12.2 **Space Deformation**

Hao Li
http://cs621.hao-li.com
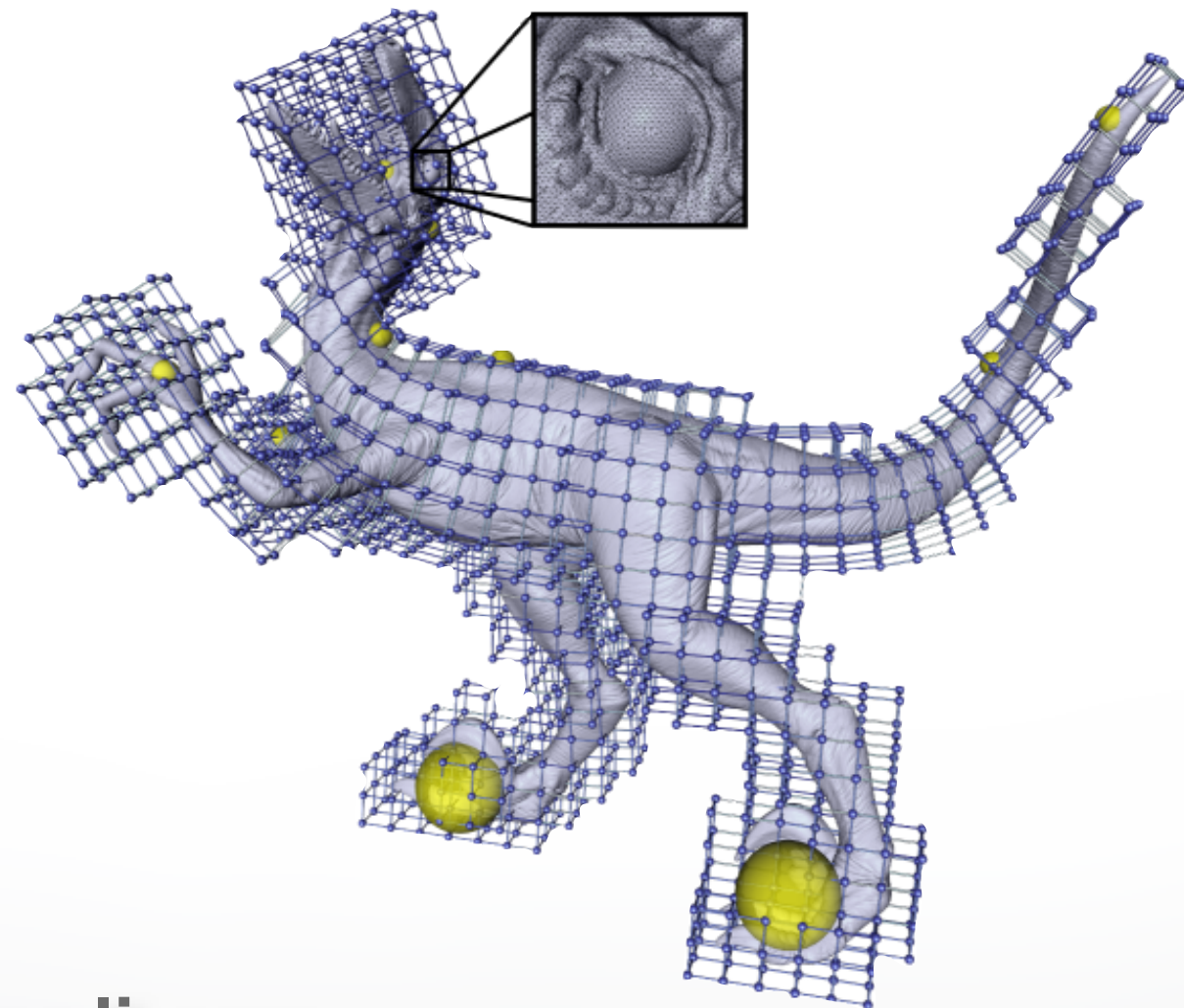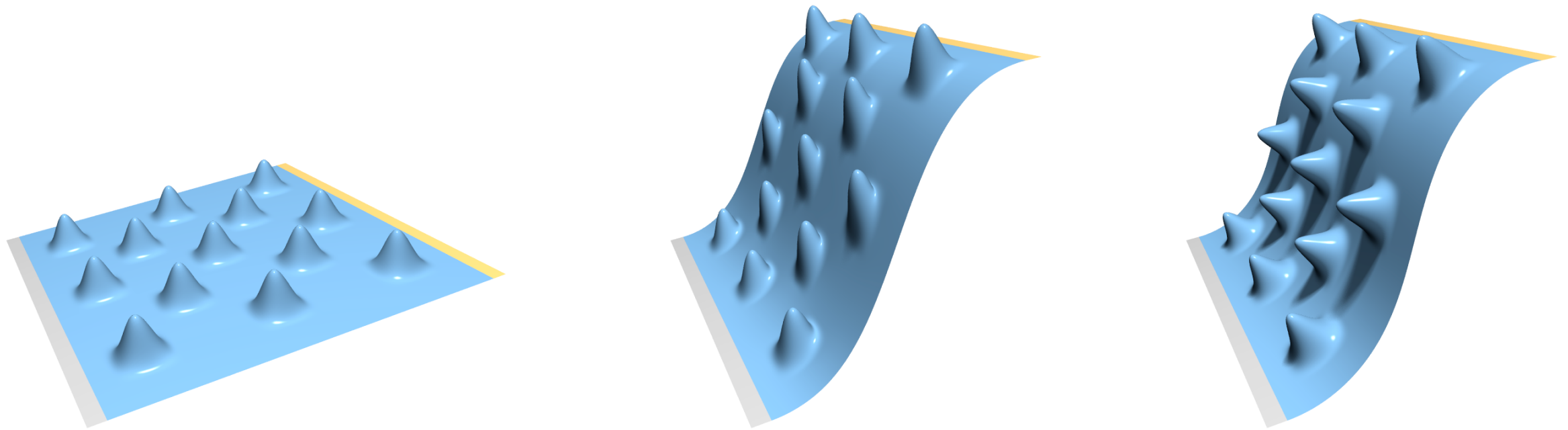
Hoooray!

Surface Deformations
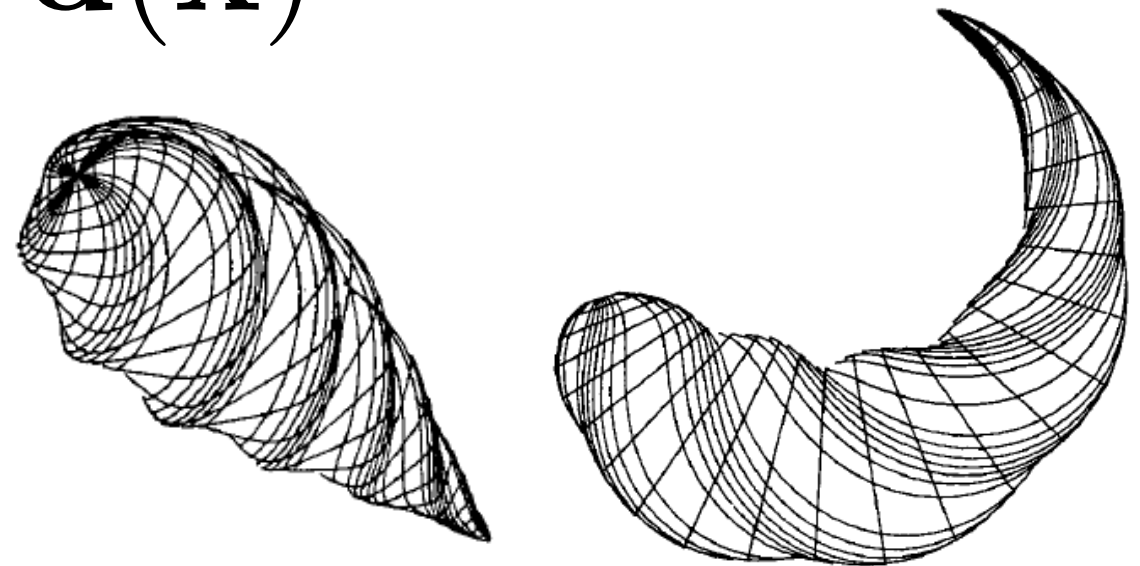
Space Deformation

- Displacement function defined on the ambient space

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

- Evaluate the function on the points of the shape embedded in the space

$$\mathbf{x}' = \mathbf{x} + \mathbf{d}(\mathbf{x})$$

Twist warp
Global and local deformation of solids
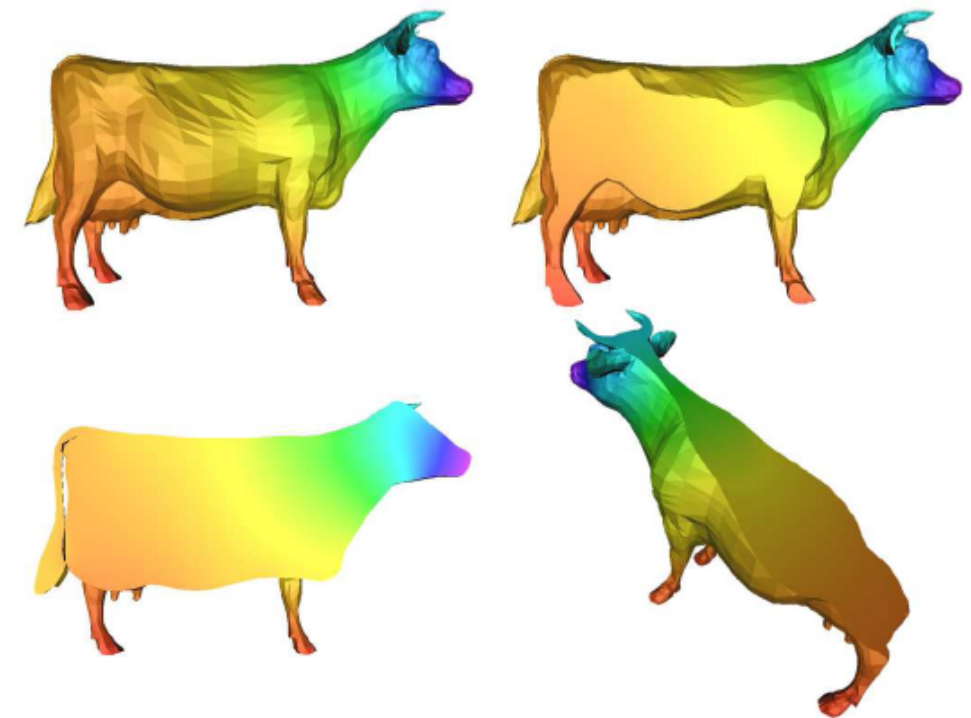[A. Barr, SIGGRAPH 84]

Freeform Deformation

- Control object

- User defines displacements $\mathbf{d}_i$ for each element of the control object

- Displacements are interpolated to the entire space using basis functions $B_i(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}$

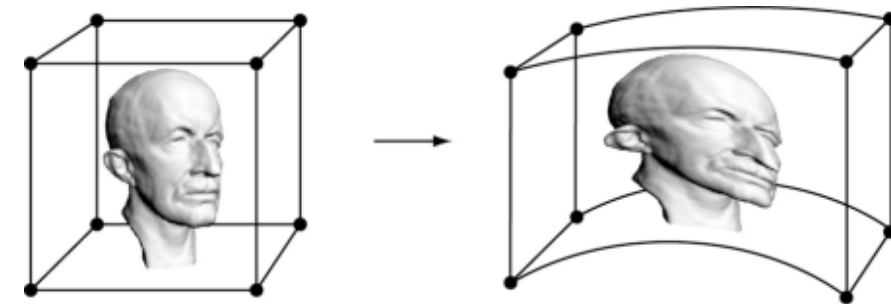$$\mathbf{d}(\mathbf{x}) = \sum_{i=1}^{k} \mathbf{d}_i \, B_i(\mathbf{x})$$
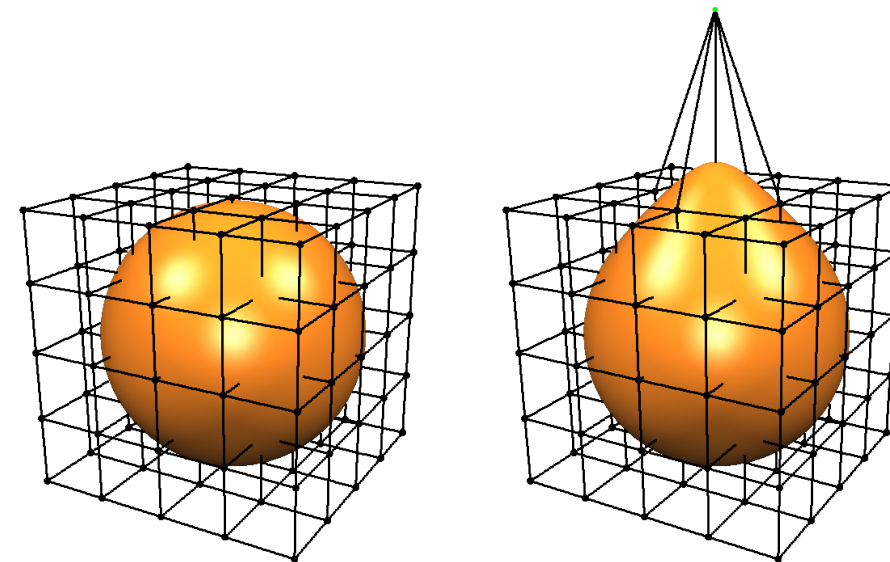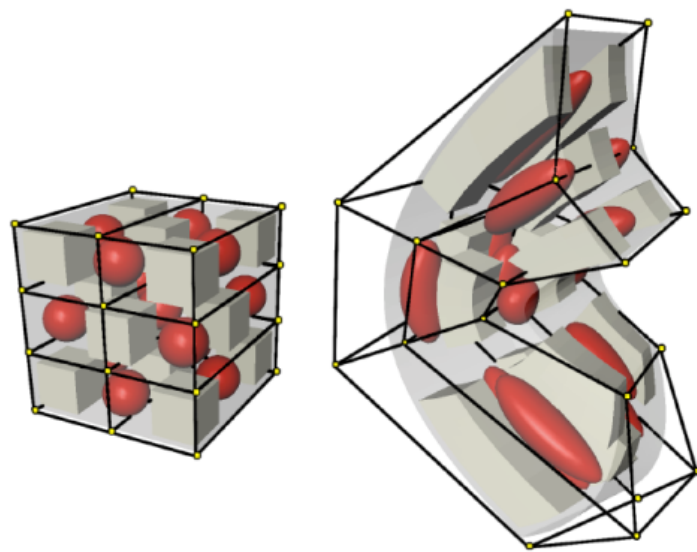
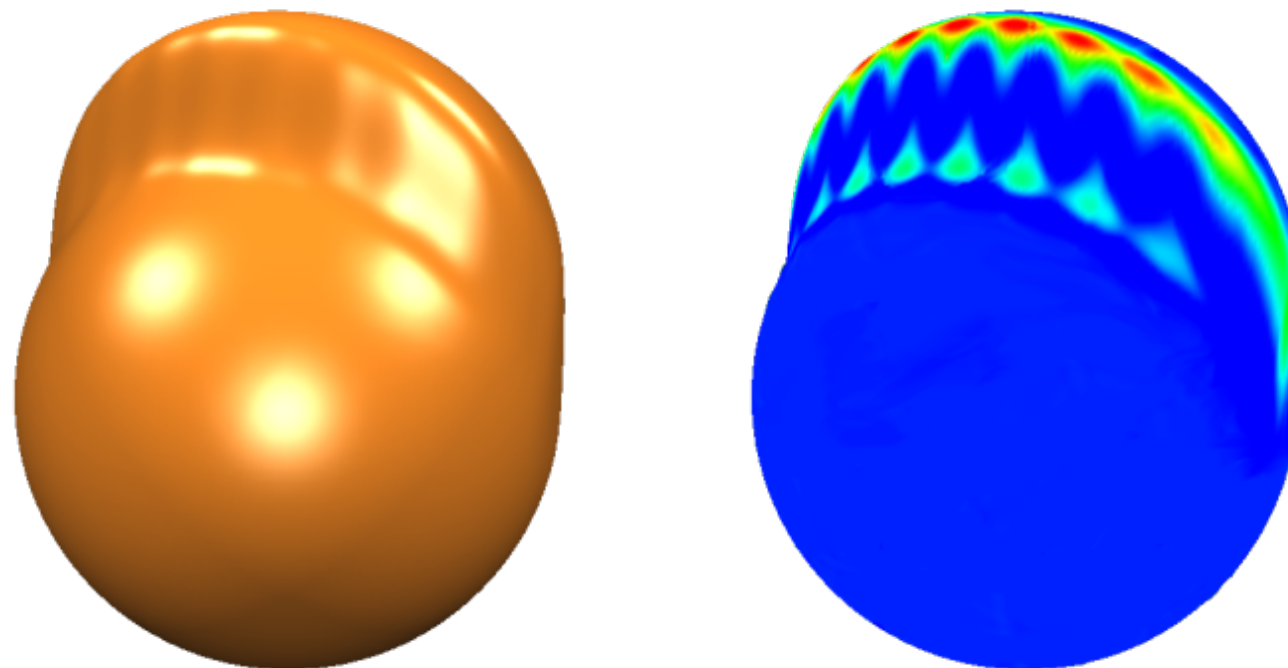- Basis functions should be smooth for aesthetic results

- Control object = lattice
- Basis functions $B_i(\mathbf{x})$ are trivariate tensor-product splines:

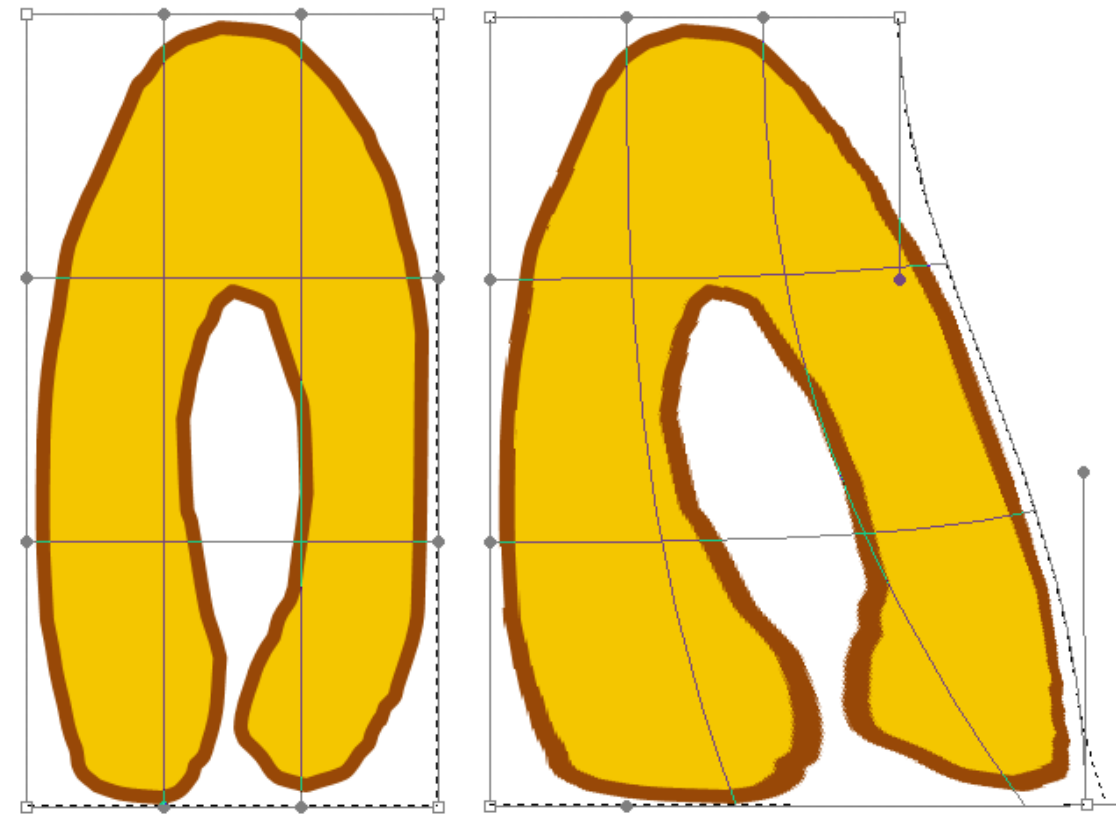$$\mathbf{d}(x, y, z) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{d}_{ijk} \, N_i(x) N_j(y) N_k(z)$$

