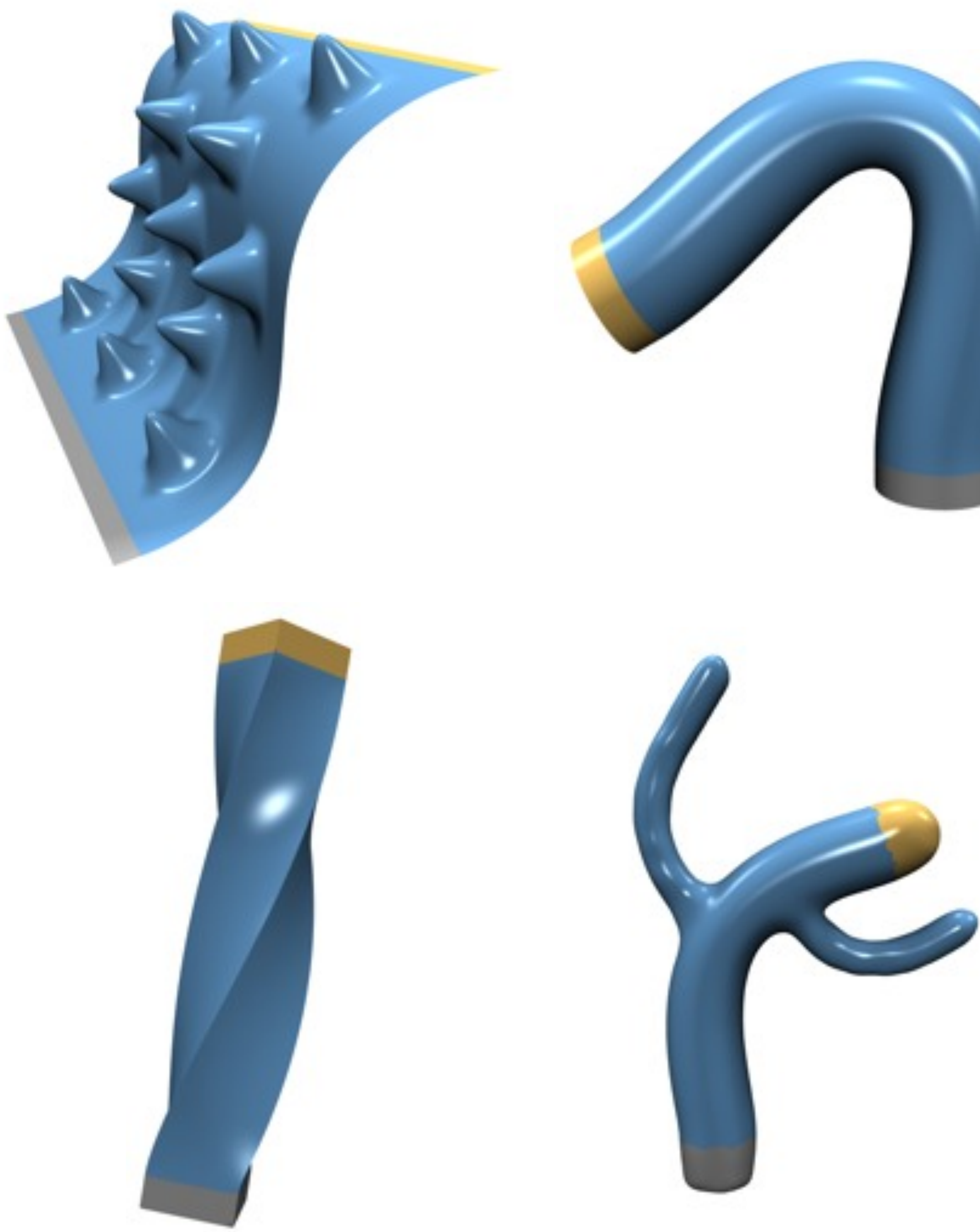
analytical derivatives

- Iterate until convergence

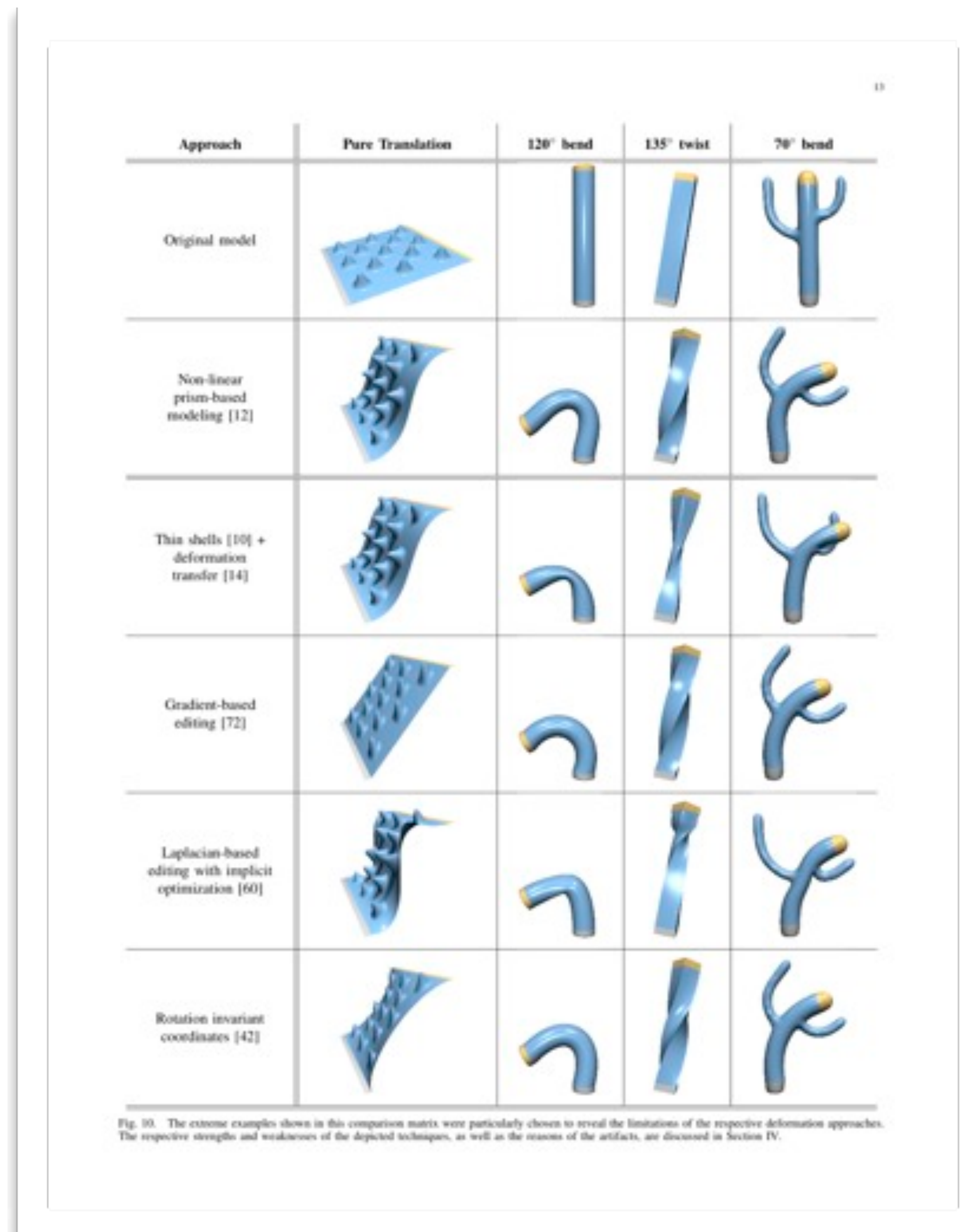
$$\mathbf{J}(\mathbf{x})^T\, \mathbf{J}(\mathbf{x})\ \boldsymbol{\delta}\ =\ -\mathbf{J}(\mathbf{x})^T\, \mathbf{f}(\mathbf{x})$$

$$\mathbf{x}\ \leftarrow\ \mathbf{x} + h\, \boldsymbol{\delta}$$

Deformation Results



[Fröhlich & Botsch, CGF 11]



