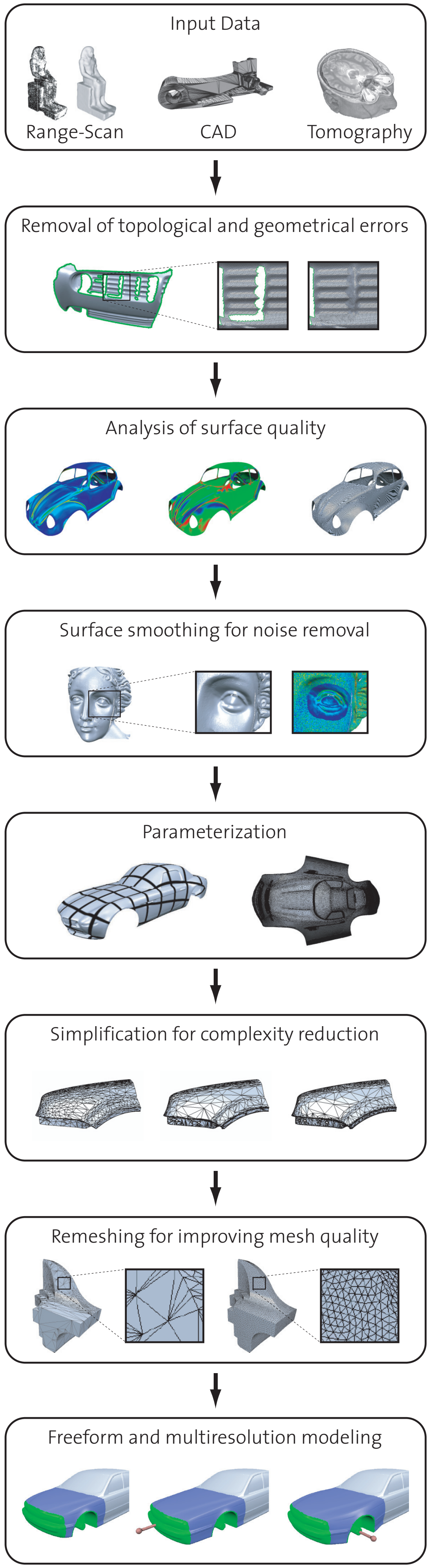
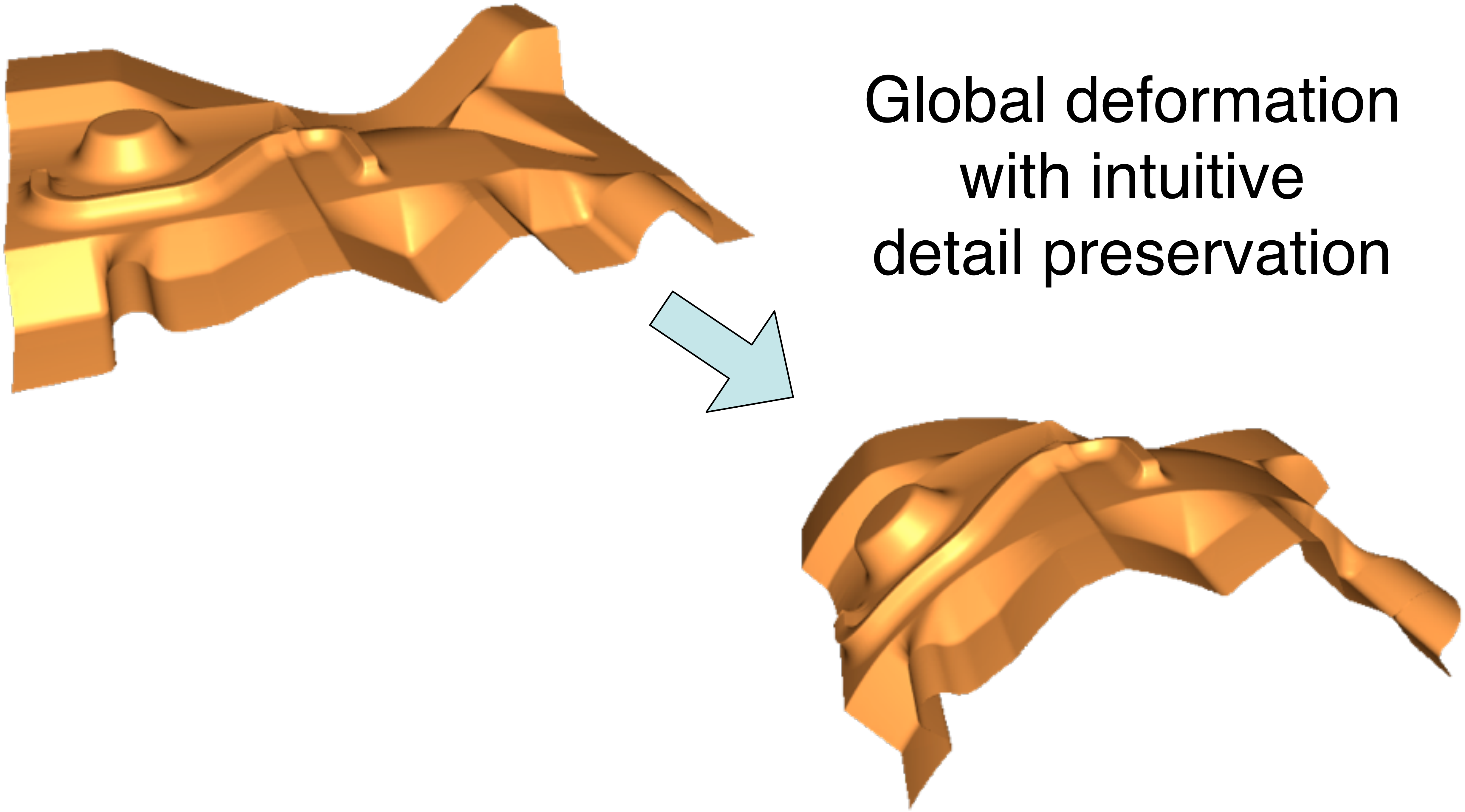


Mesh Editing

Mario Botsch

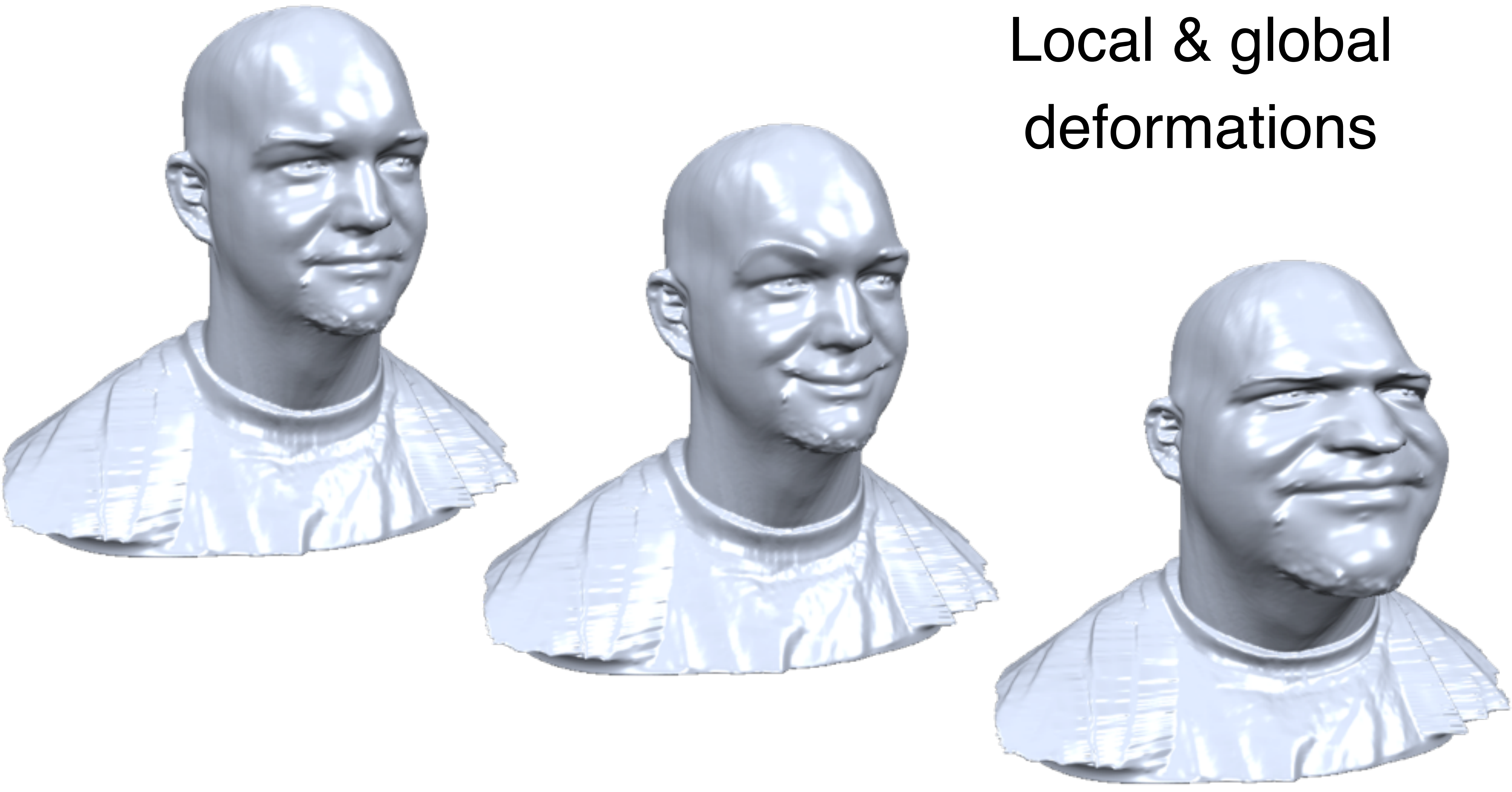
Bielefeld University

Mesh Deformation



Global deformation
with intuitive
detail preservation

Mesh Deformation



Local & global deformations

Mesh Deformation

Editing of complex meshes

Mesh Deformation

Editing of
"bad" meshes

Overview

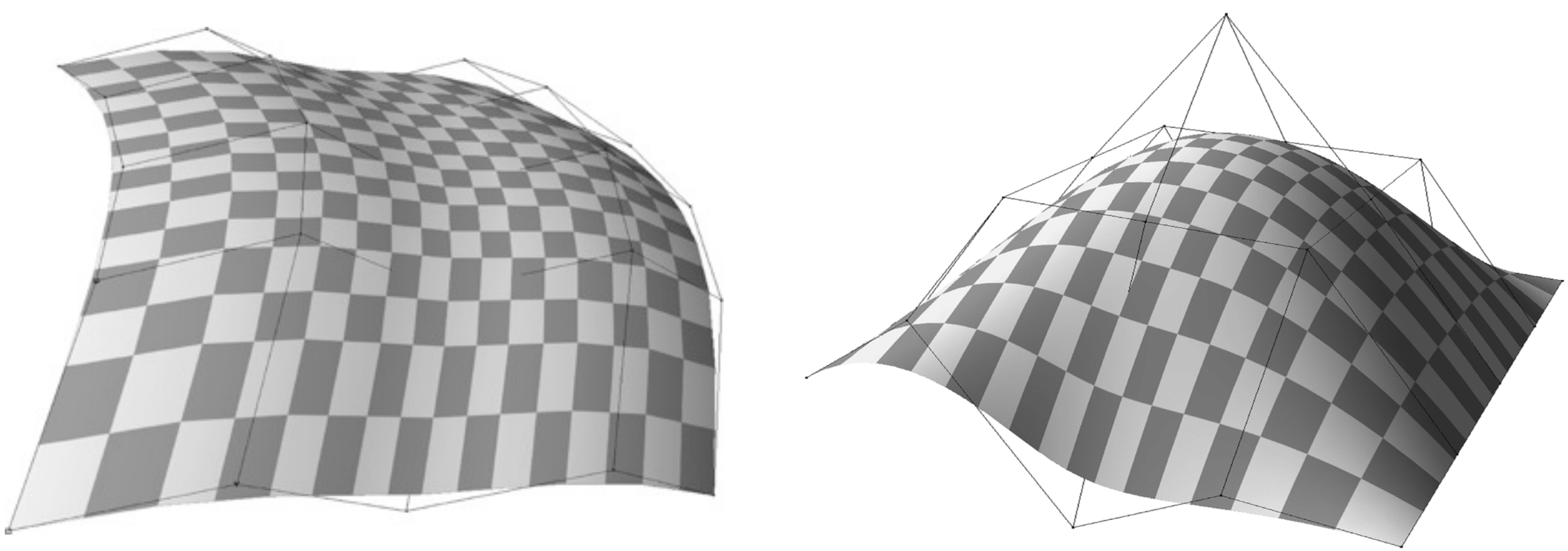
- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods

Overview

- **Surface-based deformation**
  - **Energy minimization**
  - Multiresolution editing
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods

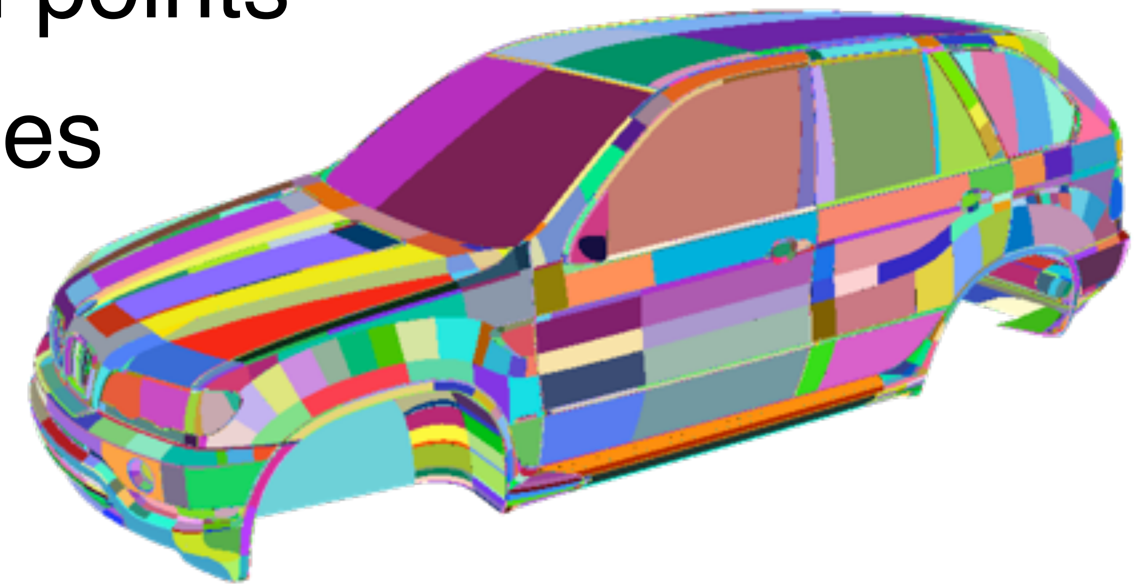
Spline Surfaces

- Tensor product surfaces
  - Rectangular grid of control points
  - Rectangular surface patches



Spline Surfaces

- Tensor product surfaces
  - Rectangular grid of control points
  - Rectangular surface patches

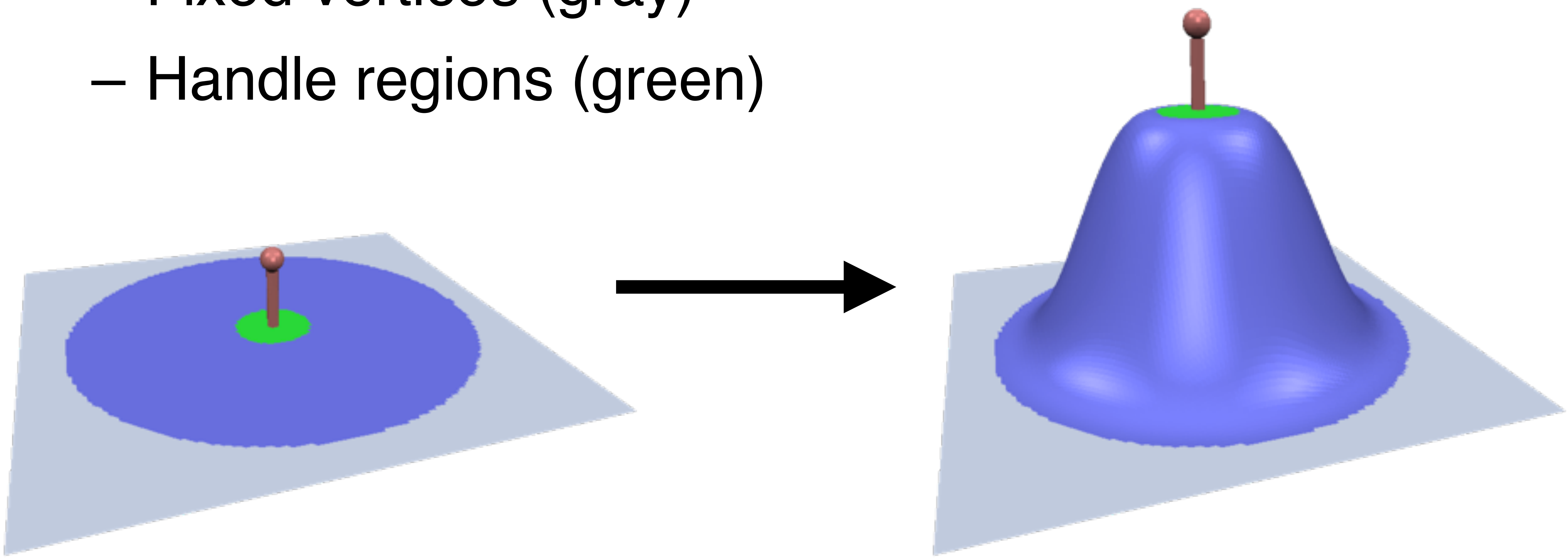


- Problems:
  - Many patches for complex models
  - Smoothness across patch boundaries

➡ Use irregular triangle meshes!

