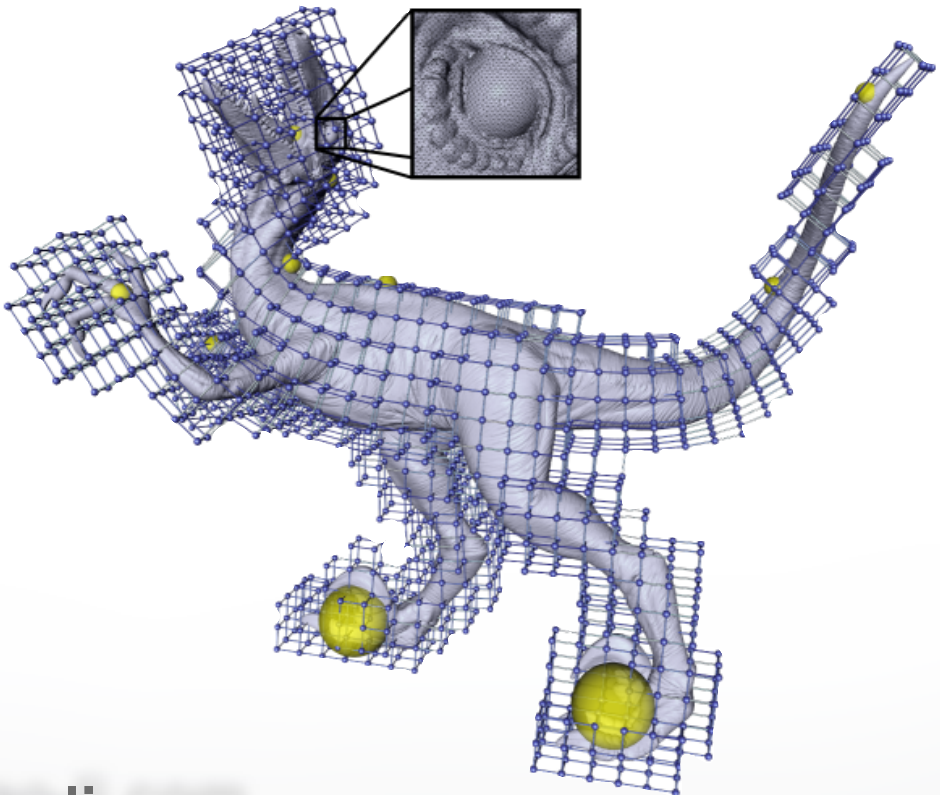
Spring 2014

CSCI 599: **Digital Geometry Processing**

11.2 **Space Deformation**

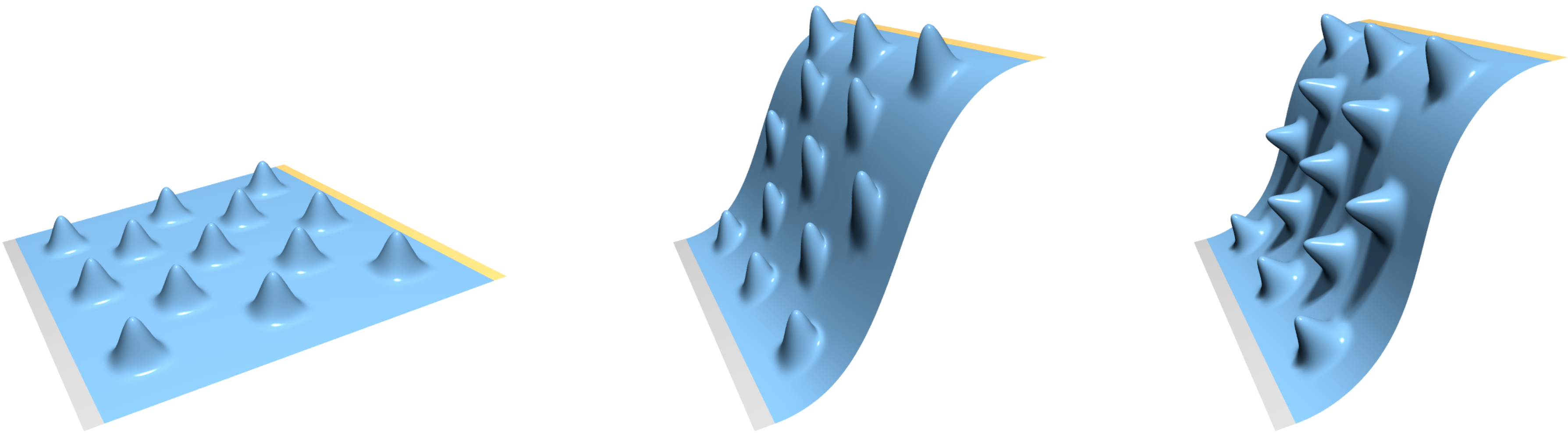


Hao Li

http://cs599.hao-li.com

Hoooray!

Last Time



Surface Deformations

Space Deformation

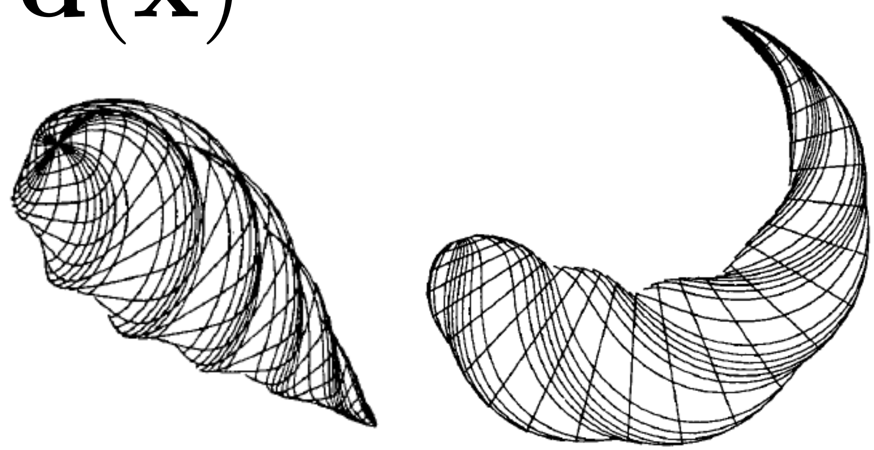
- Displacement function defined on the ambient space

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

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

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

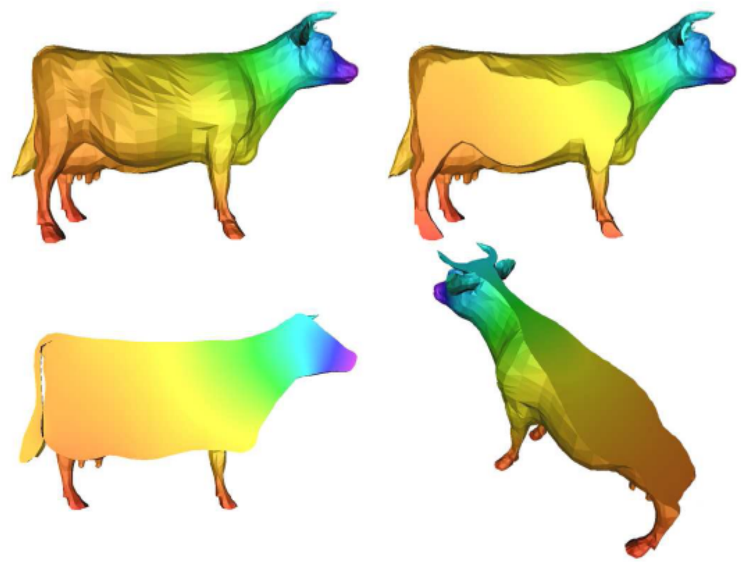
Twist warp
Global and local deformation of solids
[A. Barr, SIGGRAPH 84]