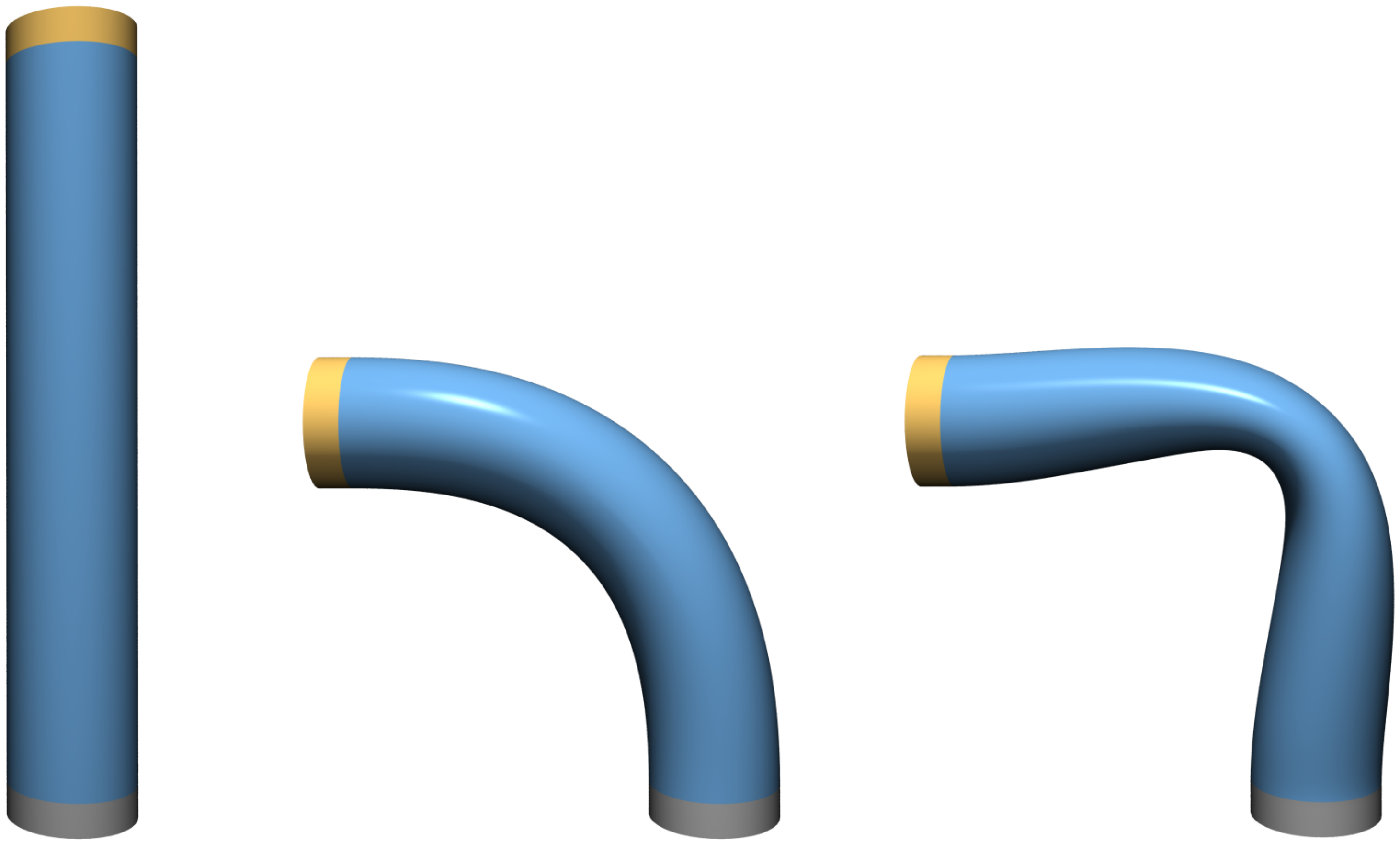
13

Approach	Pure Translation	120° bend	135° twist	70° bend
Original model				
Non-linear prism-based modeling [12]				
Thin shells [10] + deformation transfer [14]				
Gradient-based editing [72]				
Laplacian-based editing with implicit optimization [60]				
Rotation invariant coordinates [42]				

Fig. 10. The extreme examples shown in this comparison matrix were particularly chosen to reveal the limitations of the respective deformation approaches. The respective strengths and weaknesses of the depicted techniques, as well as the reasons of the artifacts, are discussed in Section IV.

[Botsch & Sorkine, TVCG 08]

Deformation Results



Global stiffness control

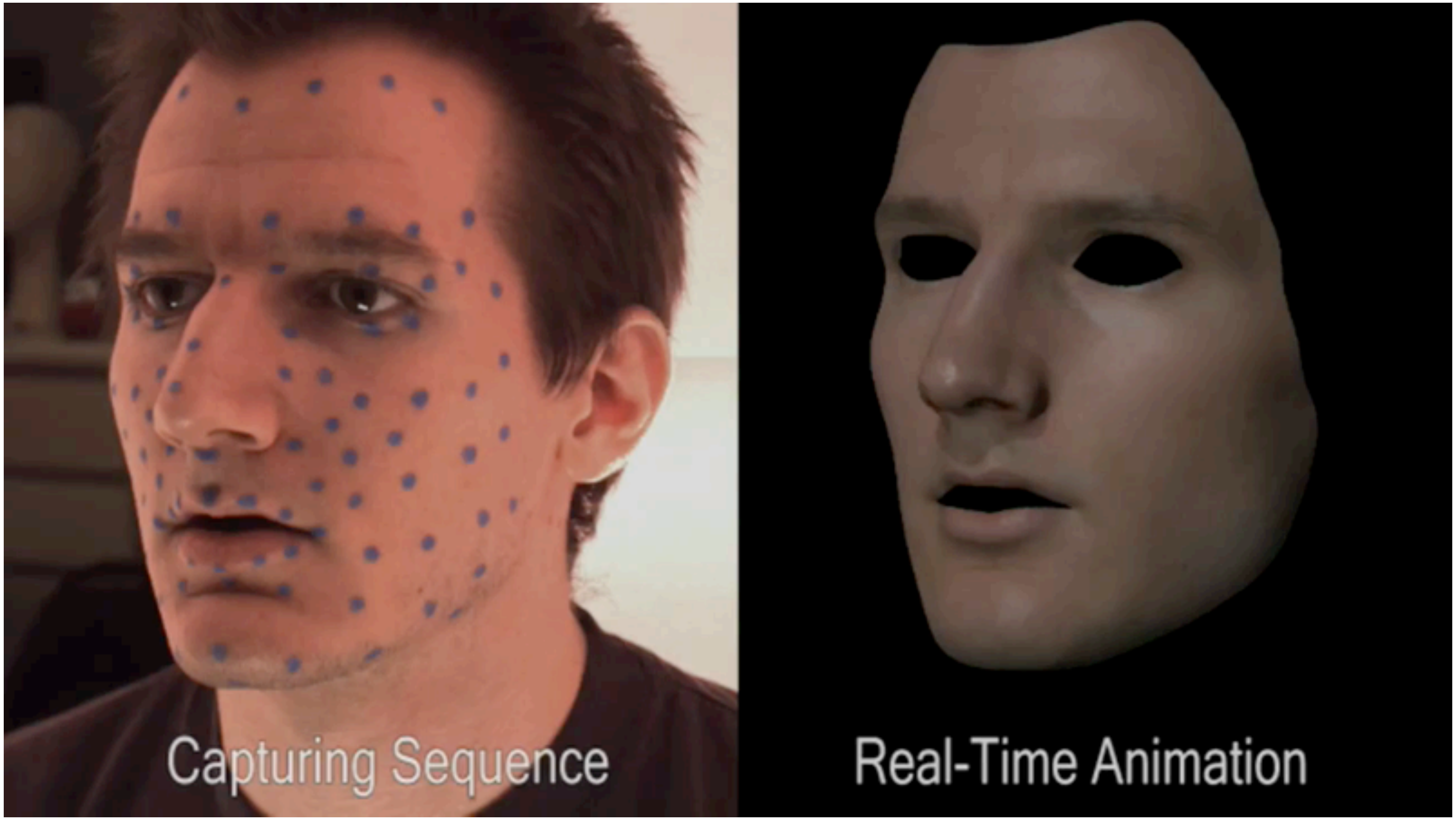
[Fröhlich & Botsch, CGF 11]

Deformation Results

Local stiffness control

[Fröhlich & Botsch, CGF 11]

Nonlinear Face Animation



Add nonlinear wrinkle effects & realistic rendering

[Bickel et al, SCA 2008]

Nonlinear Surface Deformation

- Shell-Based Deformation
- **Rigid Cells**
- As-rigid-as-possible deformation

Cells

- Qualitatively emulate thin-shell behavior
- Thin volumetric layer around center surface
- Extrude polygonal cell C_i per mesh face

[Botsch et al, SGP 06]

Rigid Cells

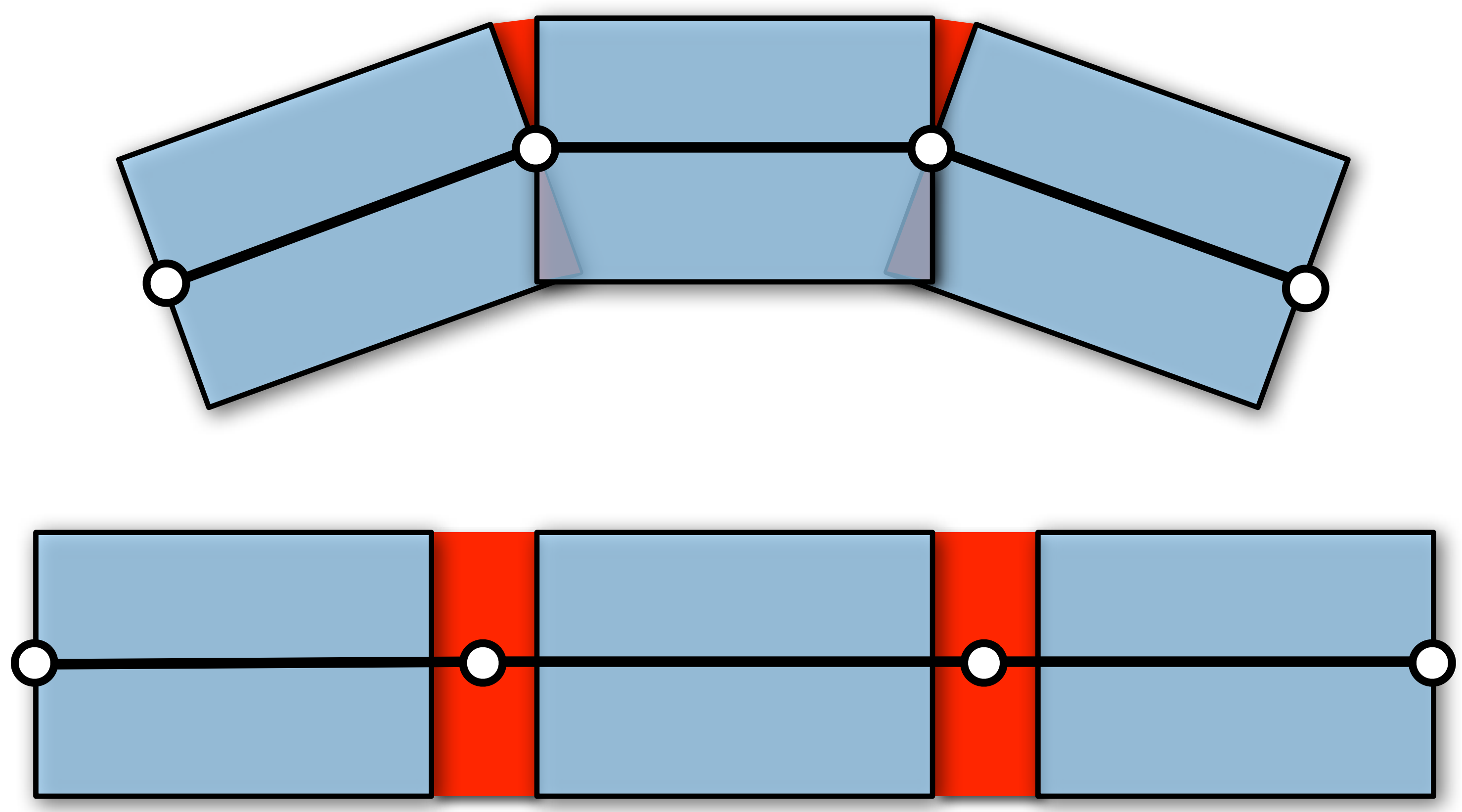
- Aim for robustness
  - Prevent cells from degenerating

  ➡ Keep cells *rigid*

[Botsch et al, SGP 06]