- Aliasing artifacts

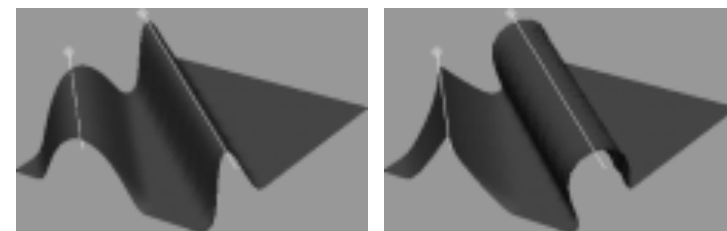- Interpolate deformation constraints?
  - Only in least squares sense

- Difficult to manipulate
- The control object is not related to the shape of the edited object
- Parts of the shape in close Euclidean distance always deform similarly, even if geodesically far

Wires

- Control objects are arbitrary space curves
- Can place curves along meaningful features of the edited object
- Smooth deformations around the curve with decreasing influence
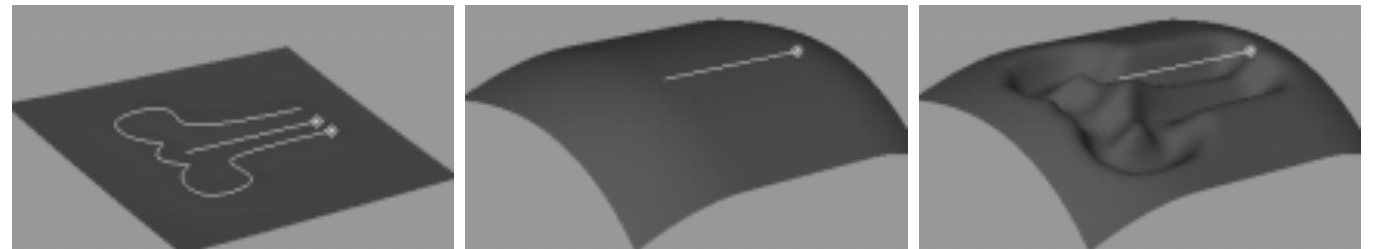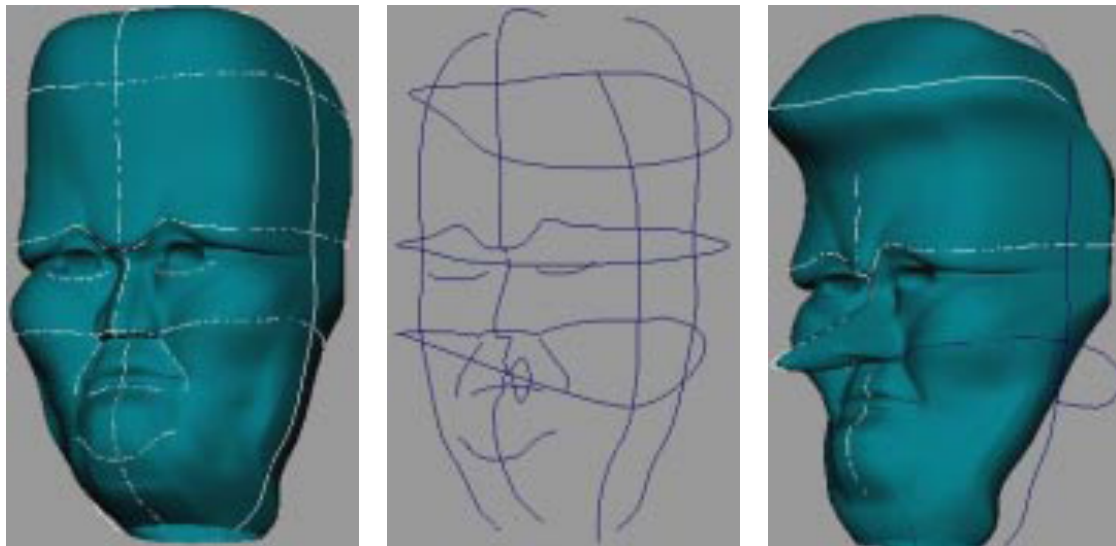
- Wish list for the displacement function $\mathbf{d}(\mathbf{x})$:
    - Interpolate prescribed constraints
    - Smooth, intuitive deformation

$$\mathbf{d}(\mathbf{x}_i) = \mathbf{d}_i$$

$$\mathbf{d} : \mathbb{R}^3 \to \mathbb{R}^3$$

$$\mathbf{x} \to \mathbf{x} + \mathbf{d}(\mathbf{x})$$
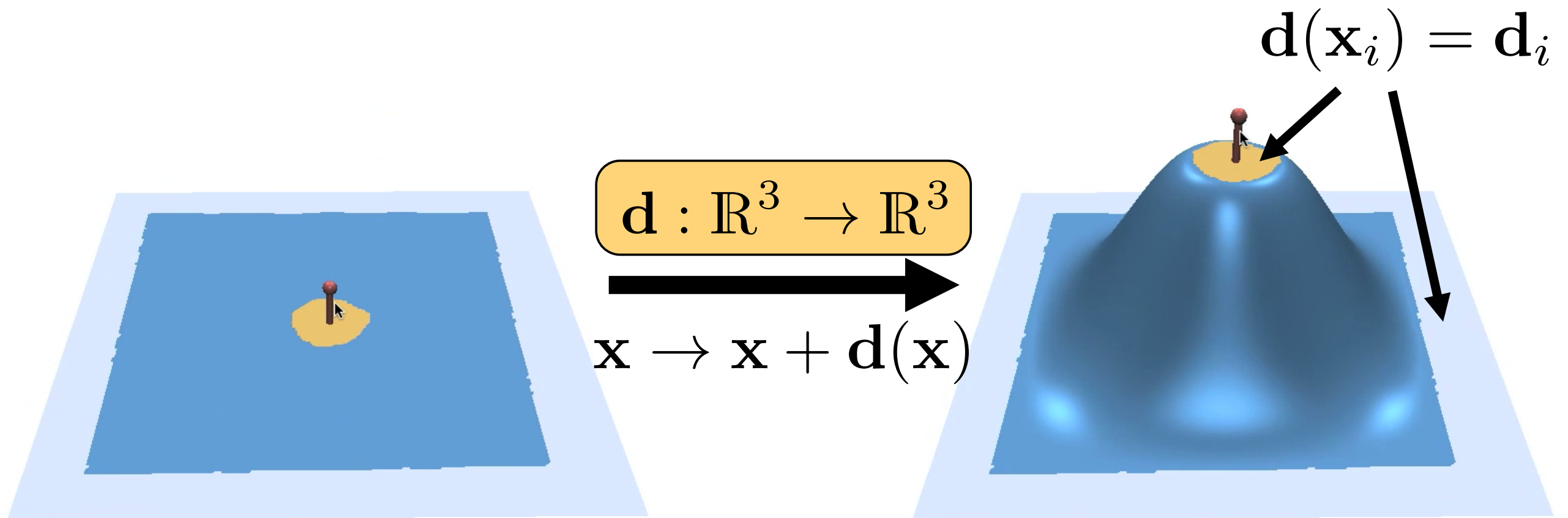
Volumetric Energy Minimization

- Minimize similar energies to surface case

$$\int_{\mathbb{R}^3} \|\mathbf{d}_{xx}\|^2 + \|\mathbf{d}_{xy}\|^2 + \ldots + \|\mathbf{d}_{zz}\|^2 \, dx \, dy \, dz \to \min$$

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

Radial Basis Functions

- Represent deformation by RBFs

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

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

$$\int_{\mathbb{R}^3} \|\mathbf{d}_{xxx}\|^2 + \|\mathbf{d}_{xyy}\|^2 + \ldots + \|\mathbf{d}_{zzz}\|^2 \, dx \, dy \, dz \to \min$$
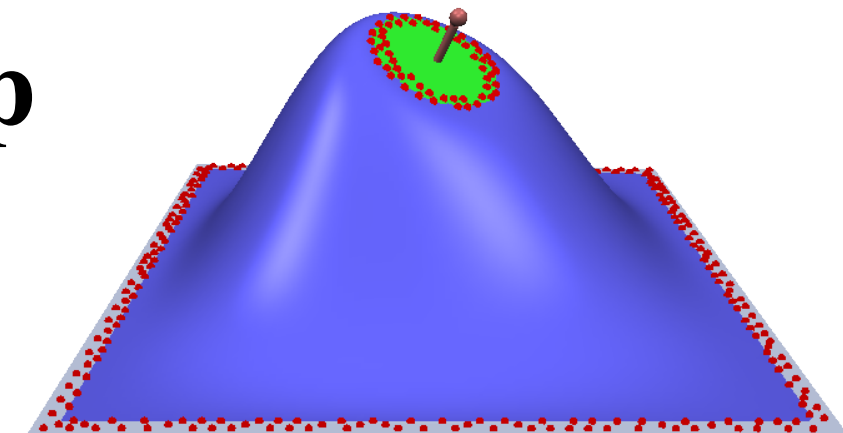
Radial Basis Functions

- Represent deformation by RBFs

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

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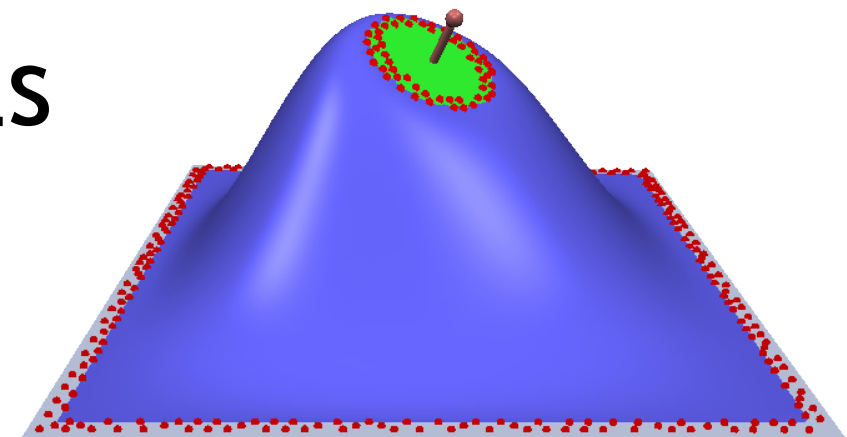
Radial Basis Functions

- ## Represent deformation by RBFs

$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \, \varphi(\|\mathbf{c}_j - \mathbf{x}\|) + \mathbf{p}(\mathbf{x})$$
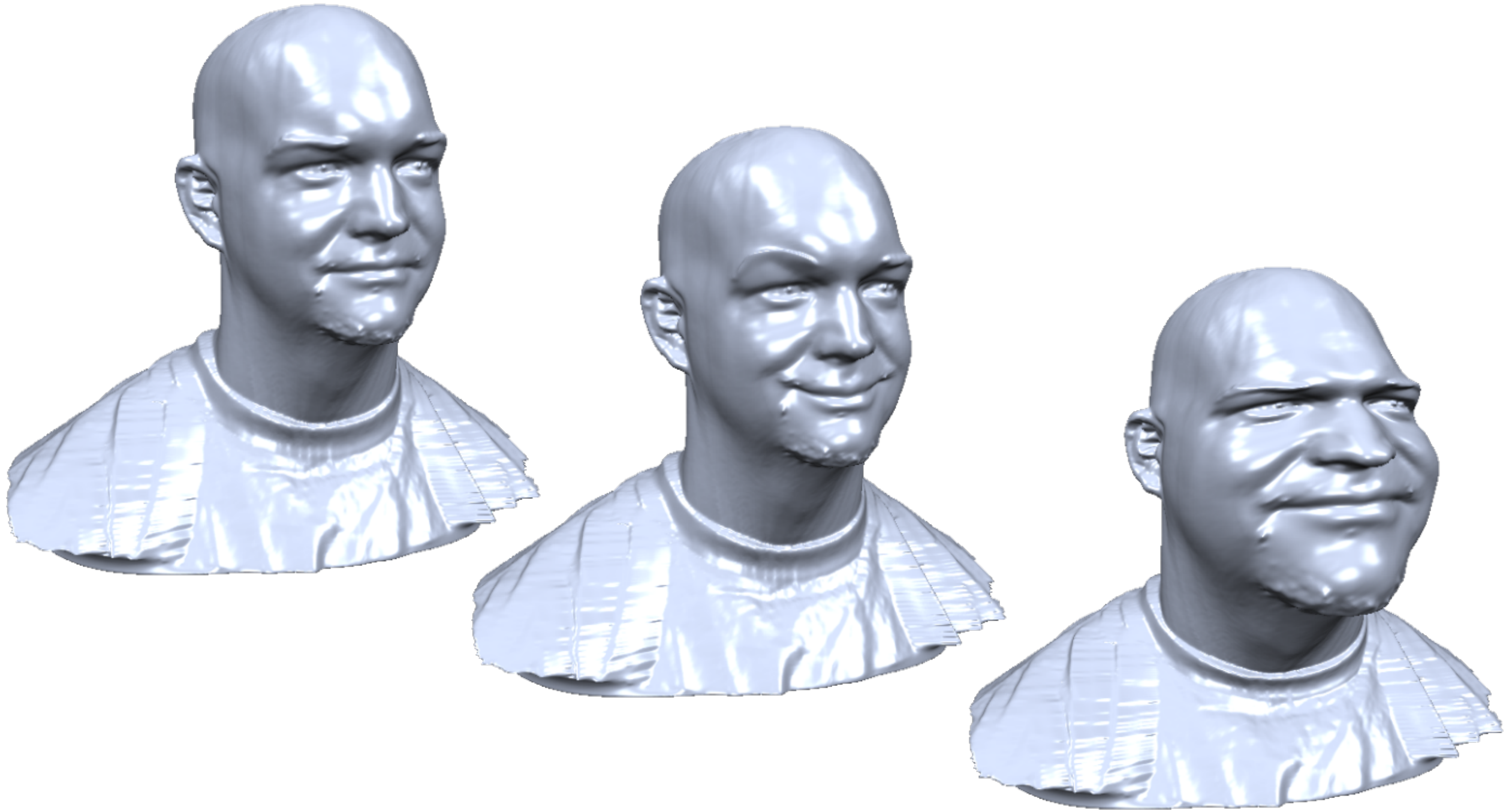
- ## RBF evaluation
  - Function $\mathbf{d}$ transforms points
  - Jacobian $\nabla\mathbf{d}$ transforms normals
  - Precompute basis functions
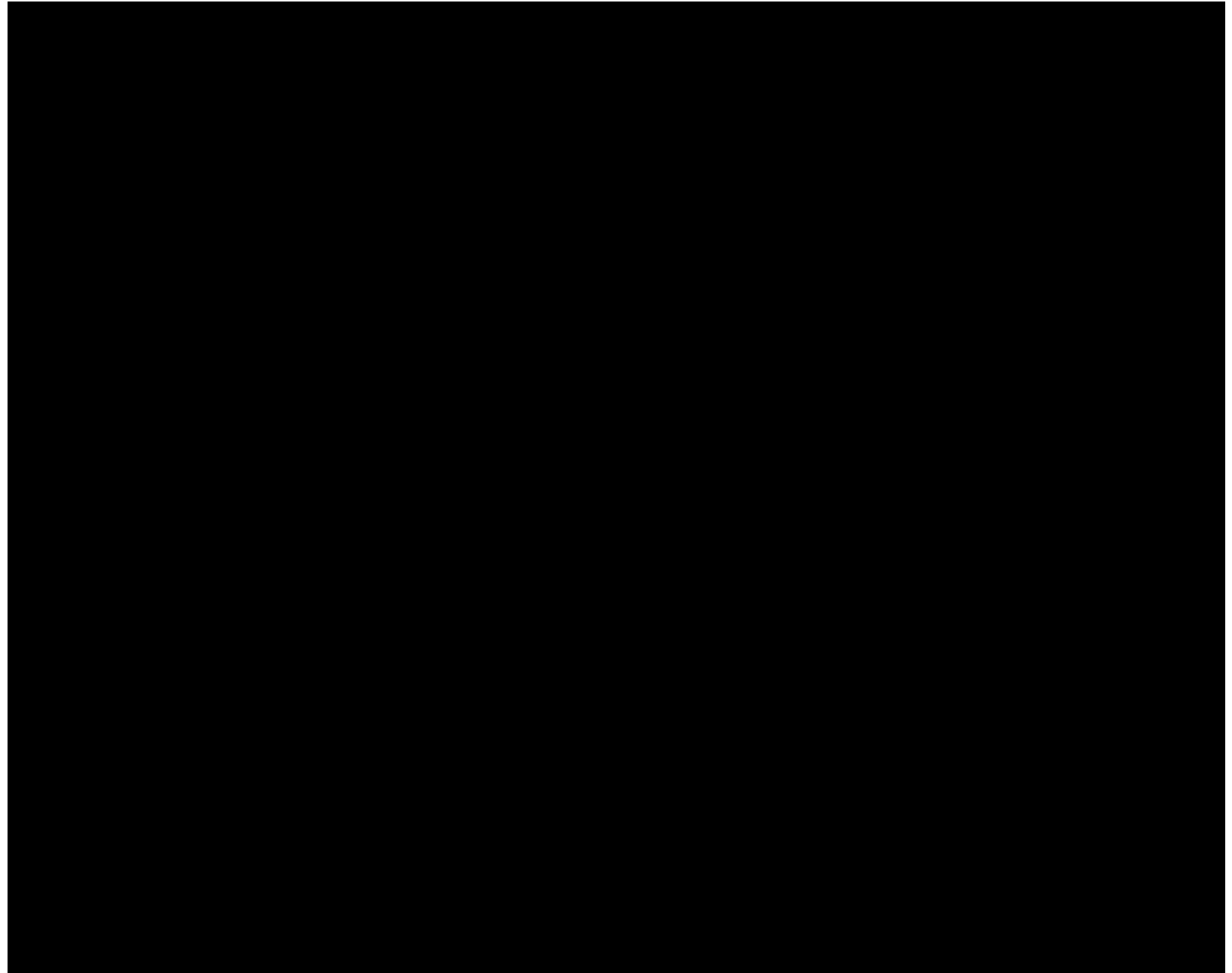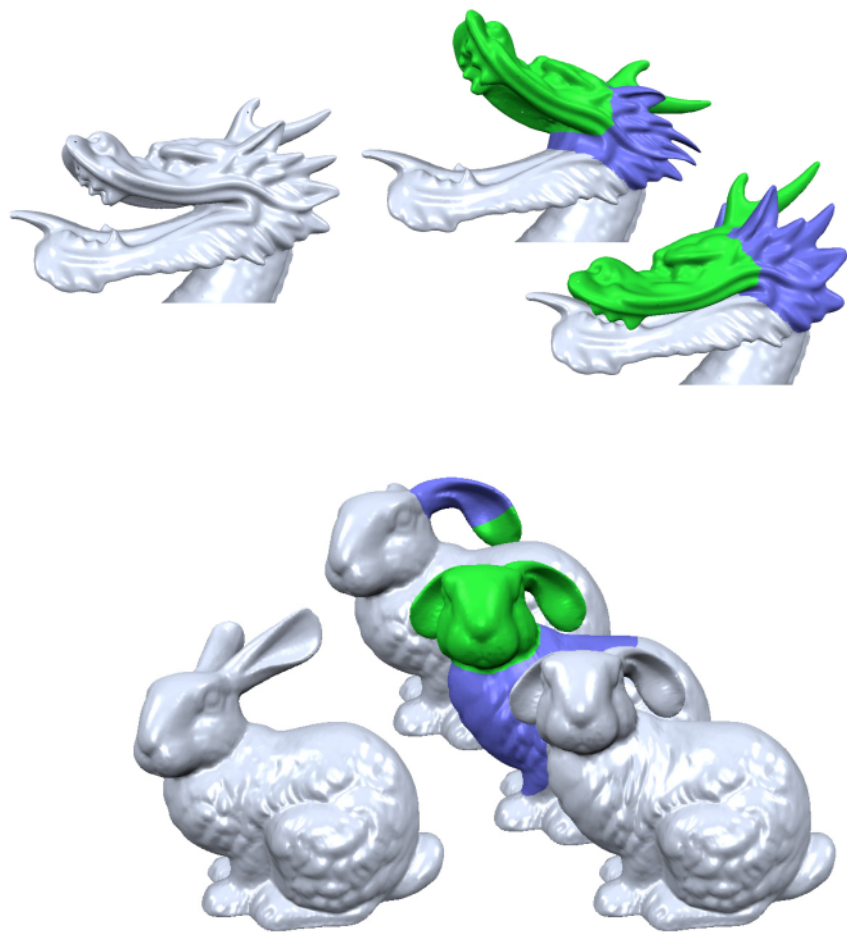 - Evaluate on the GPU!