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 $B_i(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}$

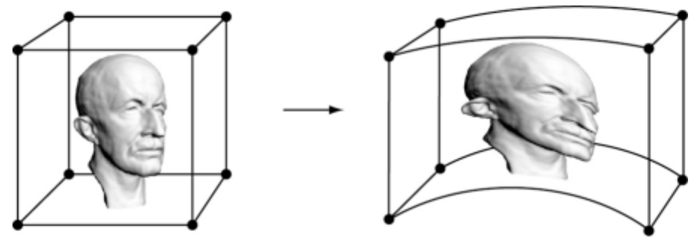
$$\mathbf{d}(\mathbf{x}) = \sum_{i=1}^{k} \mathbf{d}_i \, B_i(\mathbf{x})$$



- Basis functions should be smooth for aesthetic results

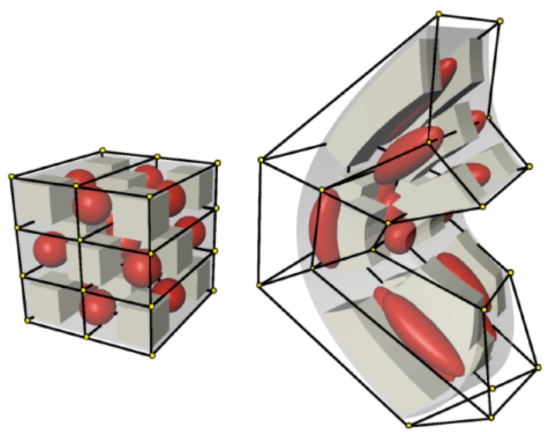
Freeform Deformation

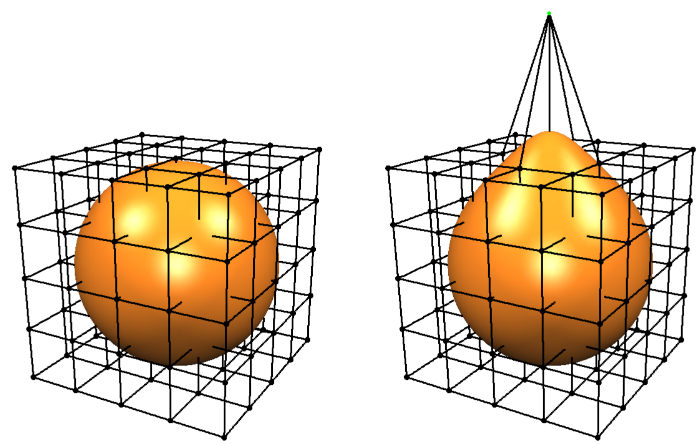
[Sederberg & Parry 86]



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

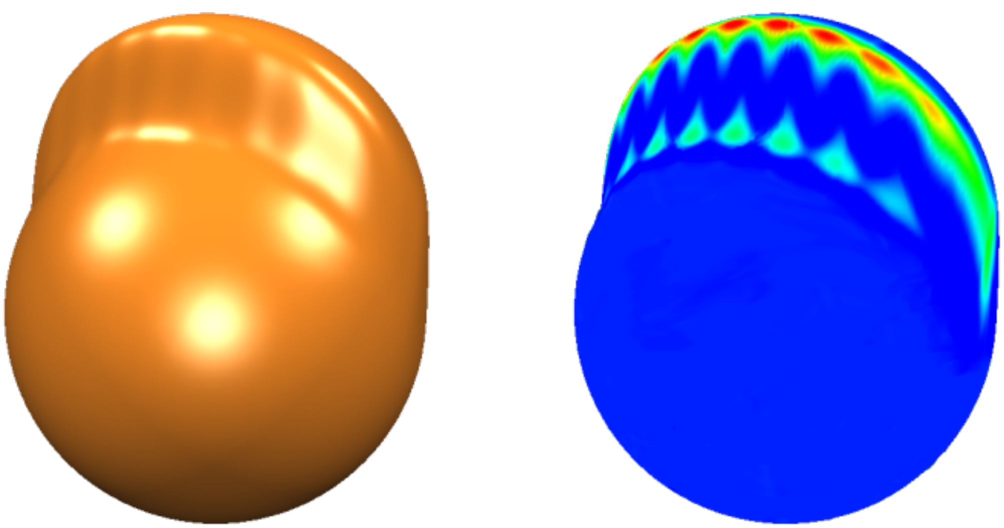
$$\mathbf{d}(x, y, z) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} \mathbf{d}_{ijk} N_i(x) N_j(y) N_k(z)$$





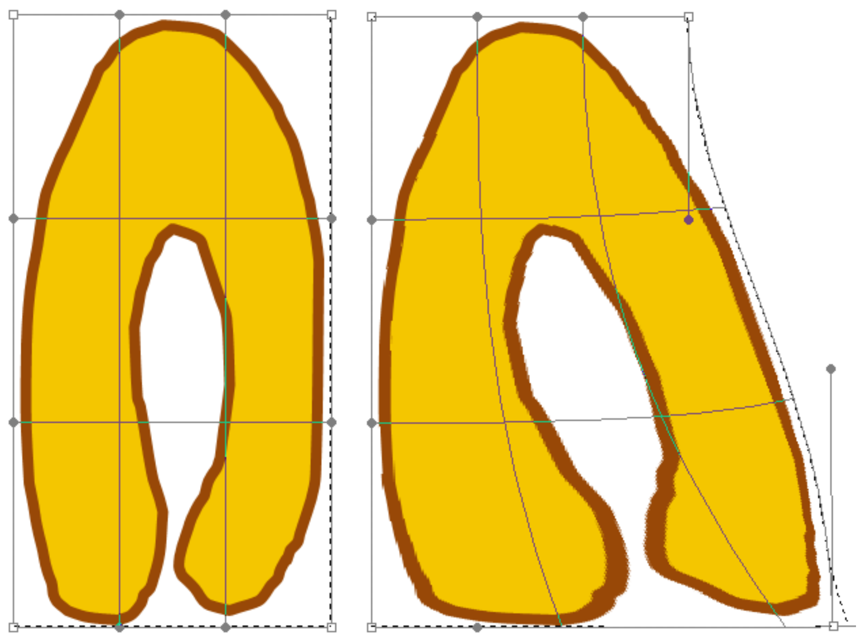
Freeform Deformation
[Sederberg & Parry 86]

- Aliasing artifacts
- Interpolate deformation constraints?
  - Only in least squares sense