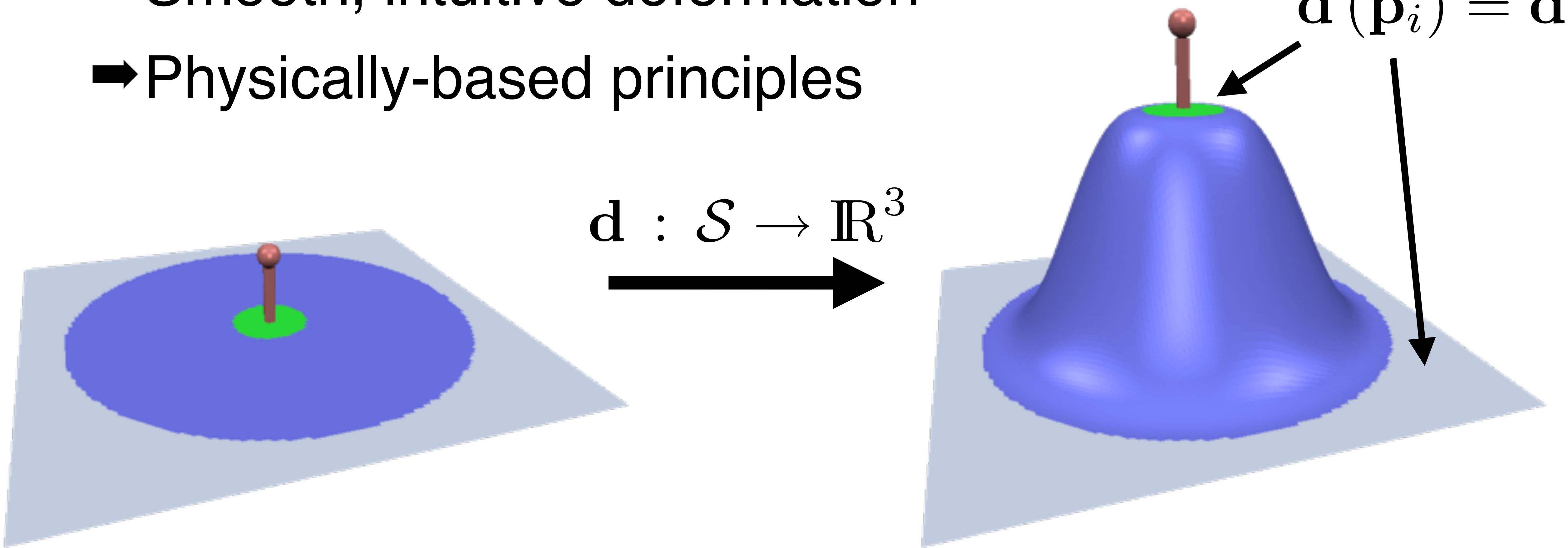
Modeling Metaphor

- Paint three surface regions
  - Support region (blue)
  - Fixed vertices (gray)
  - Handle regions (green)



Modeling Notation

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



$$\mathbf{d}\left(\mathbf{p}_i\right) = \mathbf{d}_i$$

$$\mathbf{d} \,:\, \mathcal{S} \rightarrow \mathbb{R}^3$$

$$\mathcal{S}$$

$$\mathcal{S}' = \left\{\mathbf{p} + \mathbf{d}\left(\mathbf{p}\right) \,|\, \mathbf{p} \in \mathcal{S}\right\}$$

Physically-Based Deformation

- Non-linear 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 energies

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

Physically-Based Deformation

- Minimize linearized bending energy

$$E(\mathbf{d}) \;=\; \int_{\mathcal{S}} \left\|\mathbf{d}_{uu}\right\|^2 + 2\left\|\mathbf{d}_{uv}\right\|^2 + \left\|\mathbf{d}_{vv}\right\|^2 \mathrm{d}\mathcal{S}$$

$f(x) \rightarrow \min$

- Variational calculus, Euler-Lagrange PDE

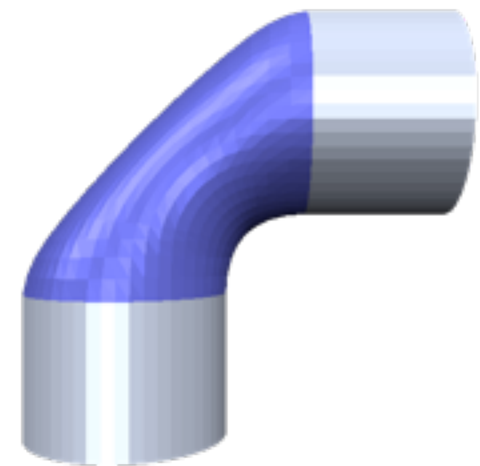
$$\Delta^2\mathbf{d} \;:=\; \mathbf{d}_{uuuu} + 2\mathbf{d}_{uuvv} + \mathbf{d}_{vvvv} \;=\; 0$$

$f'(x) = 0$

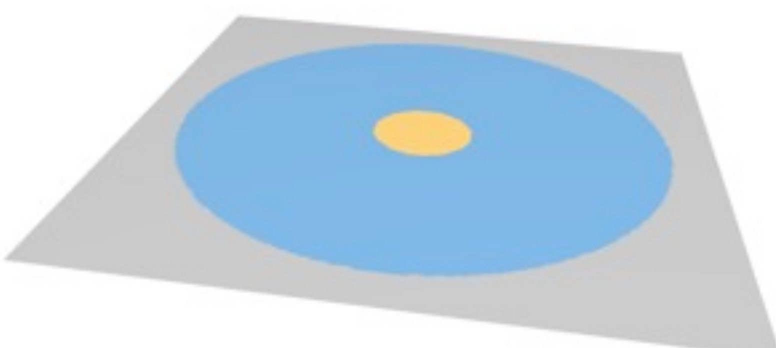
➡ "Best" deformation that satisfies constraints

Deformation Energies

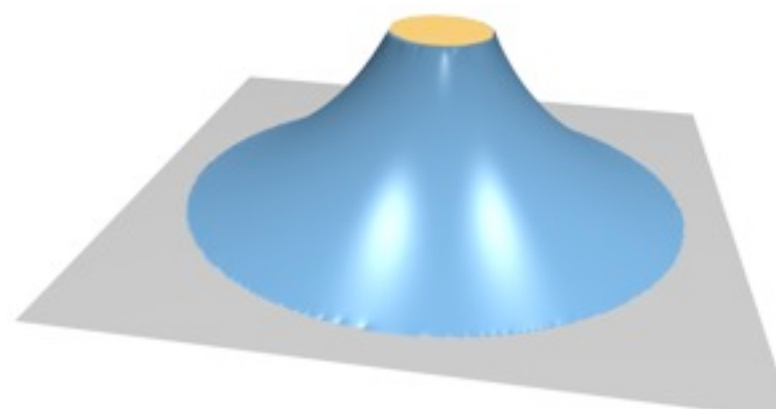
$$\Delta \mathbf{p} = 0$$



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

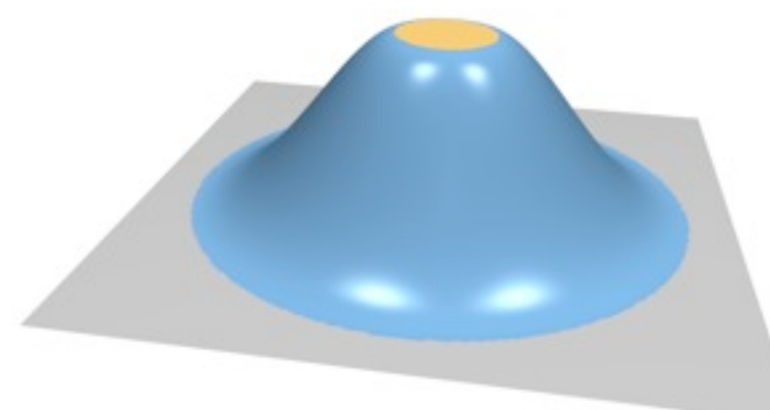


Initial state



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

(Membrane)



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

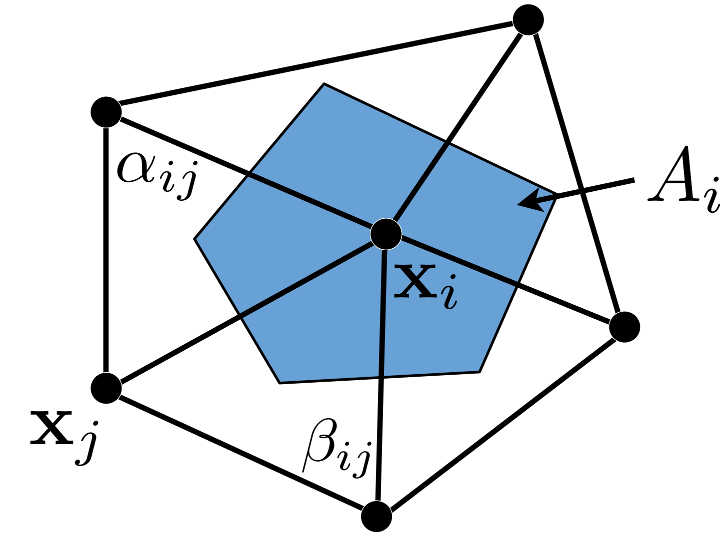
(Thin plate)

Discretization

- Laplace discretization

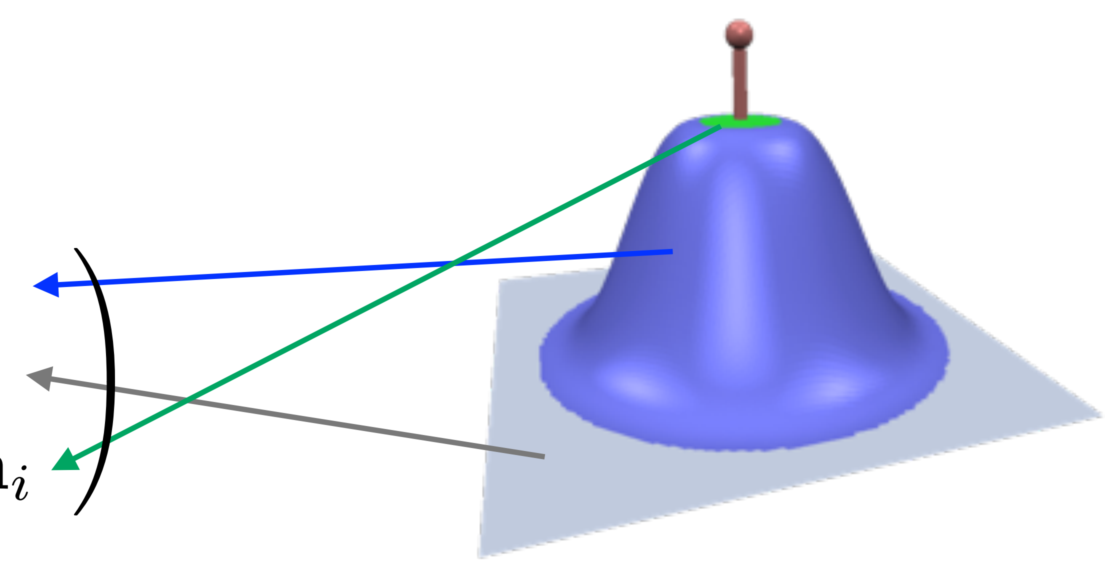
$$\Delta \mathbf{d}_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}_i} (\cot \alpha_{ij} + \cot \beta_{ij})(\mathbf{d}_j - \mathbf{d}_i)$$

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



- Sparse linear system

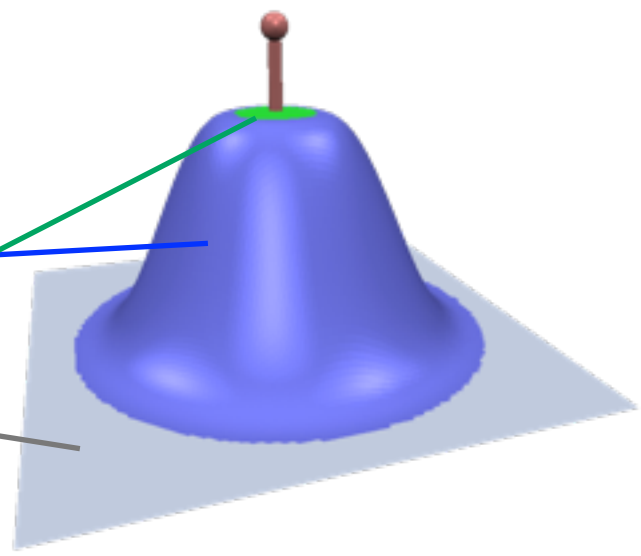
$$\underbrace{\begin{pmatrix} & \Delta^2 & \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}}_{=:\mathbf{M}} \begin{pmatrix} \vdots \\ \mathbf{d}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \delta \mathbf{h}_i \end{pmatrix}$$



Discretization

- Sparse linear system (19 nz/row)

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



- Can be turned into symm. pos. def. system
  - Right hand sides changes each frame!
  - Use efficient linear solvers...

Sparse SPD Solvers

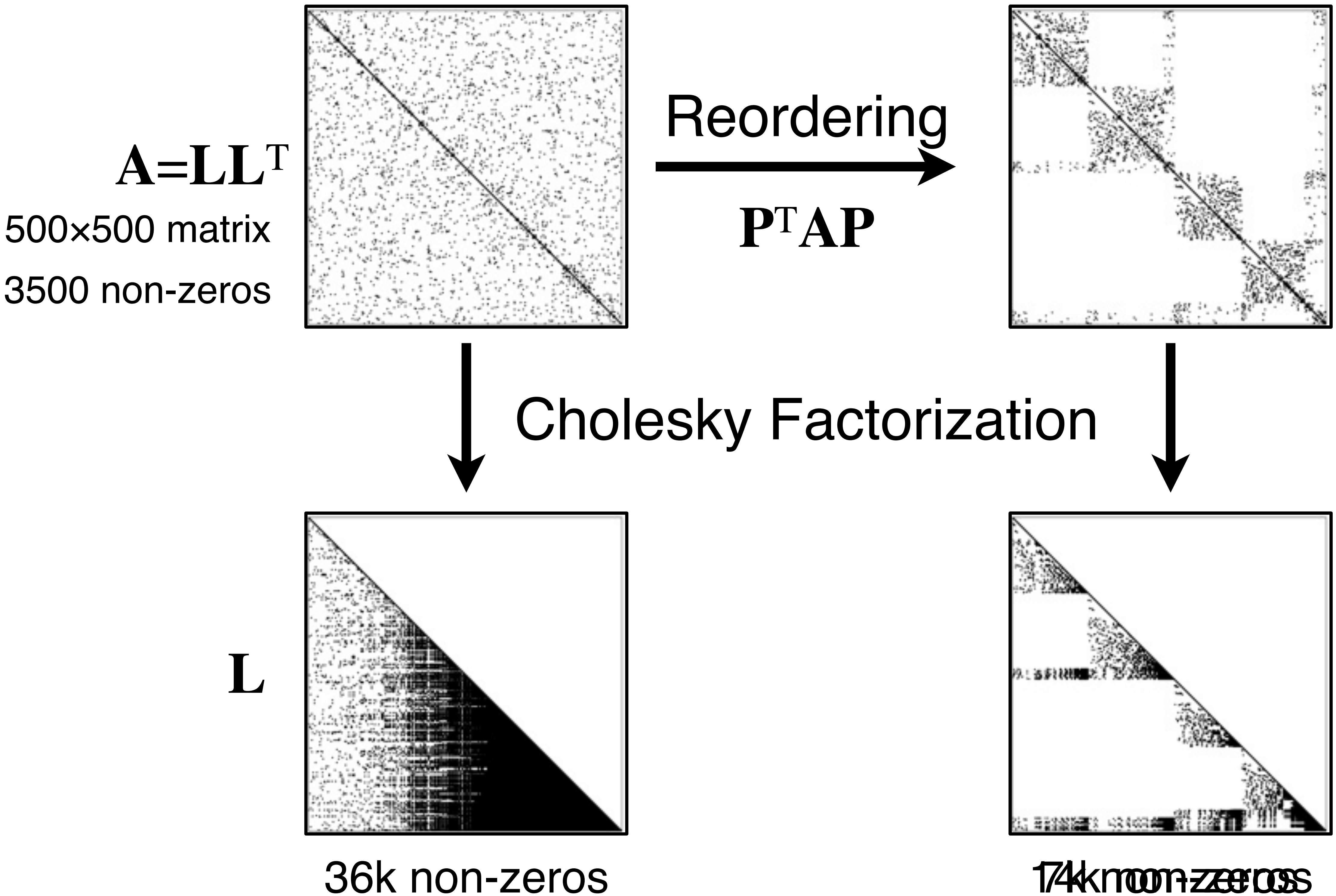
- Dense Cholesky factorization
  - Cubic complexity
  - High memory consumption (doesn't exploit sparsity)
- Iterative conjugate gradients
  - Quadratic complexity
  - Need sophisticated preconditioning
- Multigrid solvers
  - Linear complexity
  - But rather complicated to develop (and to use)
- Sparse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

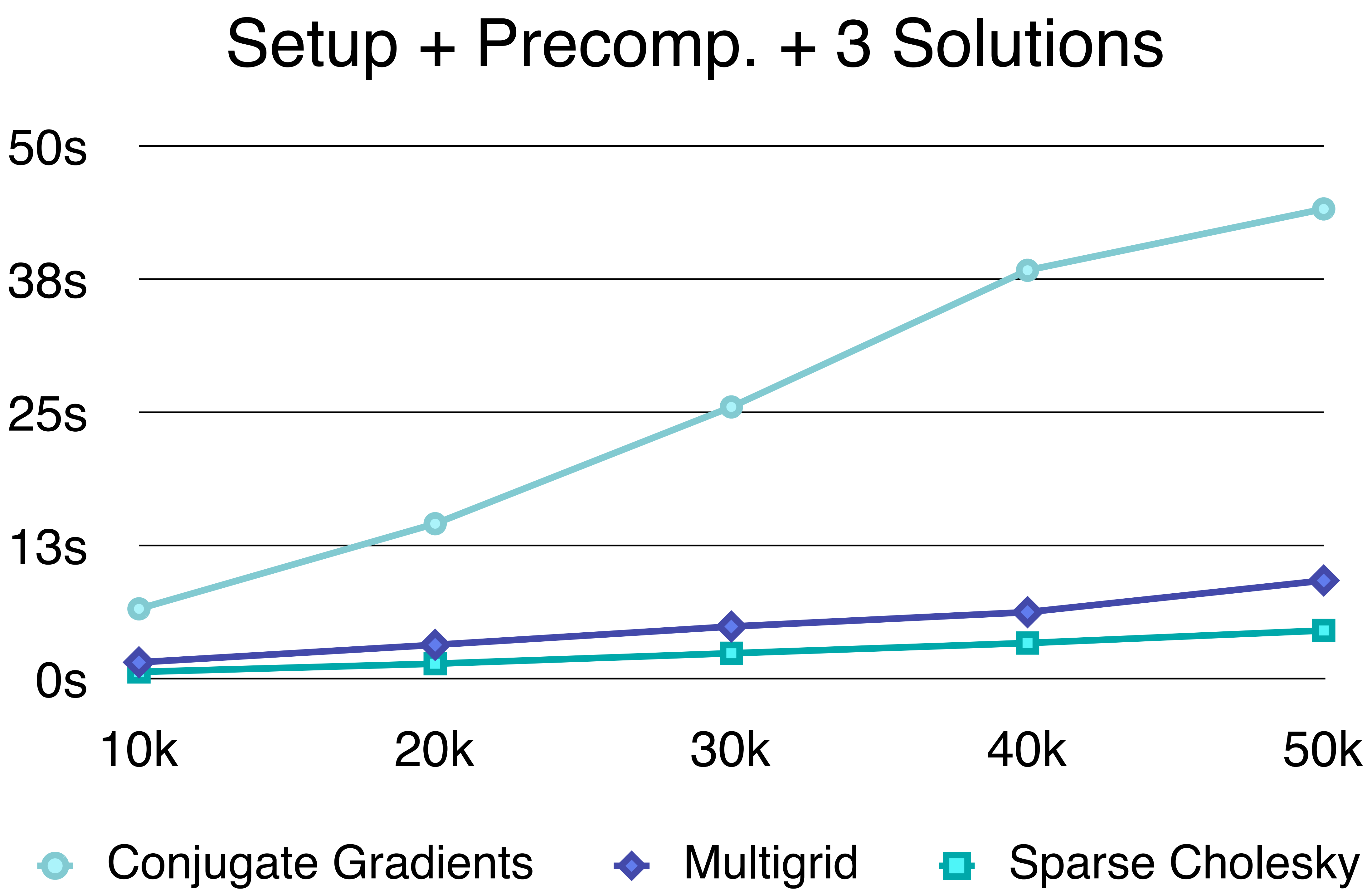
Solve $\mathbf{Ax} = \mathbf{b}$

Pre-computation

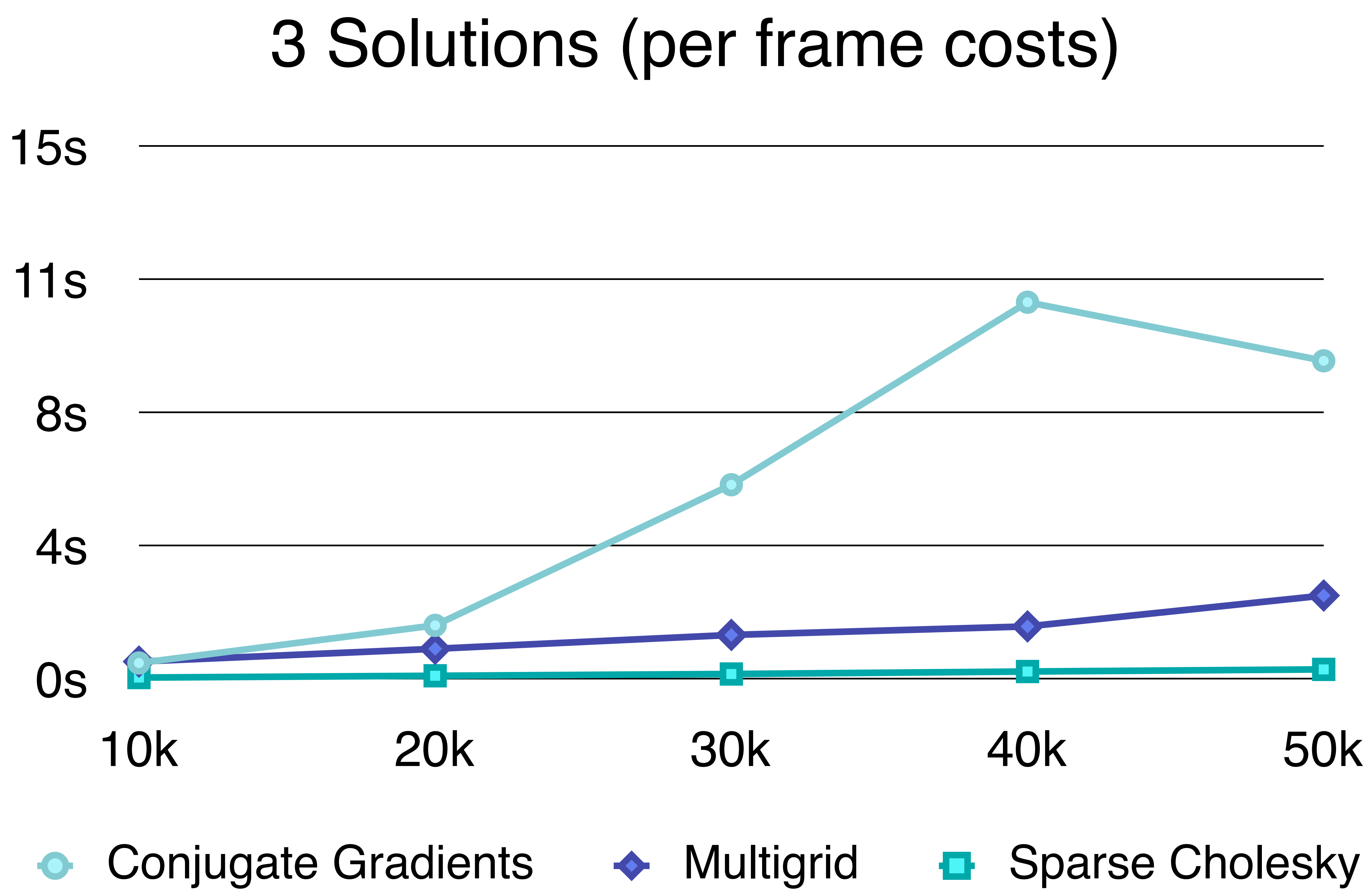
1. Matrix re-ordering $\tilde{\mathbf{A}} = \mathbf{P}^T\mathbf{AP}$
2. Cholesky factorization $\tilde{\mathbf{A}} = \mathbf{LL}^T$
3. Solve system $\mathbf{y} = \mathbf{L}^{-1}\mathbf{P}^T\mathbf{b}, \quad \mathbf{x} = \mathbf{PL}^{-T}\mathbf{y}$