Elastically Connected Rigid Cells

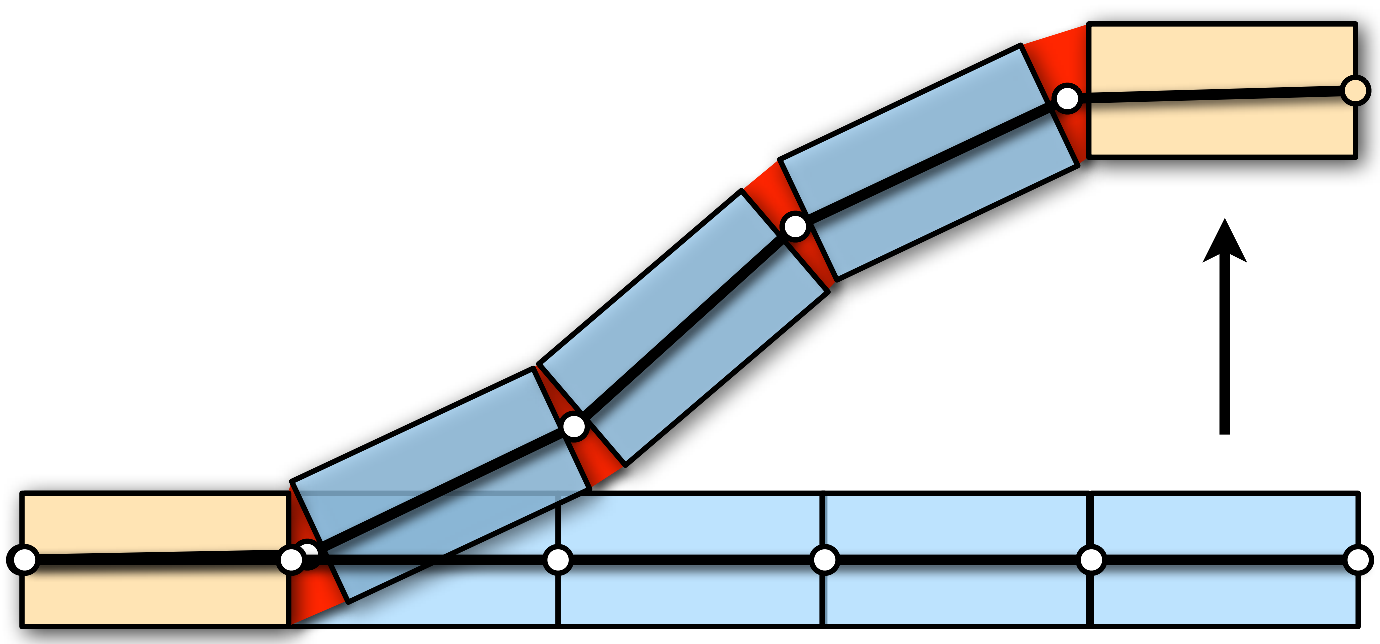
- Connect cells along their faces
  - Nonlinear elastic energy
  - Measures bending, stretching, twisting, ...



[Botsch et al, SGP 06]

Cell-Based Surface Deformation

1. Prescribes position/orientation for cells
2. Find optimal rigid motions per cell
3. Update vertices by average cell transformations

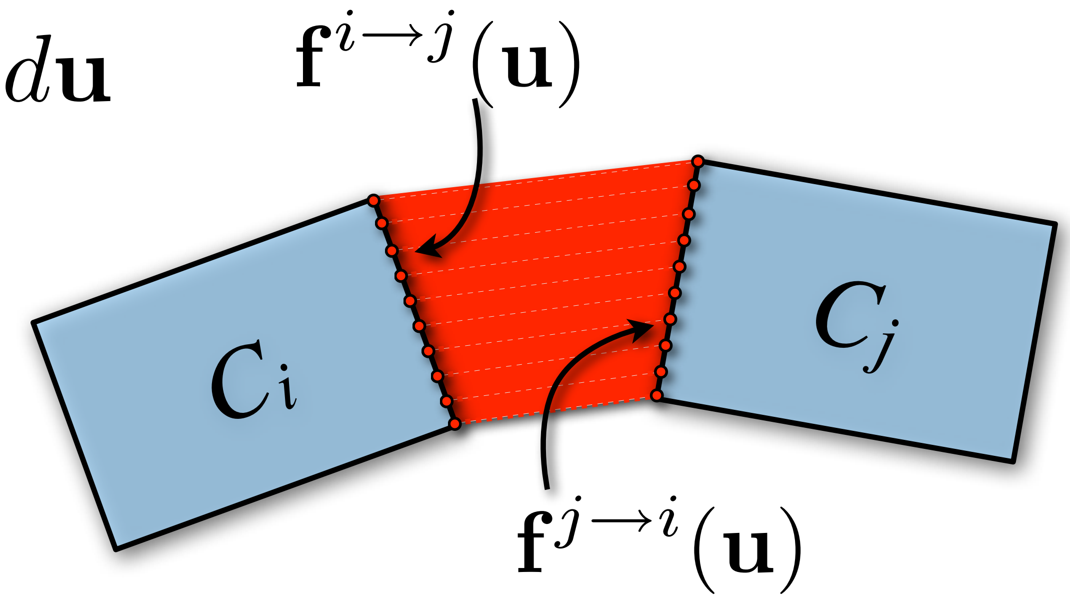


[Botsch et al, SGP 06]

Elastically Connected Rigid Cells

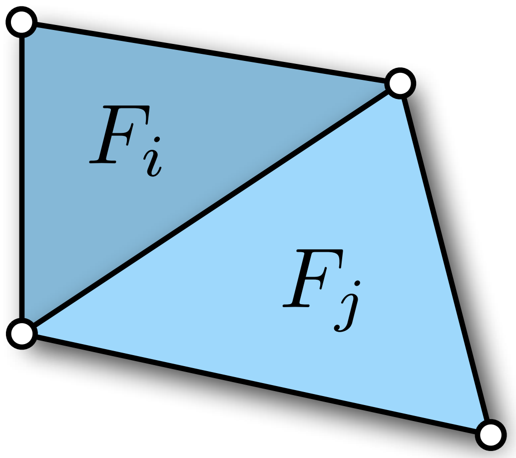
- Pairwise energy

$$E_{ij} = \int_{[0,1]^2} \left\| \mathbf{f}^{\,i \to j}(\mathbf{u}) - \mathbf{f}^{\,j \to i}(\mathbf{u}) \right\|^2 d\mathbf{u}$$



- Global energy

$$E = \sum_{\{i,j\}} w_{ij} \cdot E_{ij} \quad , \qquad w_{ij} = \frac{\|\mathbf{e}_{ij}\|^2}{|F_i| + |F_j|}$$



[Botsch et al, SGP 06]

Nonlinear Minimization

- Find *rigid* motion $\mathbf{T}_i$ per cell C_i

$$\min_{\{\mathbf{T}_i\}} \sum_{\{i,j\}} w_{ij} \int_{[0,1]^2} \left\| \mathbf{T}_i\left(\mathbf{f}^{\,i \to j}(\mathbf{u})\right) - \mathbf{T}_j\left(\mathbf{f}^{\,j \to i}(\mathbf{u})\right) \right\|^2 \mathrm{d}\mathbf{u}$$

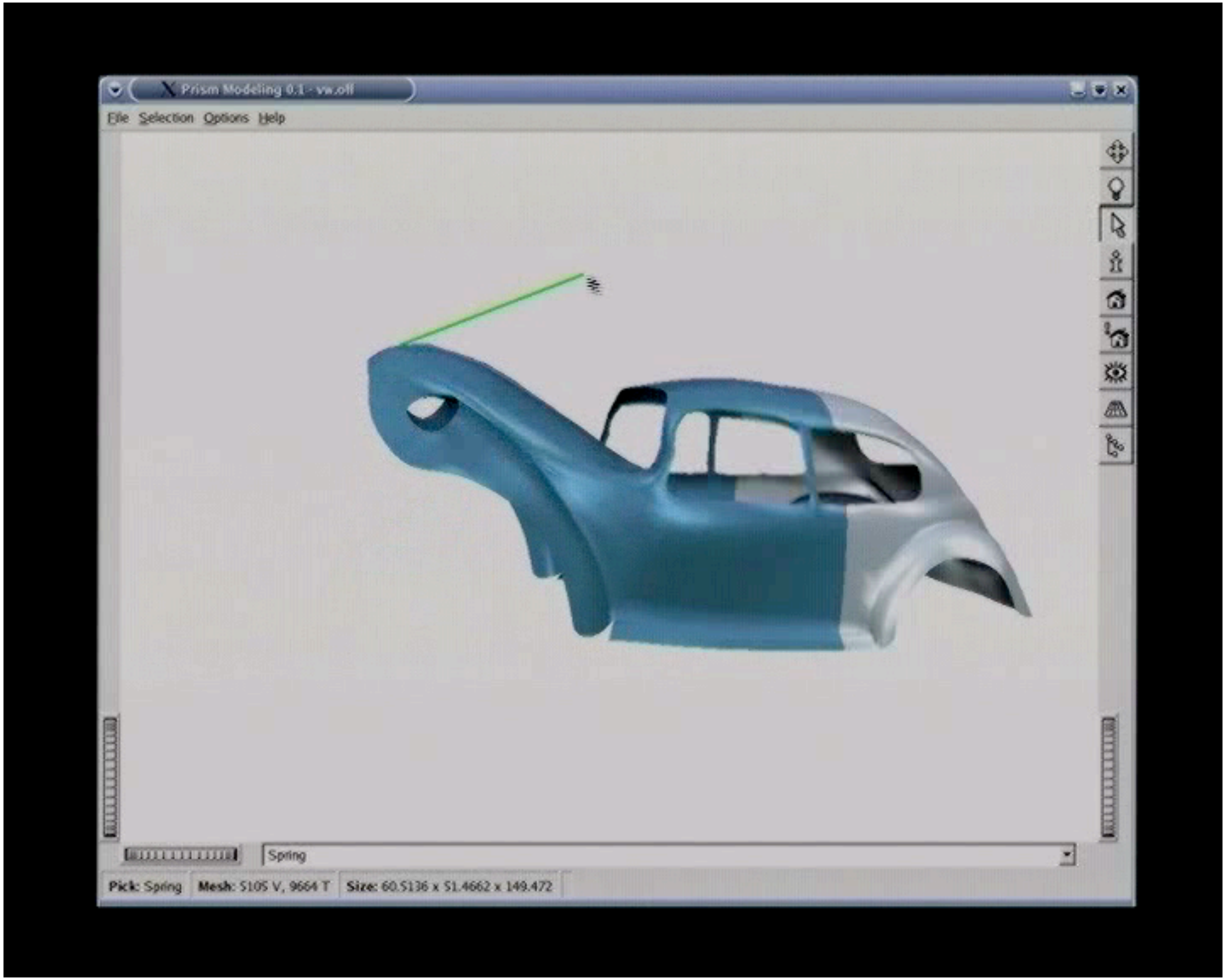
- Generalized global *shape matching* problem
  - Robust geometric optimization
  - Nonlinear Newton-type minimization
  - Hierarchical multi-grid solver