Limitations of Lattices as Control Objects

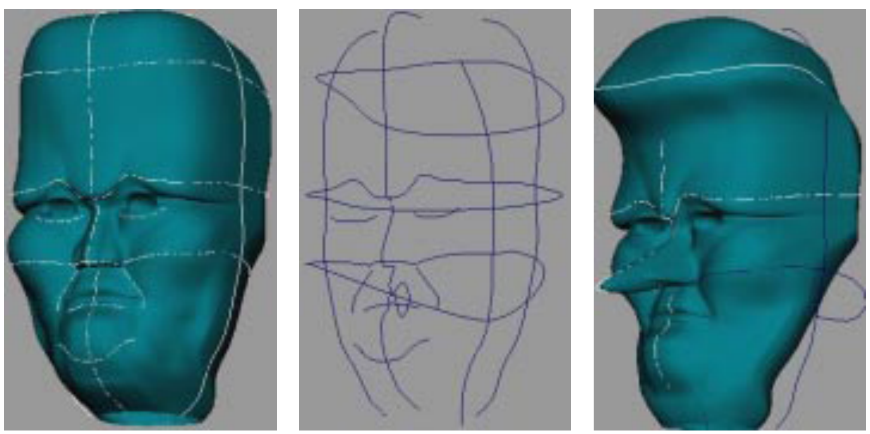
- Difficult to manipulate
- The control object is not related to the shape of the edited object
- Parts of the shape in close Euclidean distance always deform similarly, even if geodesically far

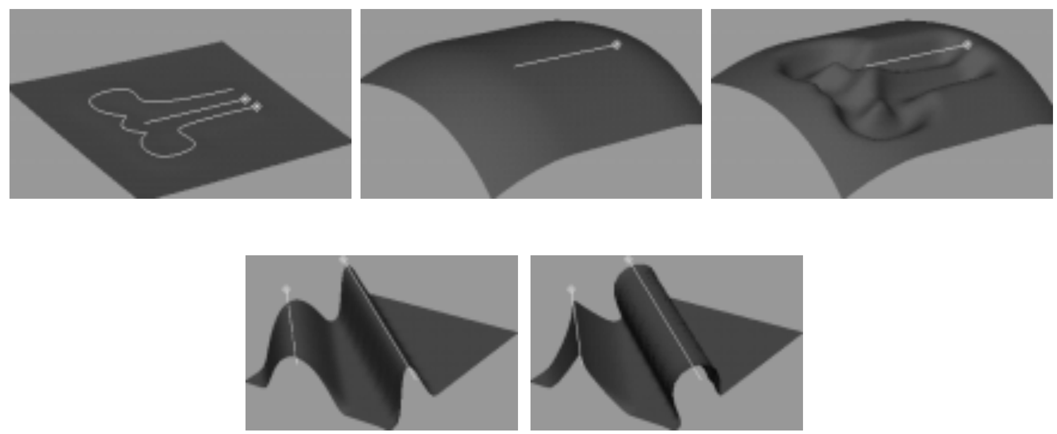


Wires

[Singh & Fiume 98]

- Control objects are arbitrary space curves
- Can place curves along meaningful features of the edited object
- Smooth deformations around the curve with decreasing influence



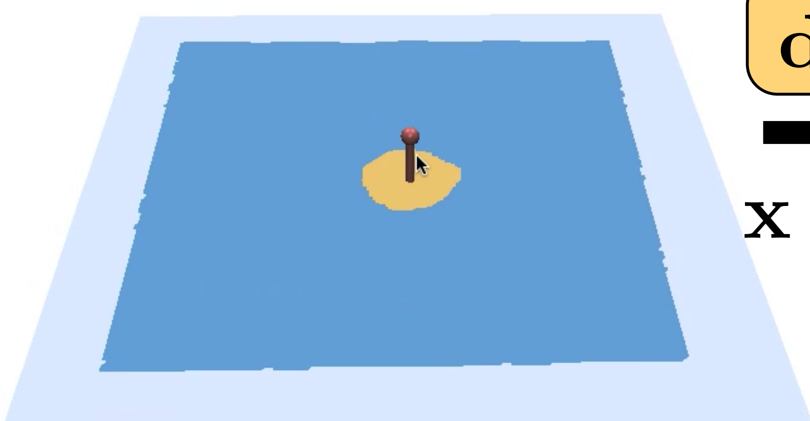


Handle Metaphor

[RBF, Botsch & Kobbelt 05]

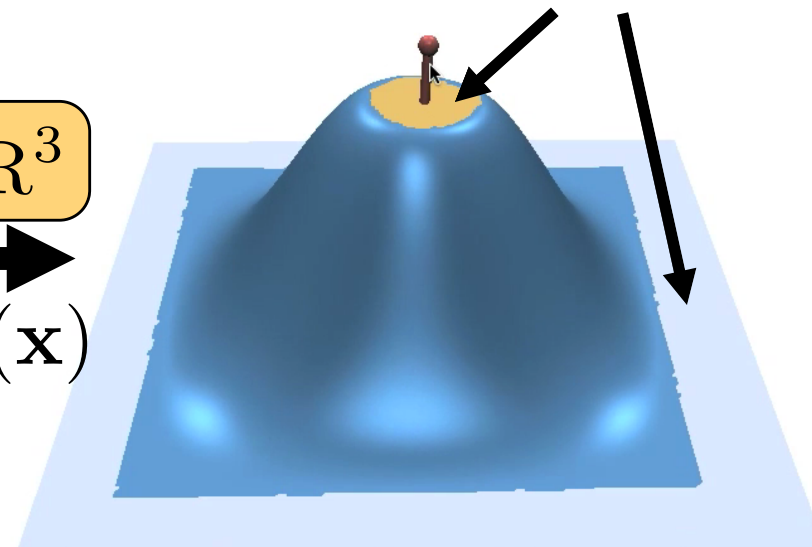
- Wish list for the displacement function $\mathbf{d}(\mathbf{x})$:
  - Interpolate prescribed constraints
  - Smooth, intuitive deformation

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



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

$$\mathbf{x} \rightarrow \mathbf{x} + \mathbf{d}(\mathbf{x})$$



Volumetric Energy Minimization

[RBF, Botsch & Kobbelt 05]

- 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

[RBF, Botsch & Kobbelt 05]

- 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

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

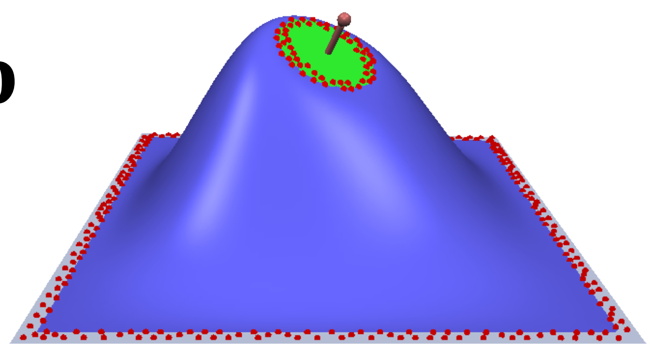
Radial Basis Functions

[RBF, Botsch & Kobbelt 05]

- 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 $\mathbf{p}$



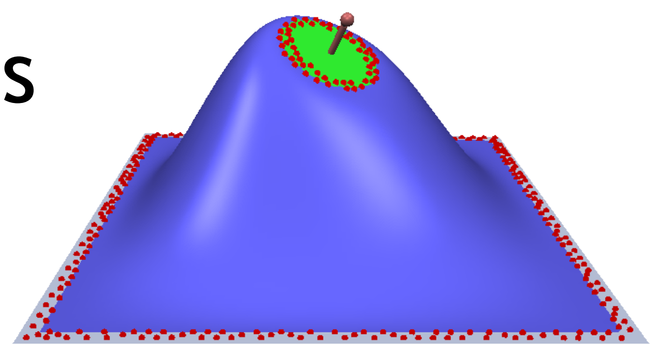
Radial Basis Functions

[RBF, Botsch & Kobbelt 05]