Per-frame computation

Bi-Laplace Systems

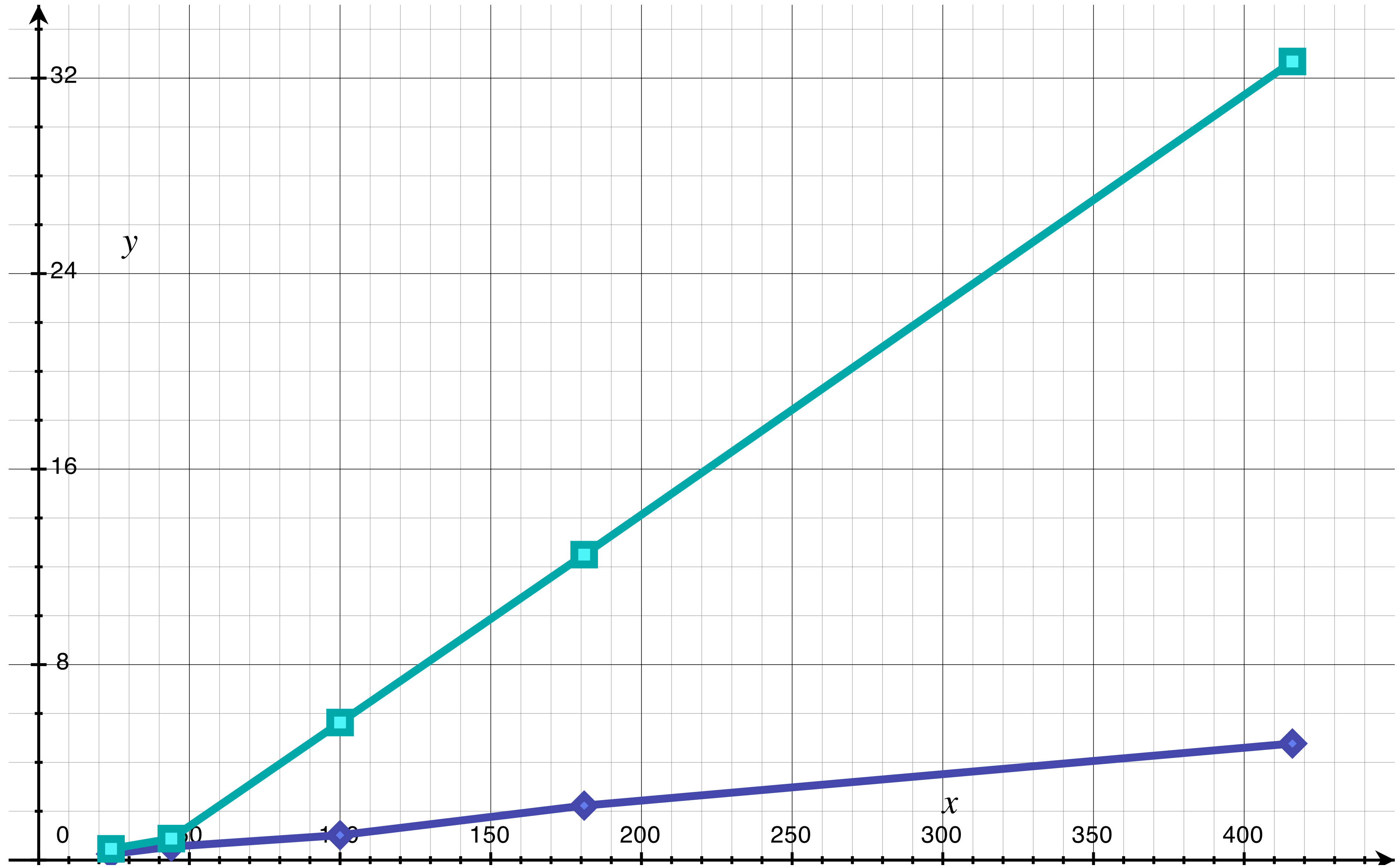


Bi-Laplace Systems



Laplace System

Setup + Precomp. + 3 Solutions



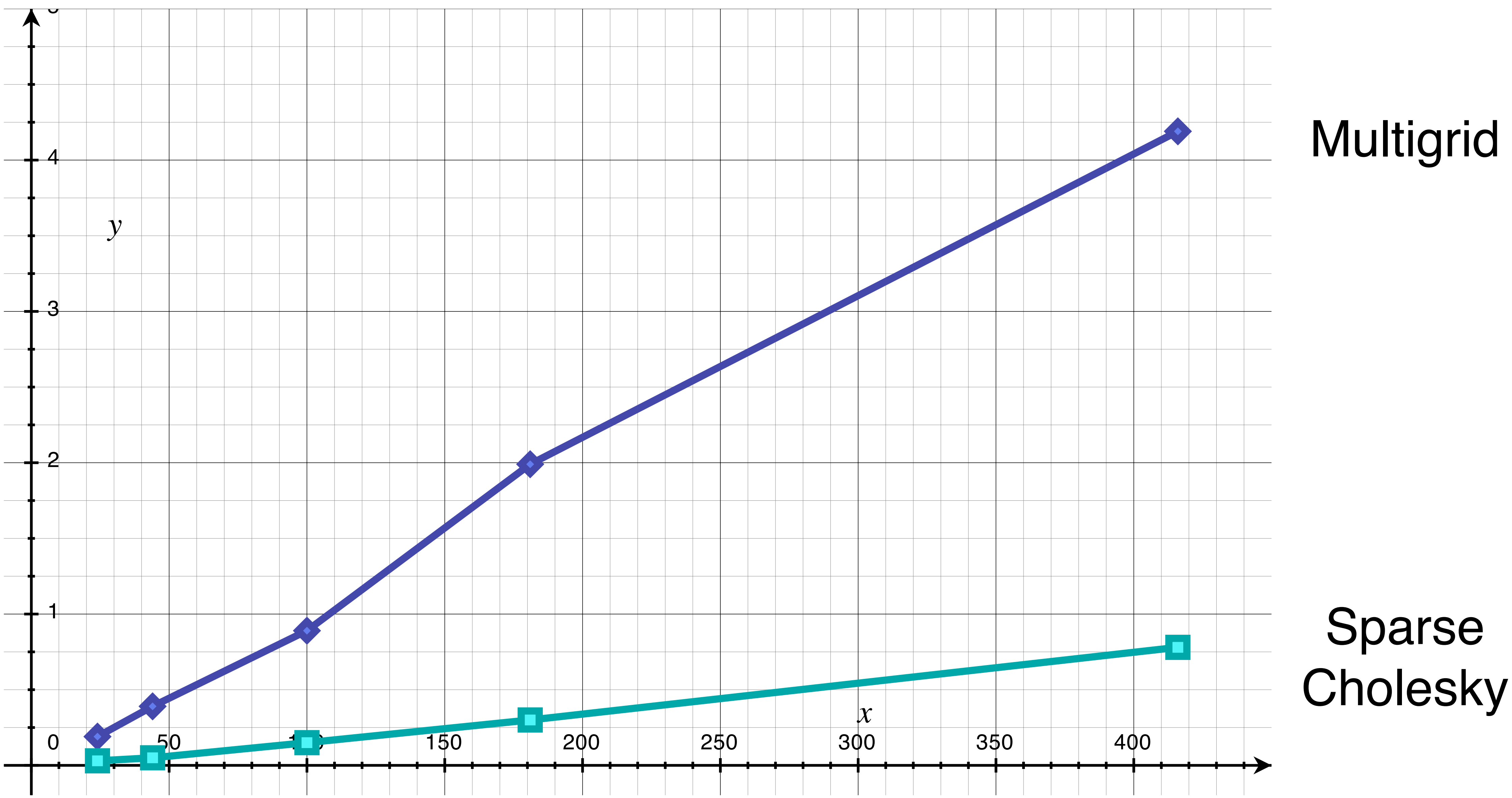
Sparse Cholesky

Multigrid

[Shi et al, SIGGRAPH 06]

Laplace System

3 Solutions (per frame costs)

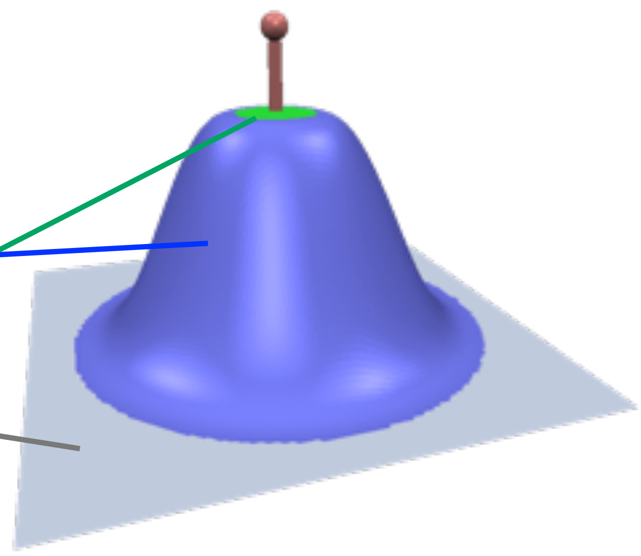


[Shi et al, SIGGRAPH 06]

Discretization

- Sparse linear system

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



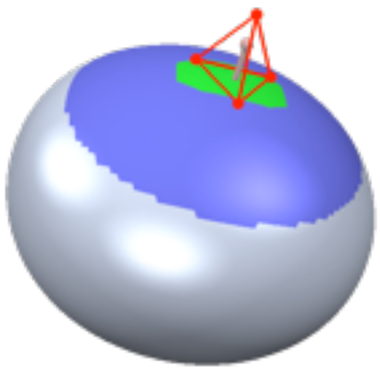
- Can be turned into symm. pos. def. system
  - Right hand sides changes each frame
  - Sparse Cholesky factorization
  - Very efficient implementations publicly available

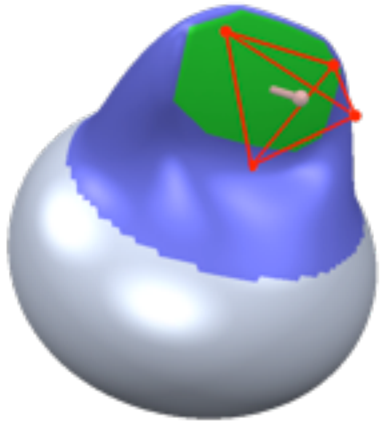
More Efficient Solution

- Handle is transformed affinely

$$(\ldots, \mathbf{h}_i, \ldots) = \mathbf{Q}\,(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})^T$$

$$(\ldots, \delta\mathbf{h}_i, \ldots) = \mathbf{Q}\,(\delta\mathbf{a}, \delta\mathbf{b}, \delta\mathbf{c}, \delta\mathbf{d})^T$$

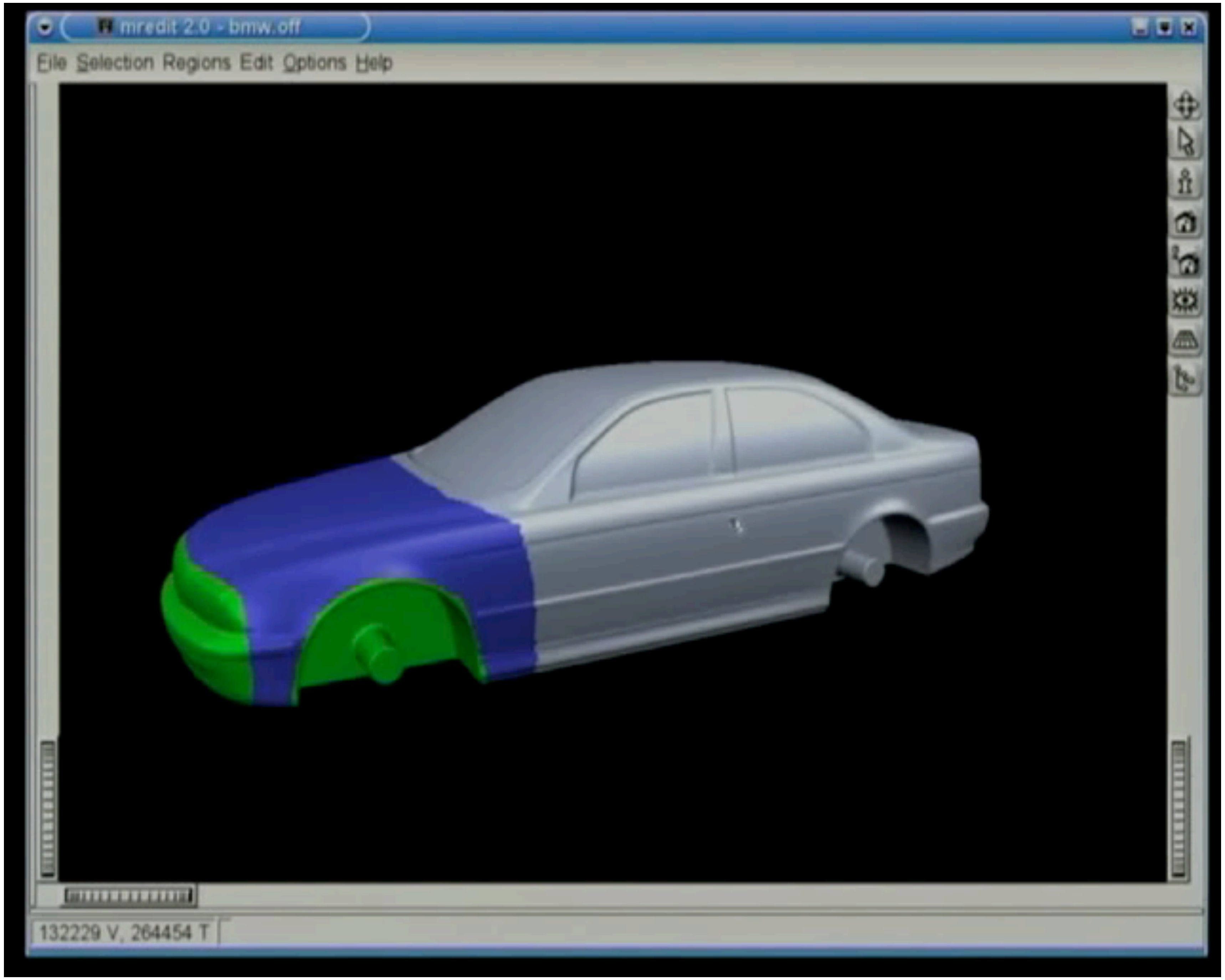




- Precompute linear basis functions

$$\begin{pmatrix} \vdots \\ \mathbf{d}_i \\ \vdots \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \delta\mathbf{h}_i \end{pmatrix} = \underbrace{\mathbf{M}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{Q} \end{pmatrix}}_{\mathbf{B} \in \mathbb{R}^{n \times 4}} (\delta\mathbf{a}, \delta\mathbf{b}, \delta\mathbf{c}, \delta\mathbf{d},)^T$$

Front Deformation



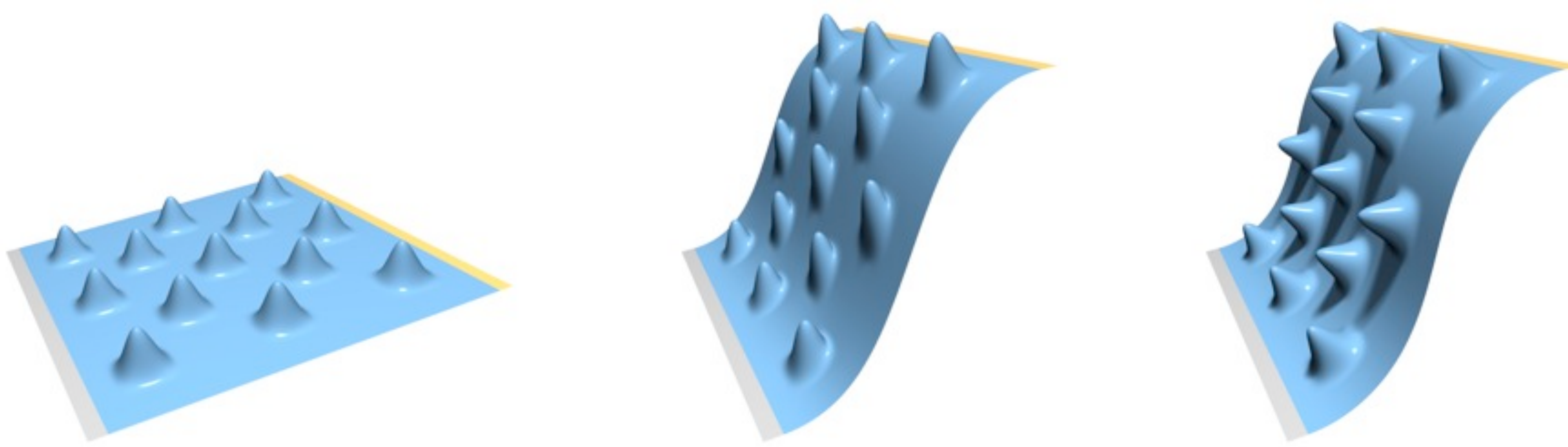
Overview

- **Surface-based deformation**
  - Energy minimization
  - **Multiresolution editing**
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods

Multiresolution Modeling

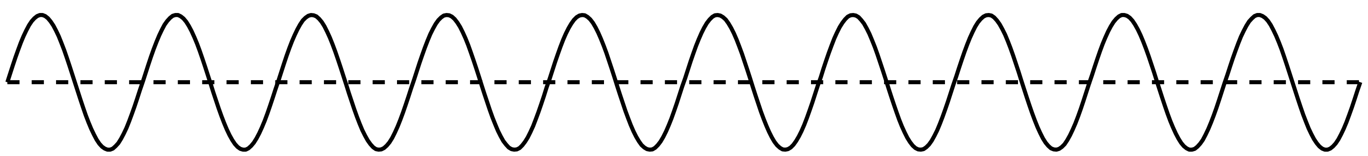
- Even pure translations induce local rotations!
  - Inherently <u>nonlinear</u> coupling

- Or: Linear model + multi-scale decomposition...