[Botsch et al, SGP 06]

Robustness

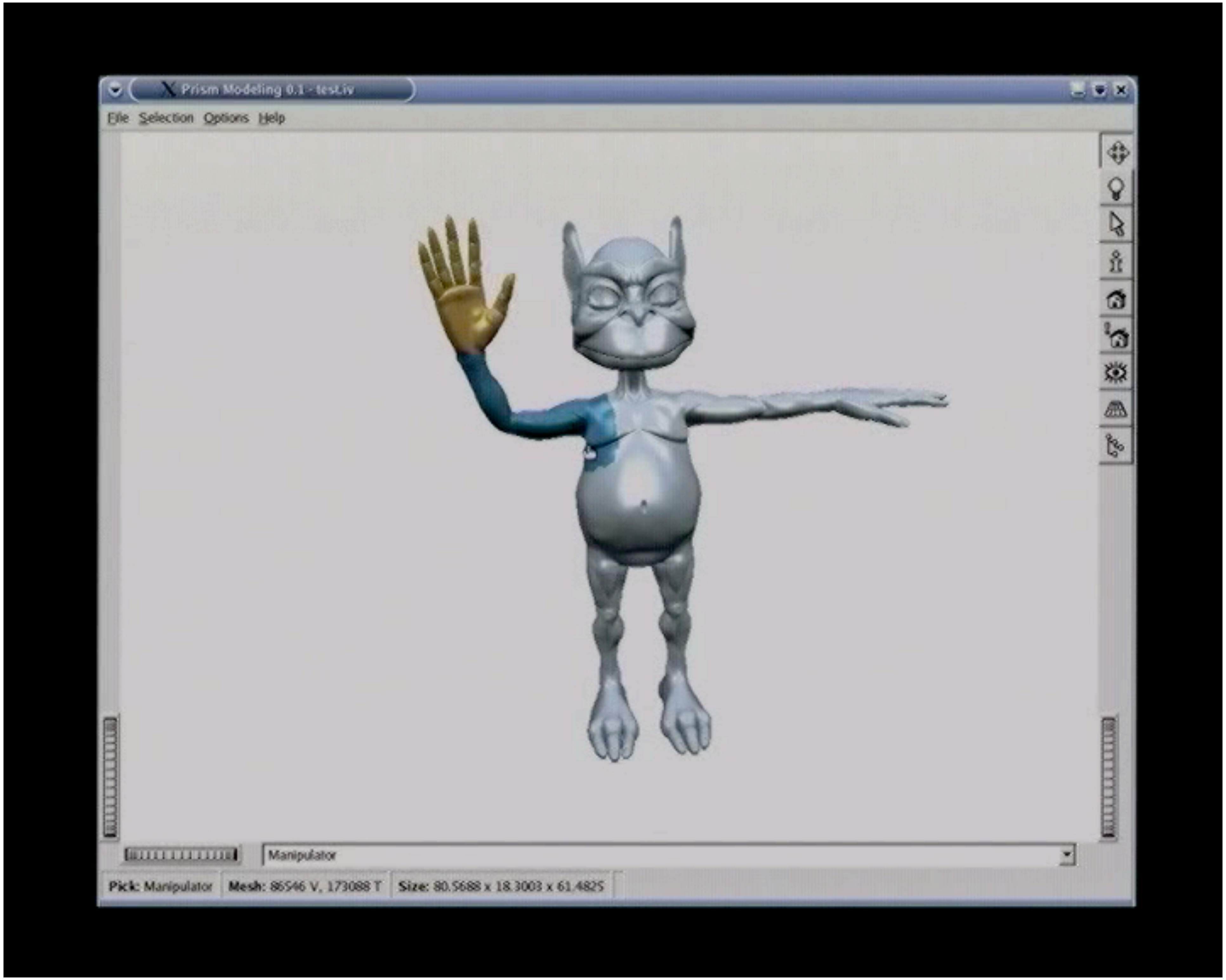
[Botsch et al, SGP 06]

PriMo



[Botsch et al, SGP 06]

Character Posing



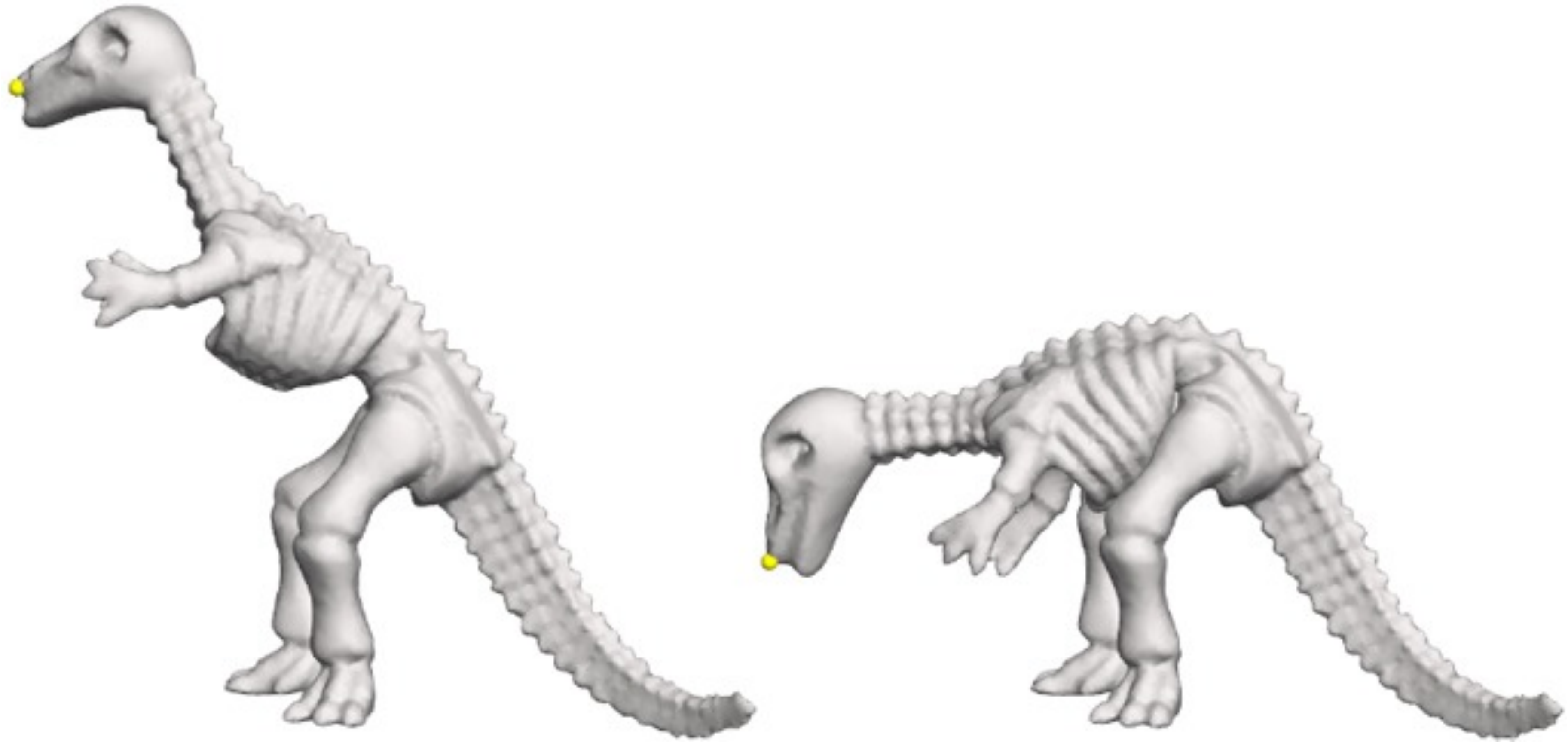
[Botsch et al, SGP 06]

Nonlinear Surface Deformation

- Shell-Based Deformation
- Rigid Cells
- **As-rigid-as-possible deformation**

Surface Deformation

- Smooth large scale deformation
- Local as-rigid-as-possible behavior
  - Preserves small-scale details

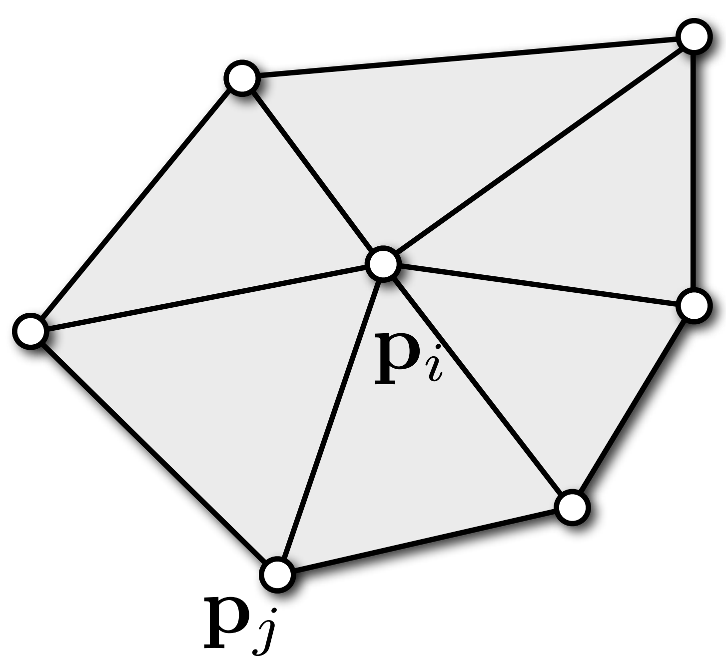


[Sorkine & Alexa, SGP 07]