- Represent deformation by RBFs

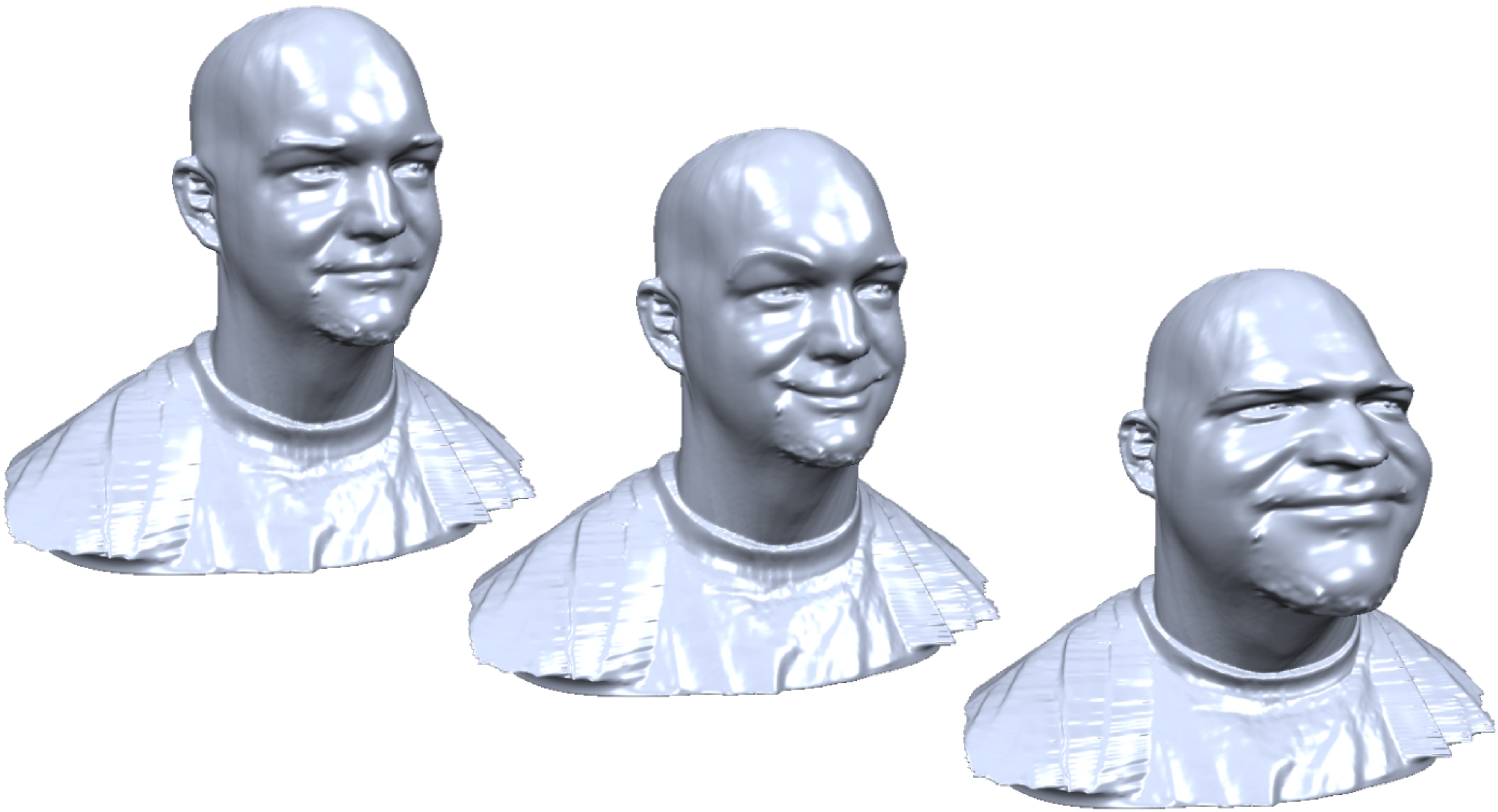
$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \, \varphi(\|\mathbf{c}_j - \mathbf{x}\|) + \mathbf{p}(\mathbf{x})$$

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



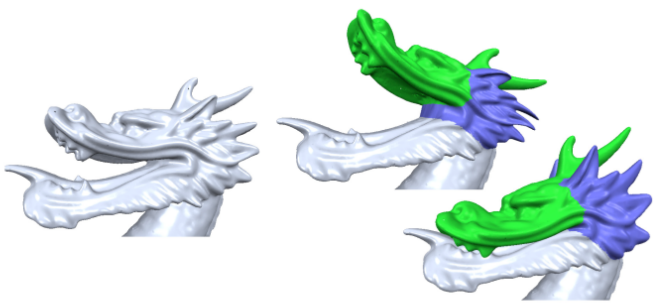
Local & Global Deformations

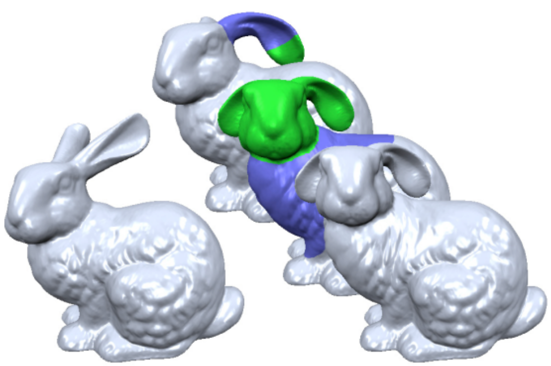
[RBF, Botsch & Kobbelt 05]



Local & Global Deformations

[RBF, Botsch & Kobbelt 05]