Local & Global Deformations

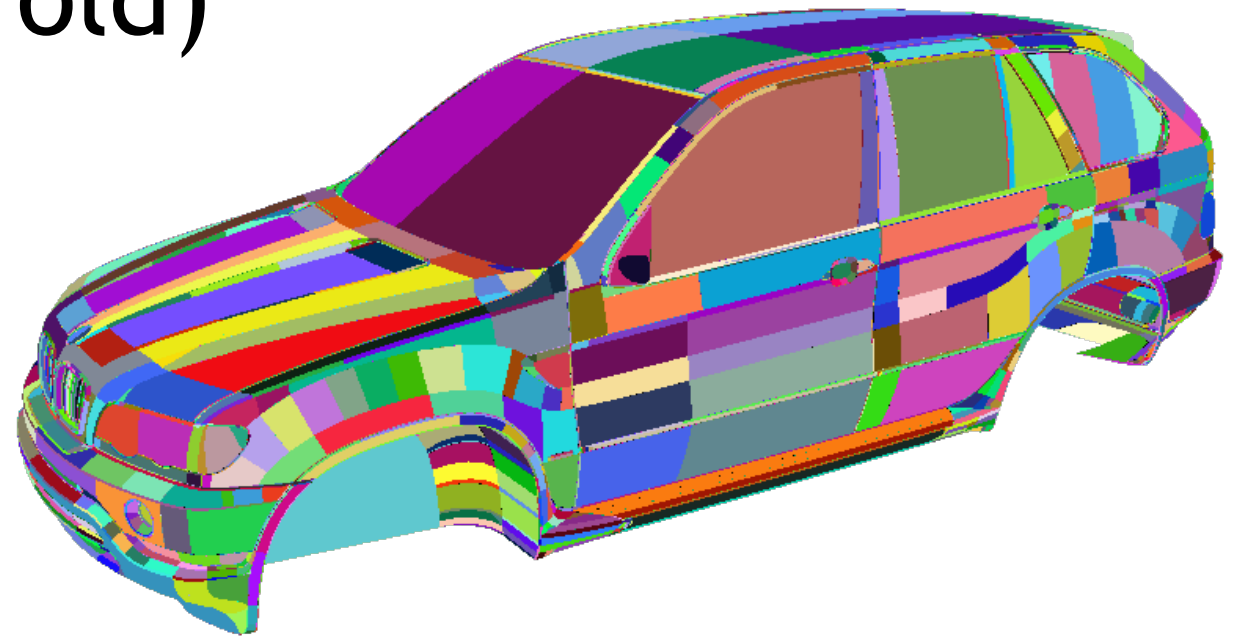[RBF, Botsch & Kobbelt 05]

Local & Global Deformations

[RBF, Botsch & Kobbelt 05]

1M vertices
movie

- Handle arbitrary input
  - Meshes (also non-manifold)
  - Point sets
  - Polygonal soups
  - ...

- Complexity mainly depends on the control object, not the surface

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

Space Deformation

- Handle arbitrary input
    - Meshes (also non-manifold)
    - Point sets
    - Polygonal soups
    - ...

- Easier to analyze:
functions on Euclidean domain
    - Volume preservation: |Jacobian| = 1

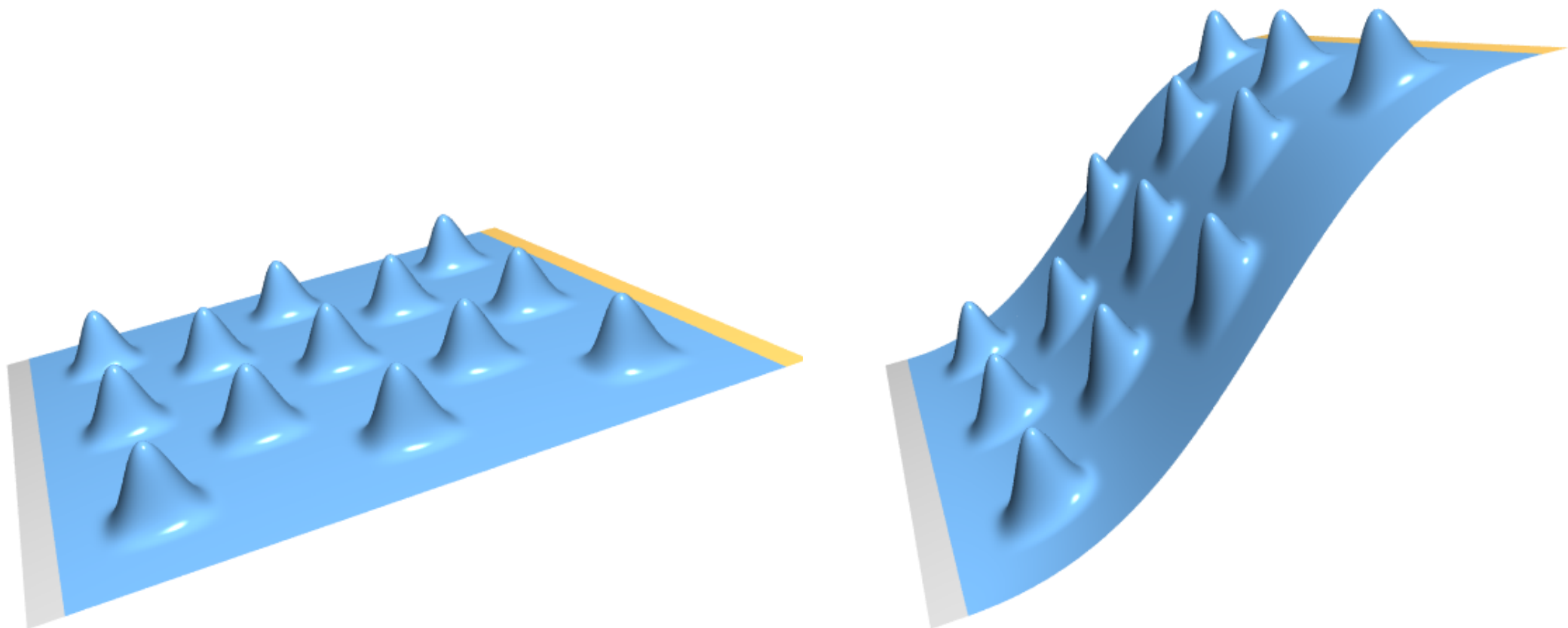$$\mathbf{F}(x,y,z) = (F(x,y,z), G(x,y,z), H(x,y,z))$$

then the Jacobian is the determinant

$$Jac(\mathbf{F}) = \begin{vmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} & \frac{\partial F}{\partial z} \\ \frac{\partial G}{\partial x} & \frac{\partial G}{\partial y} & \frac{\partial G}{\partial z} \\ \frac{\partial H}{\partial x} & \frac{\partial H}{\partial y} & \frac{\partial H}{\partial z} \end{vmatrix}$$

Space Deformation
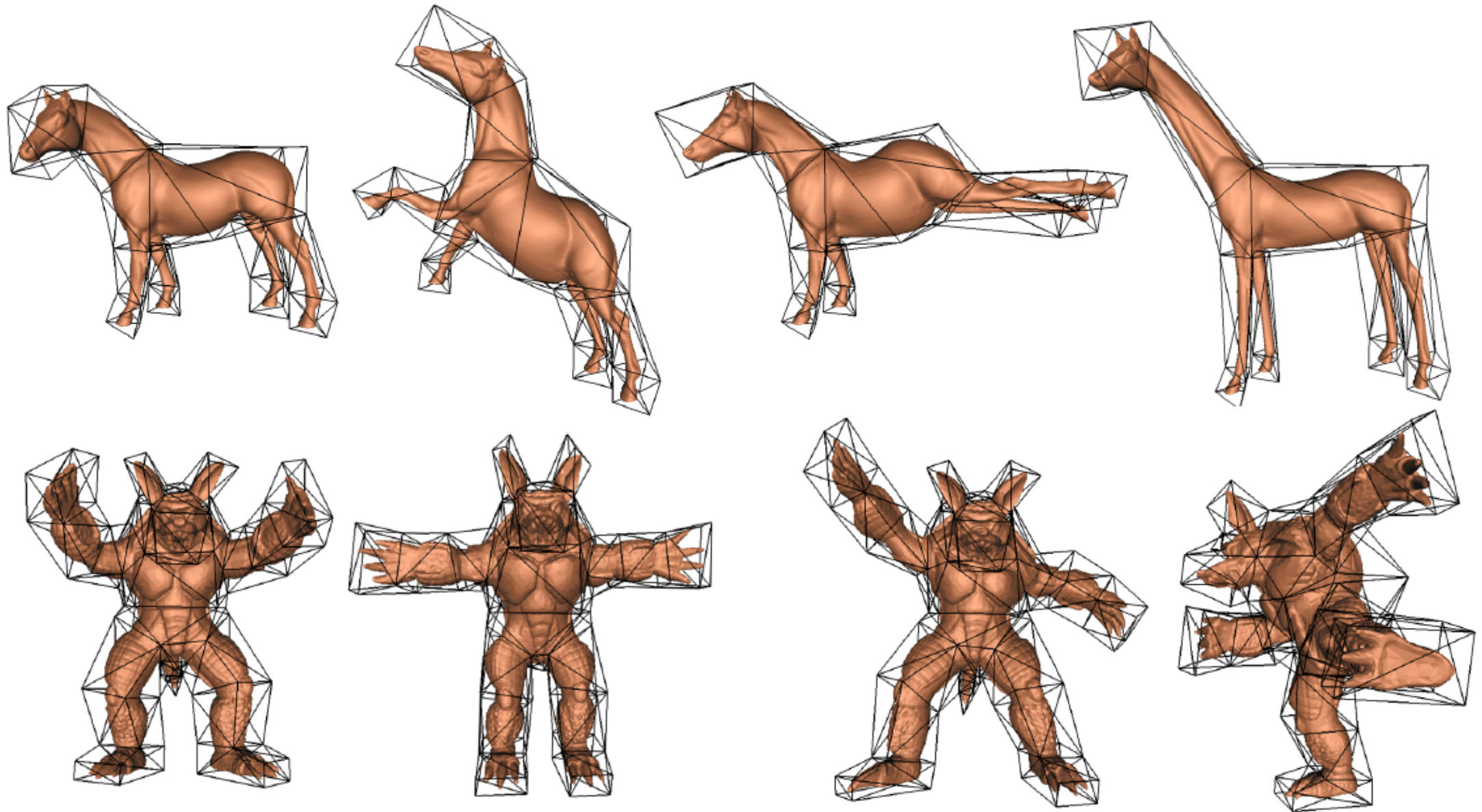
- The deformation is only loosely aware of the shape that is being edited
- Small Euclidean distance → similar deformation
- Local surface detail may be distorted

[Ju et al. 05]

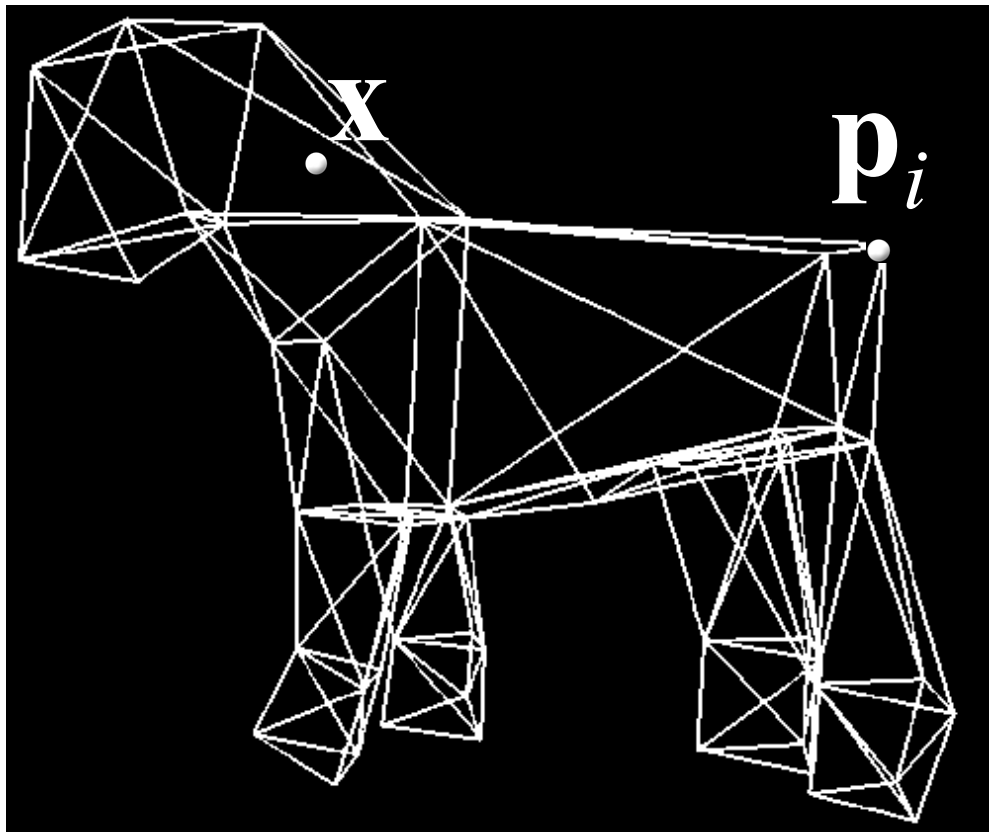- Cage = crude version of the input shape
- Polytope (not a lattice)

[Ju et al. 05]

- Each point $\mathbf{x}$ in space is represented w.r.t. to the cage elements using coordinate functions

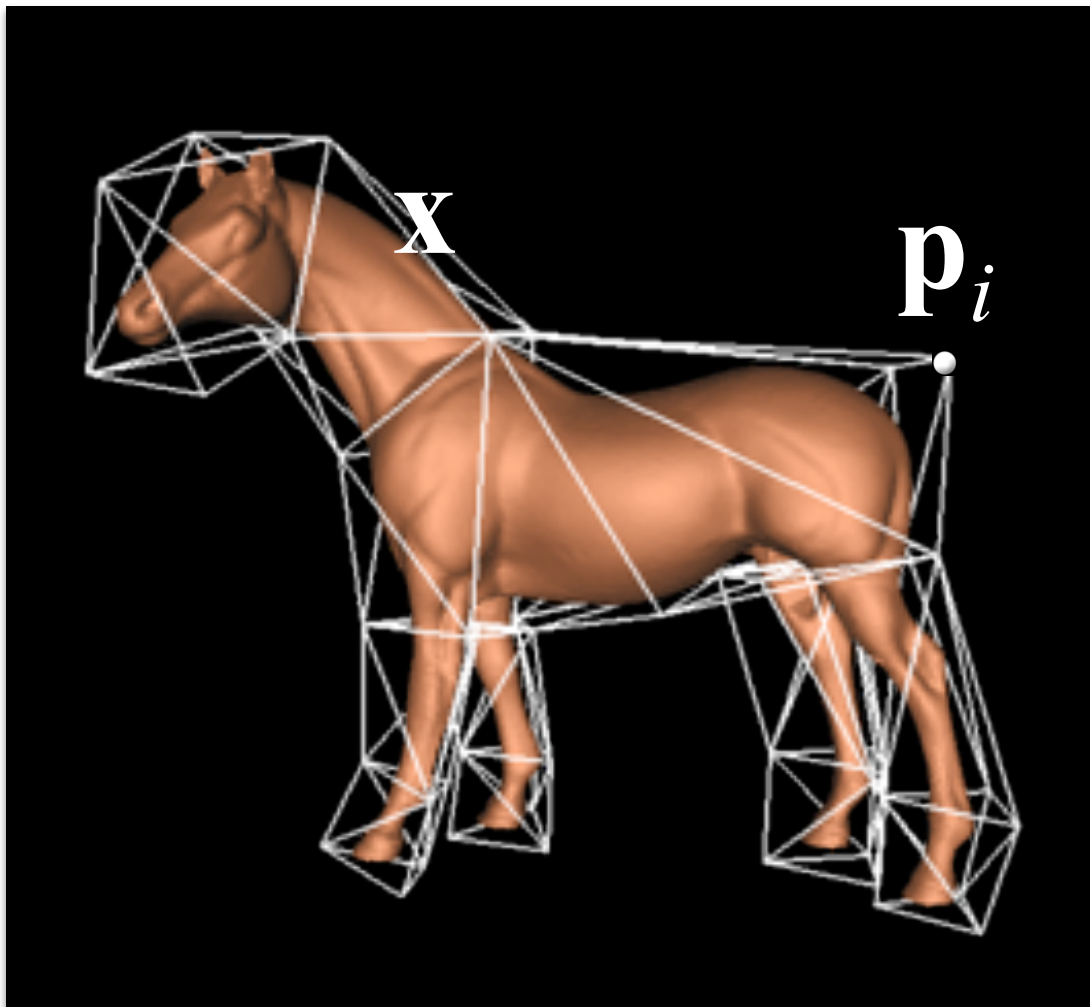$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\, \mathbf{p}_i$$