Deformation Energy

- Vertex neighborhoods should deform rigidly

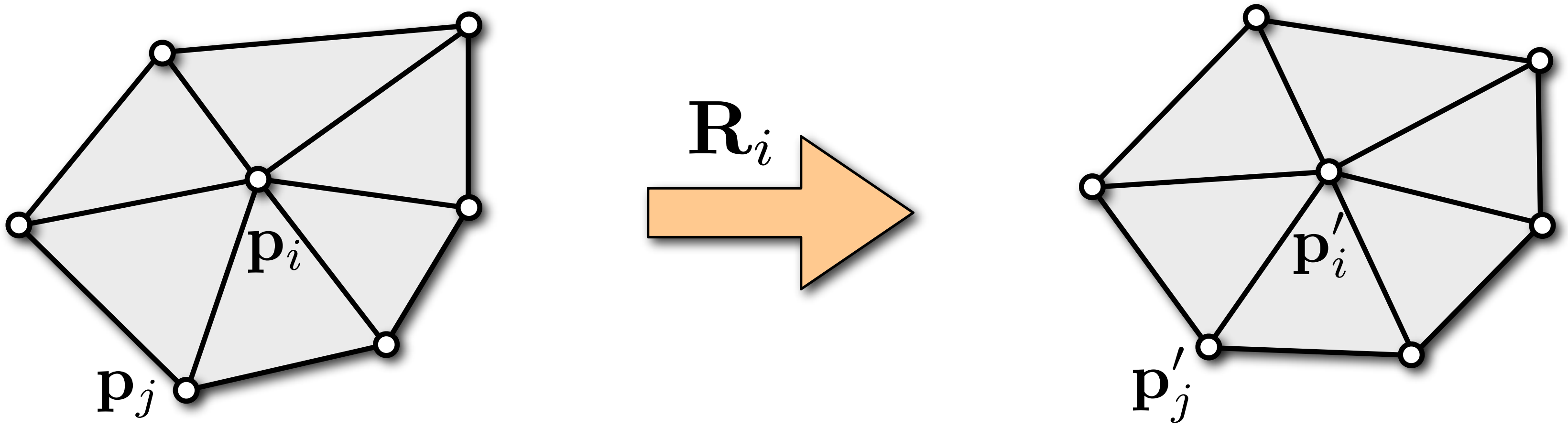
$$\sum_{j \in N(i)} \left\| \left(\mathbf{p}'_j - \mathbf{p}'_i\right) - \mathbf{R}_i \left(\mathbf{p}_j - \mathbf{p}_i\right) \right\|^2 \rightarrow \min$$



[Sorkine & Alexa, SGP 07]

Cell Deformation Energy

- If $\mathbf{p}, \mathbf{p}'$ are known then $\mathbf{R}_i$ is uniquely defined



➡ *Shape matching* problem

[Sorkine & Alexa, SGP 07]

Total Deformation Energy

- Sum over all vertex

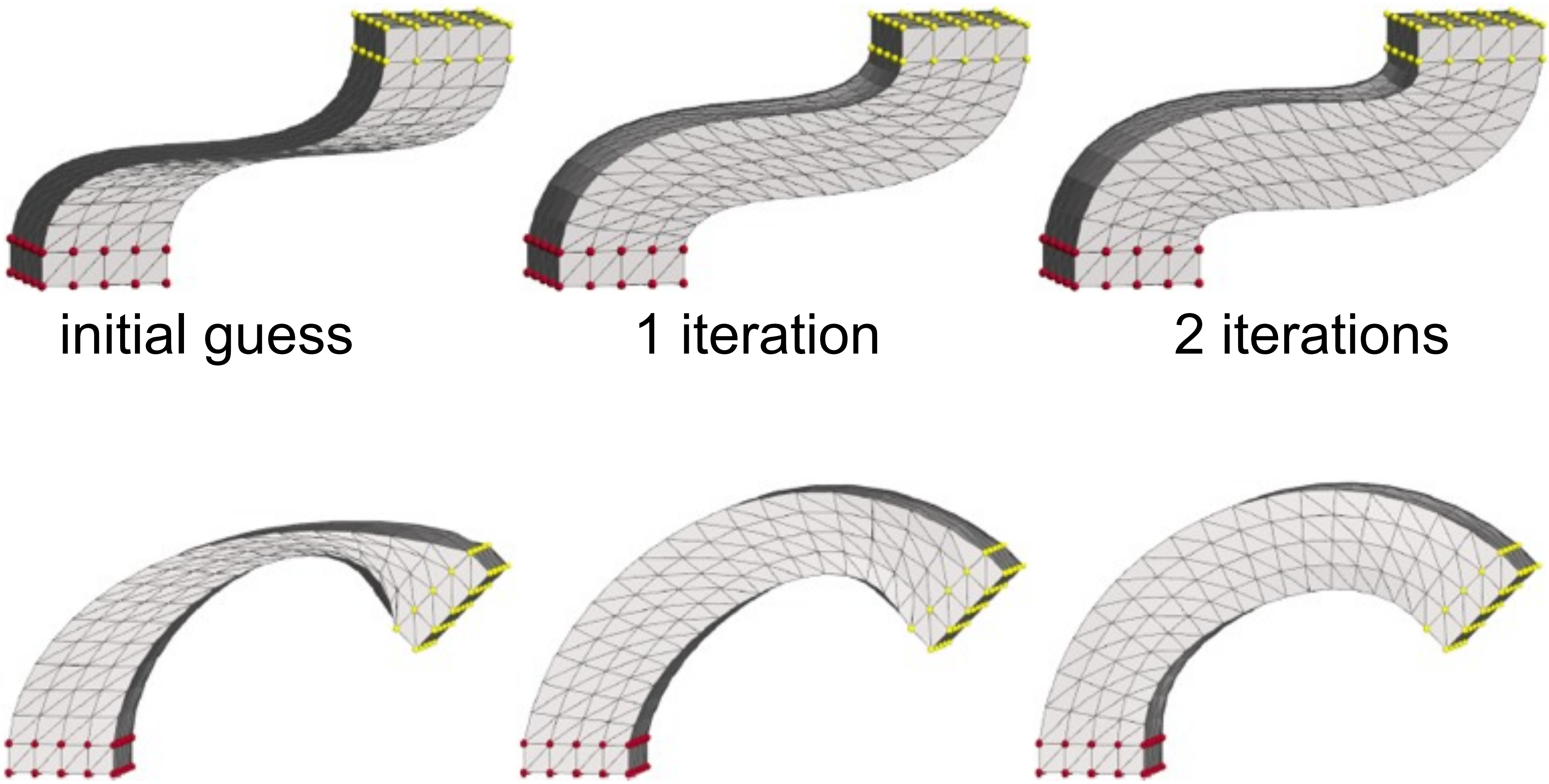
$$\min_{\mathbf{p}'} \sum_{i=1}^{n} \sum_{j \in N(i)} \left\| \left(\mathbf{p}'_j - \mathbf{p}'_i\right) - \mathbf{R}_i \left(\mathbf{p}_j - \mathbf{p}_i\right) \right\|^2$$

- Treat $\mathbf{p}'$ and $\mathbf{R}_i$ as separate variables
- Allows for alternating optimization
  - Fix $\mathbf{p}'$, find $\mathbf{R}_i$: Local shape matching per one-ring
  - Fix $\mathbf{R}_i$, find $\mathbf{p}'$: Solve Laplacian system

[Sorkine & Alexa, SGP 07]

As-Rigid-As-Possible Modeling

- Start from naïve Laplacian editing as initial guess

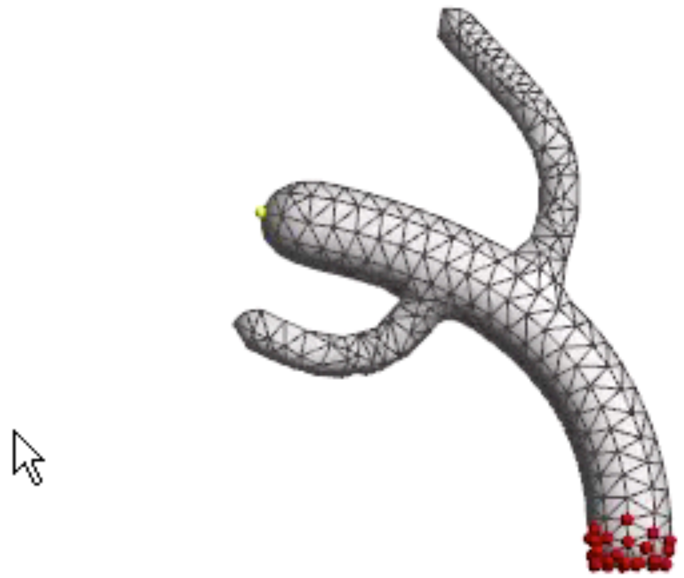


initial guess | 1 iteration | 2 iterations

initial guess | 1 iterations | 4 iterations

[Sorkine & Alexa, SGP 07]

As-Rigid-As-Possible Modeling

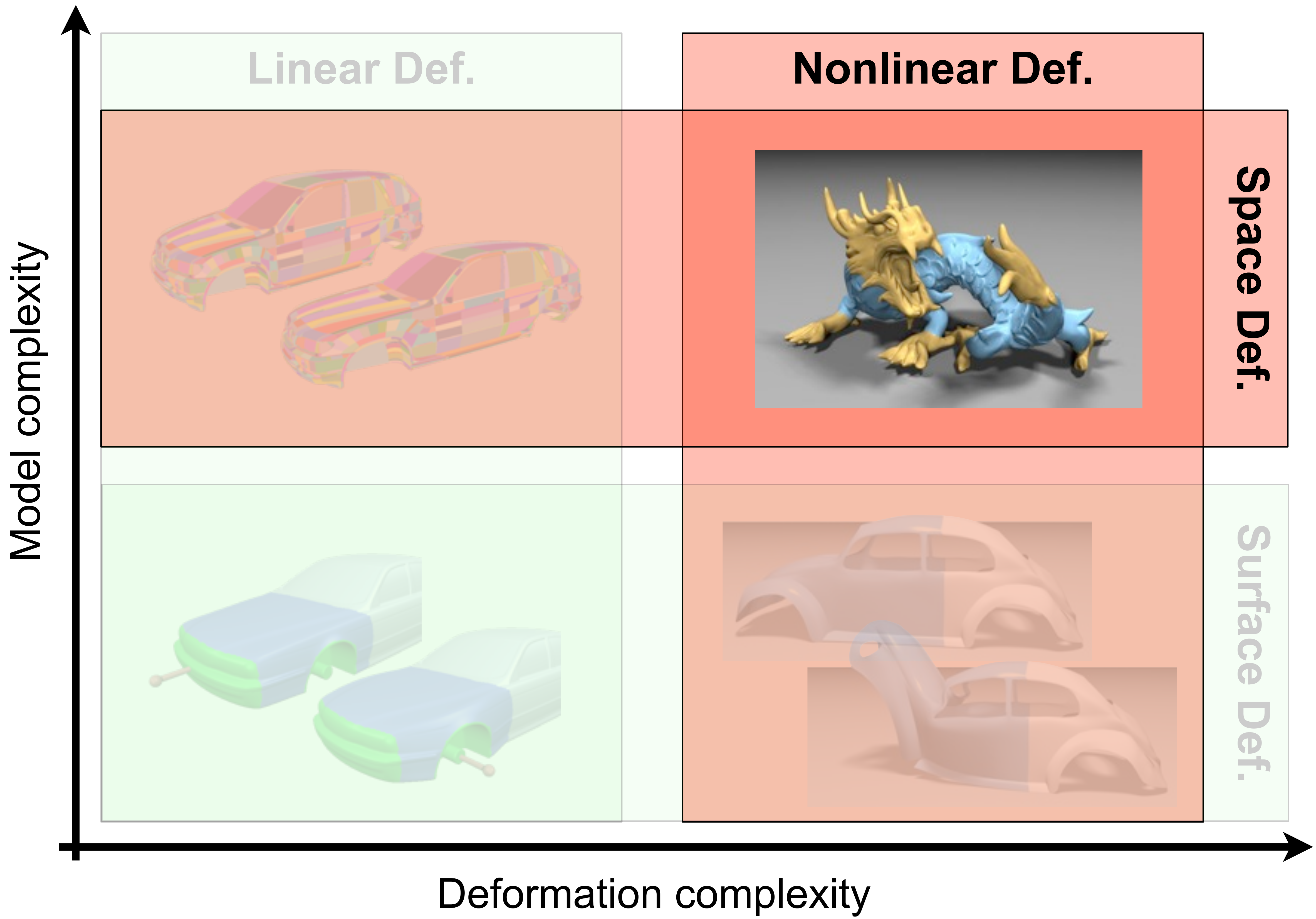


[Sorkine & Alexa, SGP 07]