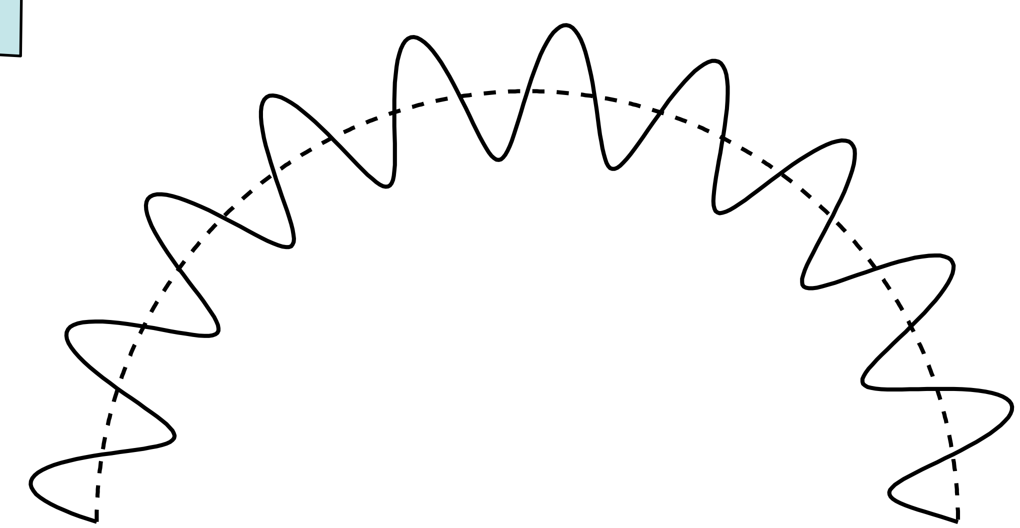
Original　　Linear deform.　　Non-linear deform.

Multiresolution Editing



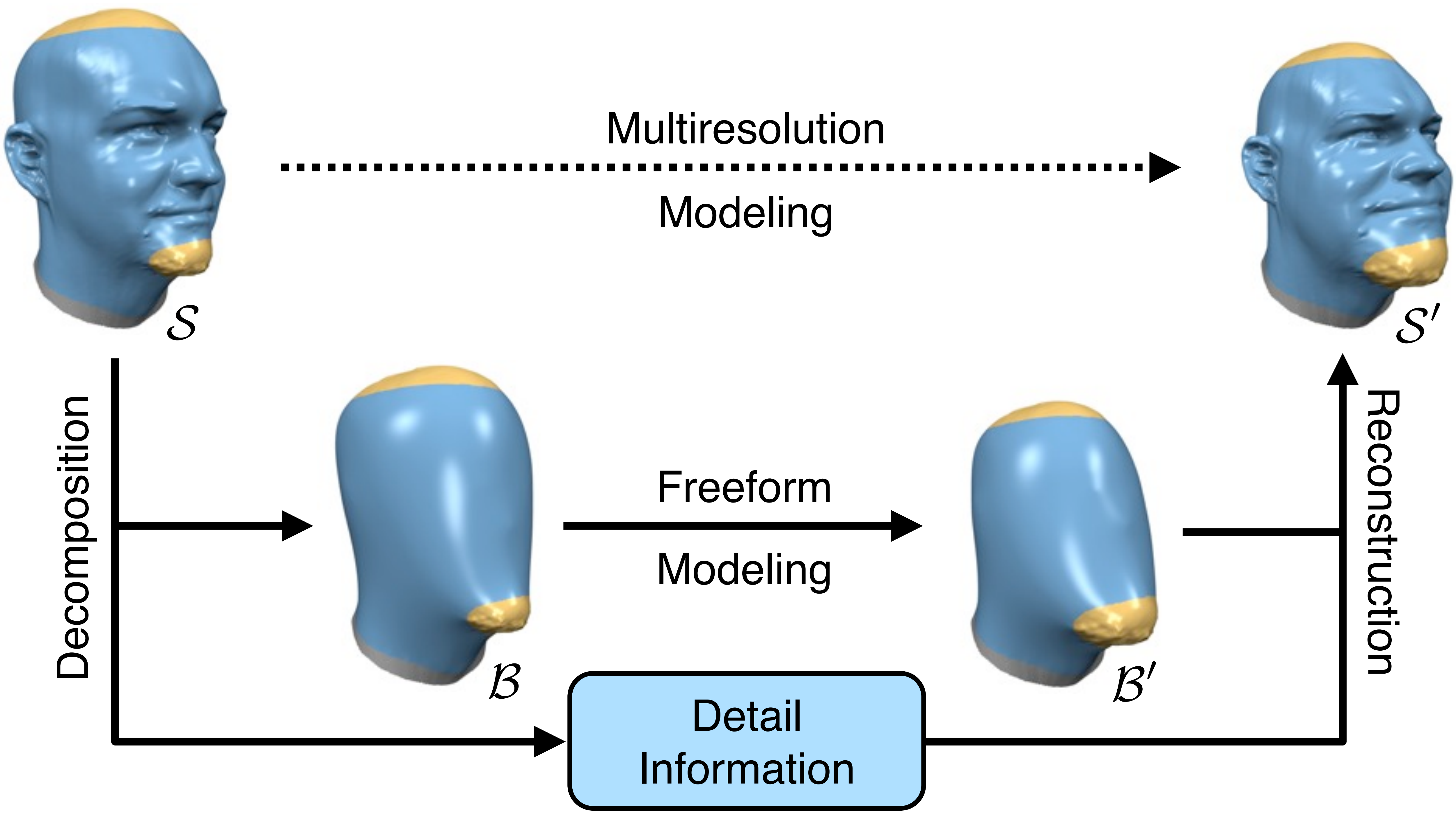
Frequency decomposition

Change low
frequencies

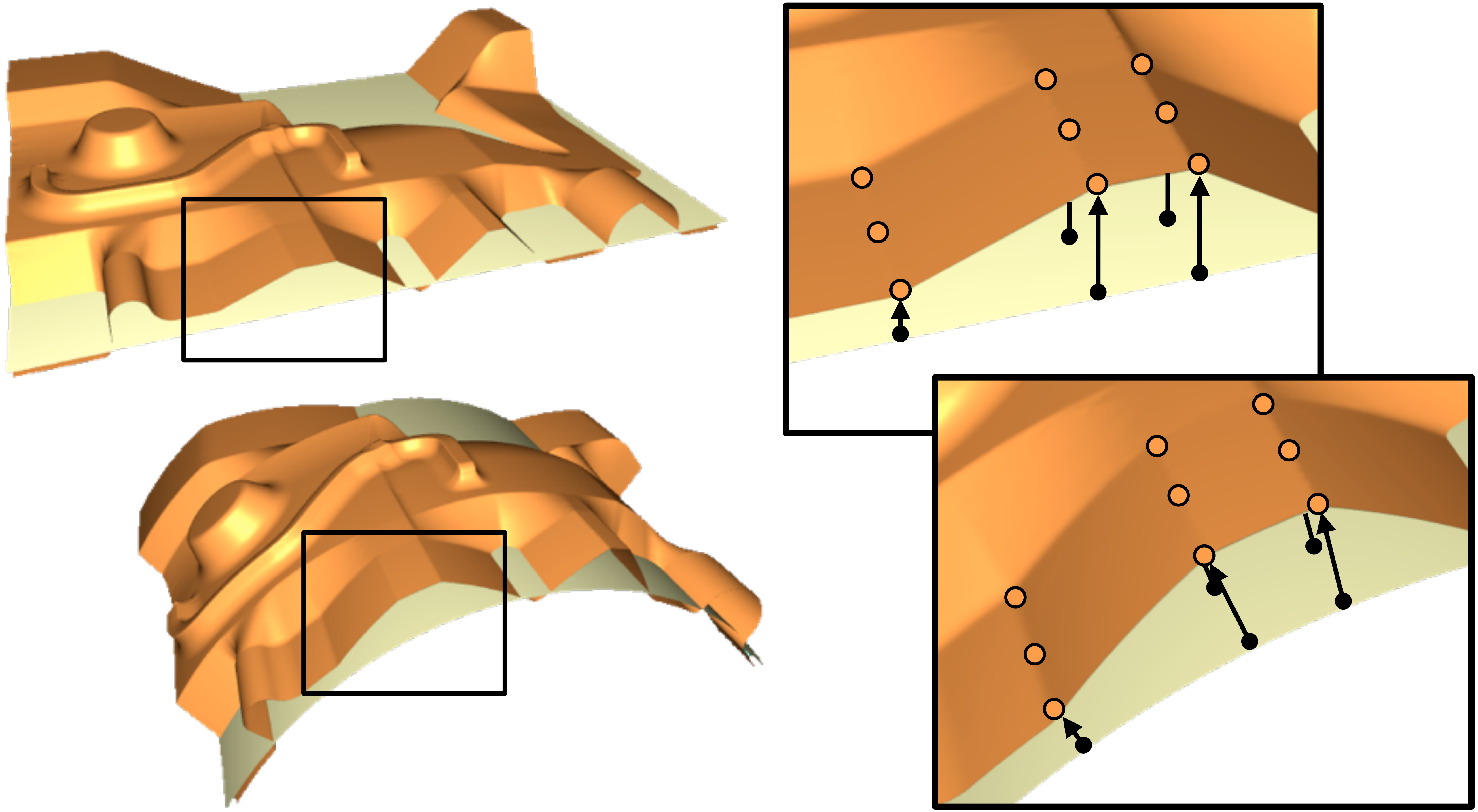


Add high frequency details,
stored in local frames

Multiresolution Editing

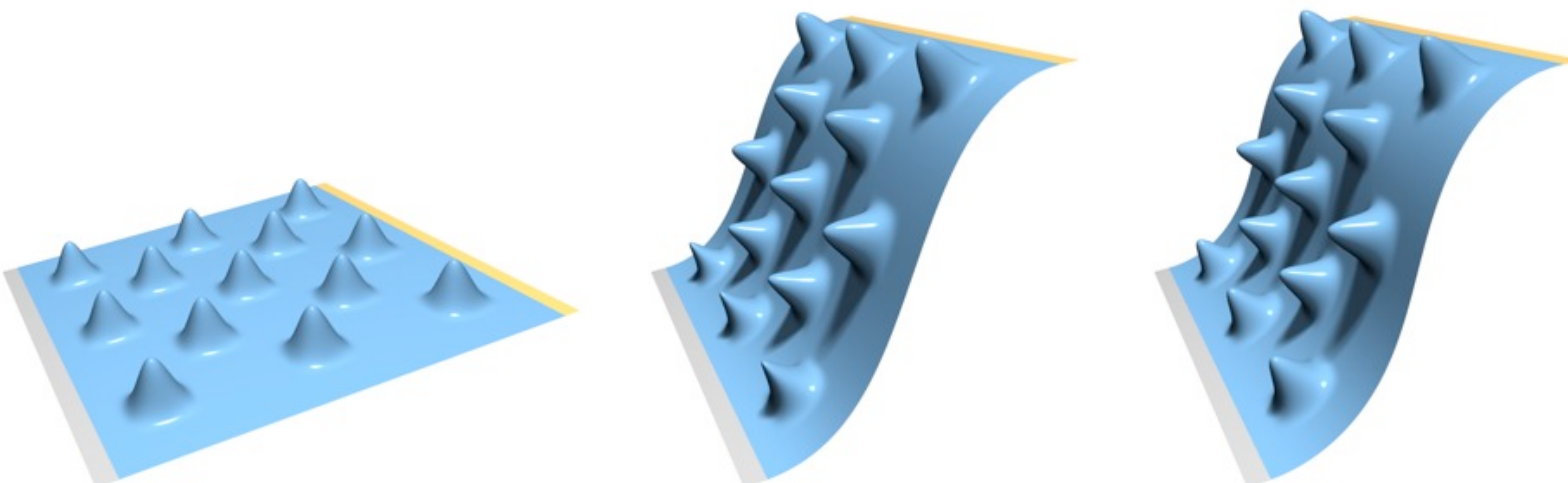


Normal Displacements



Limitations

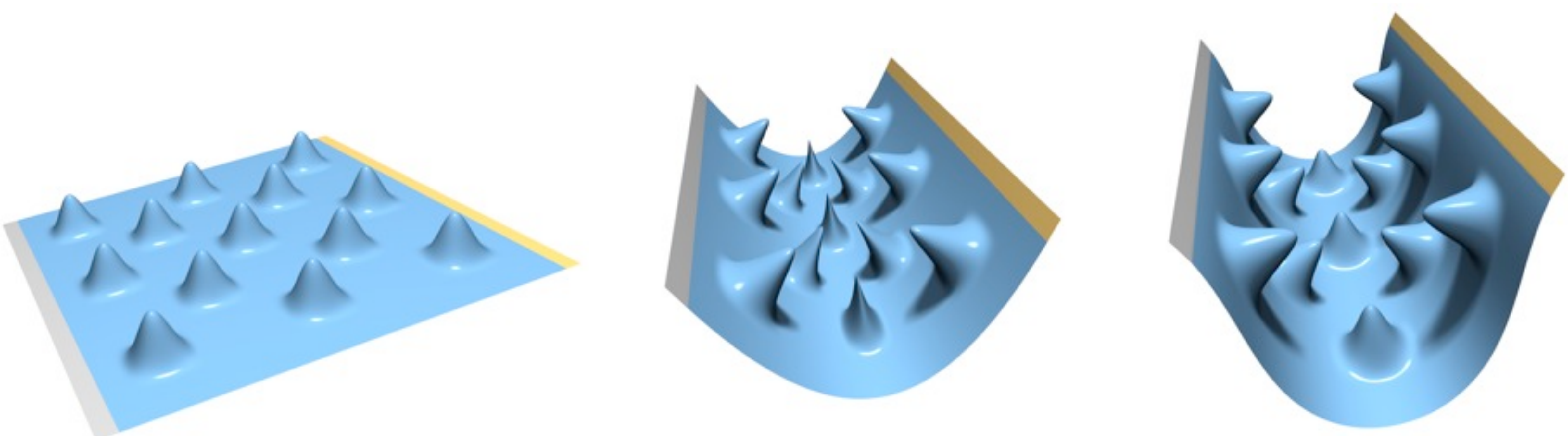
- Neighboring displacements are not coupled
  - Surface bending changes their angle
  - Leads to volume changes or self-intersections



Original　　　Normal Displ.　　　Nonlinear

Limitations

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



Original | Normal Displ. | Nonlinear

Limitations

- Neighboring displacements are not coupled
  - Surface bending changes their angle
  - Leads to volume changes or self-intersections
  - See course notes for some other techniques...

- Multiresolution hierarchy difficult to compute for meshes of complex topology / geometry
  - Might require more hierarchy levels

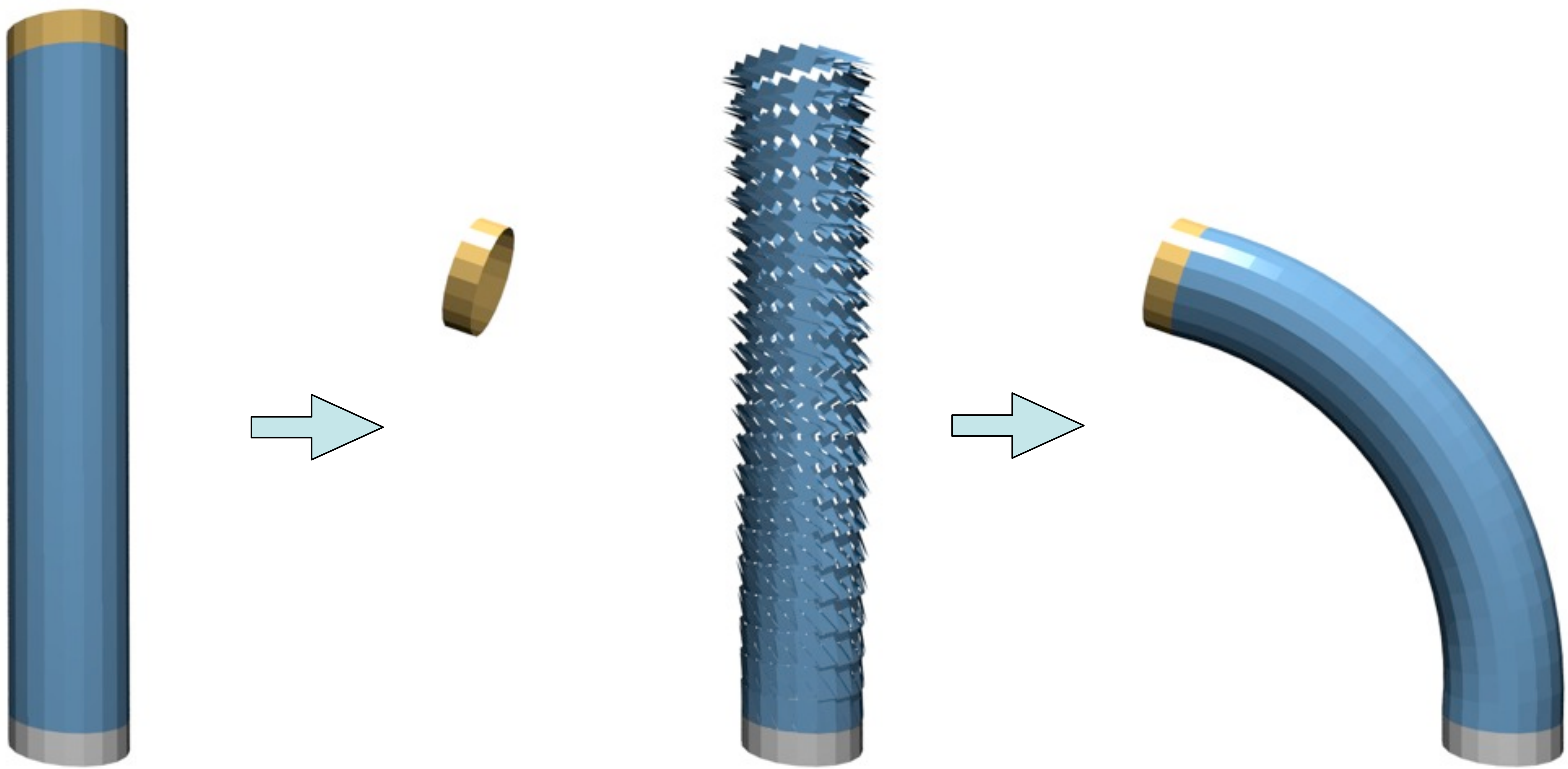
Overview

- **Surface-based deformation**
  - Energy minimization
  - Multiresolution editing
  - **Differential coordinates**
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods

Differential Coordinates

1. Manipulate differential coordinates instead of spatial coordinates
   - Gradients, Laplacians, ...
2. Then find mesh with desired differential coords
   - Basically an integration step

Gradient-Based Editing



Original Rotated DiffCoords Reconstructed Mesh

Gradient-Based Editing

- Manipulate gradient field of a function (surface)

$$\mathbf{g} = \nabla \mathbf{p} \qquad \mathbf{g} \mapsto \mathbf{g}'$$

- Find function $\mathbf{f}$' whose gradient is (close to) $\mathbf{g}$'

$$\int_{\mathcal{S}} \|\nabla \mathbf{p}' - \mathbf{g}'\|^2 \, \mathrm{d}\mathcal{S} \;\rightarrow\; \min$$