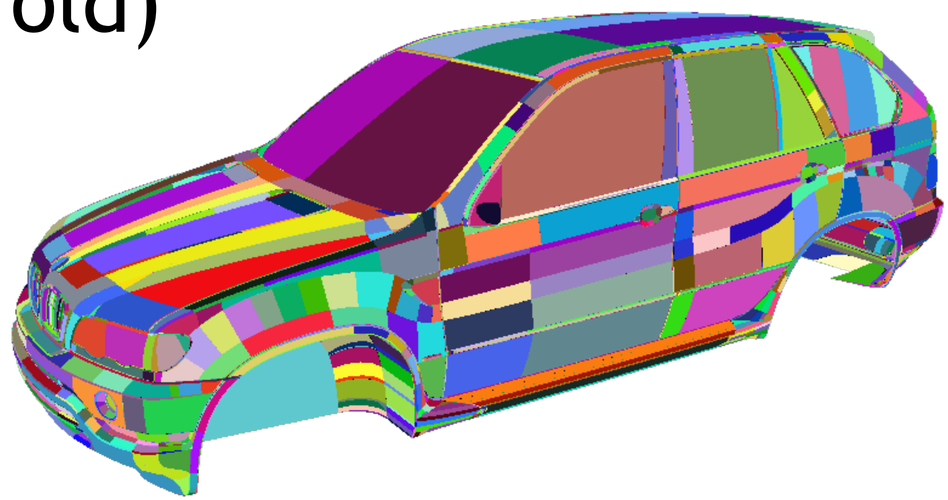




1M vertices
movie

Space Deformation

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

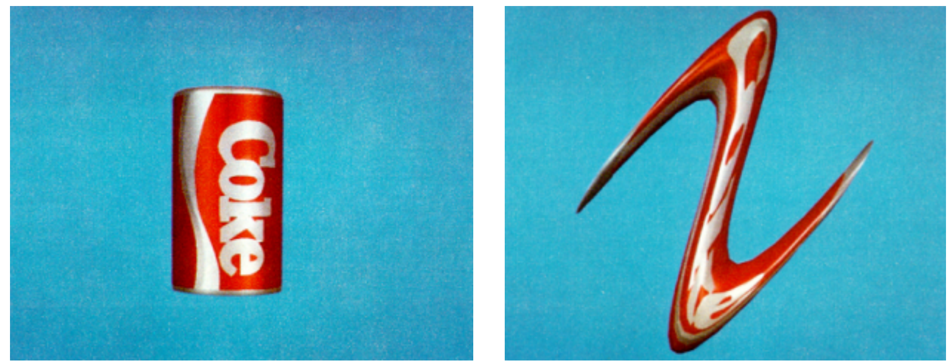


- 3M triangles
- 10k components
- Not oriented
- Not manifold

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

Space Deformation

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



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

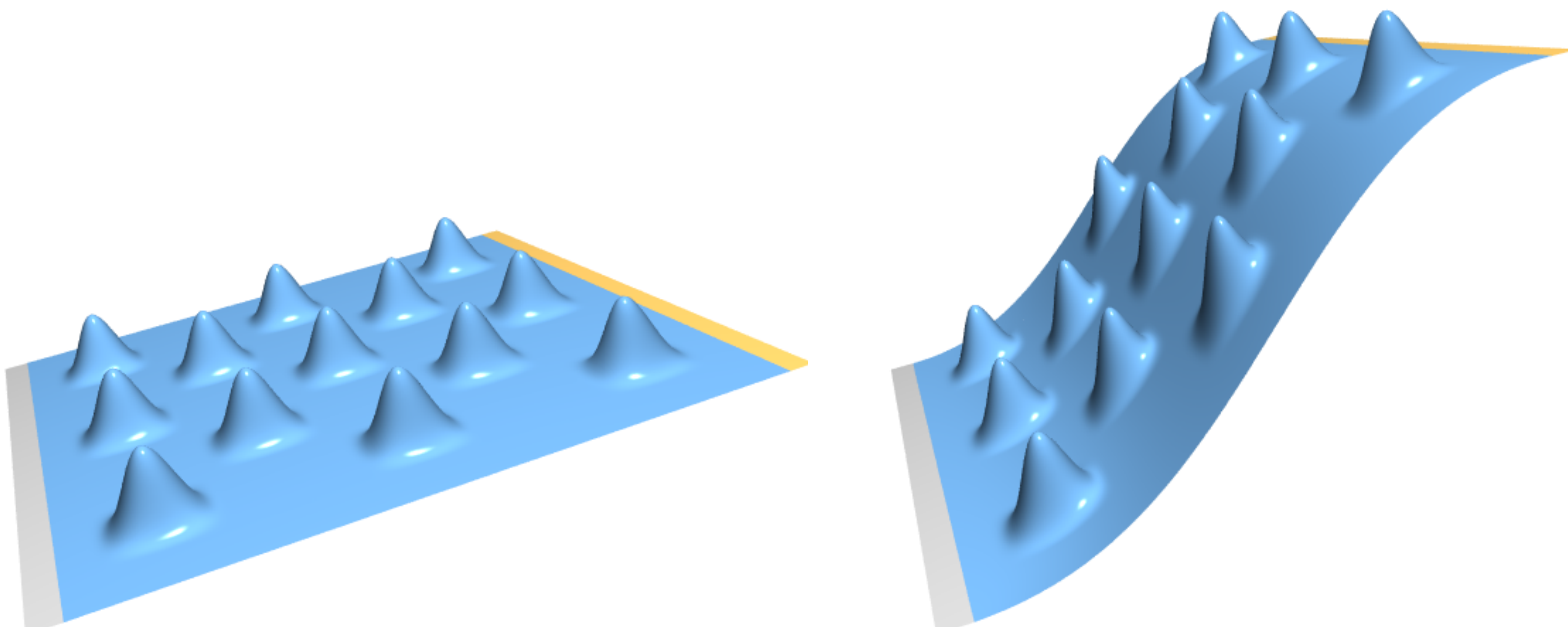
then the Jacobian is the determinant

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

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

Space Deformation

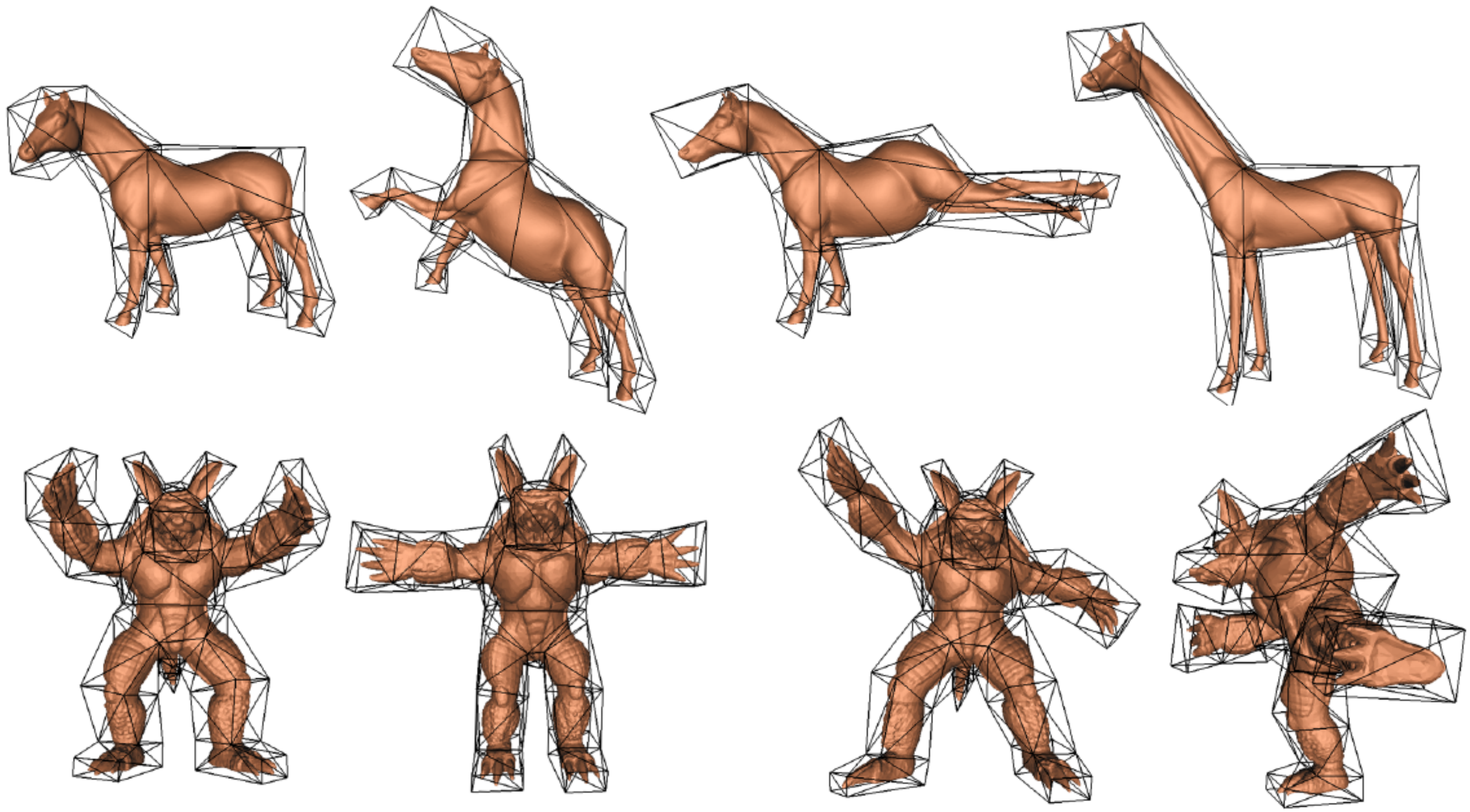
- The deformation is only loosely aware of the shape that is being edited
- Small Euclidean distance → similar deformation
- Local surface detail may be distorted