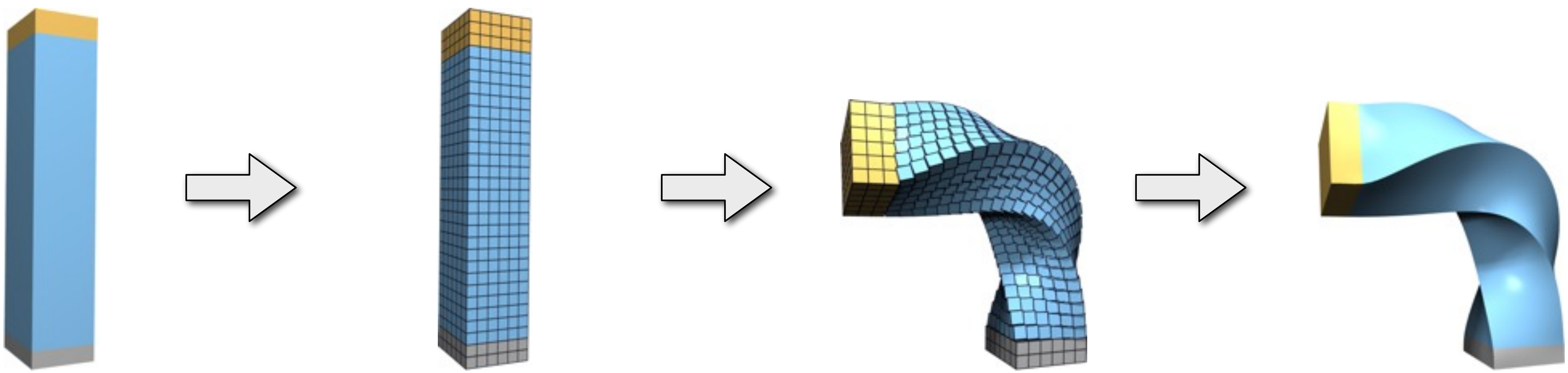
Literature

- Botsch et al, *PriMo: Coupled prisms for intuitive surface modeling*, SGP 2006
- Sorkine & Alexa, *As-rigid-as-possible surface editing*, SGP 2007
- Grinspun et al, *Discrete shells*, SCA 2003
- Fröhlich & Botsch, *Example-driven deformations based on discrete shells*, CGF 2011

Shape Deformation



Space PriMo



Volumetric Discretization

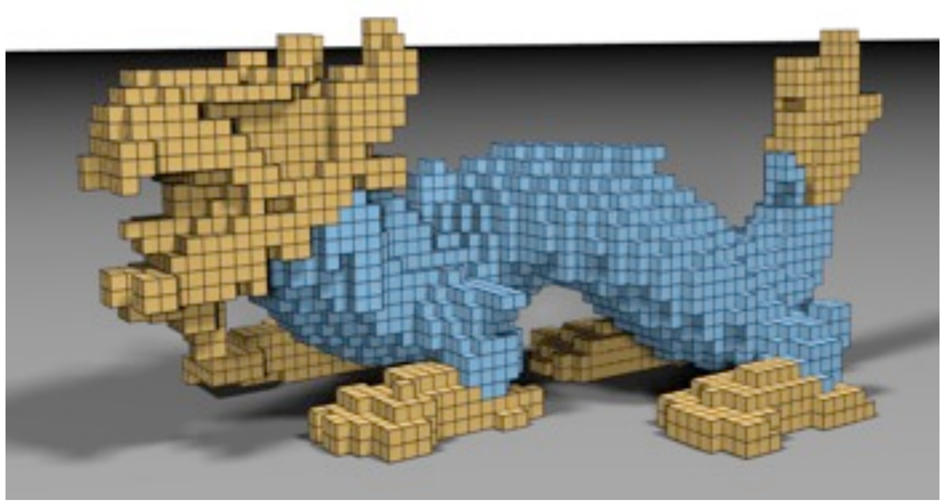
Cell-Based Deformation

Space Deformation

[Botsch et al, EG 07]

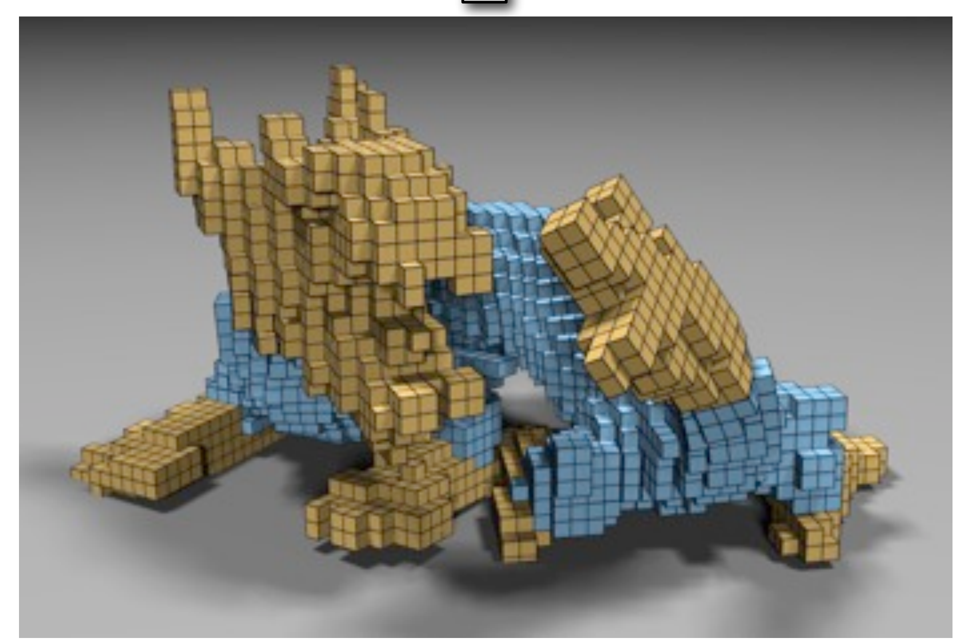
Space PriMo

100k



5.5k

100k



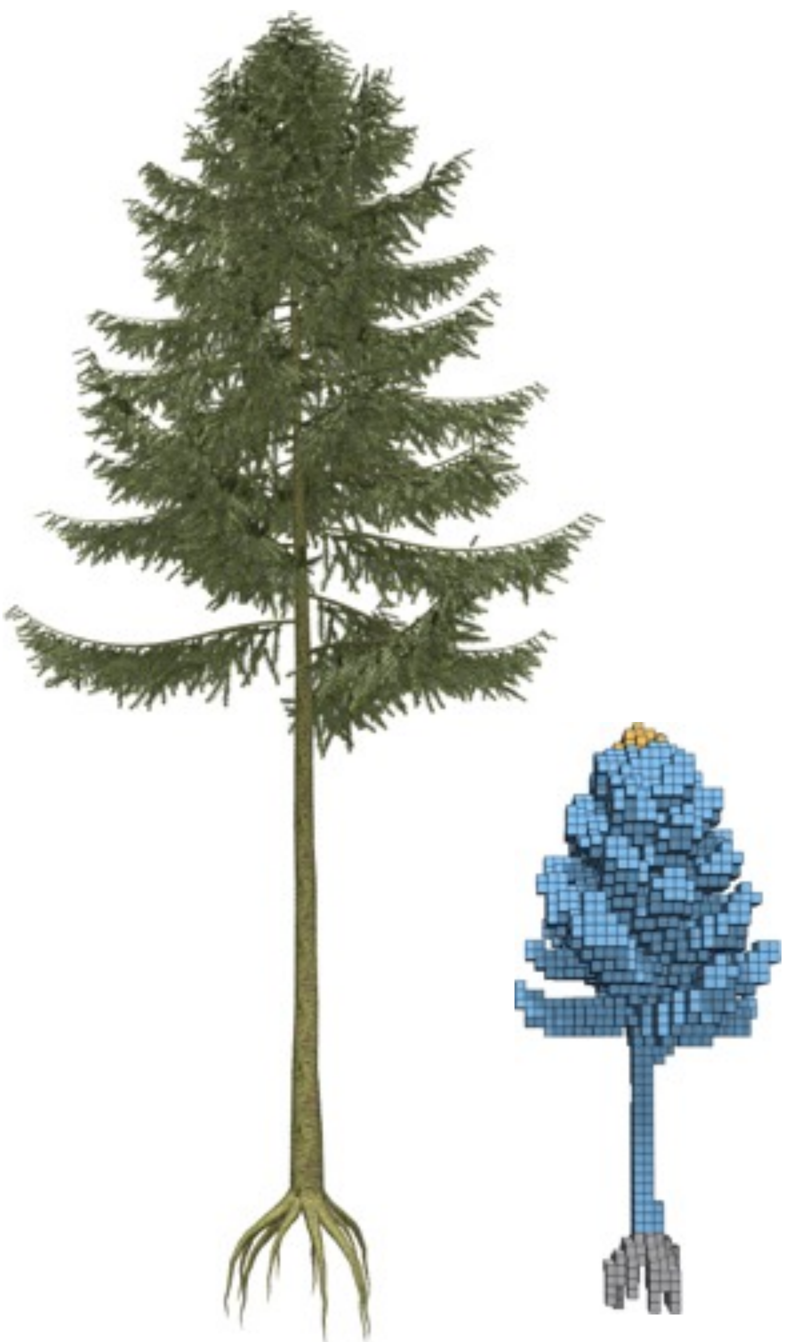
5.5k

[Botsch et al, EG 07]