[Ju et al. 05]

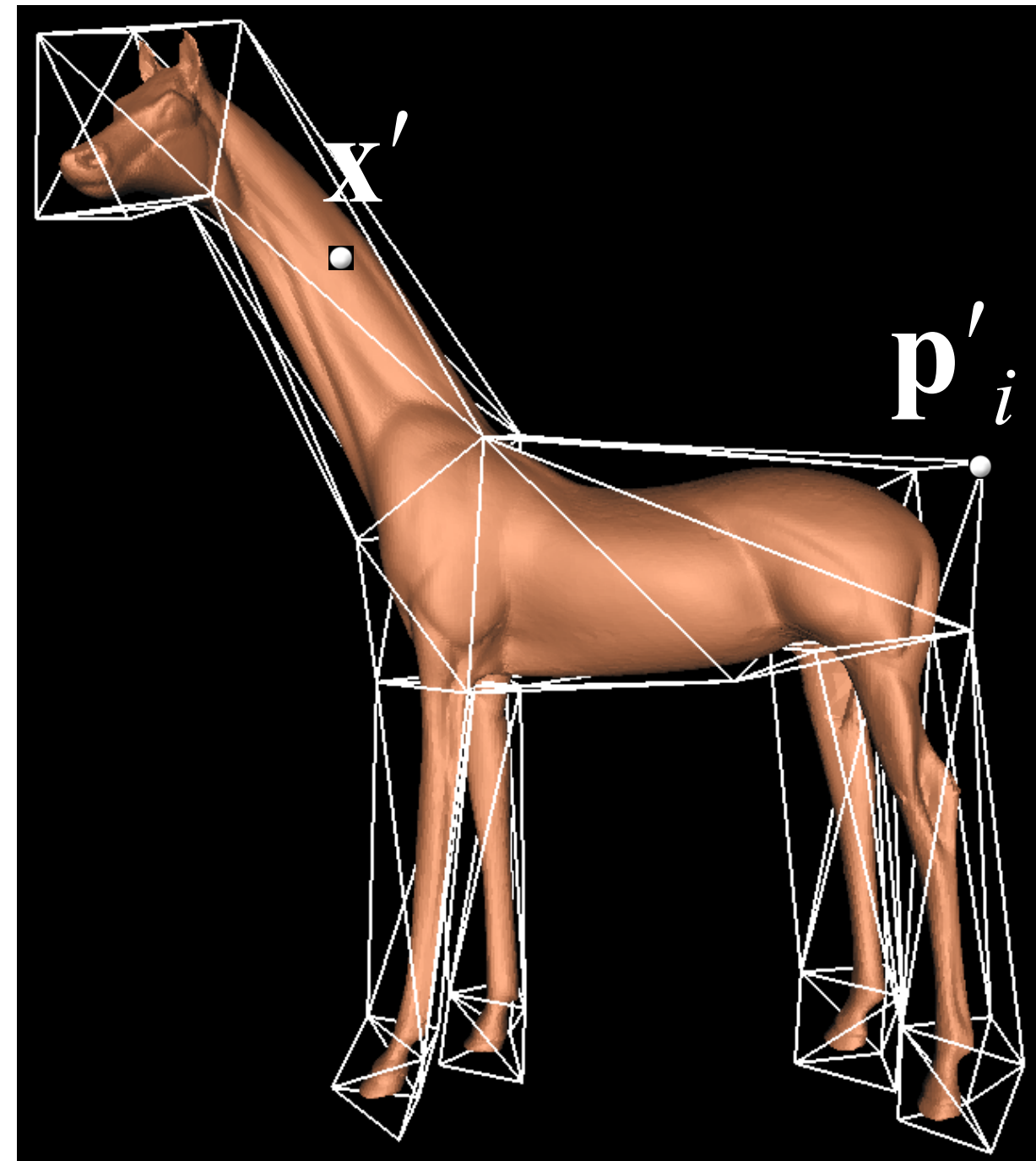- Each point $\mathbf{x}$ in space is represented w.r.t. to the cage elements using coordinate functions
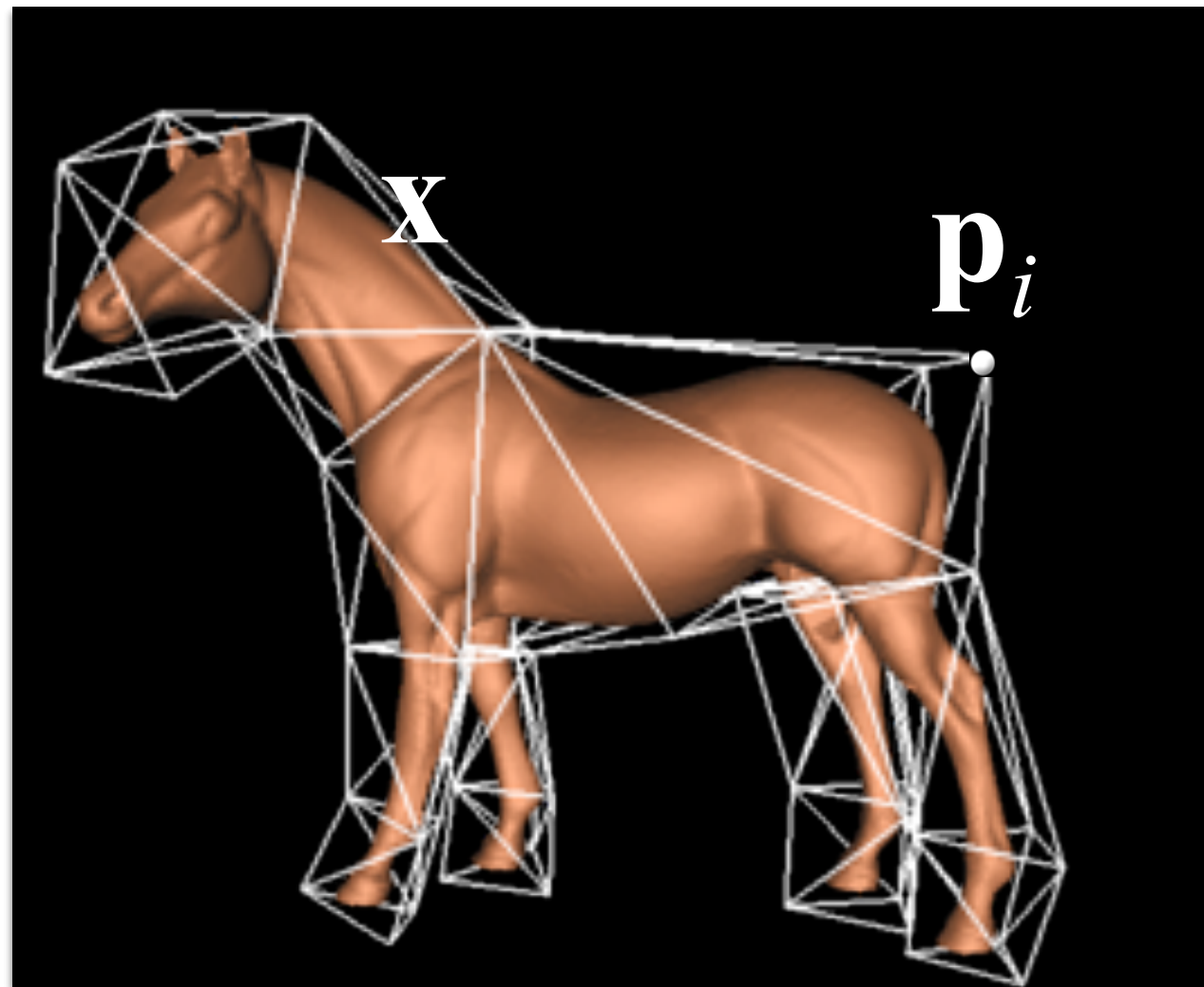

$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\, \mathbf{p}_i$$

Cage-Based Deformation

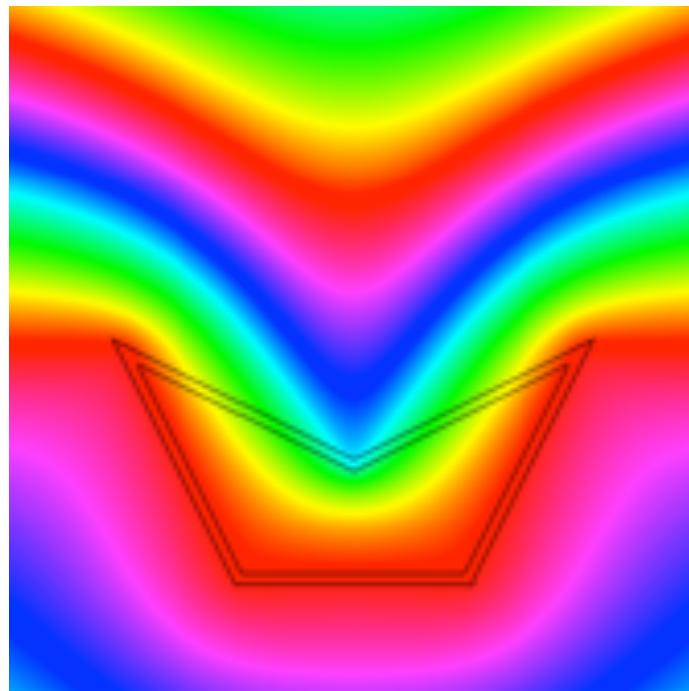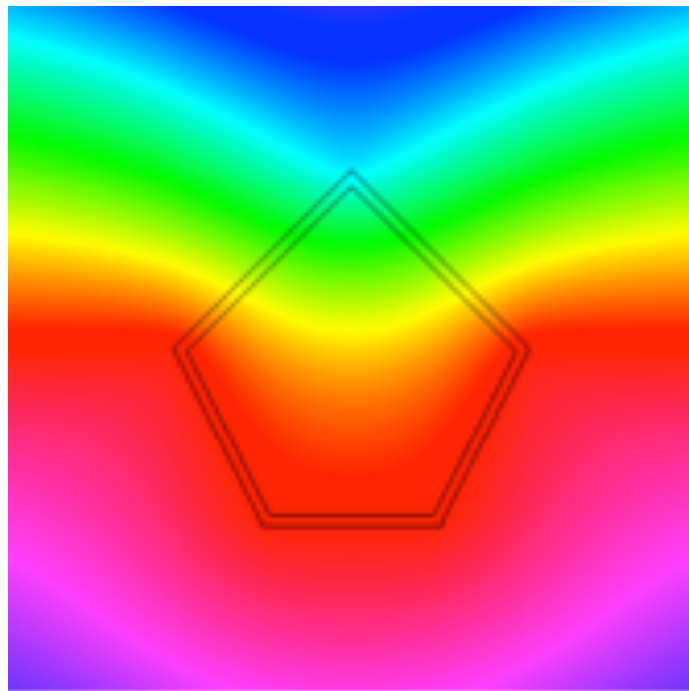$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\, \mathbf{p}'_i$$

Generalized Barycentric Coordinates

- Lagrange property: $\quad w_i(\mathbf{p}_j) = \delta_{ij}$

- Reproduction: $\quad \forall \mathbf{x}, \; \sum_{i=1}^{k} w_i(\mathbf{x}) \, \mathbf{p}_i = \mathbf{x}$

- Partition of unity: $\quad \forall \mathbf{x}, \; \sum_{i=1}^{k} w_i(\mathbf{x}) = 1$
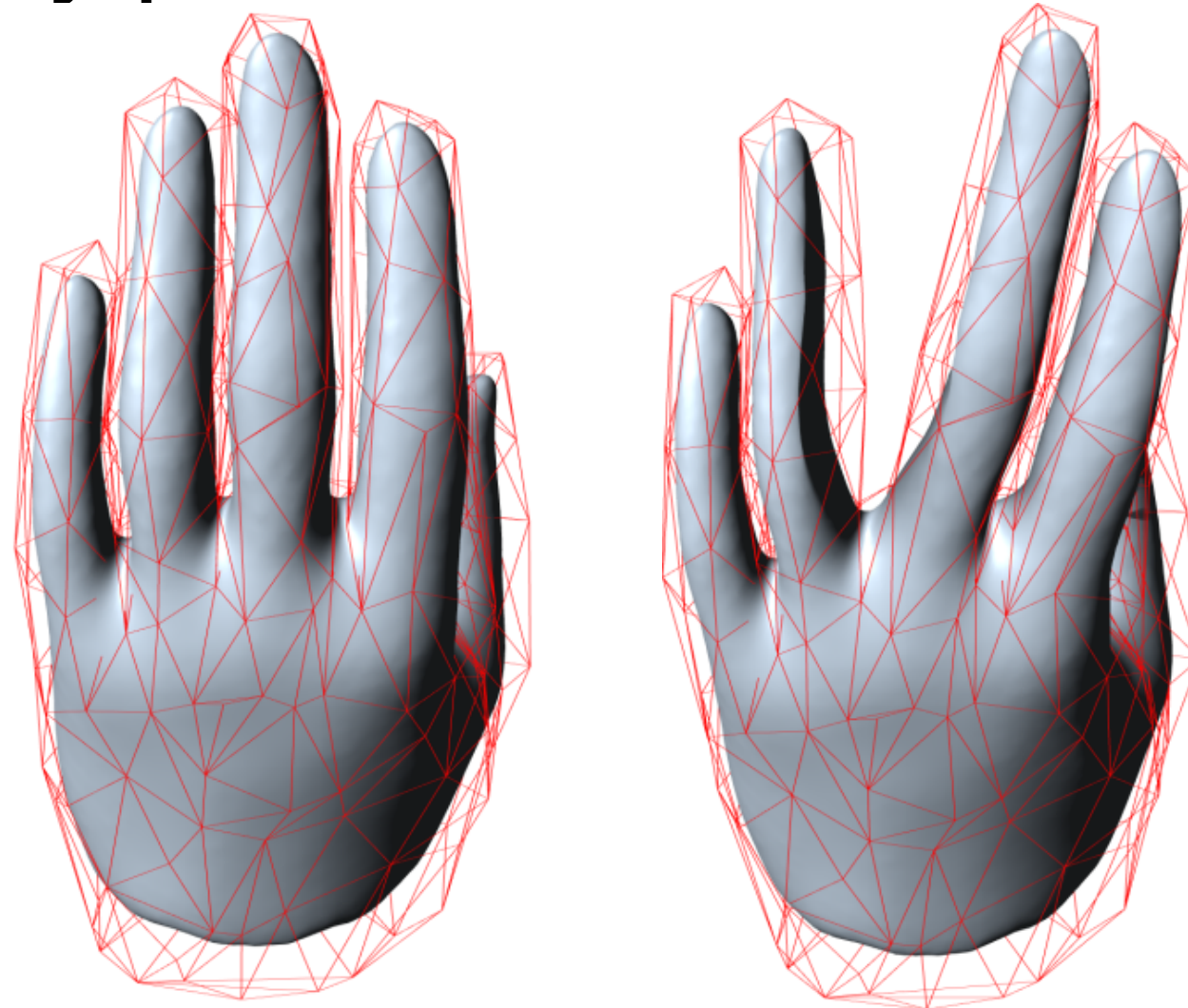
- Mean-value coordinates [Floater 2003, Ju et al. 2005]
  - Generalization of barycentric coordinates
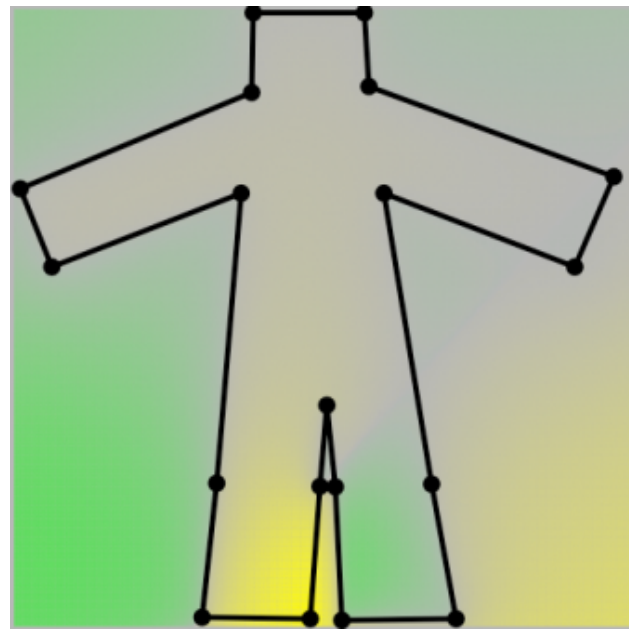 - Closed-form solution for $w_i(\mathbf{x})$

- # Mean-value coordinates
  [Floater, Ju et al. 2005]
  - Not necessarily positive on non-convex domains

Coordinate Functions

- Harmonic coordinates ([Joshi et al. 2007](#))
  - Harmonic functions $h_i(\mathbf{x})$ for each cage vertex $\mathbf{p}_i$
  - Solve $\quad \Delta h = 0$

  subject to: h_i linear on the boundary s.t. $h_i(\mathbf{p}_j) = \delta_{ij}$
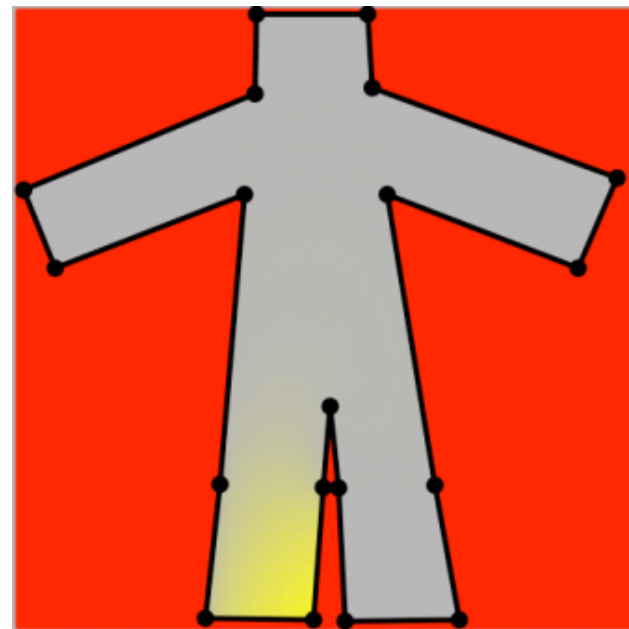

MVC   HC

Coordinate Functions

- Harmonic coordinates ([Joshi et al. 2007](#))
  - Harmonic functions $h_i(\mathbf{x})$ for each cage vertex $\mathbf{p}_i$
  - Solve $\quad \Delta h = 0$