Cage-Based Deformation

[Ju et al. 05]

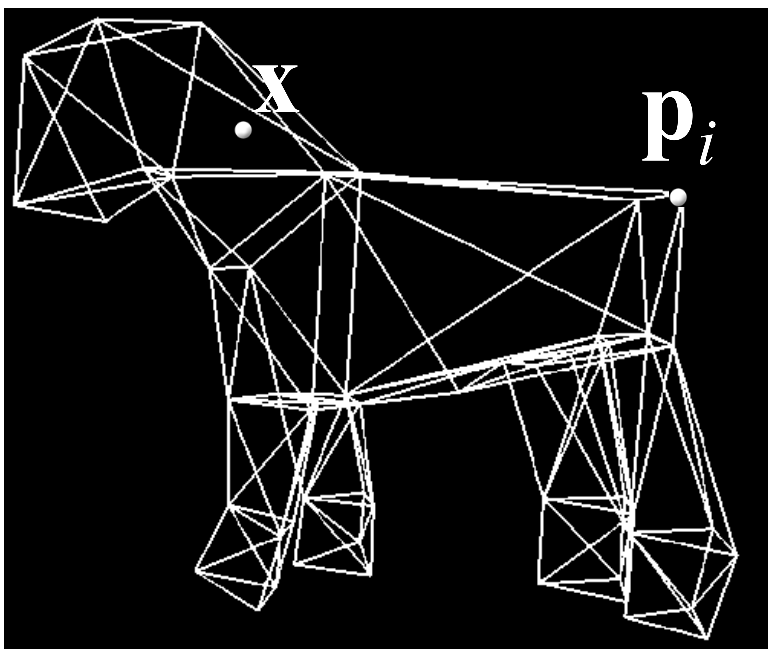
- Cage = crude version of the input shape
- Polytope (not a lattice)



Cage-Based Deformation

[Ju et al. 05]

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

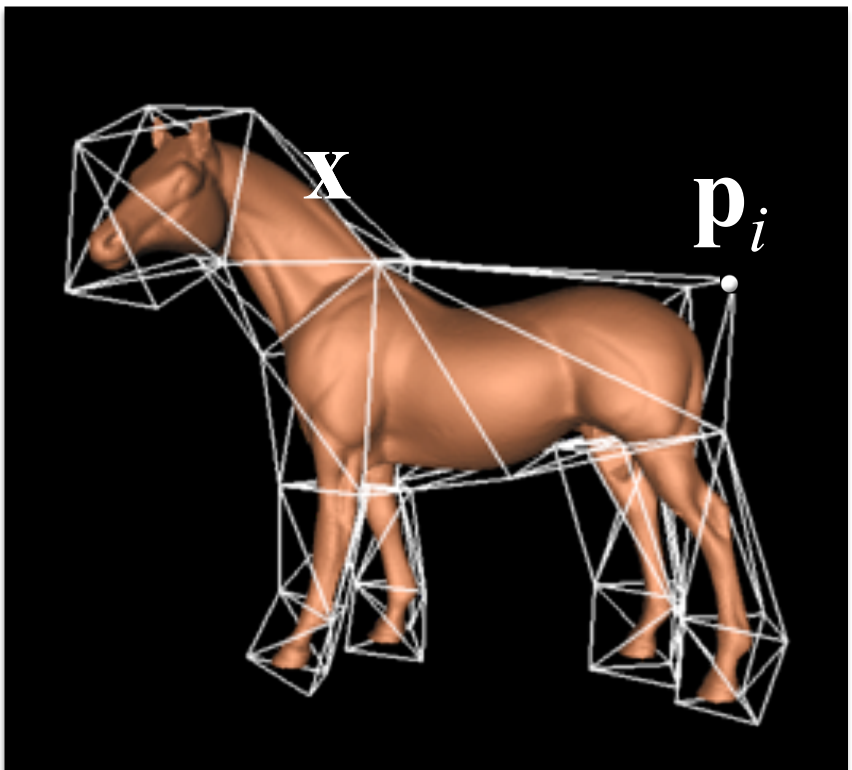


$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i$$

Cage-Based Deformation

[Ju et al. 05]