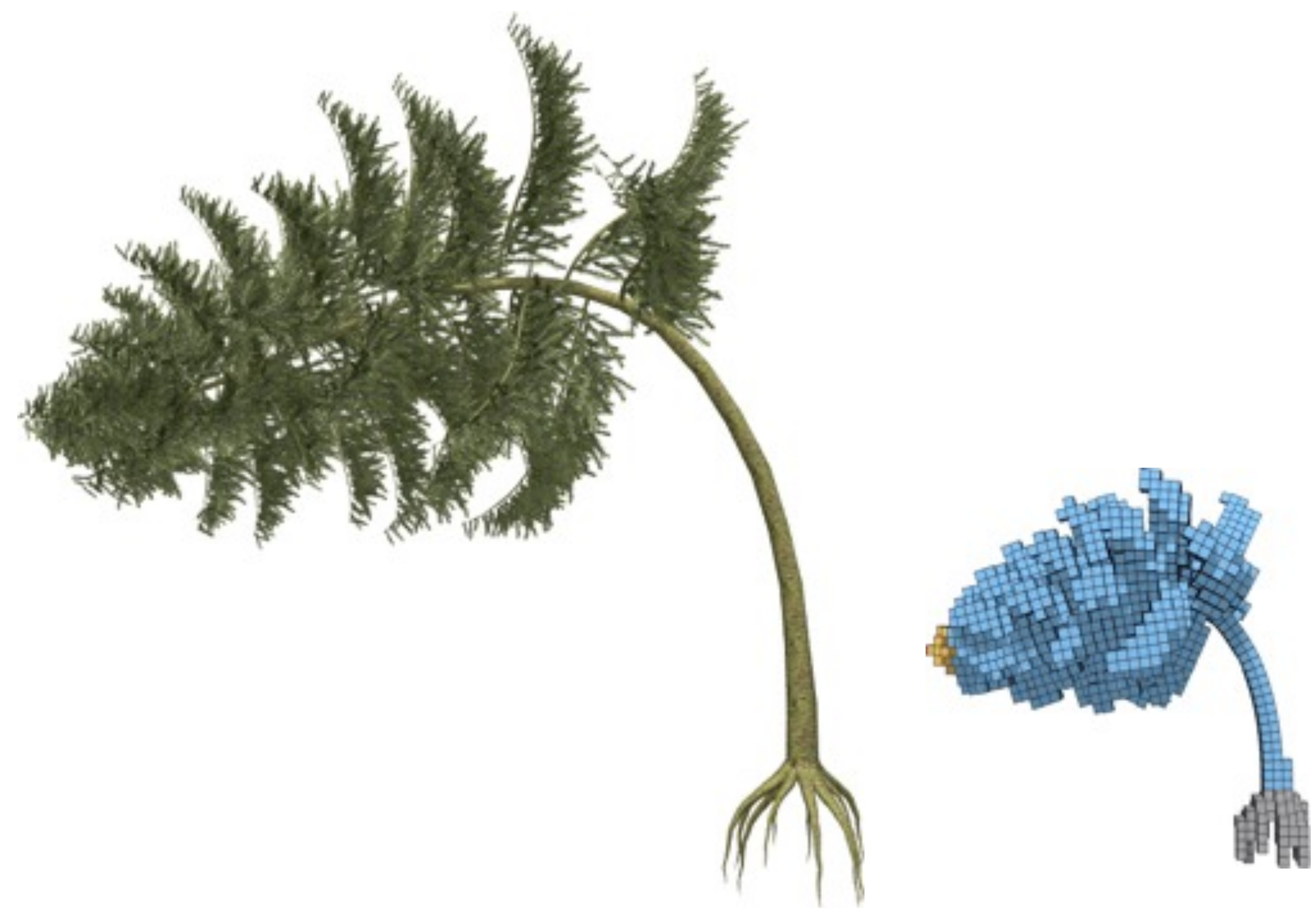
Space PriMo

[Botsch et al, EG 07]

Space PriMo



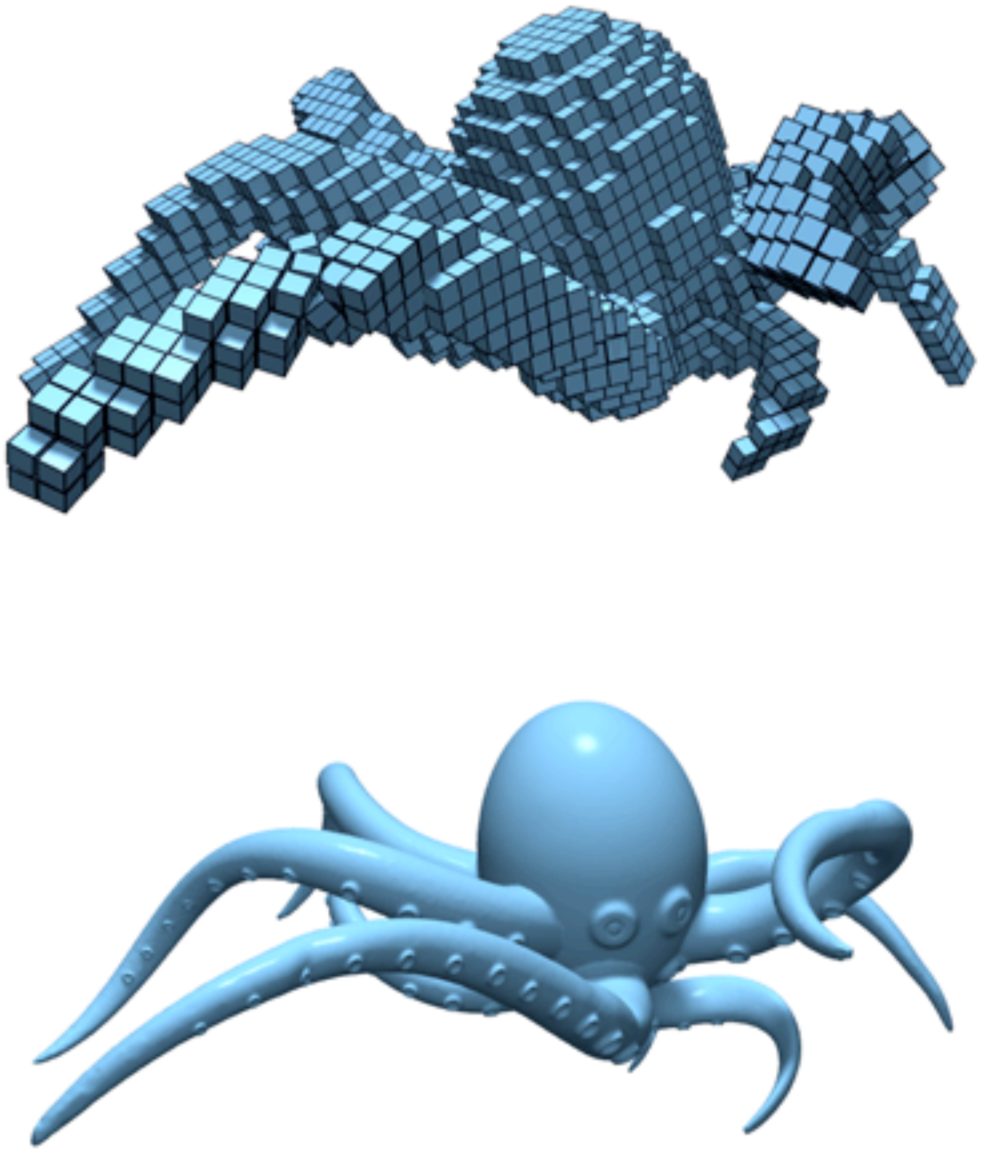
14k components



Deformed triangle soup

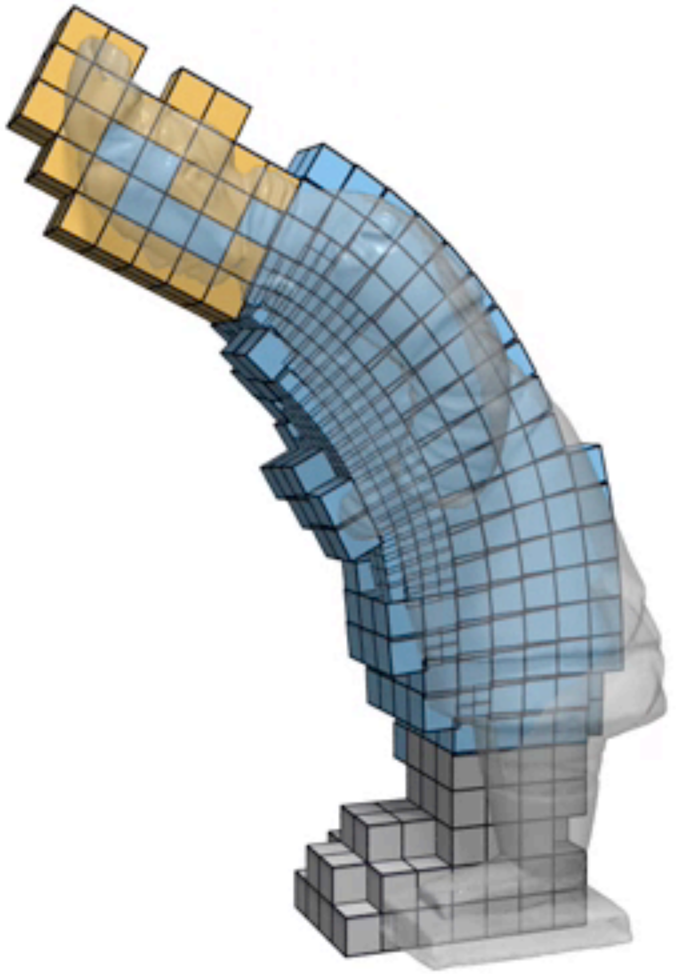
[Botsch et al, EG 07]

Space PriMo



[Botsch et al, EG 07]