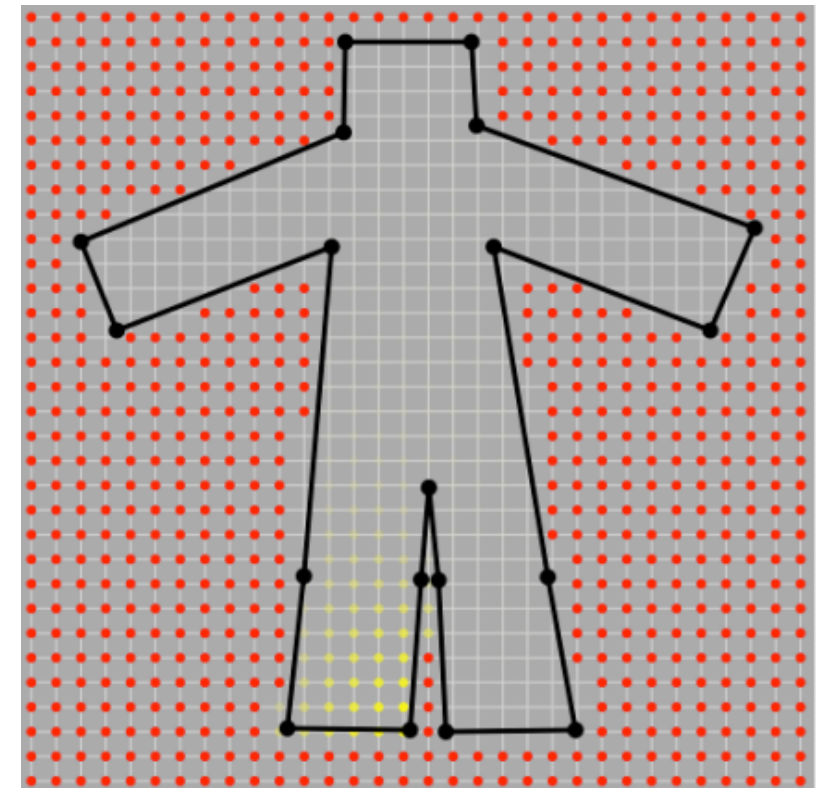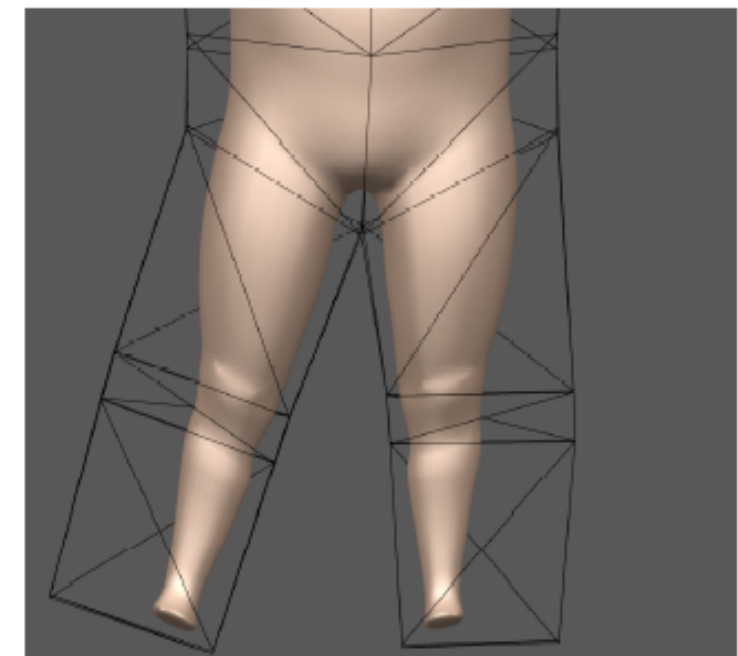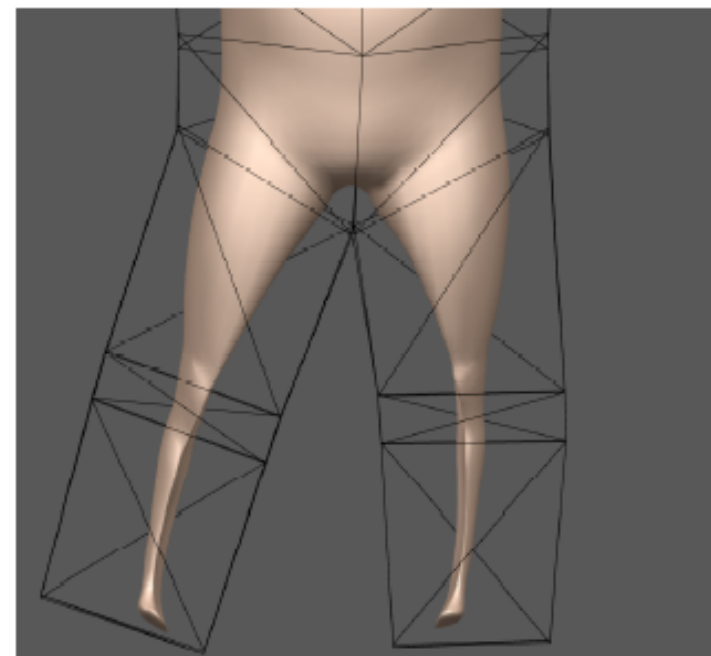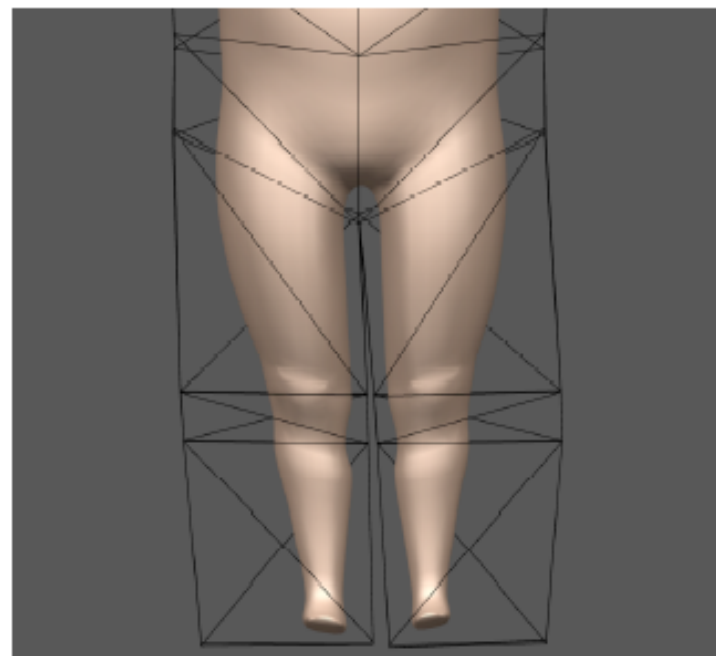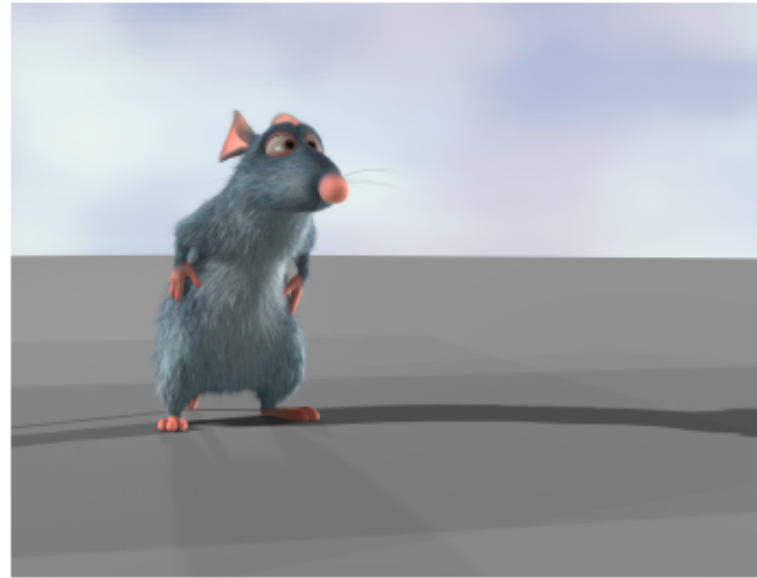 subject to: h_i linear on the boundary s.t. $h_i(\mathbf{p}_j) = \delta_{ij}$

- Volumetric Laplace equation
- Discretization, no closed-form

- Harmonic coordinates ([Joshi et al. 2007](#))

MVC

HC
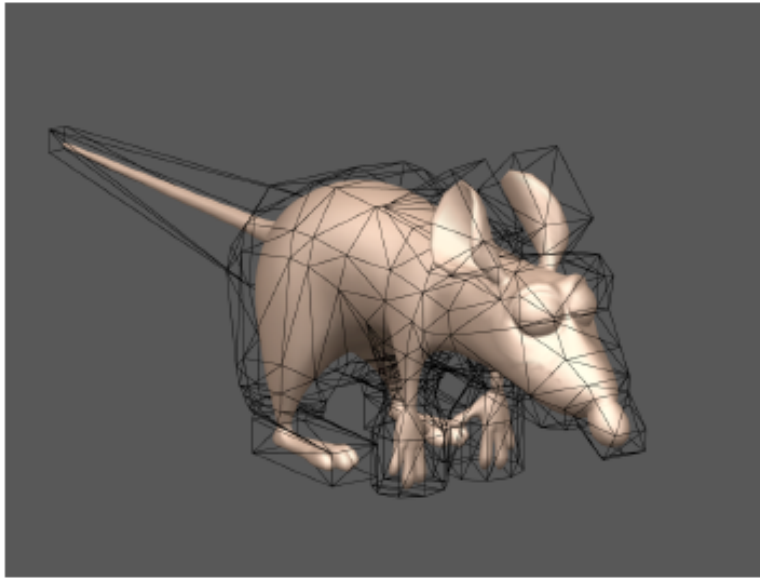
Coordinate Functions

- Green coordinates (<u>Lipman et al. 2008</u>)
- Observation: previous vertex-based basis functions always lead to affine-invariance!
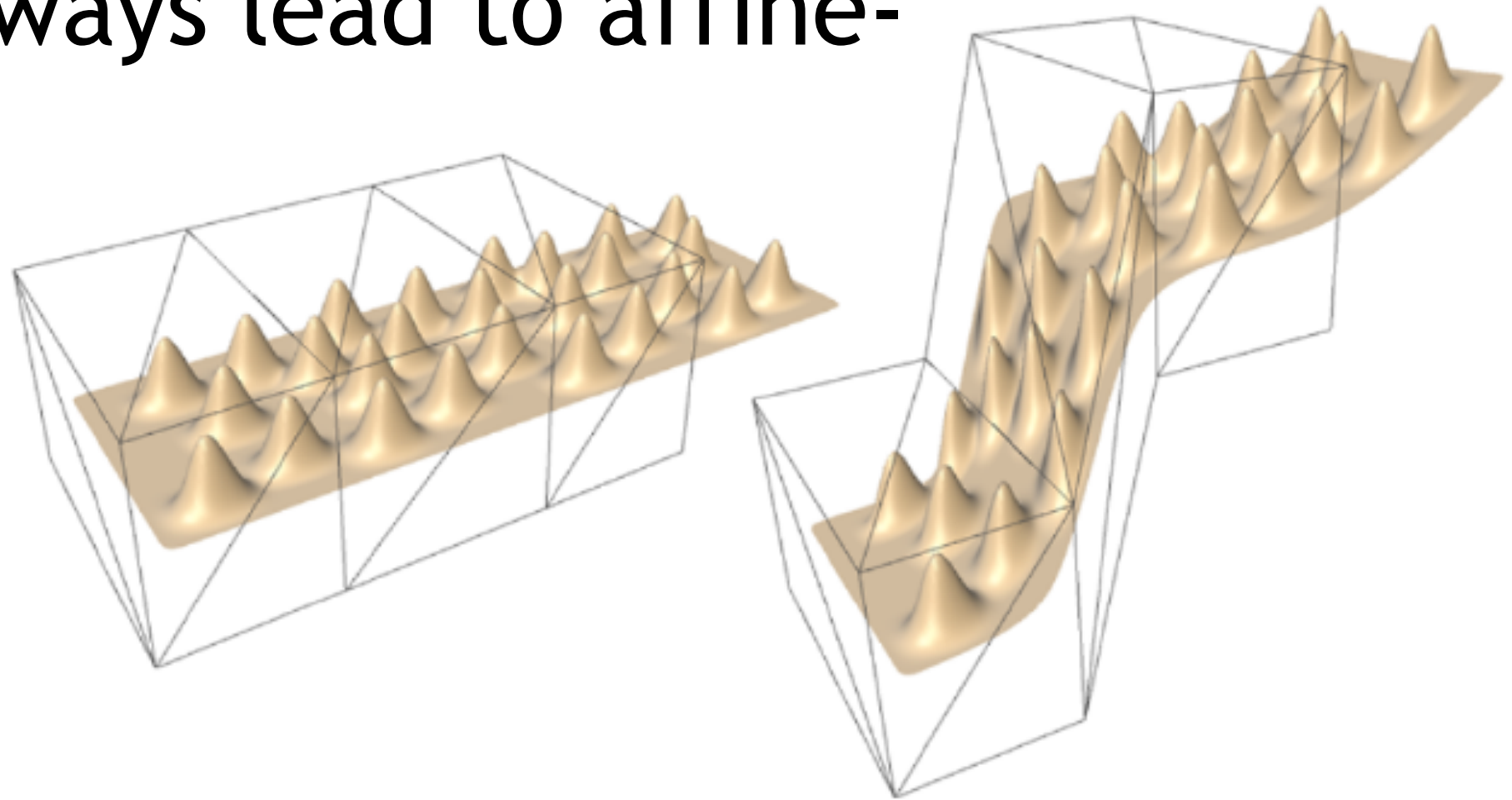
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\, \mathbf{p}'_i$$

Coordinate Functions

- Green coordinates (Lipman et al. 2008)
- Correction: Make the coordinates depend on the cage faces as well
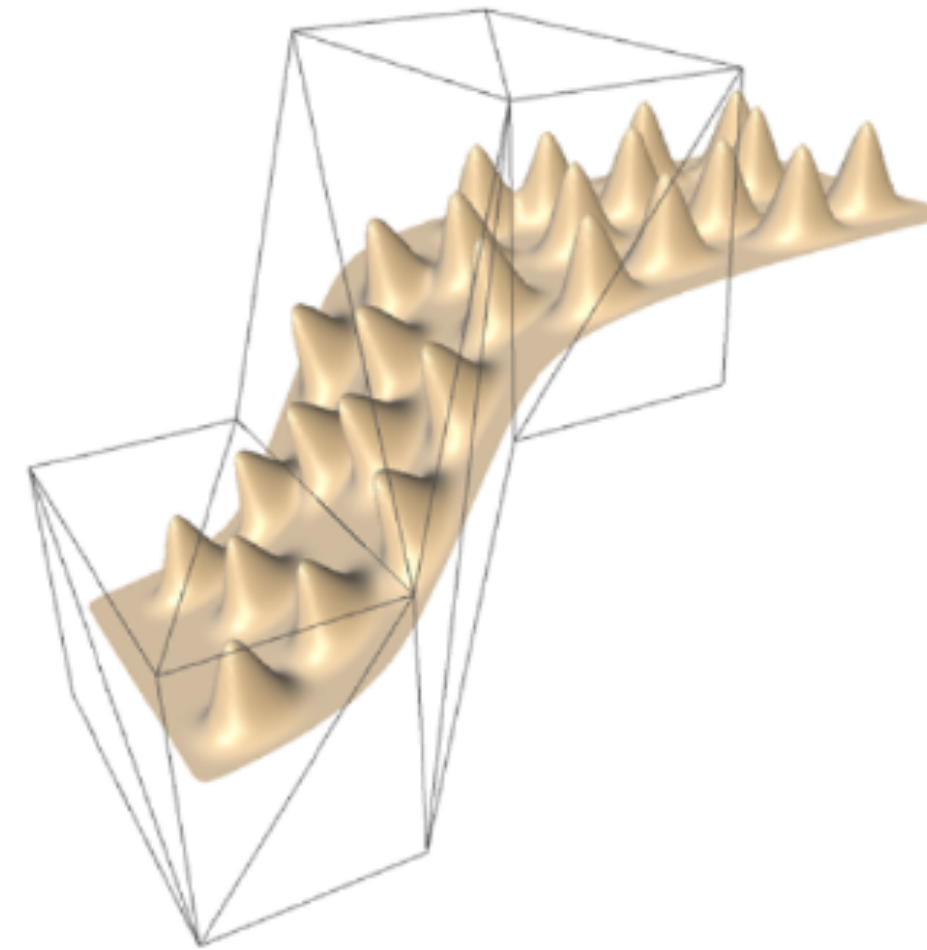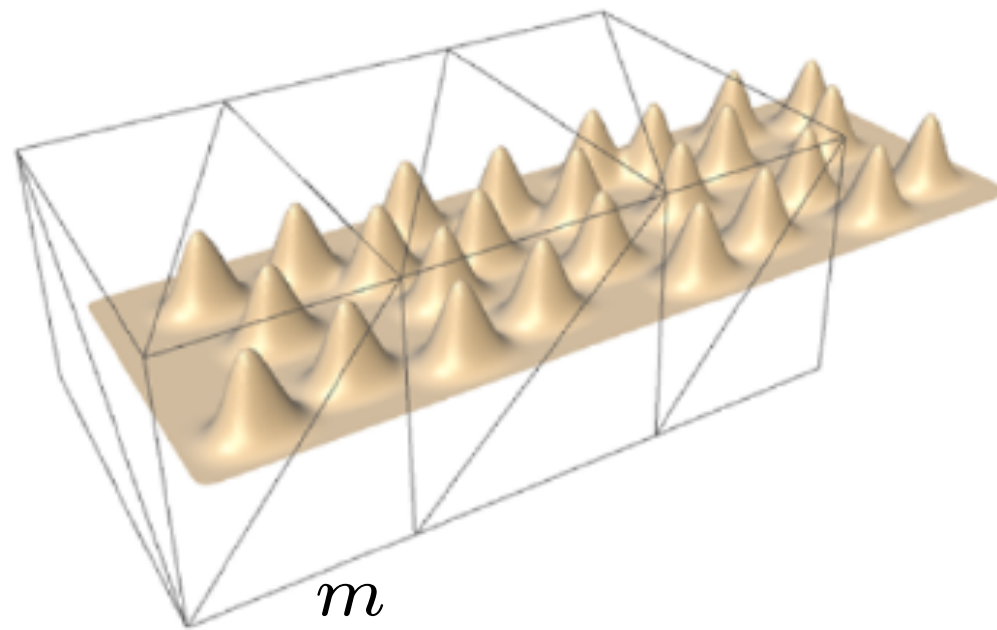
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i' + \sum_{j=1}^{m} \psi_j(\mathbf{x})\,\mathbf{n}_j'$$

Coordinate Functions

- Green coordinates ([Lipman et al. 2008](#))
- Closed-form solution
- Conformal in 2D, quasi-conformal in 3D