- Variational calculus yields Euler-Lagrange PDE

$$\Delta \mathbf{p}' = \operatorname{div} \mathbf{g}'$$

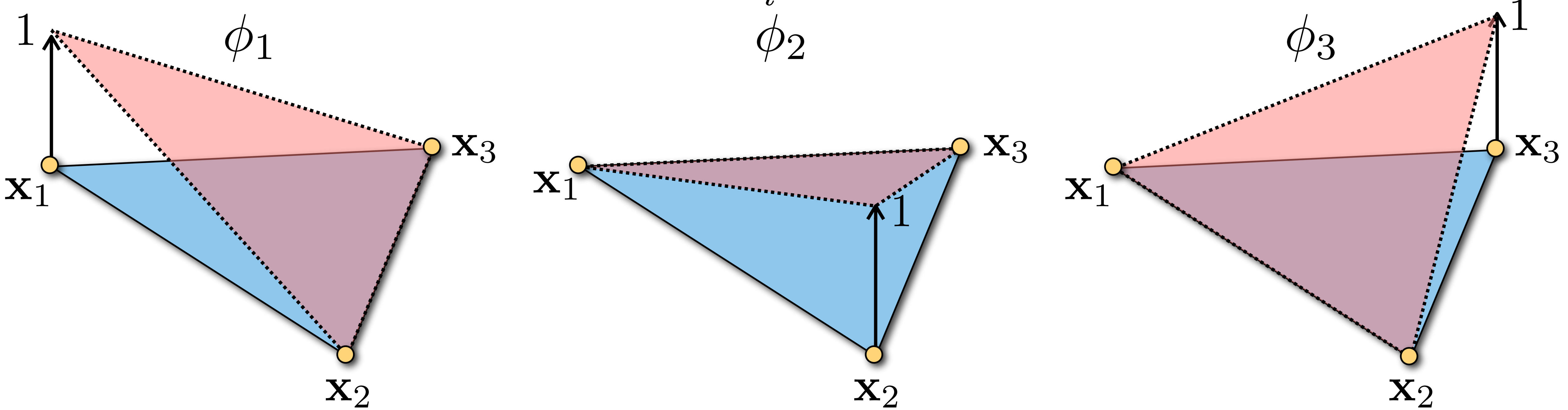
Gradient-Based Editing

- Use 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

- Use 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

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

Gradient-Based Editing

- Constant per triangle $\nabla \mathbf{p}|_{f_j} \;=:\; \mathbf{G}_j \in \mathbb{R}^{3\times 3}$

$$\begin{pmatrix} \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_F \end{pmatrix} = \underbrace{\mathbf{G}}_{\in \mathbb{R}^{3F\times V}} \cdot \begin{pmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_V^T \end{pmatrix}$$

- Manipulate per-face gradients

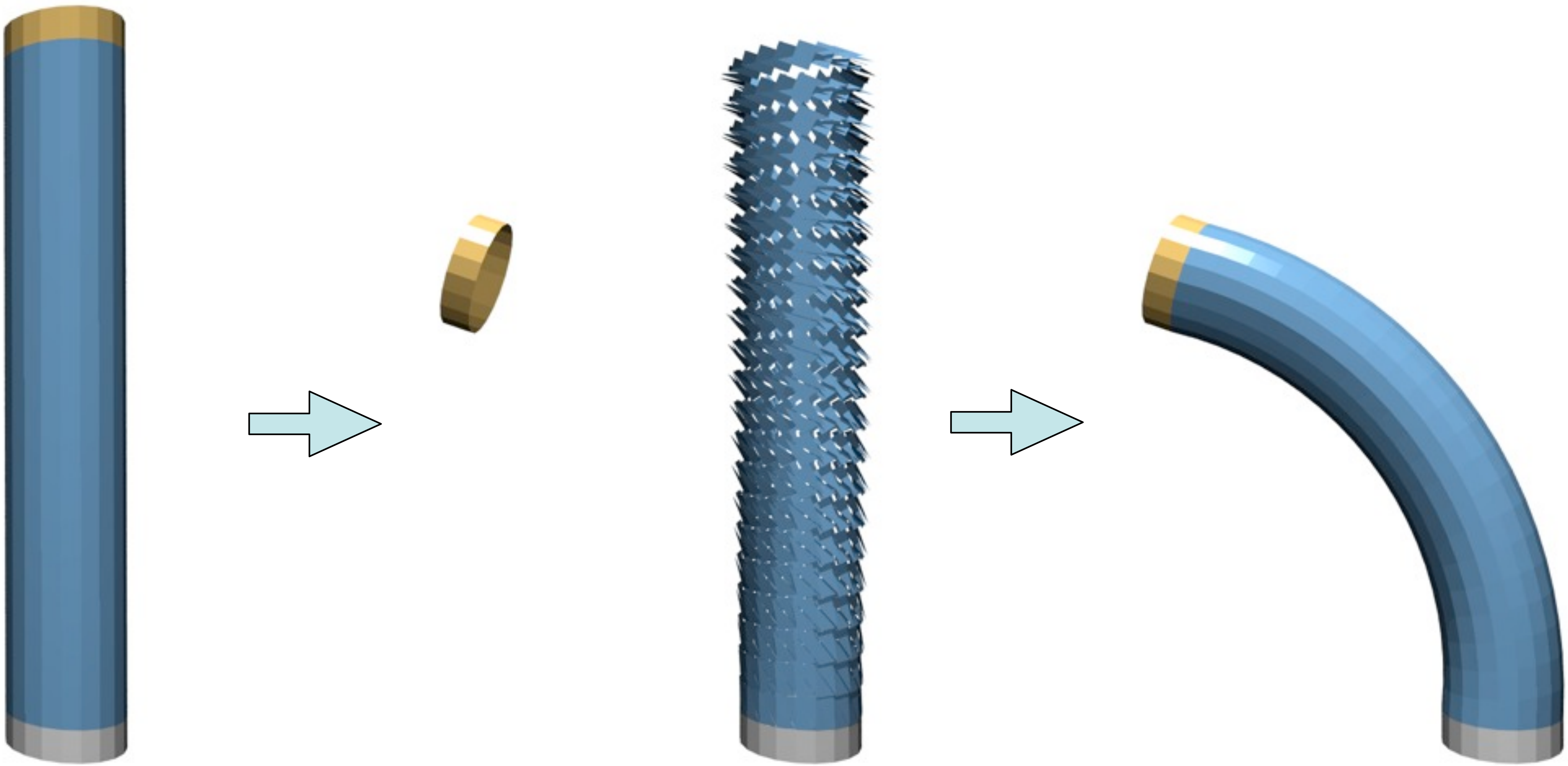
$$\mathbf{G}_j \mapsto \mathbf{G}_j'$$

Gradient-Based Editing

- Reconstruct mesh from changed gradients
  - Overdetermined problem $\mathbf{G} \in \mathbb{R}^{3F \times V}$
  - Weighted least squares system
  - Linear Poisson (Laplace) system

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

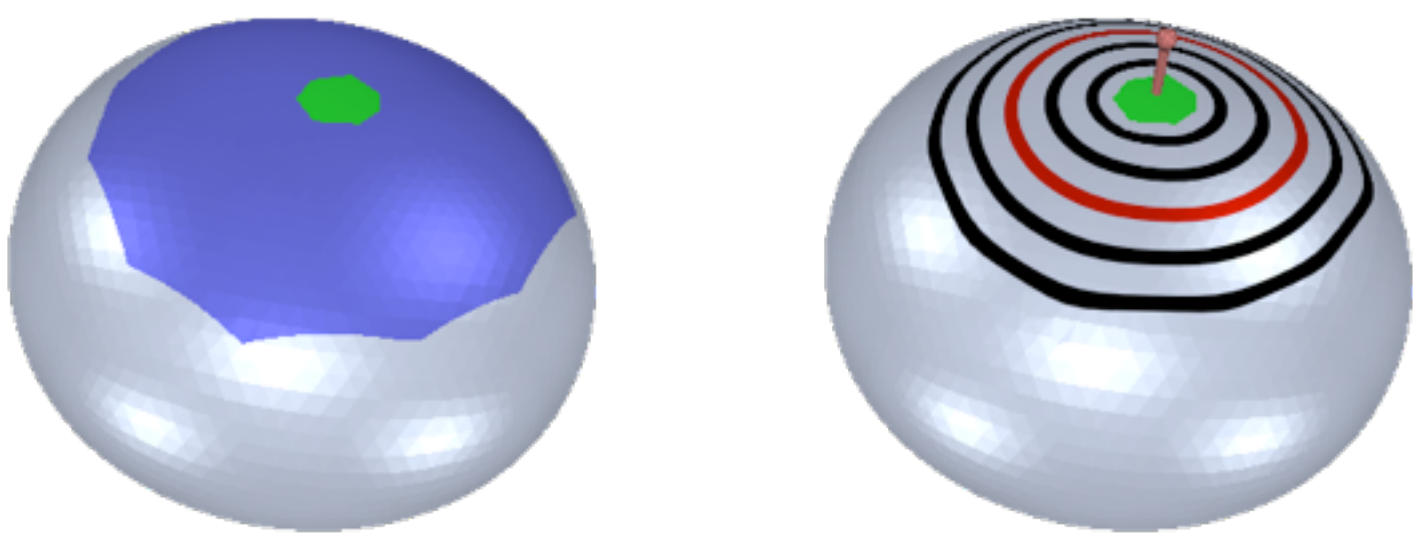
Gradient-Based Editing



Original Rotated DiffCoords Reconstructed Mesh

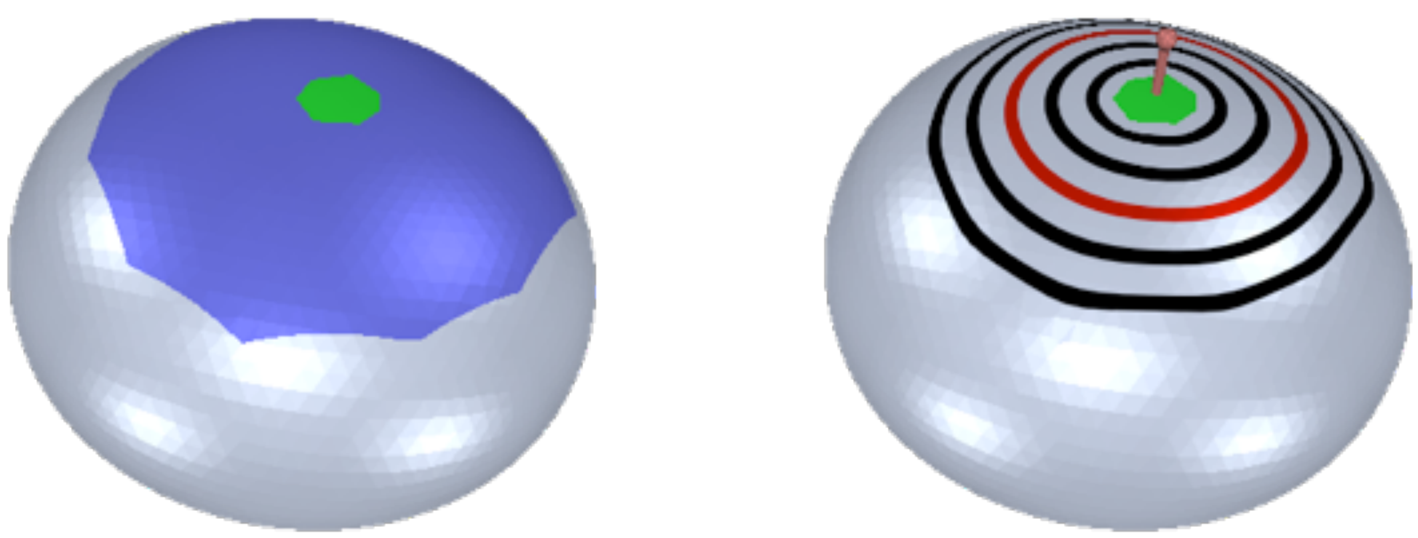
Construct Scalar Field

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



Construct Scalar Field

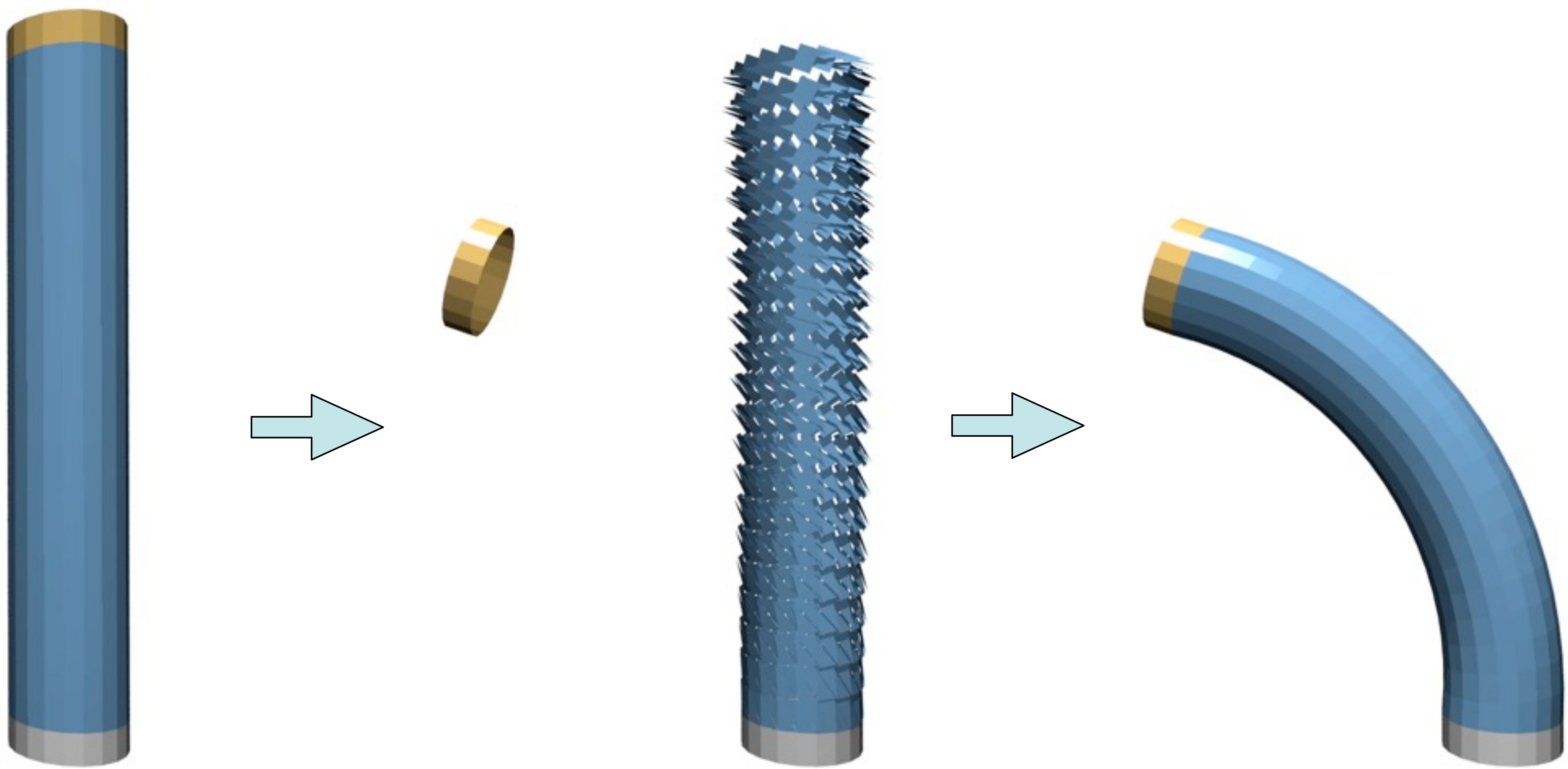
- Construct a smooth harmonic field
  - Solve $\Delta(s) = 0$
  - with $s\left(\mathbf{p}\right) = \begin{cases} 1 & \mathbf{p} \in \text{handle} \\ 0 & \mathbf{p} \in \text{fixed} \end{cases}$



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: $S(\lambda\cdot s + (1-\lambda)\cdot 1)$

Gradient-Based Editing



Original　Rotated　Gradients　Reconstructed Mesh

Laplacian-Based Editing

- Manipulate Laplacians of a surface

$$\boldsymbol{\delta}_i = \Delta(\mathbf{p}_i) \ , \quad \boldsymbol{\delta}_i \mapsto \boldsymbol{\delta}'_i$$

- Find surface whose Laplacian is (close to) $\boldsymbol{\delta}$'

$$\int_{\mathcal{S}} \left\| \Delta \mathbf{p}' - \boldsymbol{\delta}' \right\|^2 \mathrm{d}\mathcal{S} \ \rightarrow \ \min$$

- Variational calculus yields Euler-Lagrange PDE

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

Discretization

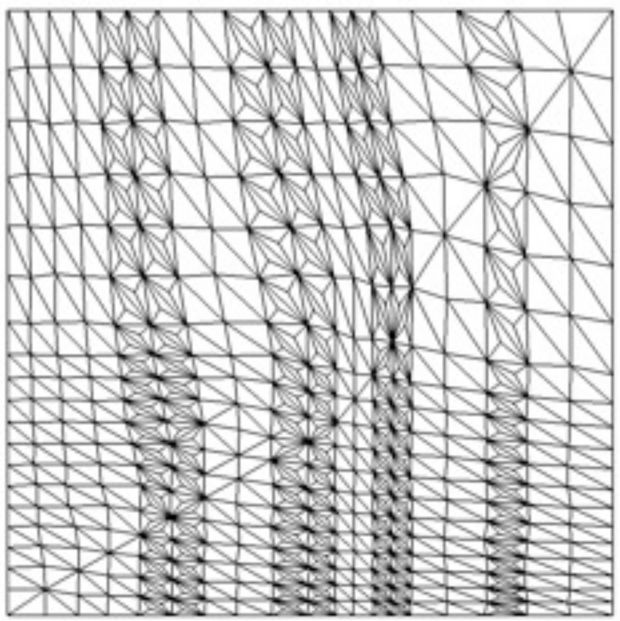
- Discretize Euler-Lagrange PDE

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

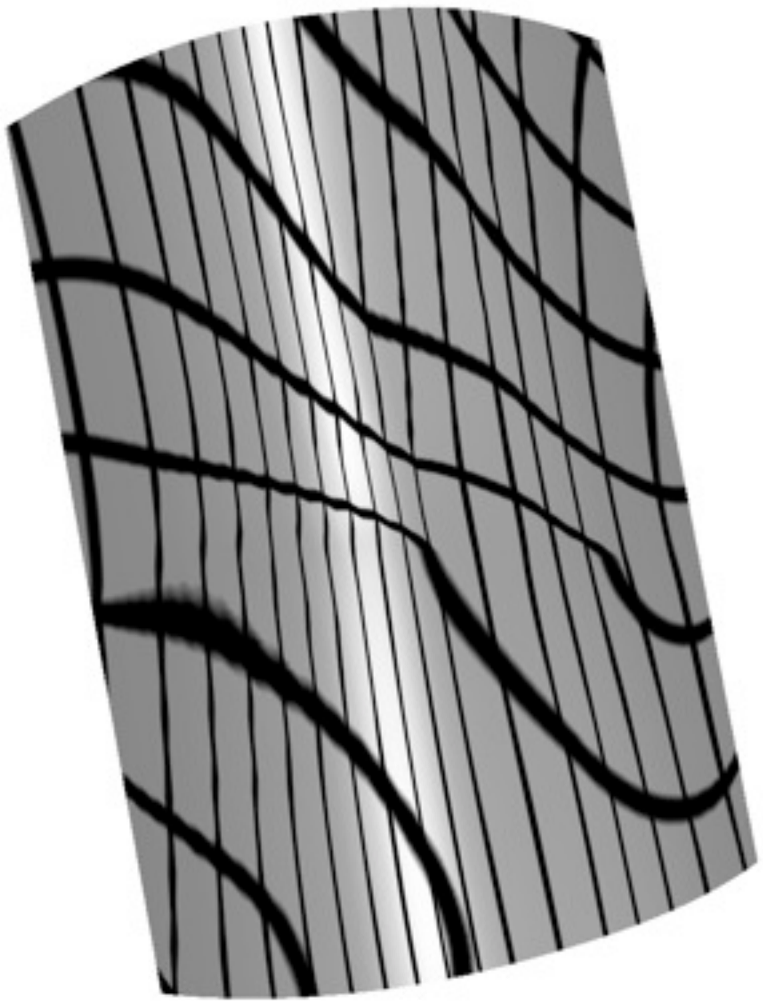
- Frequently used (wrong) version

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

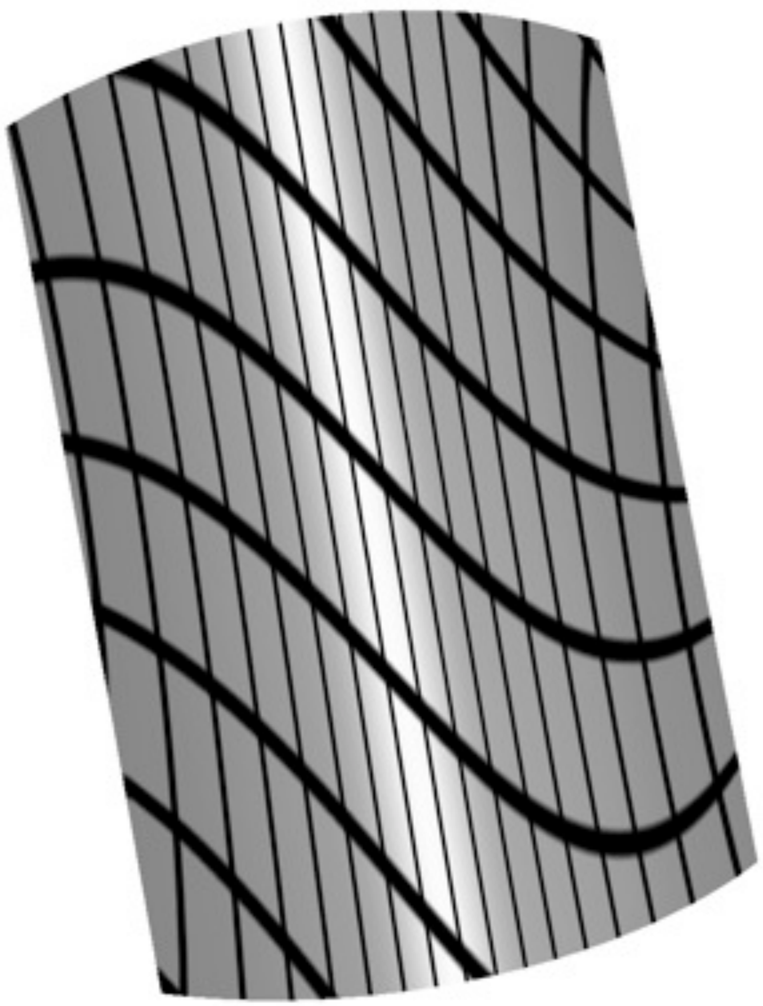
Discretization



Irregular mesh



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



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

Connection to Plate Energy?