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

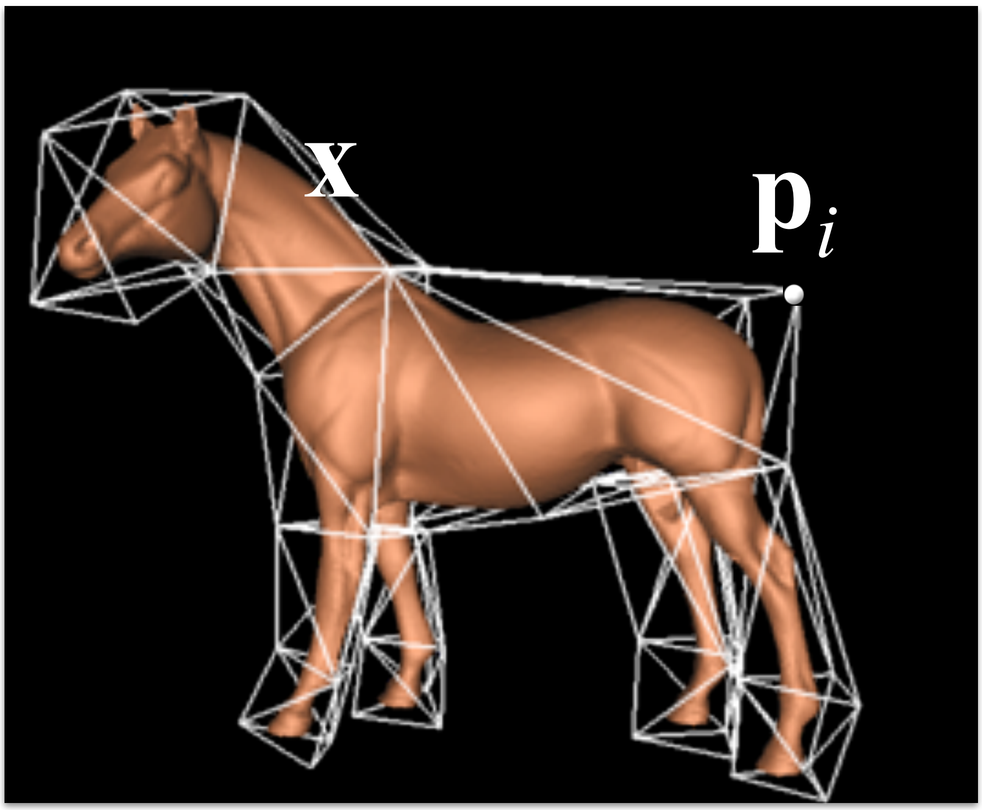


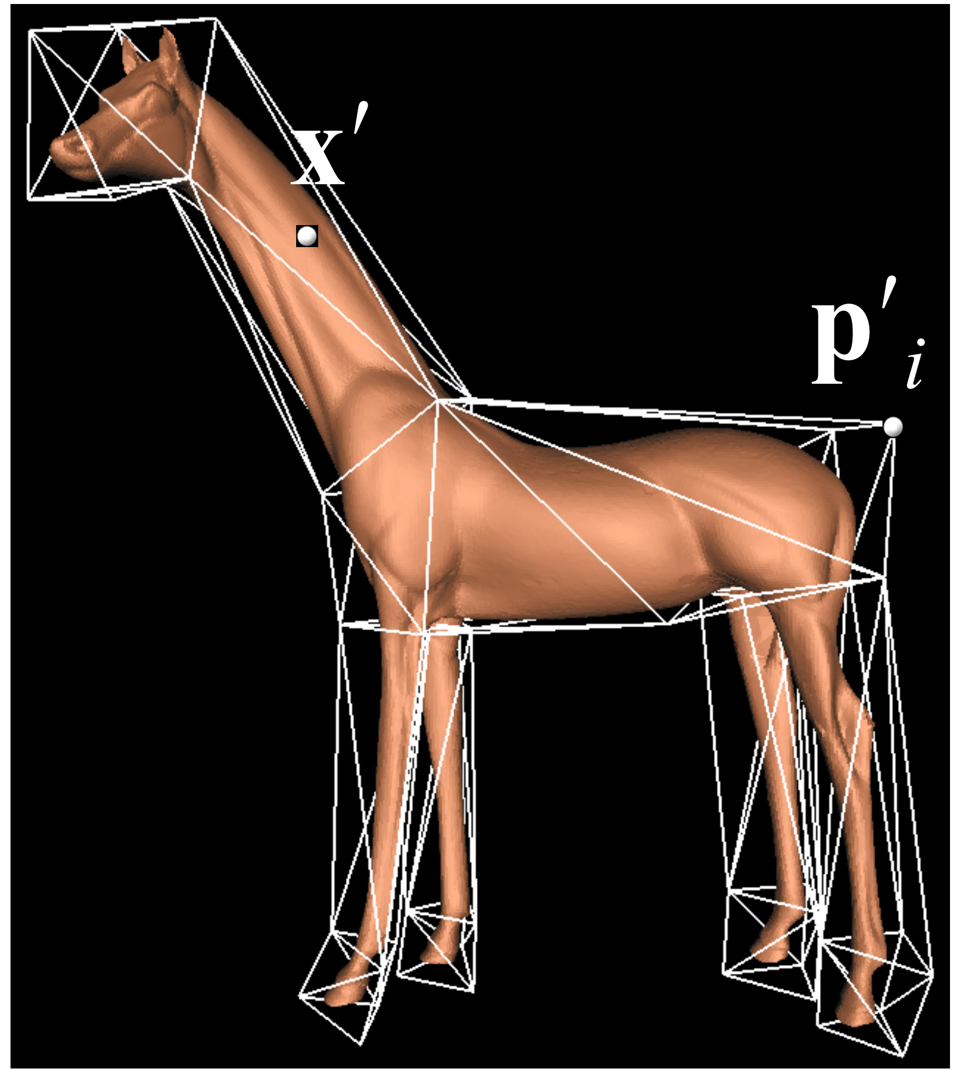
$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i$$

Cage-Based Deformation

[Ju et al. 05]

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



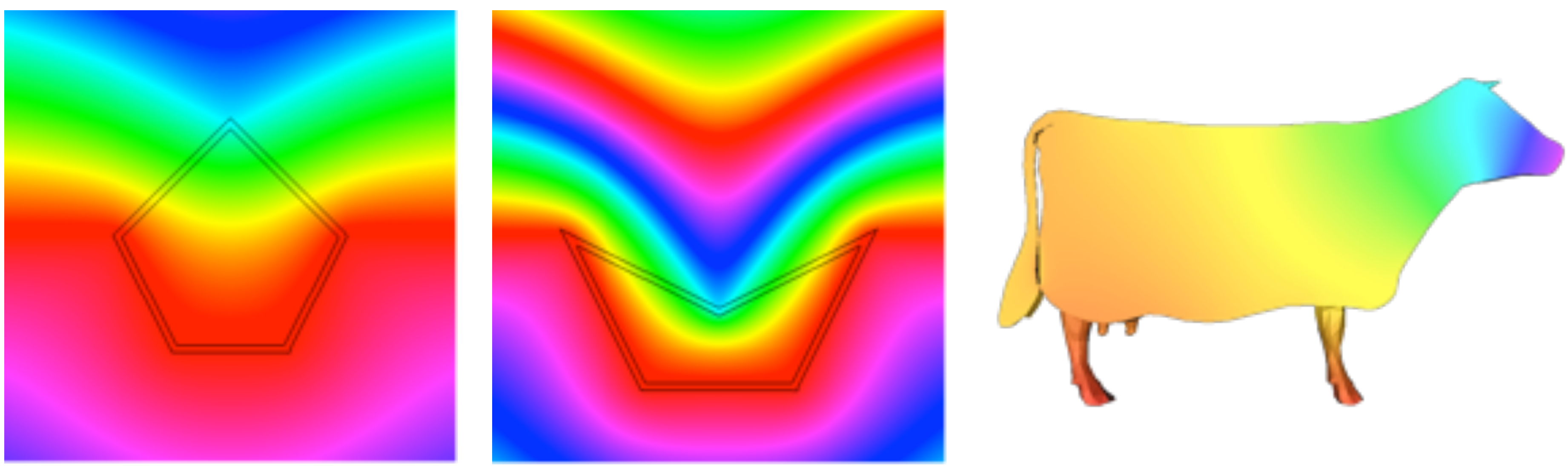


Generalized Barycentric Coordinates

- Lagrange property: $$w_i(\mathbf{p}_j) = \delta_{ij}$$
- Reproduction: $$\forall \mathbf{x},\ \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i = \mathbf{x}$$
- Partition of unity: $$\forall \mathbf{x},\ \sum_{i=1}^{k} w_i(\mathbf{x}) = 1$$