GC   MVC   GC
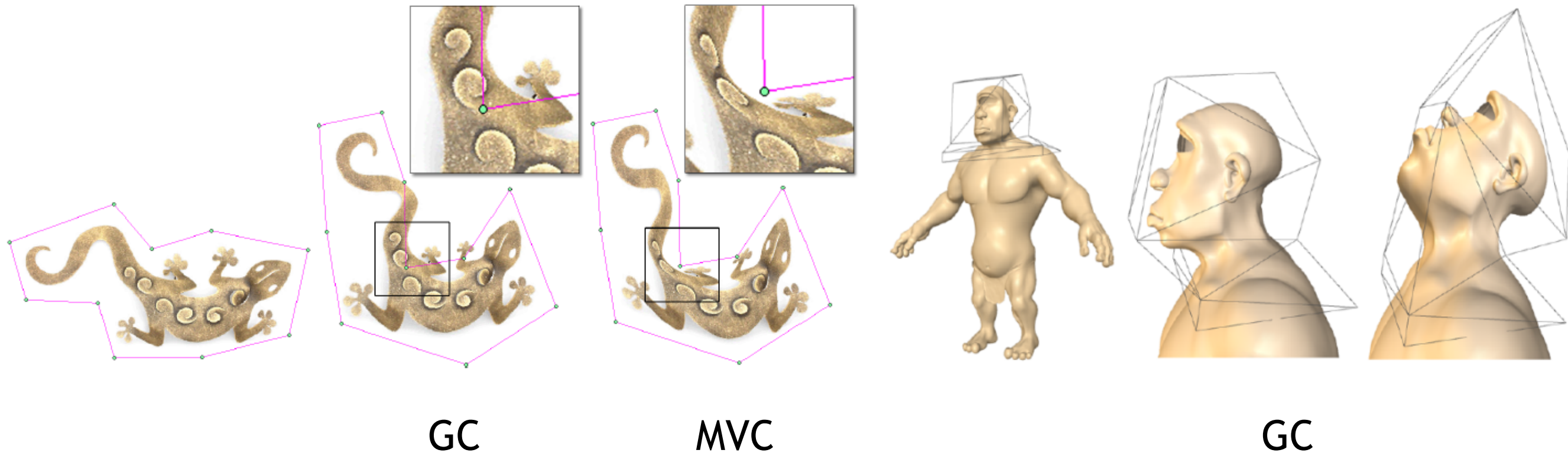
Coordinate Functions

- Green coordinates (Lipman et al. 2008)
- Closed-form solution
- Conformal in 2D, quasi-conformal in 3D

Alternative interpretation in 2D via holomorphic functions and extension to point handles : **Weber et al. Eurographics 2009**
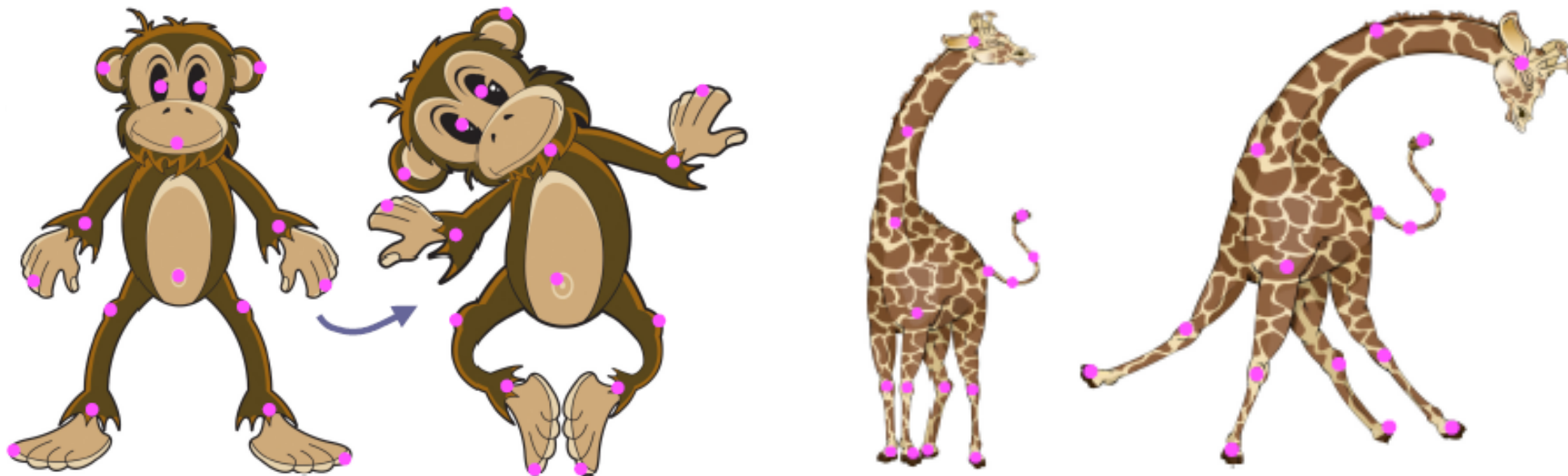
Pros:

- Nice control over volume
  - Squish/stretch

Cons:

- Hard to control details of embedded surface

Non-Linear Space Deformation

- Involve nonlinear optimization
- Enjoy the advantages of space warps
- Additionally, have shape-preserving properties

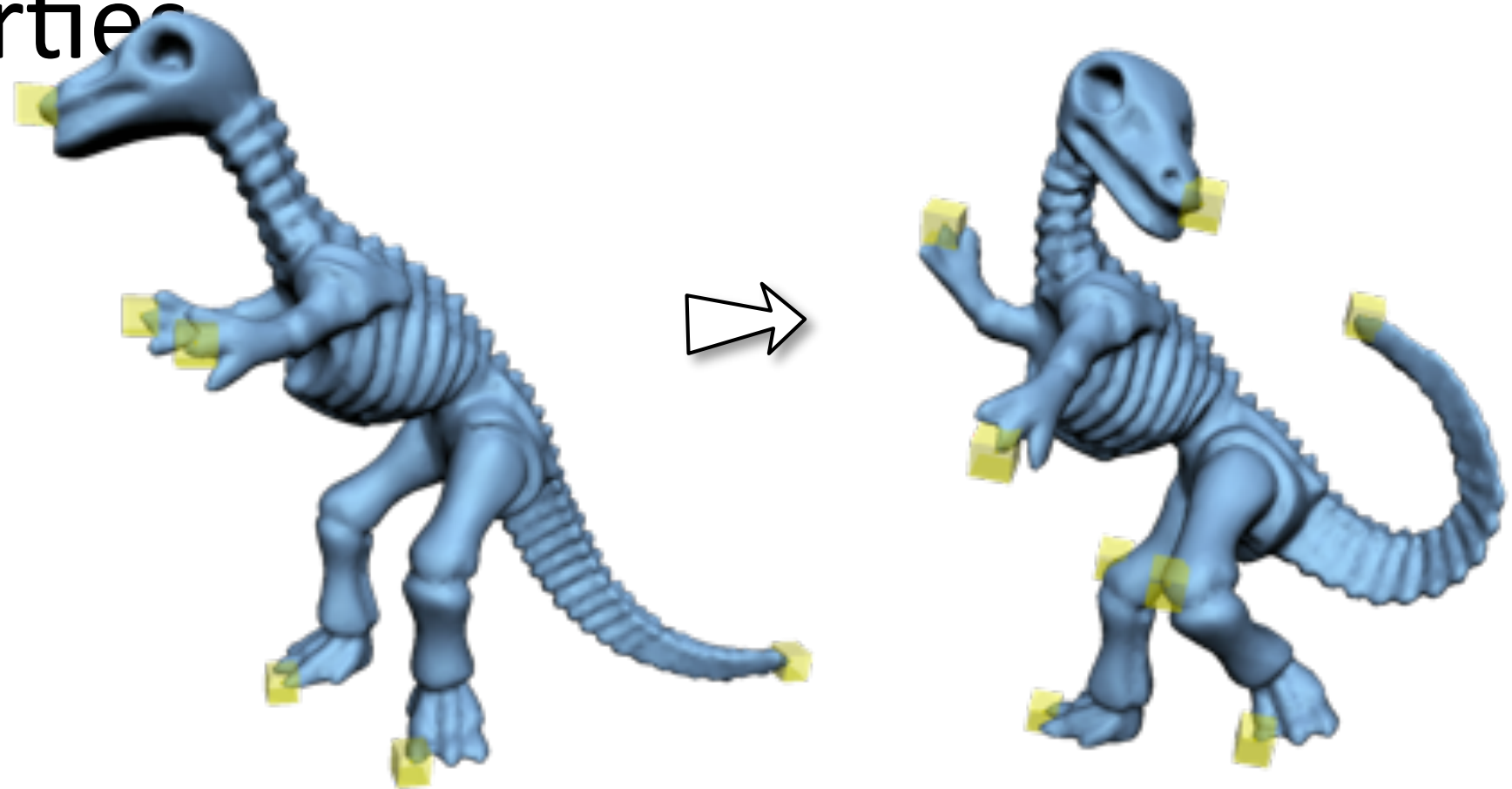Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Points or segments as control objects
- First developed in 2D and later extended to 3D by Zhu and Gortler (2007)

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Attach an affine transformation to each point $\mathbf{x} \in R^3$:

$$A_\mathbf{x}(\mathbf{p}) = M_\mathbf{x}\mathbf{p} + \mathbf{t}_\mathbf{x}$$

- The space warp:

$$\mathbf{x} \rightarrow A_\mathbf{x}(\mathbf{x})$$

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]
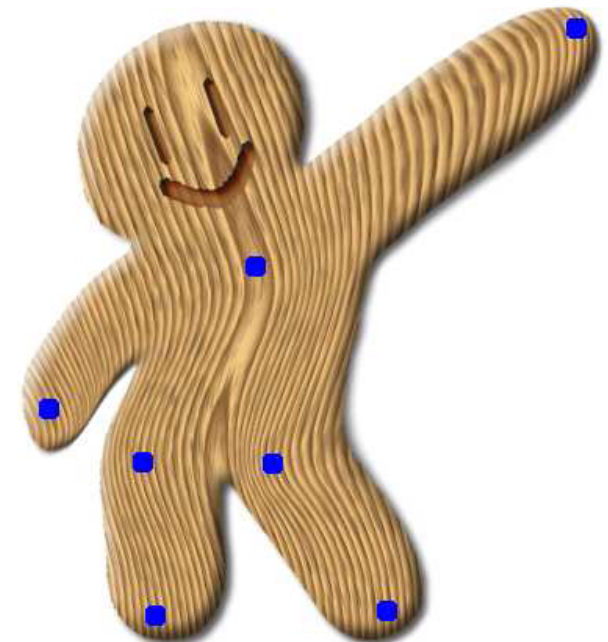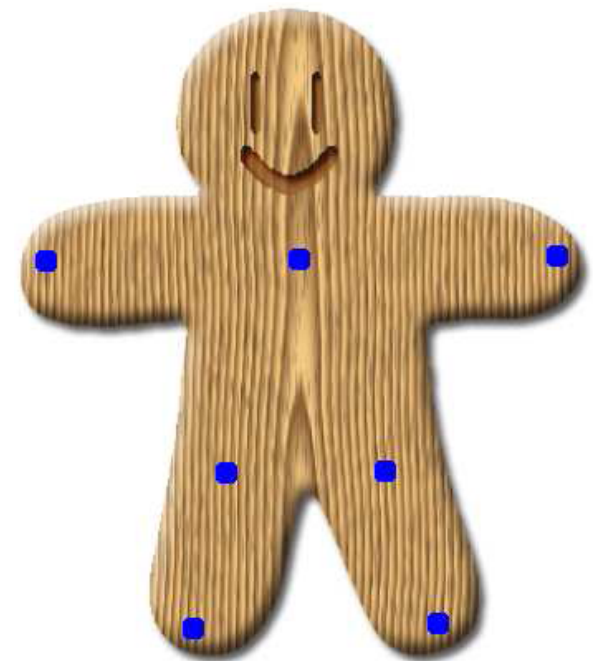
- Handles $\mathbf{p}_i$ are displaced to $\mathbf{q}_i$
- The local transformation at $\mathbf{x}$:

$$A_{\mathbf{x}}(\mathbf{p}) = M_{\mathbf{x}}\mathbf{p} + \mathbf{t}_{\mathbf{x}} \quad \text{s.t.}$$

$$\sum_{i=1}^{k} w_i(\mathbf{x}) \left\| A_{\mathbf{x}}(\mathbf{p}_i) - \mathbf{q}_i \right\|^2 \;\rightarrow\; \min$$

- The weights depend on $\mathbf{x}$:

$$w_i(\mathbf{x}) = \left\| \mathbf{p}_i - \mathbf{x} \right\|^{-2\alpha}$$

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- No additional restriction on $A_x(\cdot)$ — affine local transformations

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Restrict $A_x(\cdot)$ to similarity

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Restrict $A_{\mathbf{x}}(\cdot)$ to similarity

$$M_{\mathbf{x}} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Restrict $A_x(\cdot)$ to rigid

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Restrict $A_{\mathbf{x}}(\cdot)$ to rigid

$$M_{\mathbf{x}} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

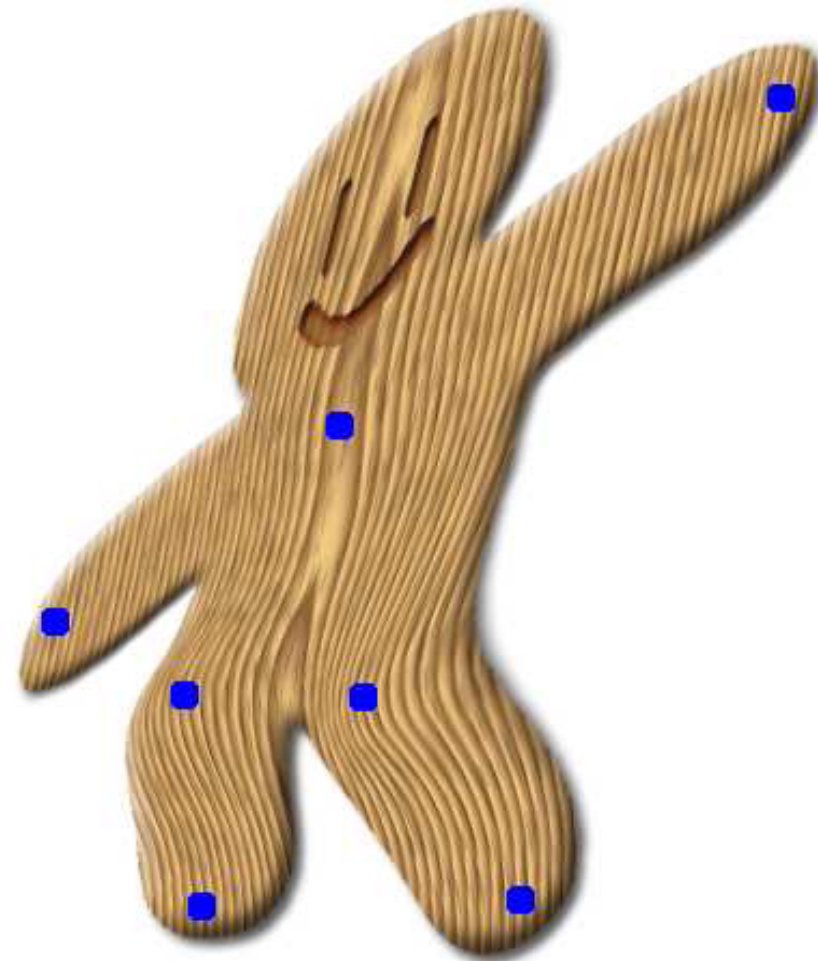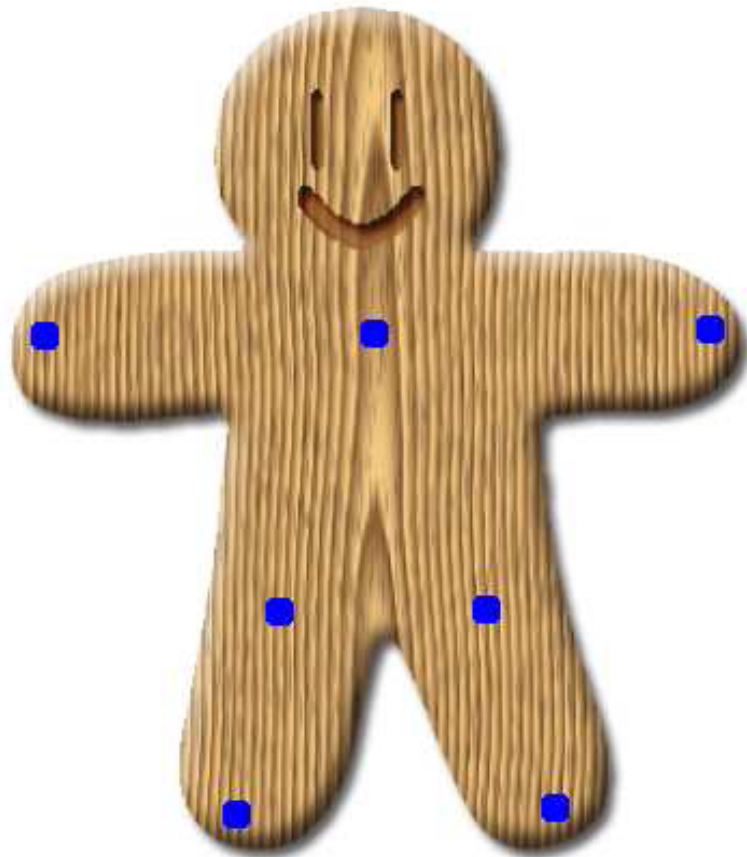Solve for $M_{\mathbf{x}}$ like similarity and then normalize

Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- Examples

Moving-Least-Squares (MLS) extension to 3D [Zhu & Gortler 07]

- No linear expression for similarity in 3D
- Instead, can solve for the minimizing rotation

$$\underset{R \in SO(3)}{\arg\min} \sum_{i=1}^{k} w_i(\mathbf{x}) \left\| R\mathbf{p}_i - \mathbf{q}_i \right\|^2$$
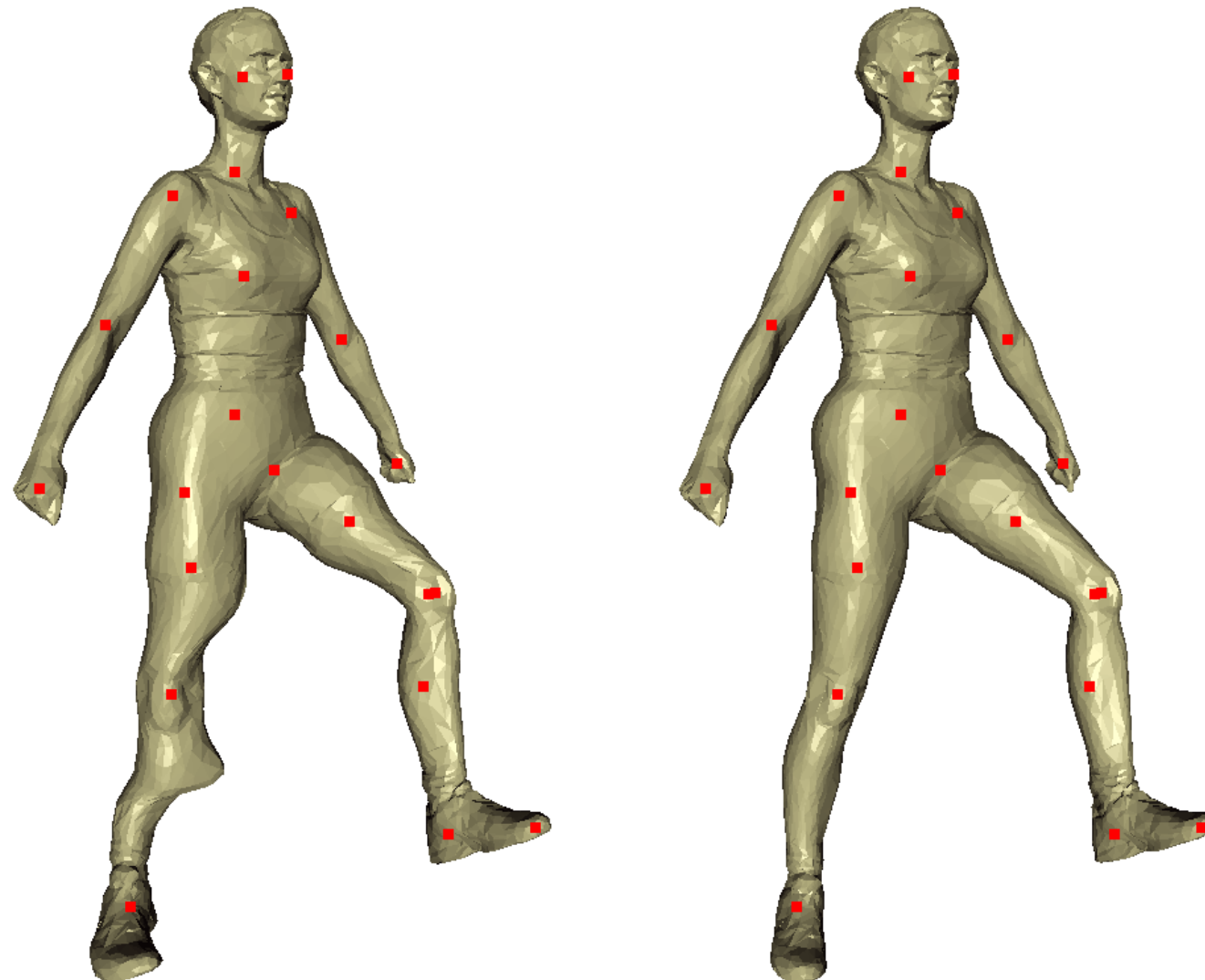
by polar decomposition of the 3×3 covariance matrix