- Neglect change of Laplacians for a moment...

$$\int \|\Delta \mathbf{p}' - \boldsymbol{\delta}\|^2 \to \min \quad \longrightarrow \quad \Delta^2 \mathbf{p}' = \Delta \boldsymbol{\delta}$$

$$\downarrow \quad \mathbf{p}' = \mathbf{p} + \mathbf{d}, \quad \boldsymbol{\delta} = \Delta \mathbf{p}$$

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

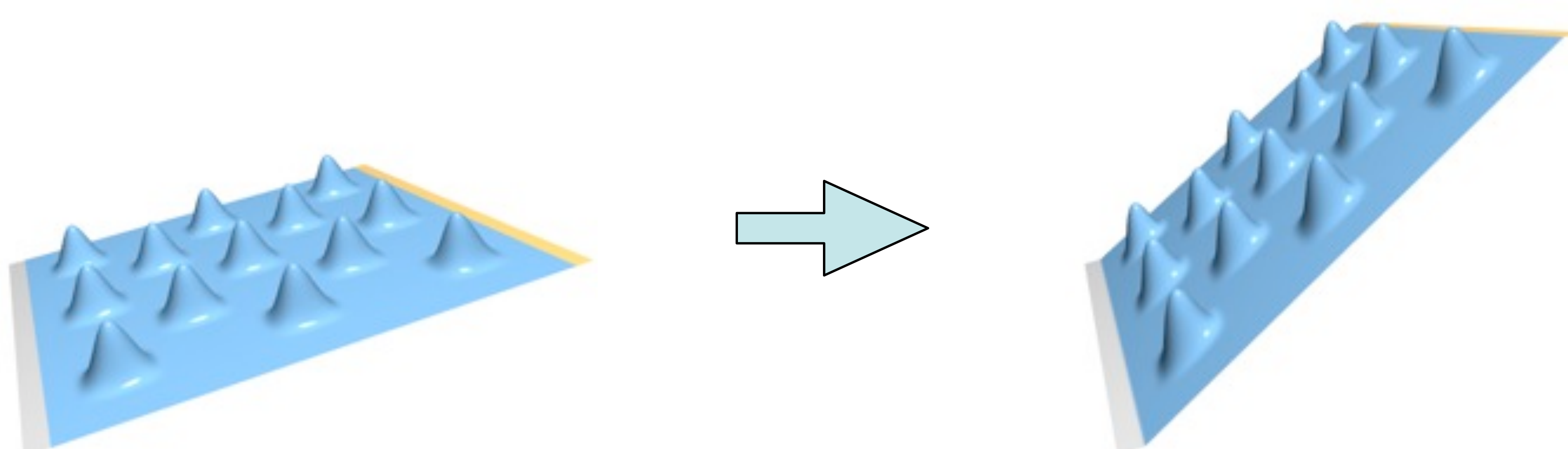
$$\downarrow$$

$$\int \|\mathbf{d}_{uu}\|^2 + 2\|\mathbf{d}_{uv}\|^2 + \|\mathbf{d}_{vv}\|^2 \to \min \quad \longleftarrow \quad \Delta^2 \mathbf{d} = 0$$

- Basic formulations equivalent!
- Differ in detail preservation
  - Rotation of Laplacians
  - Multi-scale decomposition

Limitations

- Differential coordinates work well for rotations
  - Apply damped handle rotations to diff. coords.
- Pure translations don't have rotation component
  - Translations don't change differential coordinates
  - *"Translation insensitivity"*



Overview

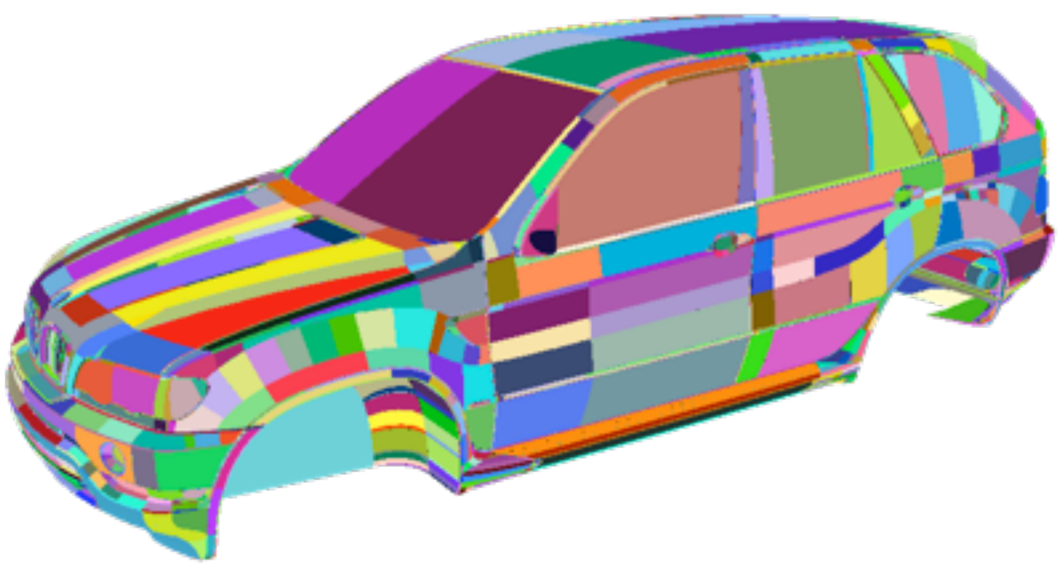
- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates

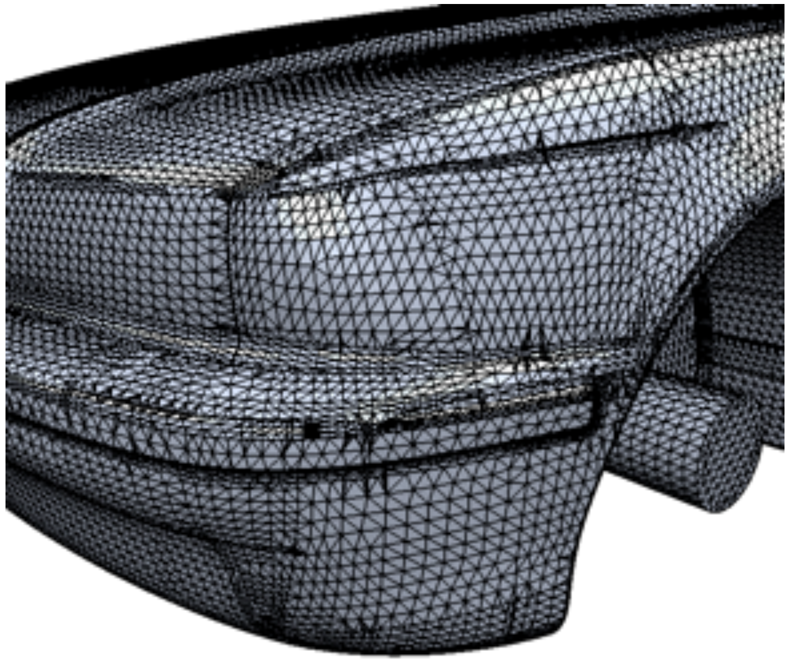
- **Space deformation**
  - **Freeform deformation**
  - Energy minimization

- Linear vs. nonlinear methods

Surface-Based Deformation

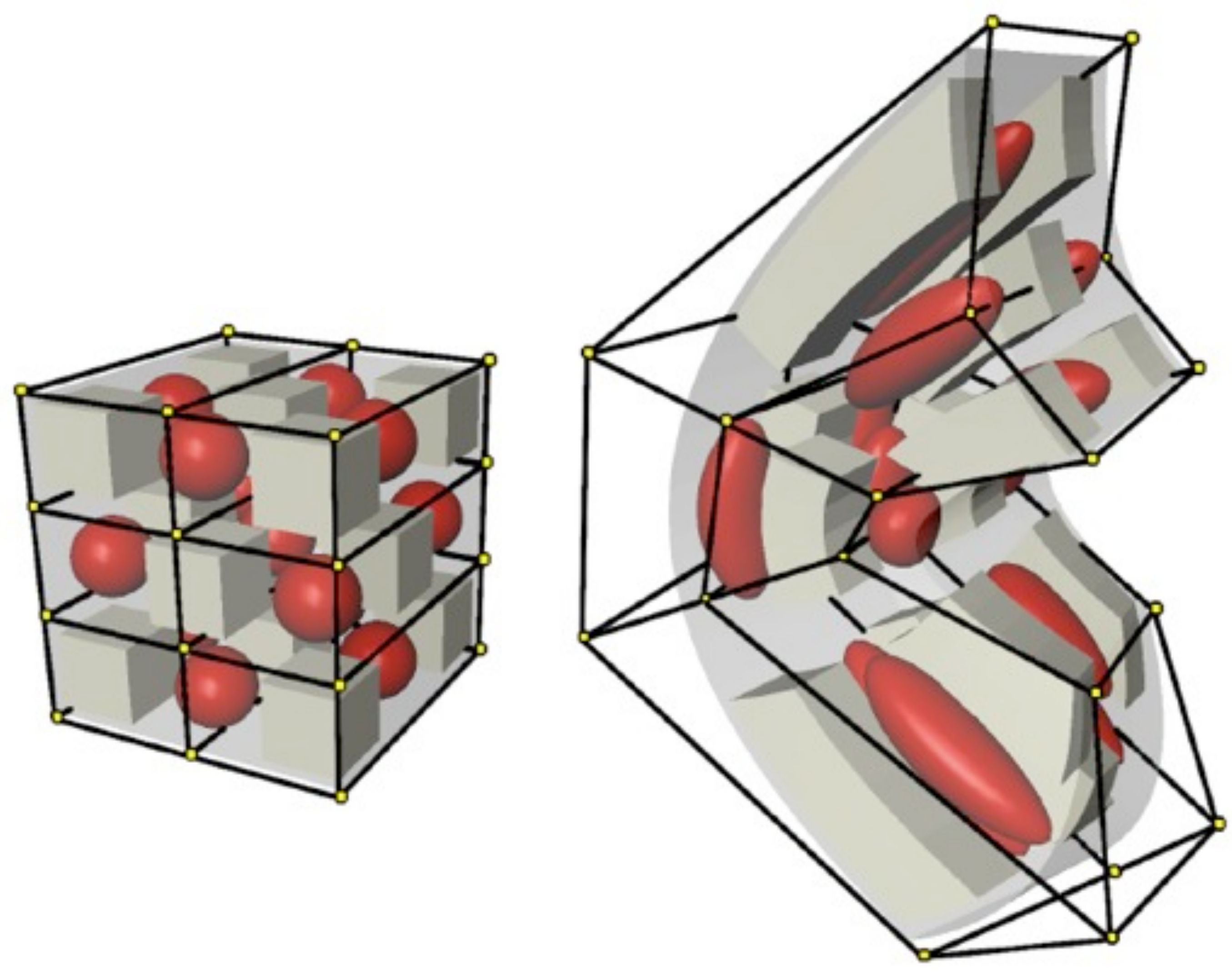
- Problems with
  - Highly complex models
  - Topological inconsistencies
  - Geometric degeneracies





Freeform Deformation

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



Freeform Deformation

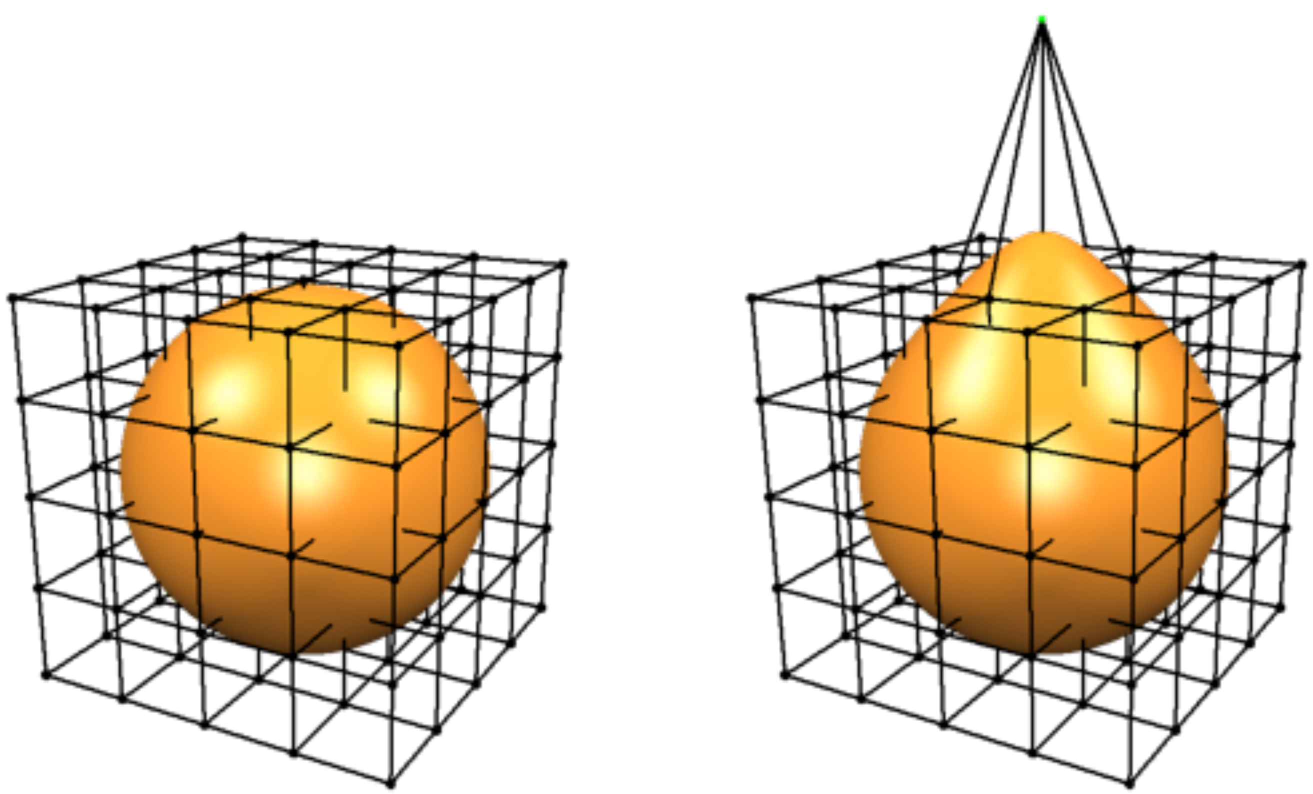
- Deform object's bounding box
  - Implicitly deforms embedded objects

- Tri-variate tensor-product spline

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

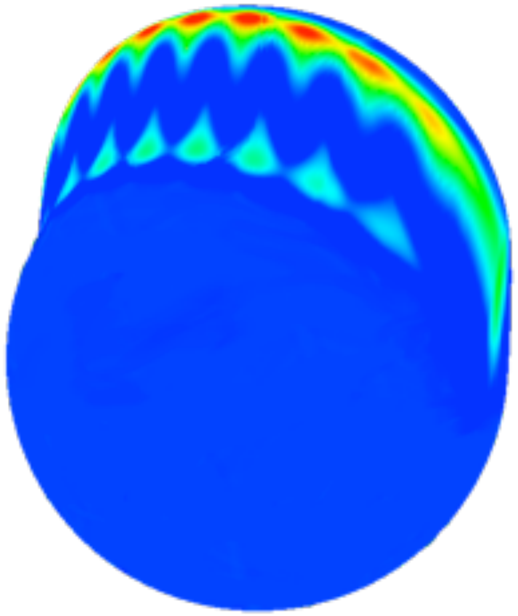
Freeform Deformation

- Deform object's bounding box
  - Implicitly deforms embedded objects
- Tri-variate tensor-product spline



Freeform Deformation

- Deform object's bounding box
  - Implicitly deforms embedded objects
- Tri-variate tensor-product spline
  - Aliasing artifacts
- Interpolate deformation constraints?
  - Only in least squares sense

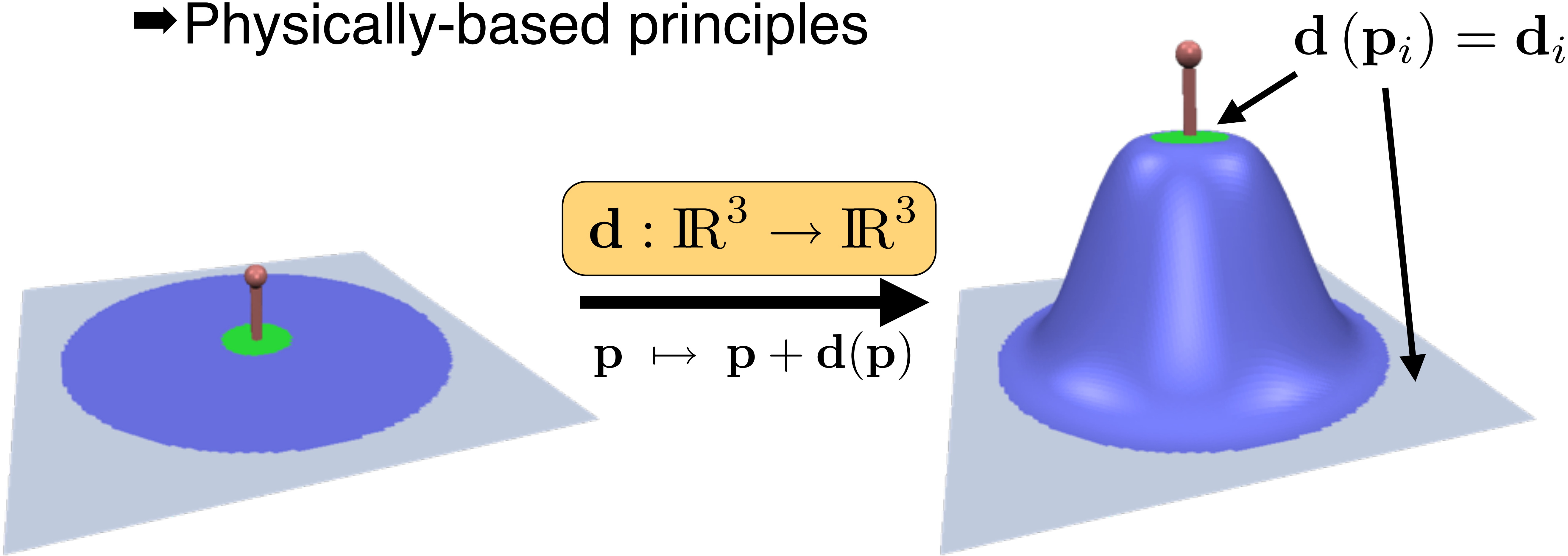


Overview

- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates
- **Space deformation**
  - Freeform deformation
  - **Energy minimization**
- Linear vs. nonlinear methods