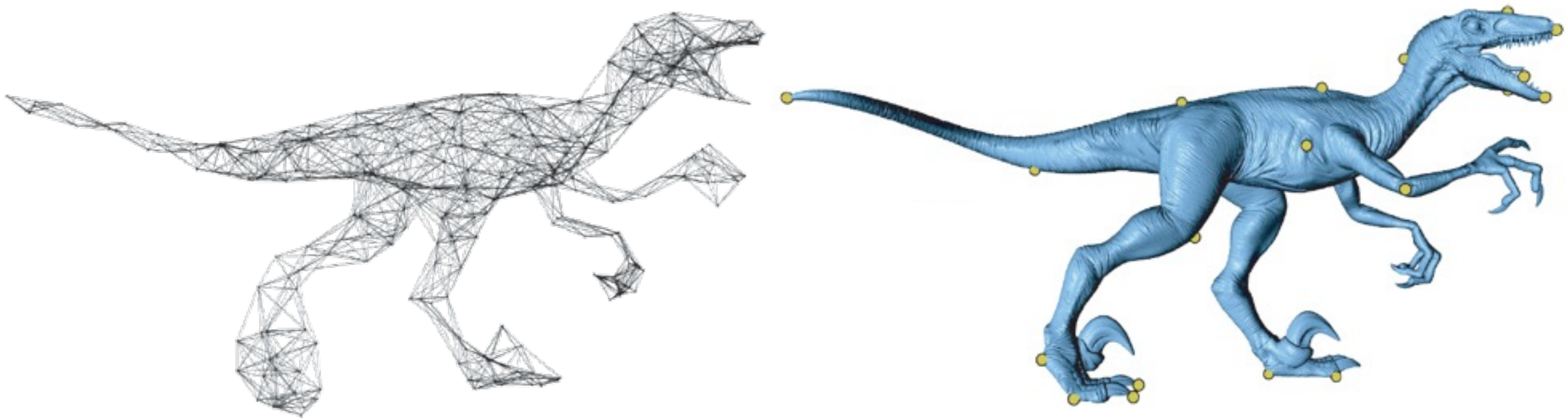
Space PriMo



[Botsch et al, EG 07]

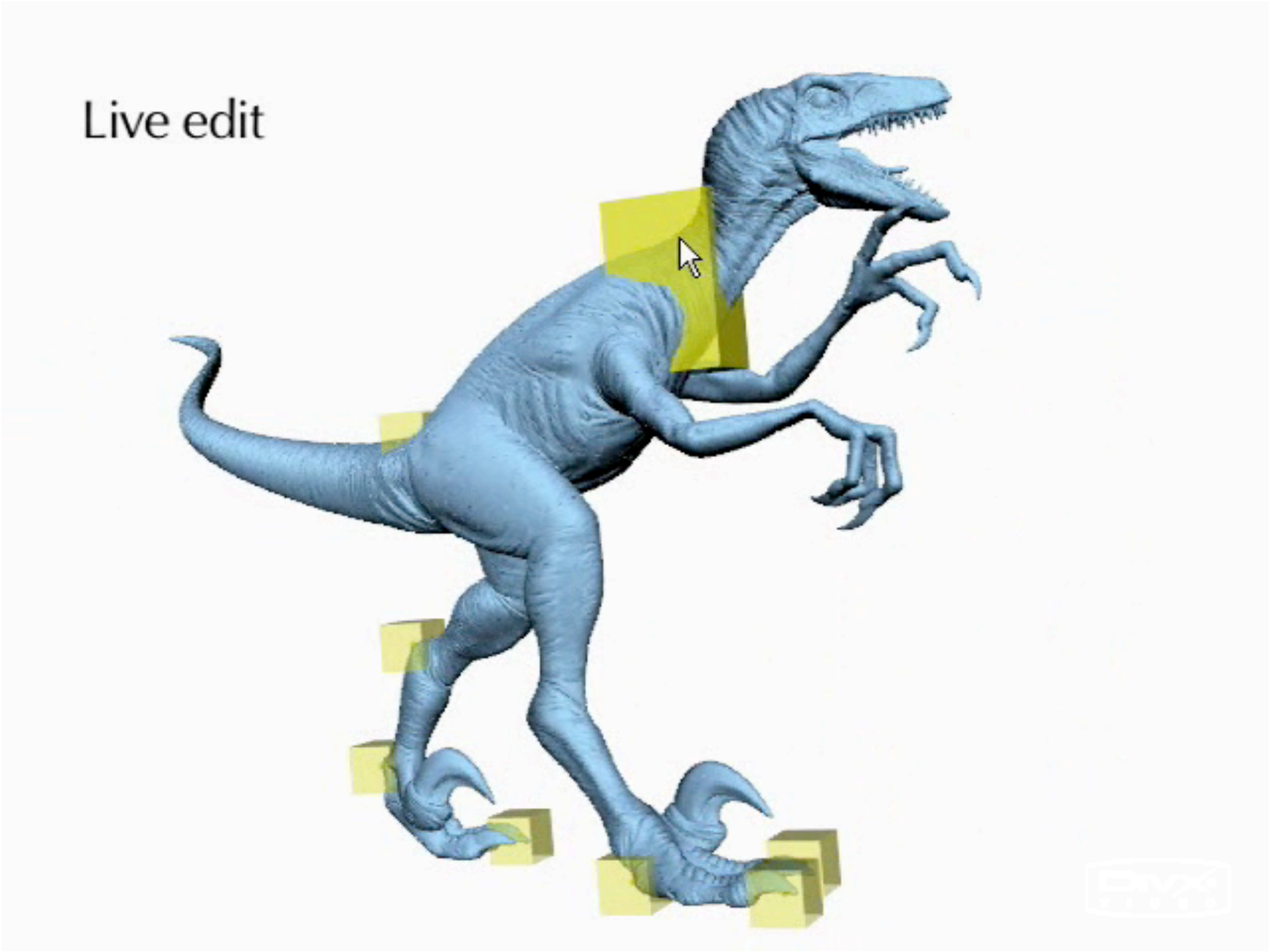
Embedded Deformation

- Parameterize model with *deformation graph*
- Find optimal transformation for each node
  - Affine transformation per node
  - Weakly enforce rigidity on matrices



[Sumner et al, SIGGRAPH 07]

Embedded Deformation

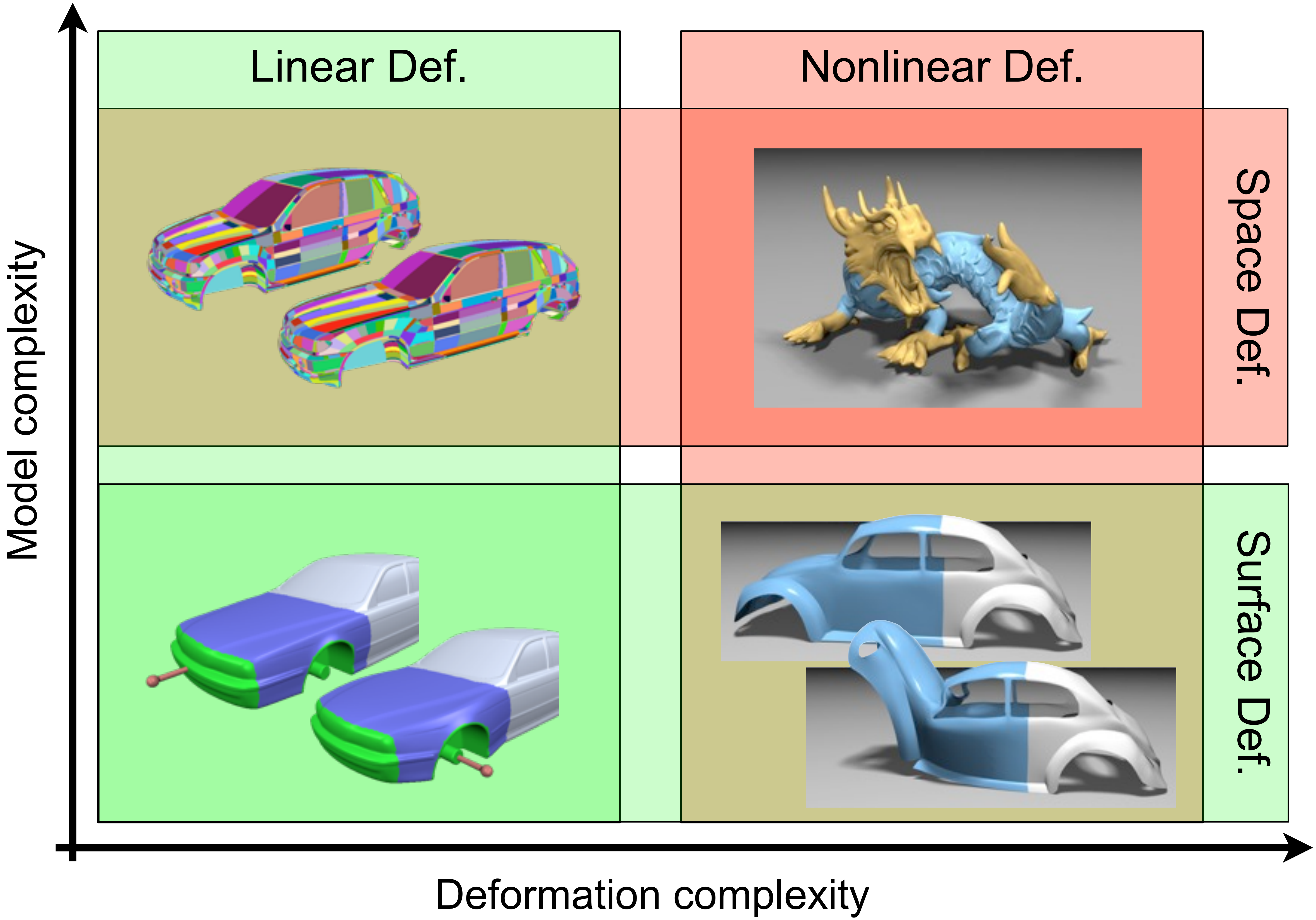


[Sumner et al, SIGGRAPH 07]

Literature

- Botsch et al, *Adaptive space deformations based on rigid cells*, Eurographics 2007
- Sumner et al, *Embedded deformations for shape manipulation*, SIGGRAPH 2007

Shape Deformation



Summary

Bending Energy

- Precise control of continuity
- Requires multi-resolution hierarchy
- Problems with large rotations

vs.

Differential Coords

- Designed for large rotations
- Problems with translations
- How to determine local rotations?

Summary

Surface-Based

+ More precise control of surface properties

– Depends on surface complexity & quality

vs.

Space Deformation

– Doesn’t know about embedded surface

+ Works for complex and “bad” input

Summary

Linear

+ Highly efficient & numerically robust
– Many constraints for large-scale edits

vs.

Nonlinear

– Numerically much more complex
+ Easier edits, fewer constraints

Literature

- Polygon Mesh Processing, Chapter 9