As-Rigid-As-Possible Deformation

Moving-Least-Squares (MLS) extension to 3D [Zhu & Gortler 07]

- Zhu and Gortler also replace the Euclidean distance in the weights by "distance within the shape"

$$w_i(\mathbf{x}) = d(\mathbf{p}_i, \mathbf{x})^{-2\alpha}$$

Moving-Least-Squares (MLS) extension to 3D [Zhu & Gortler 07]

- More results
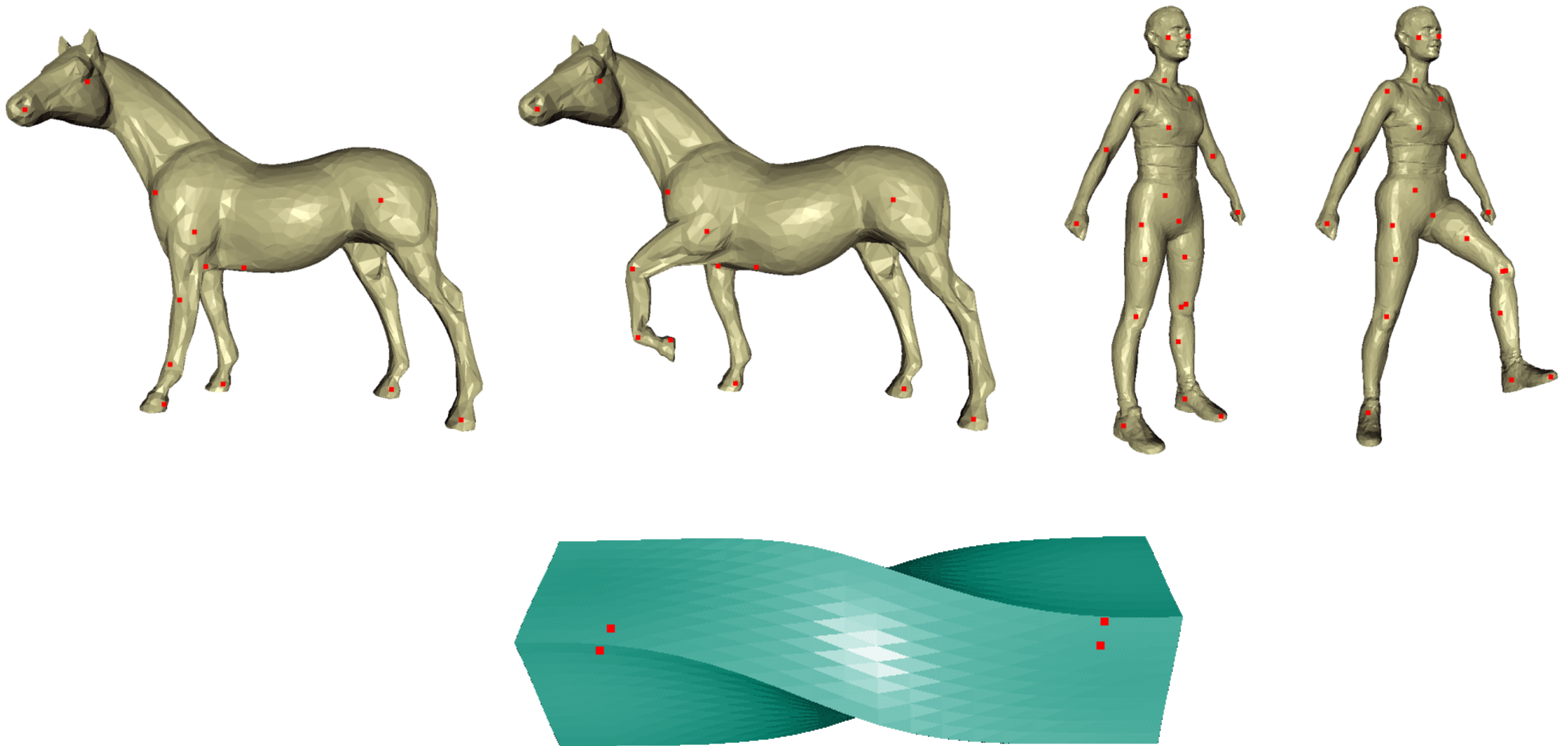
As-Rigid-As-Possible Deformation

- Surface handles as interface

- Underlying graph to represent the deformation; nodes store rigid transformations

- Decoupling of handles from def. representation
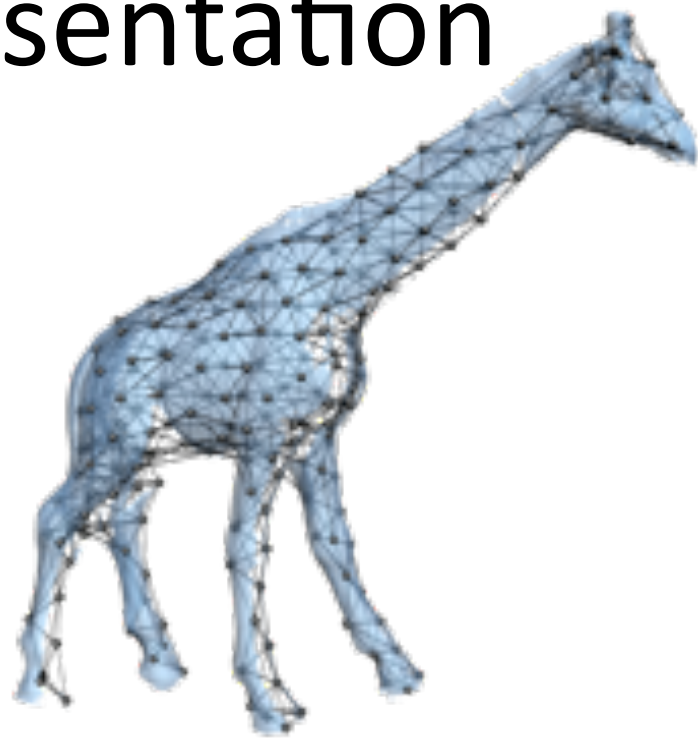
Deformation Graph
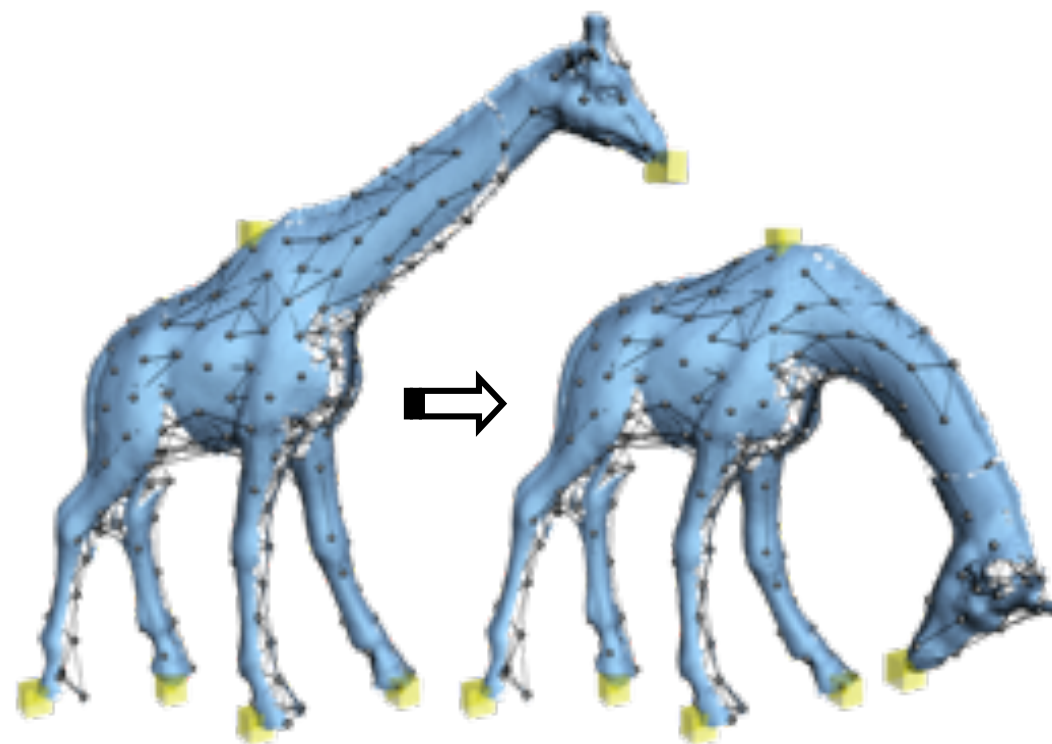
Optimization Procedure

Deformation Graph

Deformation Graph

Begin with an embedded object.

Deformation Graph

Begin with an embedded object.

Nodes selected via uniform sampling; located at $\mathbf{g}_j$

One rigid transformation for each node: $\mathbf{R}_j$, $\mathbf{t}_j$

Each node deforms nearby space.

Edges connect nodes of overlapping influence.

Deformation Graph

Begin with an embedded object.

Nodes selected via uniform sampling; located at $\mathbf{g}_j$

One rigid transformation for each node: $\mathbf{R}_j$, $\mathbf{t}_j$

Each node deforms nearby space.

Edges connect nodes of overlapping influence.

Deformation Graph

Influence of nearby transformations is blended.

point $\mathbf{x}$ transformed by node j

$$\mathbf{x}' = \sum_{j=1}^{m} w_j(\mathbf{x})\left[\mathbf{R}_j(\mathbf{x} - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j\right]$$

$$w_j(\mathbf{x}) = (1 - \left\|\mathbf{x} - \mathbf{g}_j\right\| / d_{\max})^2$$

Optimization

Select & drag vertices of embedded object.

Optimization

Select & drag vertices of embedded object.

Optimization finds

deformation parameters R_j , t_j.

Optimization

$$\min_{\mathbf{R}_1,\mathbf{t}_1,\ldots,\mathbf{R}_m,\mathbf{t}_m} \quad w_{\text{rot}} \mathrm{E}_{\text{rot}} \quad + \quad w_{\text{reg}} \mathrm{E}_{\text{reg}} \quad + \quad w_{\text{con}} \mathrm{E}_{\text{con}}$$

Graph parameters

Rotation term

Regularization term

Constraint term

Select & drag vertices of embedded object.

Optimization finds

deformation parameters R_j , $\mathbf{t}_{j}$.

56

Optimization

$$\min_{\mathbf{R}_1,\mathbf{t}_1,\ldots,\mathbf{R}_m,\mathbf{t}_m} w_{rot}\mathrm{E}_{rot} + w_{reg}\mathrm{E}_{reg} + w_{con}\mathrm{E}_{con}$$

$$\mathrm{Rot}(\mathbf{R}) = (\mathbf{c}_1 \cdot \mathbf{c}_2)^2 + (\mathbf{c}_1 \cdot \mathbf{c}_3)^2 + (\mathbf{c}_2 \cdot \mathbf{c}_3)^2 +$$

$$(\mathbf{c}_1 \cdot \mathbf{c}_1 - 1)^2 + (\mathbf{c}_2 \cdot \mathbf{c}_2 - 1)^2 + (\mathbf{c}_3 \cdot \mathbf{c}_3 - 1)^2$$
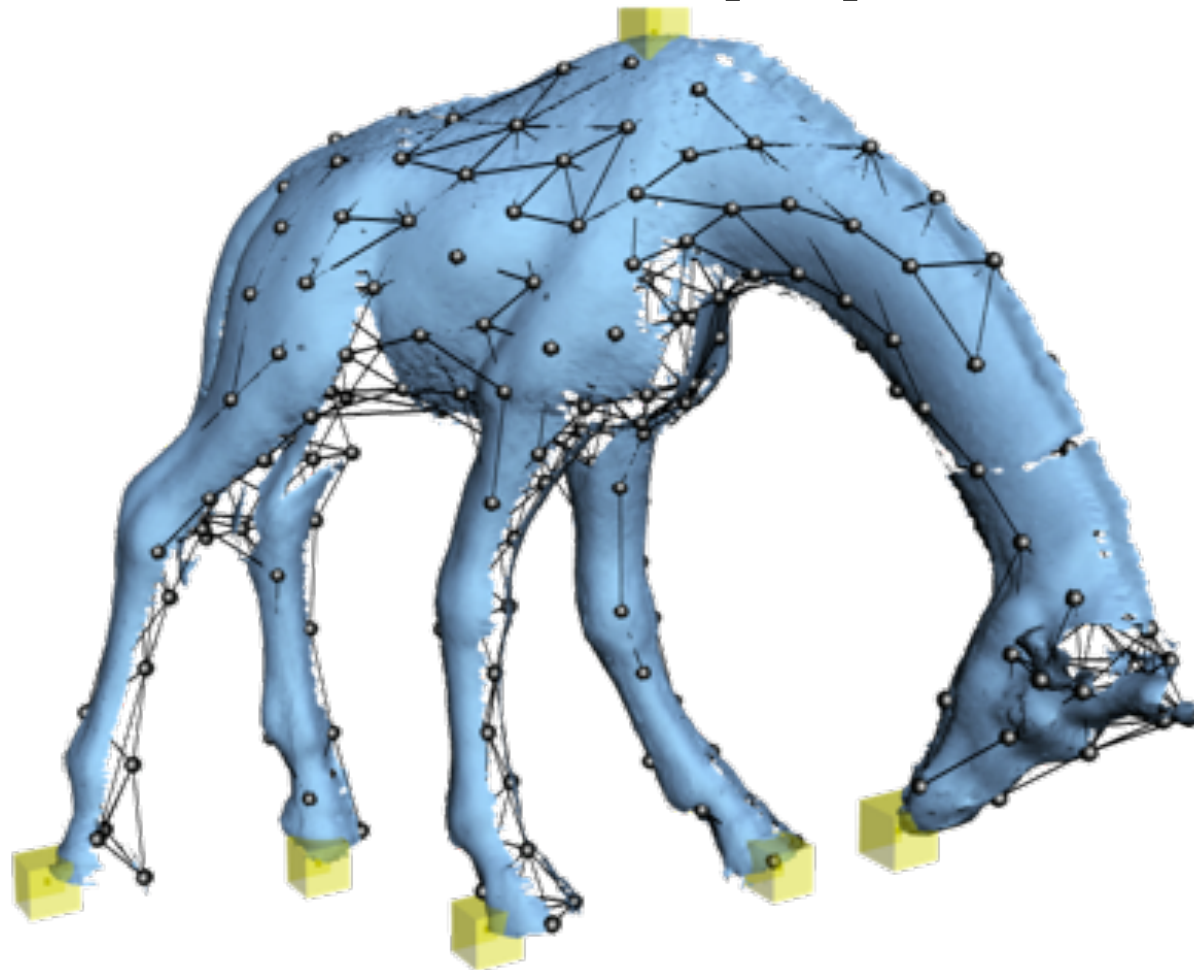
$$\mathrm{E}_{rot} = \sum_{j=1}^{m} \mathrm{Rot}(\mathbf{R}_j)$$

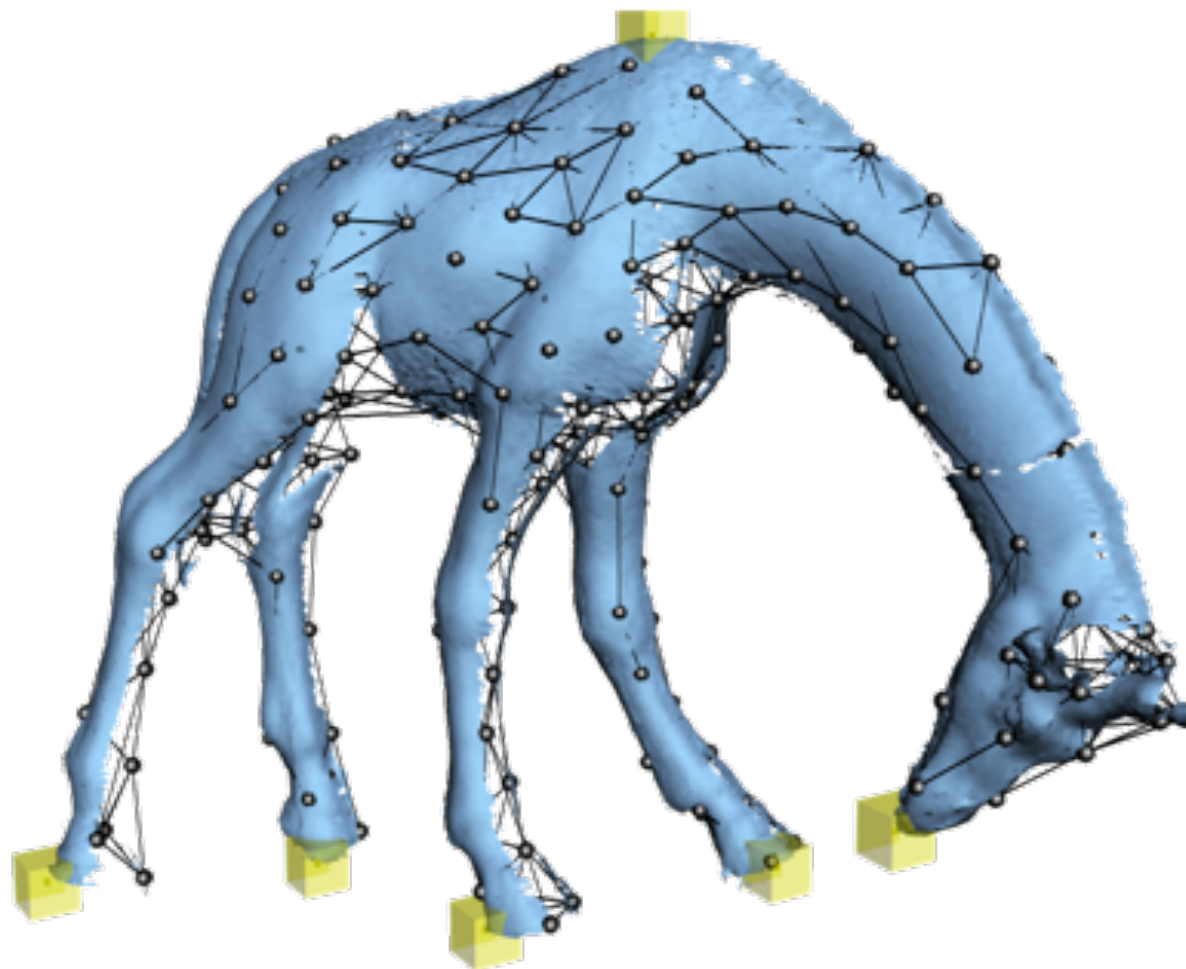For detail preservation, features should rotate and not scale or skew.