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

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

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

Radial Basis Functions

- Represent deformation by RBFs

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

- Triharmonic basis function $\varphi(r) = r^3$
  - C^2 boundary constraints
  - Highly smooth / fair interpolation

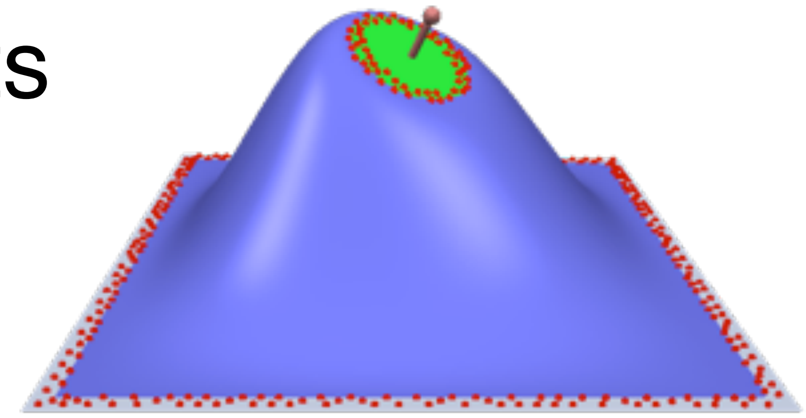
$$\int_{\mathbb{R}^3} \|\mathbf{d}_{uuu}\|^2 + \|\mathbf{d}_{vuu}\|^2 + \ldots + \|\mathbf{d}_{www}\|^2 \; \mathrm{d}u \, \mathrm{d}v \, \mathrm{d}w \;\rightarrow\; \min$$

RBF Fitting

- Represent deformation by RBFs

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

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

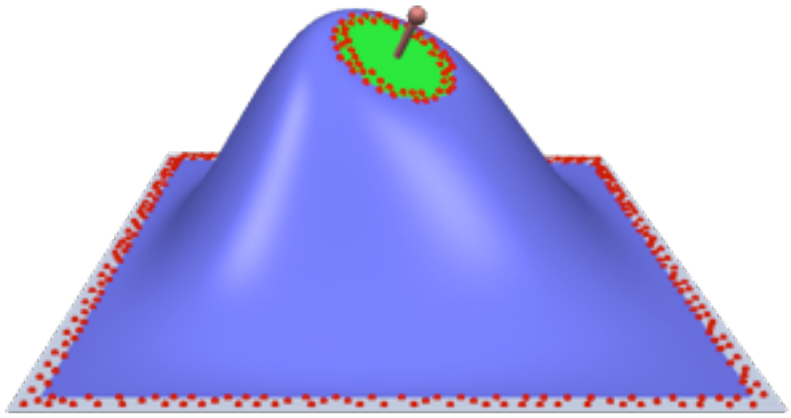


RBF Fitting

- Represent deformation by RBFs

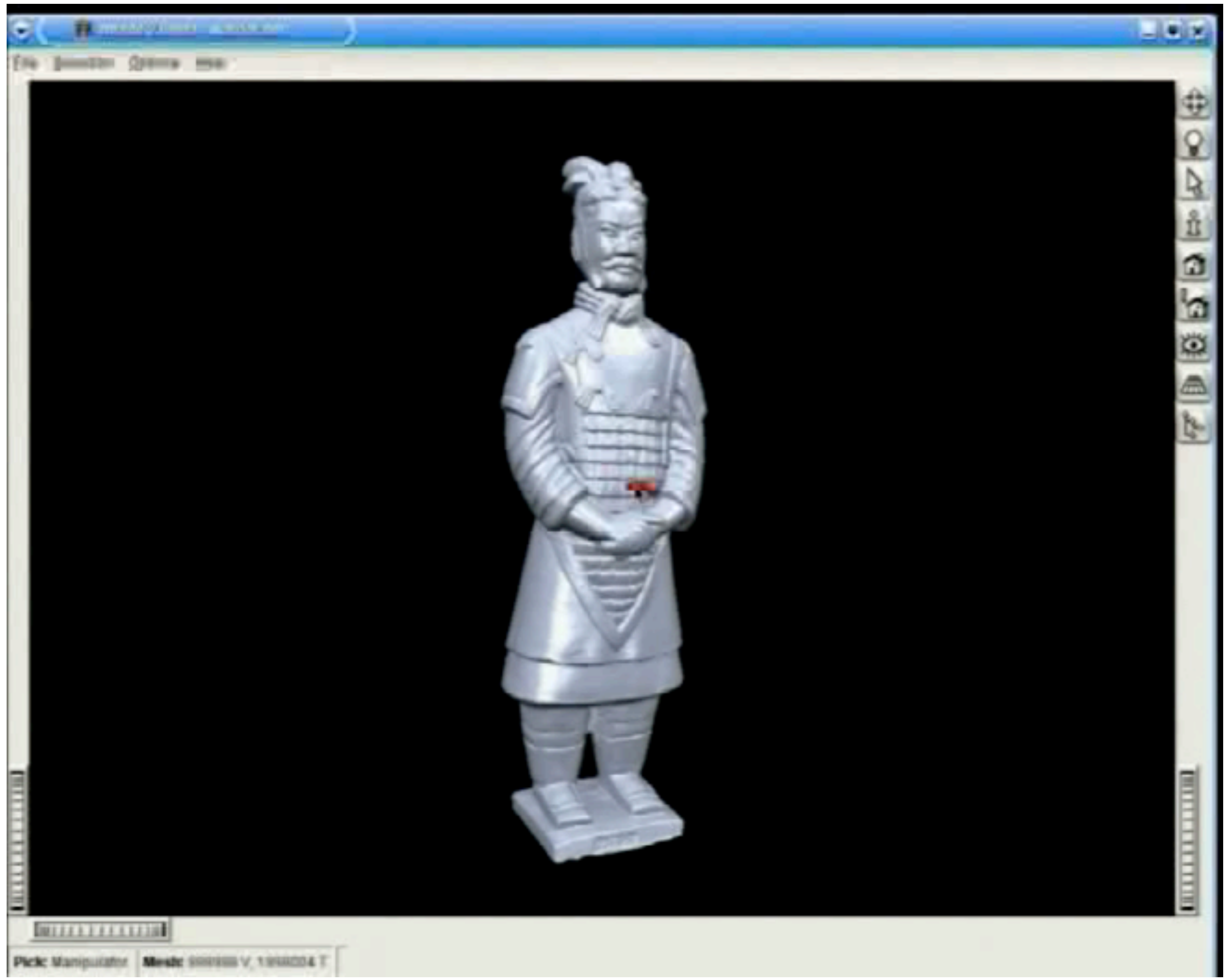
$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi\left(\|\mathbf{c}_j - \mathbf{x}\|\right) + \mathbf{p}(\mathbf{x})$$

- RBF evaluation
  - Function $\mathbf{d}$ transforms points
  - Jacobian $\nabla\mathbf{d}$ transforms normals
  - Precompute basis functions
  - Evaluate on the GPU!

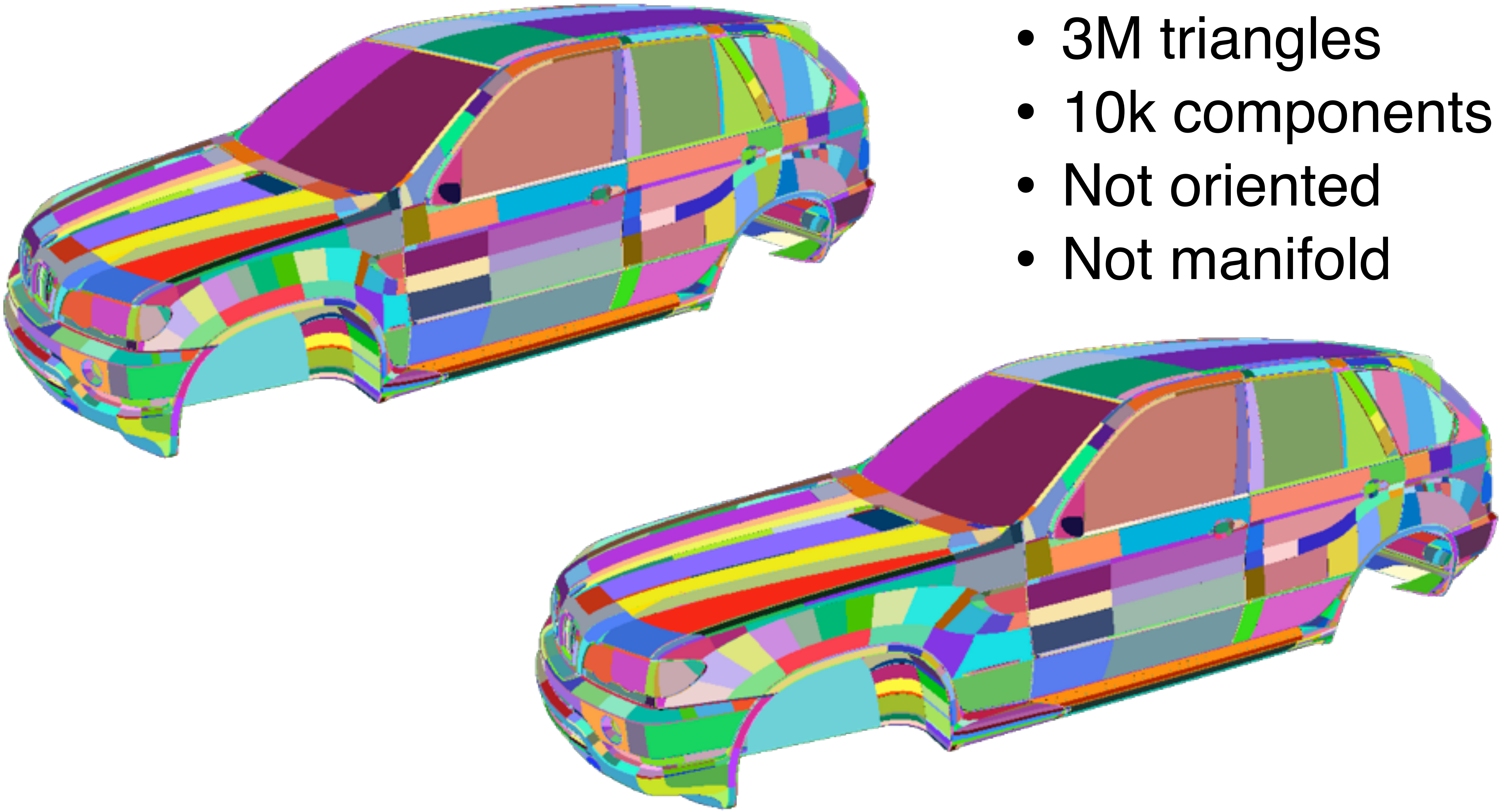


RBF Deformation

1M vertices



“Bad Meshes”



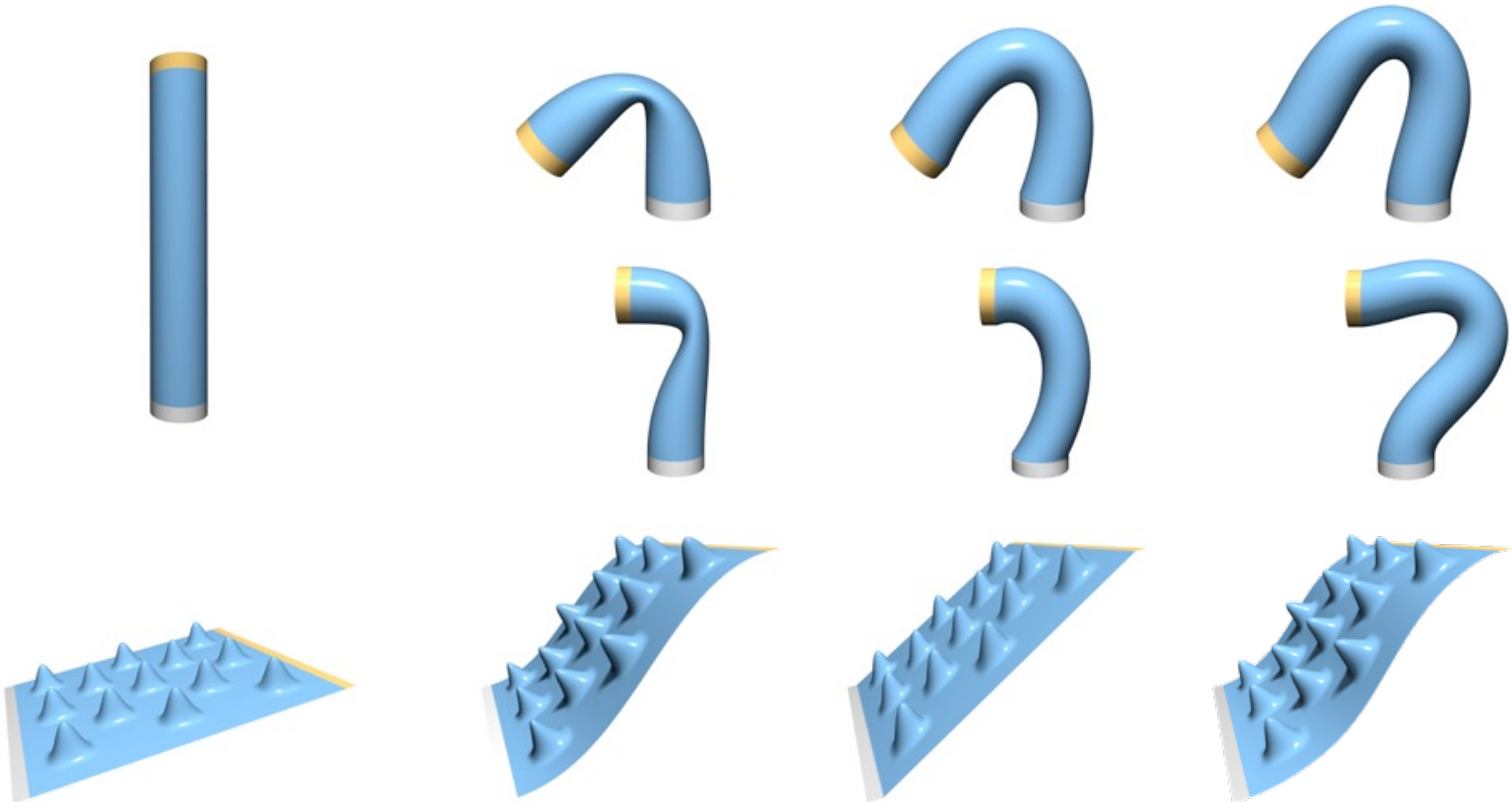
- 3M triangles
- 10k components
- Not oriented
- Not manifold

Overview

- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- **Linear vs. nonlinear methods**

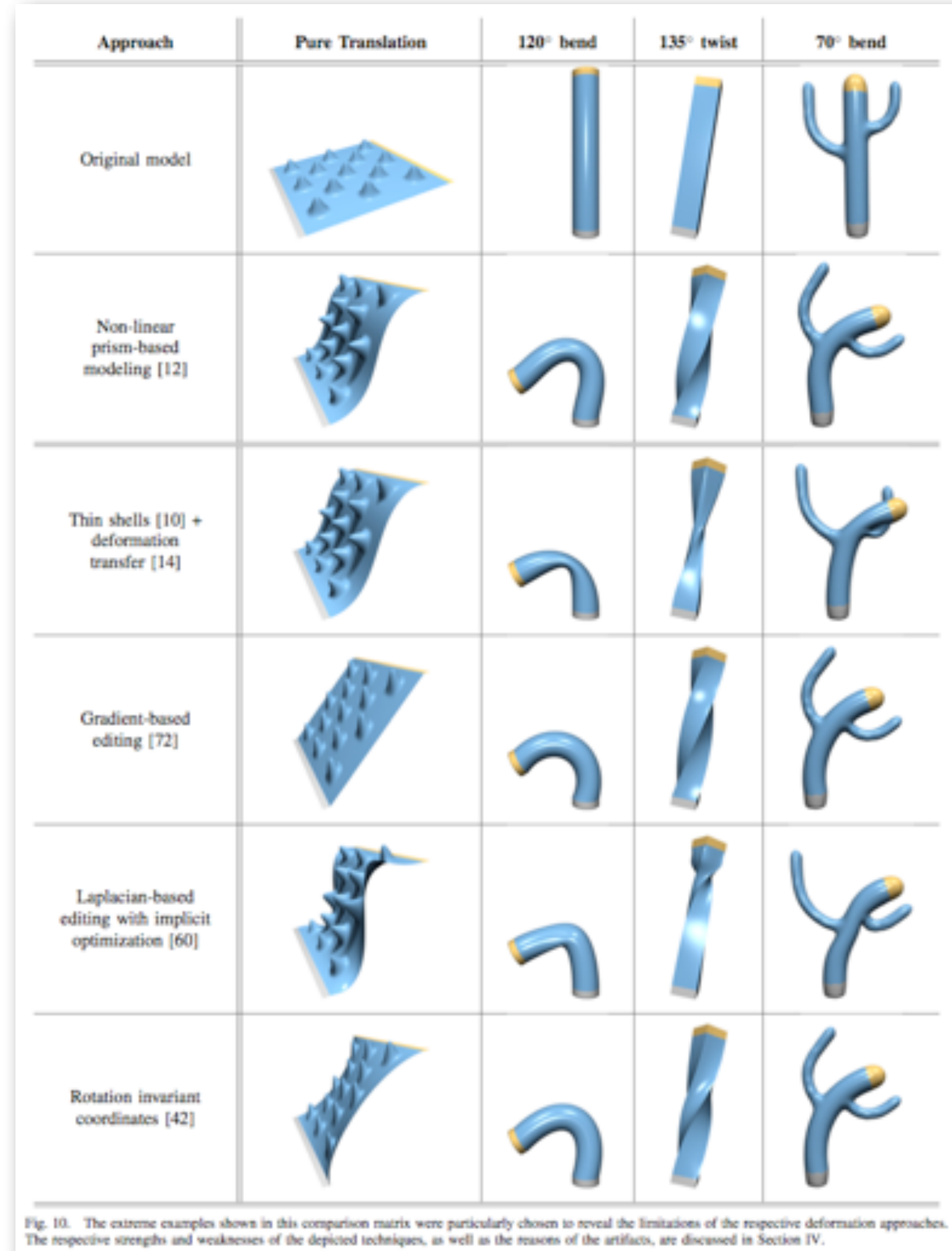
Comparison

Original	Bending Min.	Gradient	Nonlinear



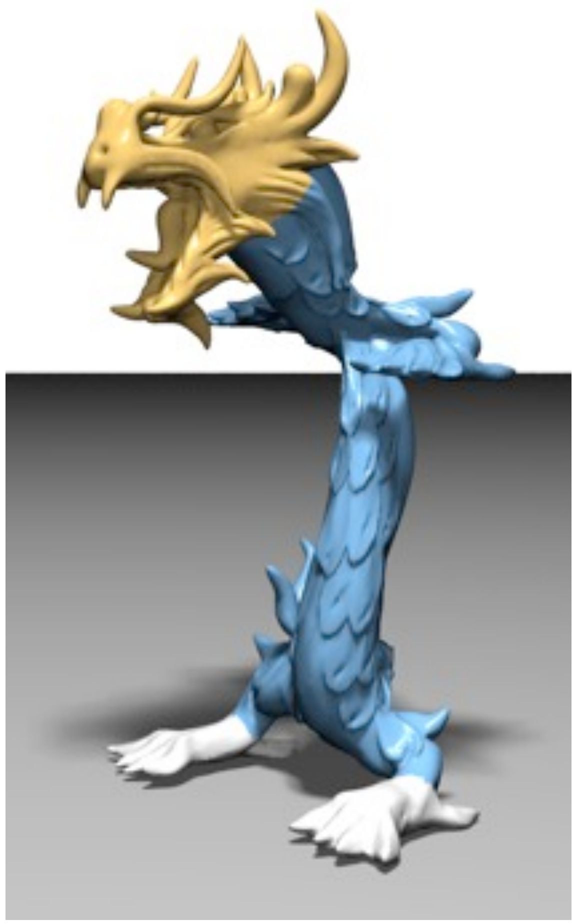
Linear vs. Nonlinear

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



[Botsch & Sorkine, TVCG 08]

Linear vs. Nonlinear



Bending Min.