Optimization

$$\min_{\mathbf{R}_1,\mathbf{t}_1,\dots,\mathbf{R}_m,\mathbf{t}_m} \quad w_{\text{rot}}\mathrm{E}_{\text{rot}} \;+\; w_{\text{reg}}\mathrm{E}_{\text{reg}} \;+\; w_{\text{con}}\mathrm{E}_{\text{con}}$$

$$\mathrm{E}_{\text{reg}} = \sum_{j=1}^{m} \sum_{k \in N(j)} \alpha_{jk} \left\| \mathbf{R}_j(\mathbf{g}_k - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_k + \mathbf{t}_k) \right\|_2^2$$

where node j thinks node k should go

where node k actually goes
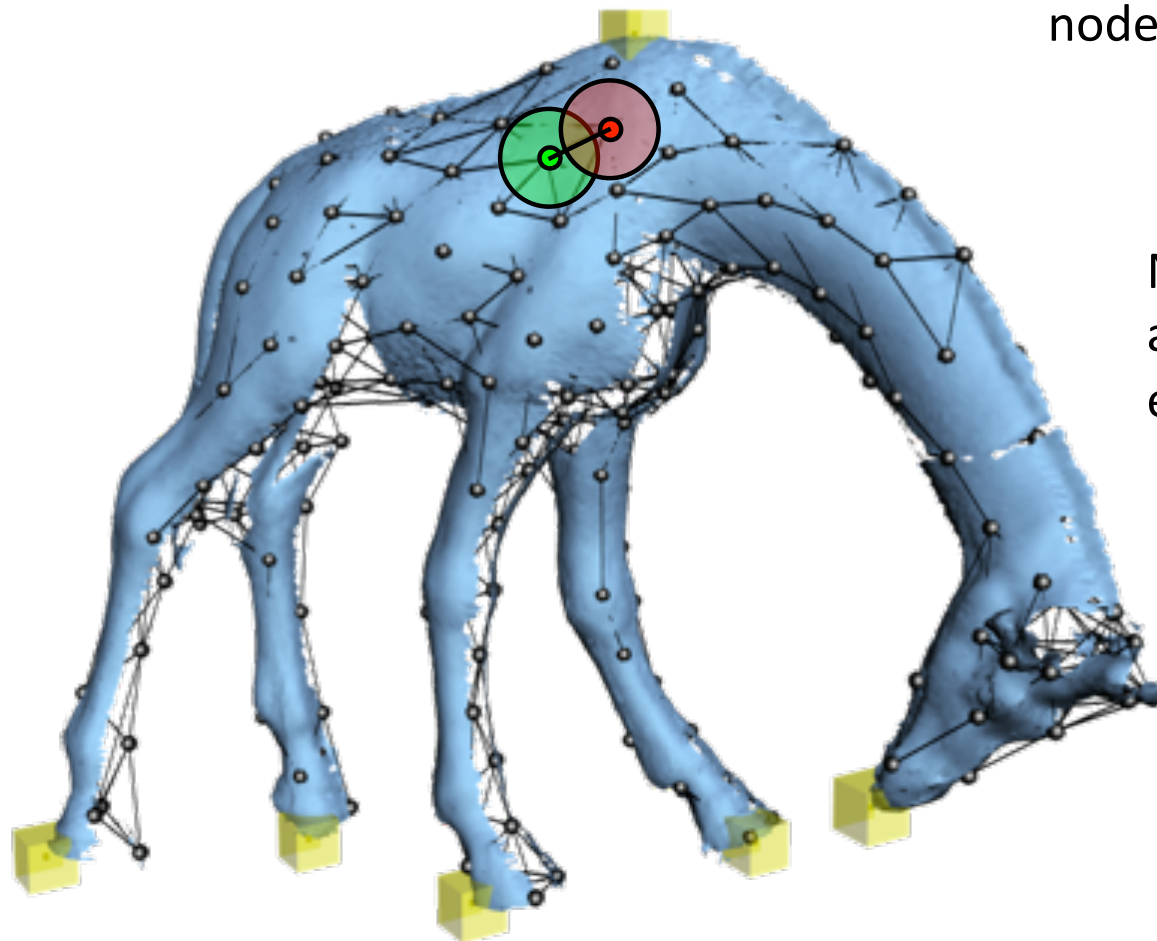
Neighboring nodes should agree on where they transform each other.

Optimization

$$\min_{\mathbf{R}_1,\mathbf{t}_1,\ldots,\mathbf{R}_m,\mathbf{t}_m} \quad w_{\text{rot}}\text{E}_{\text{rot}} \quad + \quad w_{\text{reg}}\text{E}_{\text{reg}} \quad + \quad w_{\text{con}}\text{E}_{\text{con}}$$

$$\text{E}_{\text{con}} = \sum_{l=1}^{p} \left\| \widetilde{\mathbf{v}}_{\text{index}(l)} - \mathbf{q}_l \right\|_2^2$$
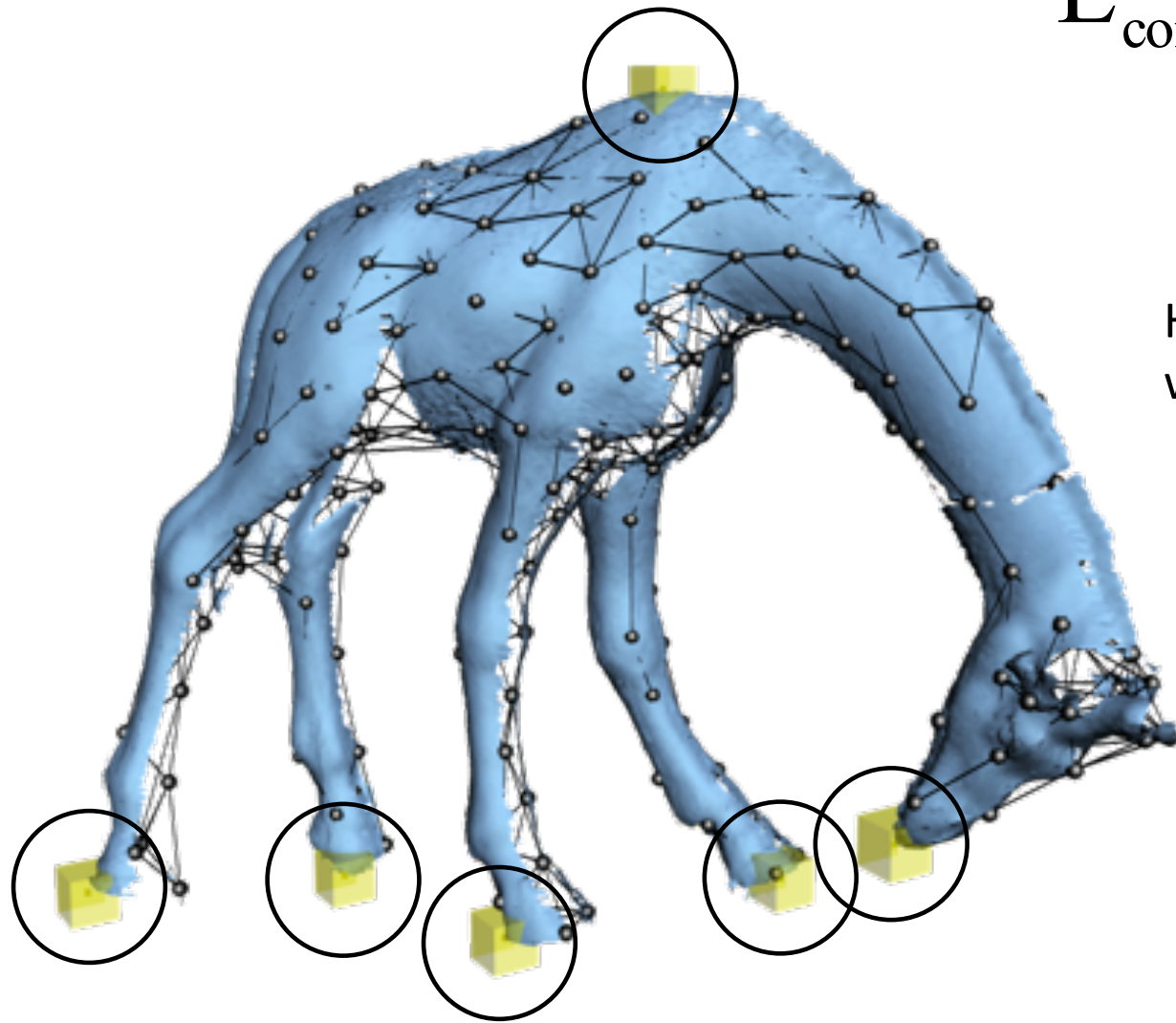
Handle vertices should go where the user puts them.

Optimization

$$\min_{\mathbf{R}_1,\mathbf{t}_1,\ldots,\mathbf{R}_m,\mathbf{t}_m} \quad w_{\text{rot}}\mathrm{E}_{\text{rot}} \quad + \quad w_{\text{reg}}\mathrm{E}_{\text{reg}} \quad + \quad w_{\text{con}}\mathrm{E}_{\text{con}}$$

Results on Polygon Soups

Embedded Deformation [Sumner et al. 07]

Detail Preservation

Discussion

- Decoupling of deformation complexity and model complexity

- Nonlinear energy optimization – results comparable to surface-based approaches
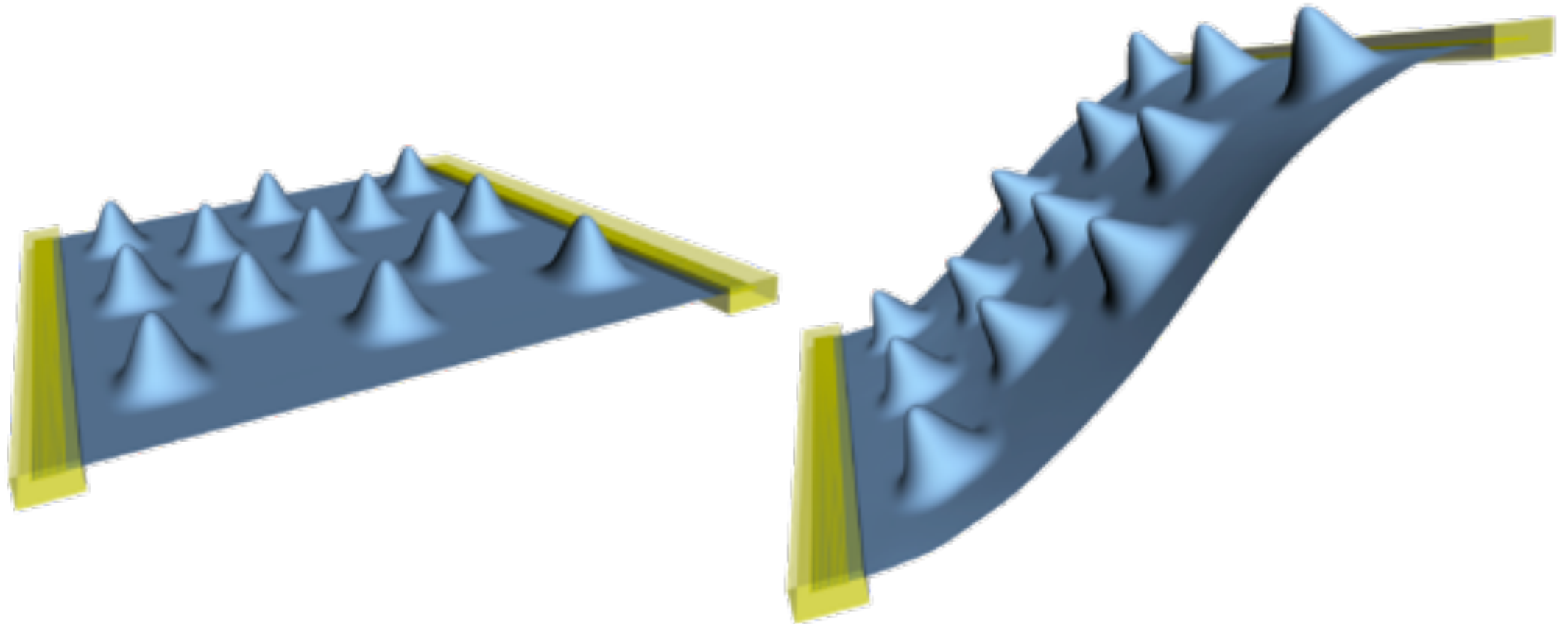
Projects

Geometry Processing Project

Goal

- Small research project

- 1 week for project proposal, **deadline April 2**

  - **choose between 3 options: A,B, or C**

- >1 month for project, **deadline May 07**

- group, size up to 2

- contributes **30%** to the final grade.

- send to zenghuan@usc.edu

Scope

A) For the disciplined

- Deformation Project, we will provide a framework
- You will implement a surface-based linear deformation algorithm (bending minimizing deformation).

B) For the creative [+10 points]

- Imagine an interesting topic around geometry processing or related to your PhD research or something you always wanted to do, and **write a proposal.**
- If it gets approved, you are good to go.

C) For the bad ass [+10 points]

- Implement a Siggraph, SGP, SCA, or Eurographics Paper.
- Geometry processing related of course ;-)

Project Submission

Deliverables for A)

- Source Code, Binary, Data

- Text files describing the project, how to run it.

Deliverables for B) and C)

- Short Presentation will be held May 7th (length TBD)

- Video / Figures

- Documentation (pdf, doc, txt file): 2 or more pages, short paper style, be rigorous and organized, must include at least **abstract**, **methodology**, and **results**.
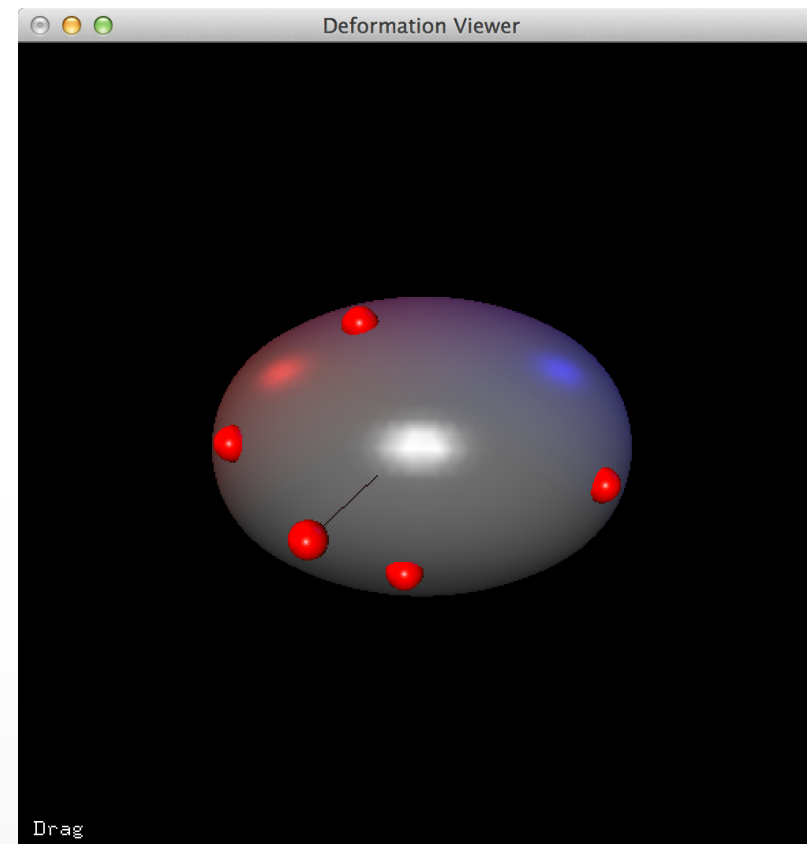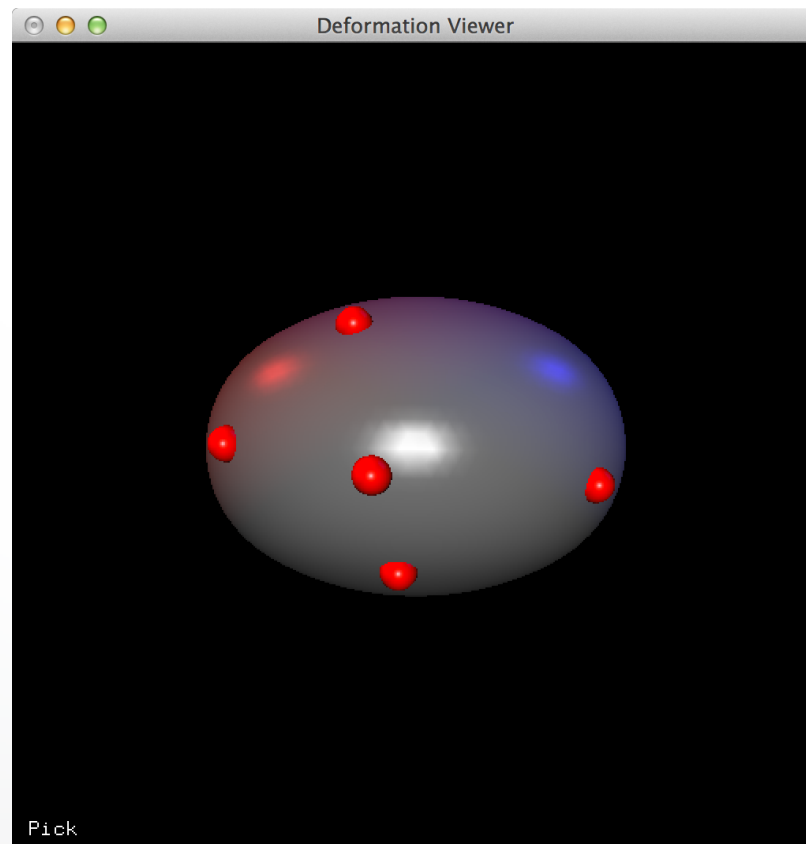
Project Proposal

Structure

- Title
- Motivation
- Goal
- Proposed Method
- References

Format

- authors' names/student IDs
- 1-2 pages
- .doc, .pdf, .txt
- figures

Deformation Framework for A)

- Inherit from MeshViewer with user interface:

  - `'p'` : pick a handle

  - `'d'` : drag a handle (last one with starting code)

  - `'m'` : move the mesh

Deformation Framework for A)

- add handle picking code to
  `DeformationViewer::mouse()`

- add deformation codes to
  `DeformationViewer::deform_mesh()`

- add extra classes and files if needed

- **gmm** is provided to solve linear systems

Some ideas for B) or C)

- **registration**: articulated / deformable motions…

- **shape matching**: RANSAC, spin images, spherical harmonics…

- **Smoothing**: implicit surface fairing…

- **parameterization**: harmonic/conformal mapping…

- **remeshing**: anisotropic, quad mesh…

- **deformation**: As-rigid-as-possible, gradient-based…

- …

http://cs621.hao-li.com

Thanks!