Gradient

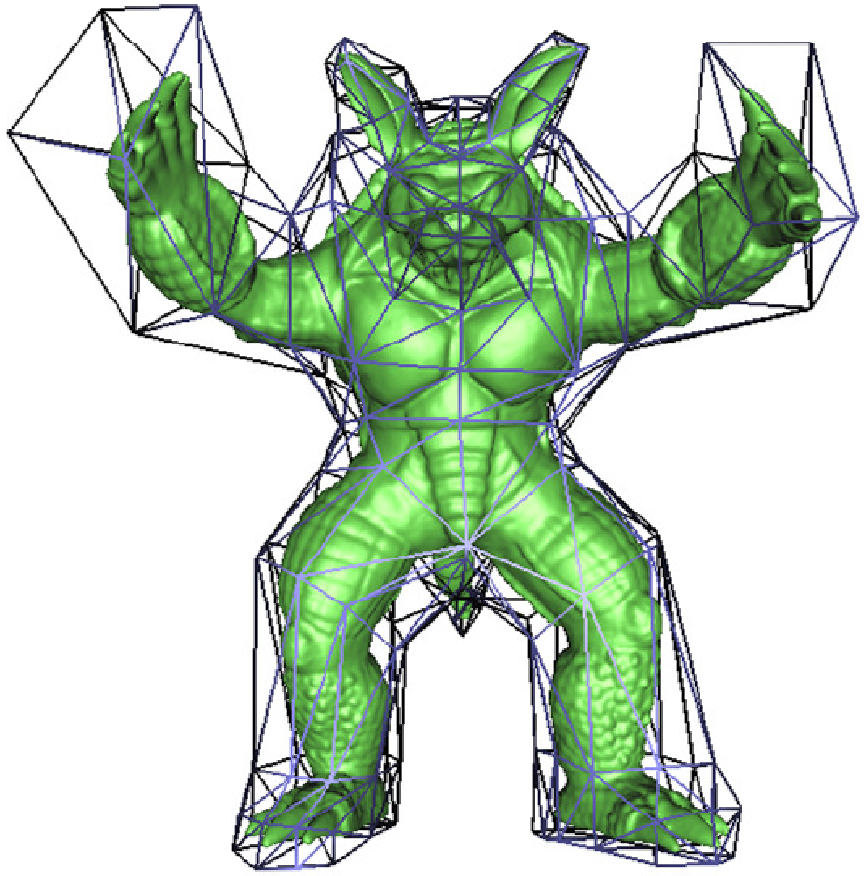
Nonlinear

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

Subspace Gradient Domain Mesh Deformation

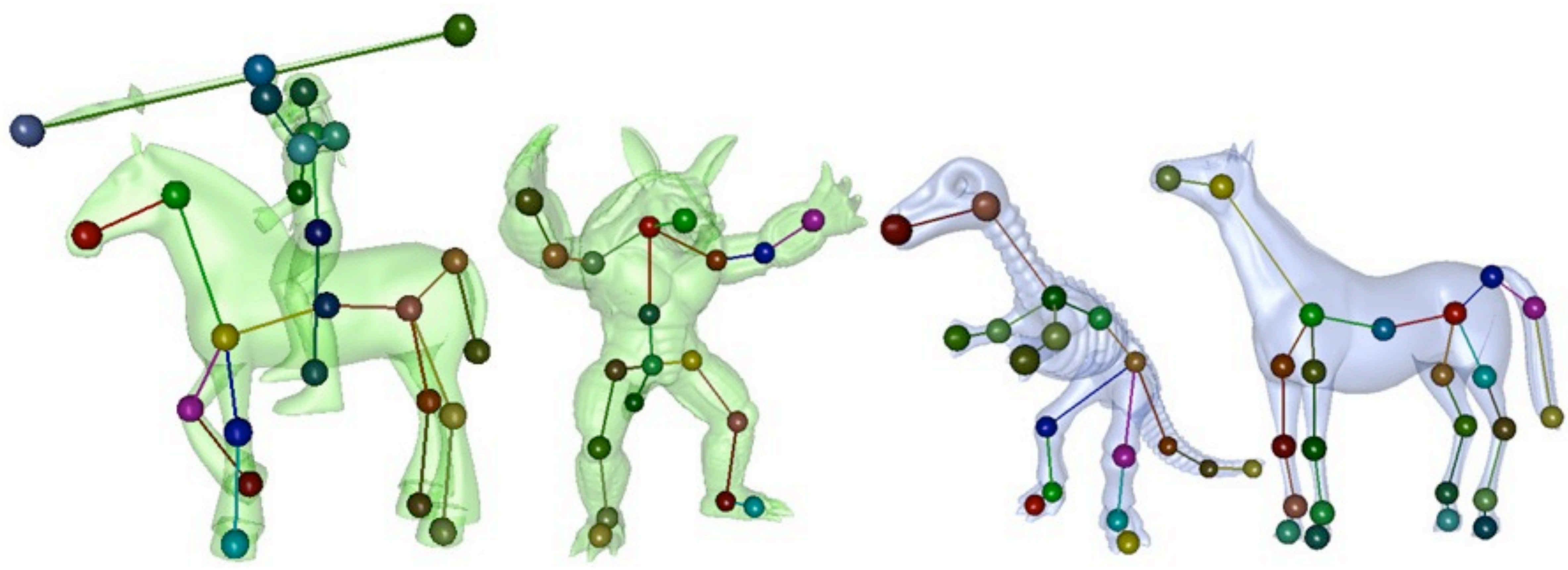
- Nonlinear Laplacian coordinates
- Least squares solution on coarse cage subspace



[Huang et al, SIGGRAPH 06]

Mesh Puppetry

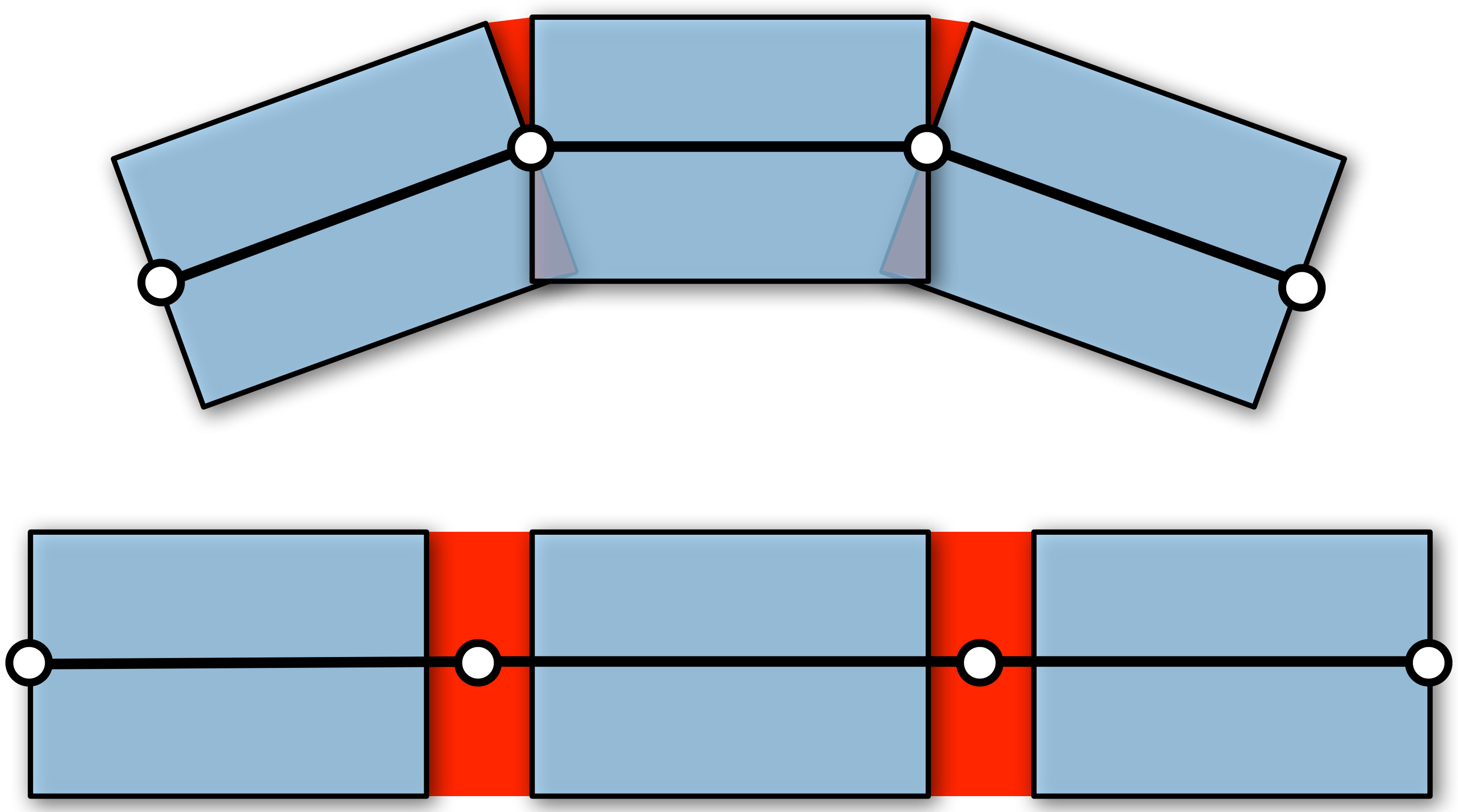
- Skeletons and Laplacian coordinates
- Cascading optimization



[Shi et al, SIGGRAPH 07]

PriMo: Coupled Prisms for Intuitive Surface Modeling

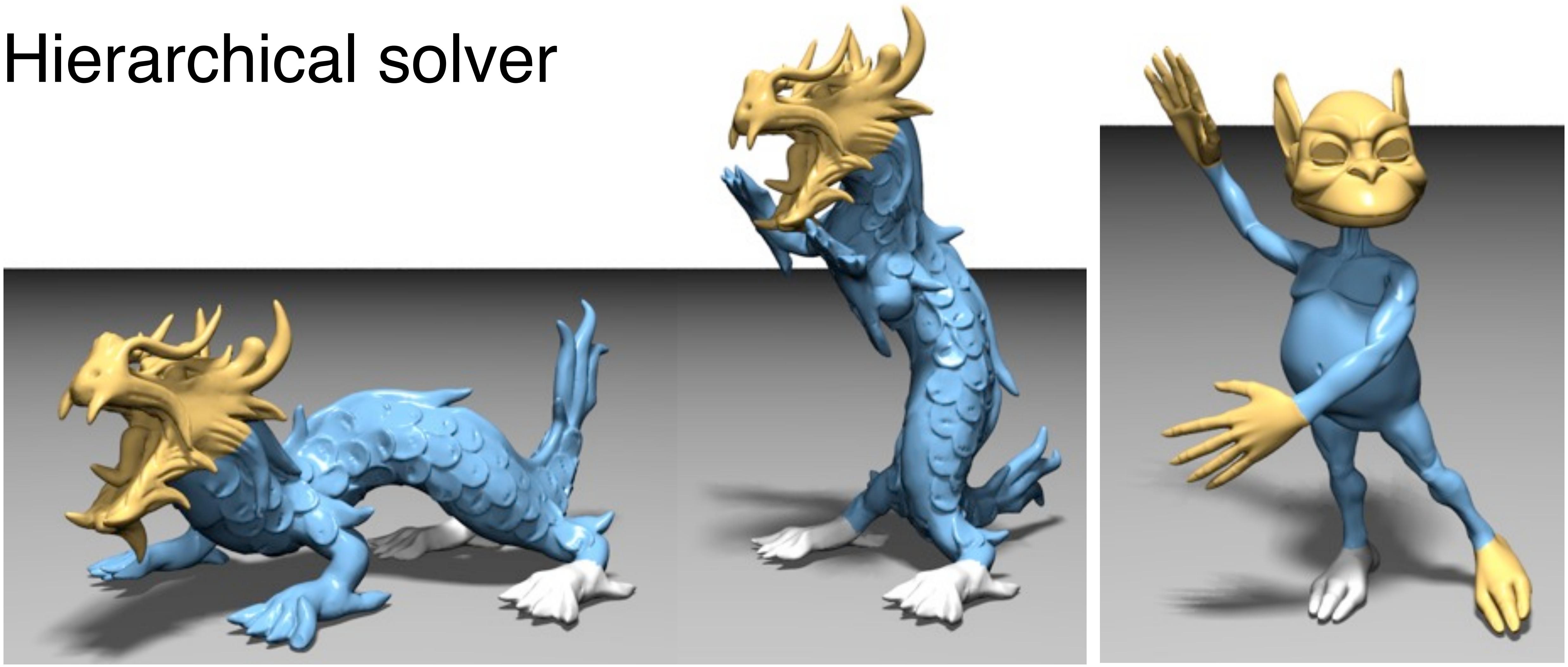
- Nonlinear shell-like energy
- Rigid cells ensure robustness
- Hierarchical solver



[Botsch et al, SGP 06]

PriMo: Coupled Prisms for Intuitive Surface Modeling

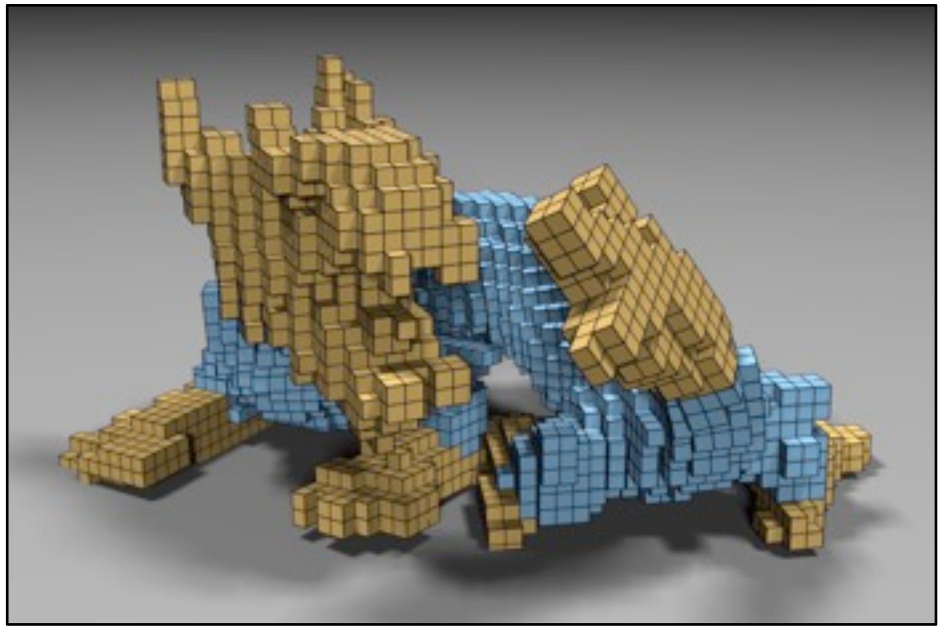
- Nonlinear shell-like energy
- Rigid cells ensure robustness
- Hierarchical solver

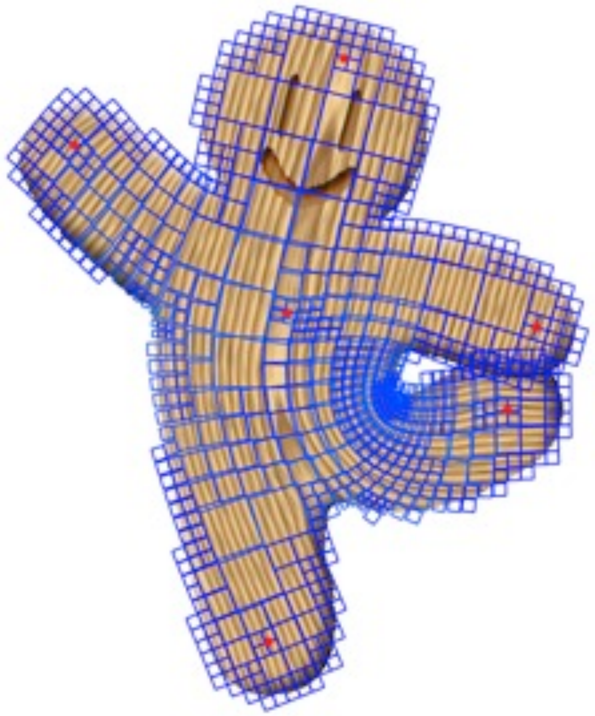


[Botsch et al, SGP 06]

Adaptive Space Deformation Based on Rigid Cells

- Nonlinear elastic energy for solids and shells
- Rigid cells ensure robustness
- Dynamic, adaptive discretization

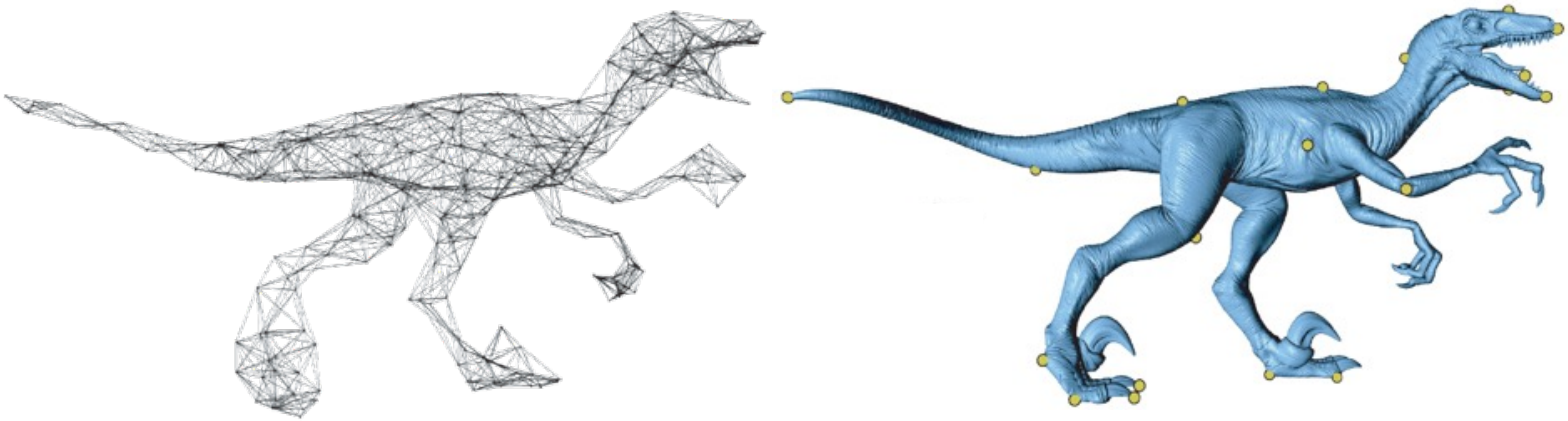




[Botsch et al, Eurographics 07]

Embedded Deformations

- As-rigid-as-possible space deformation
- Coarse deformation graph



[Sumner et al, SIGGRAPH 07]

Overview

- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods

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

Overview

- Surface-based deformation
  - Energy minimization
  - Multiresolution editing
  - Differential coordinates
- Space deformation
  - Freeform deformation
  - Energy minimization
- Linear vs. nonlinear methods