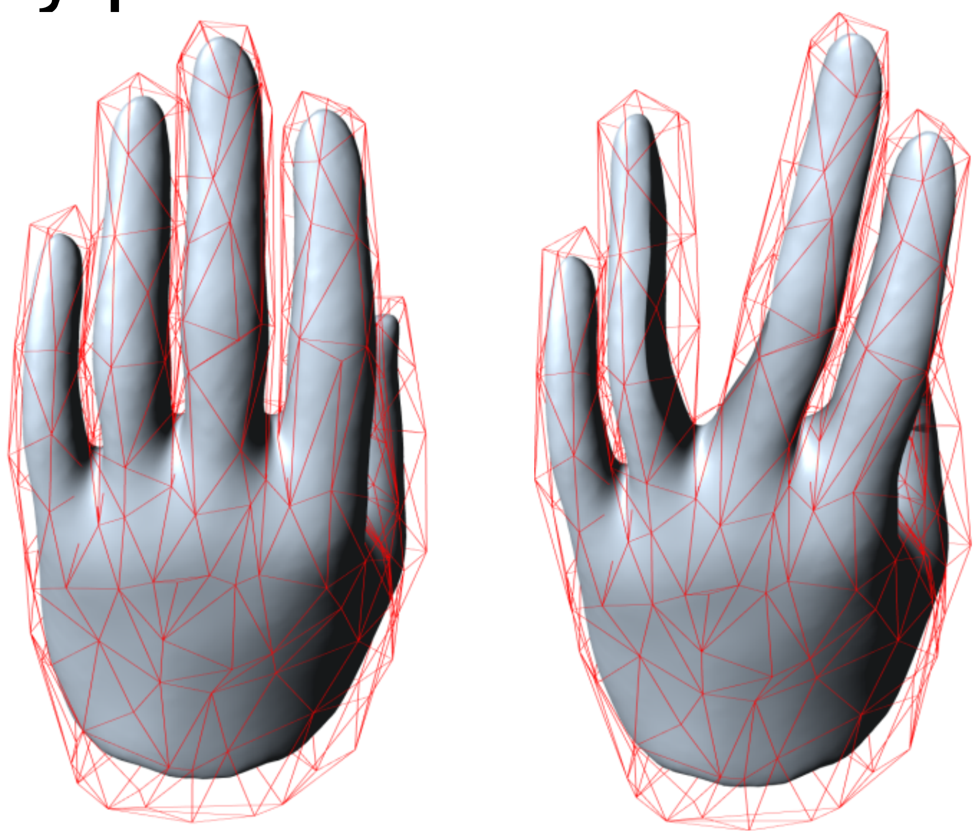
Coordinate Functions

- Mean-value coordinates [Floater 2003, Ju et al. 2005]
  - Generalization of barycentric coordinates
  - Closed-form solution for $w_i(\mathbf{x})$



Coordinate Functions

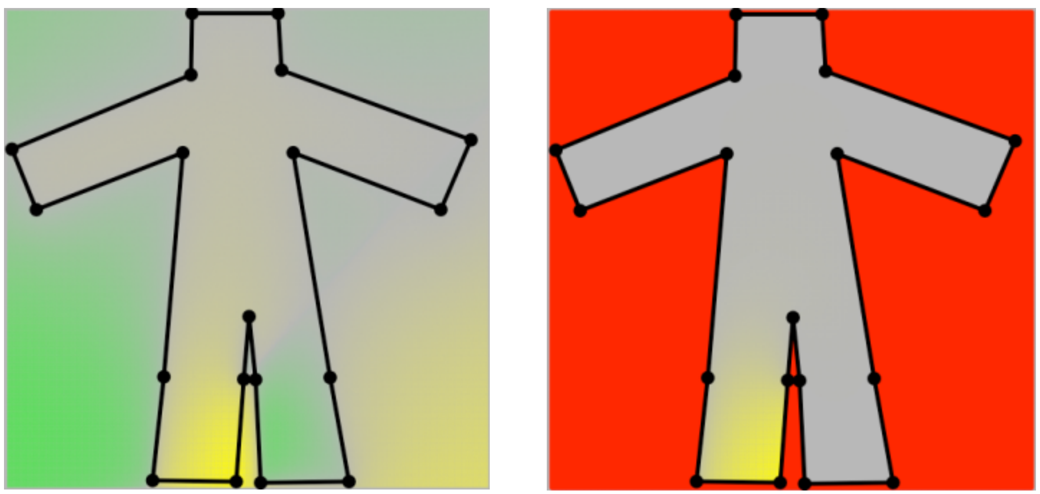
- Mean-value coordinates [Floater, Ju et al. 2005]
  - Not necessarily positive on non-convex domains



Coordinate Functions

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

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

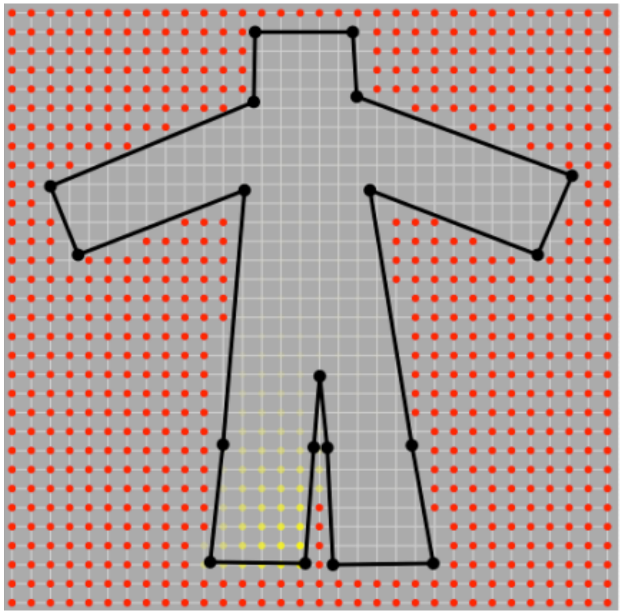


MVC HC

Coordinate Functions

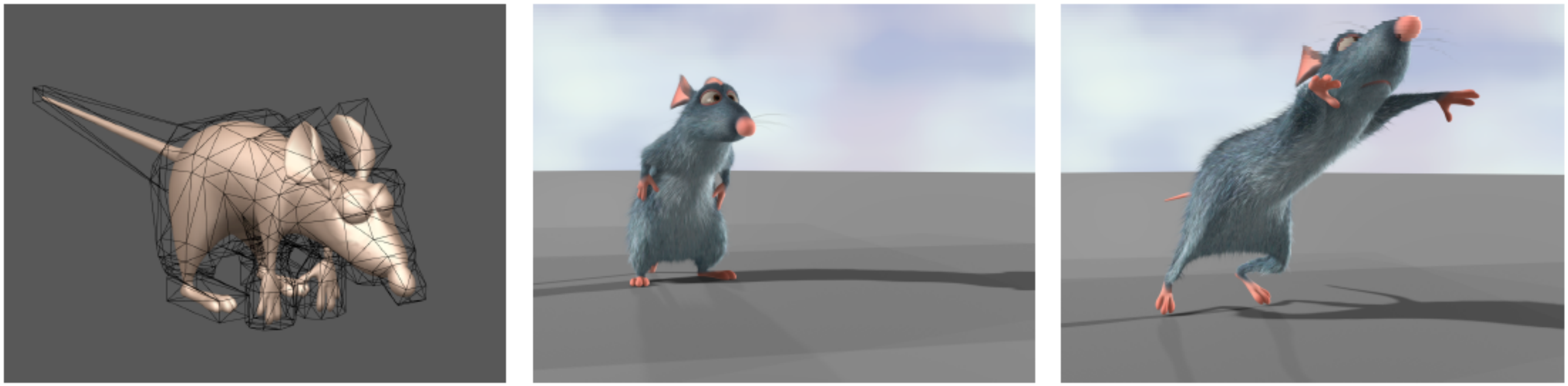
- Harmonic coordinates (Joshi et al. 2007)
  - Harmonic functions $h_i(\mathbf{x})$ for each cage vertex $\mathbf{p}_i$
  - Solve $\Delta\, h = 0$

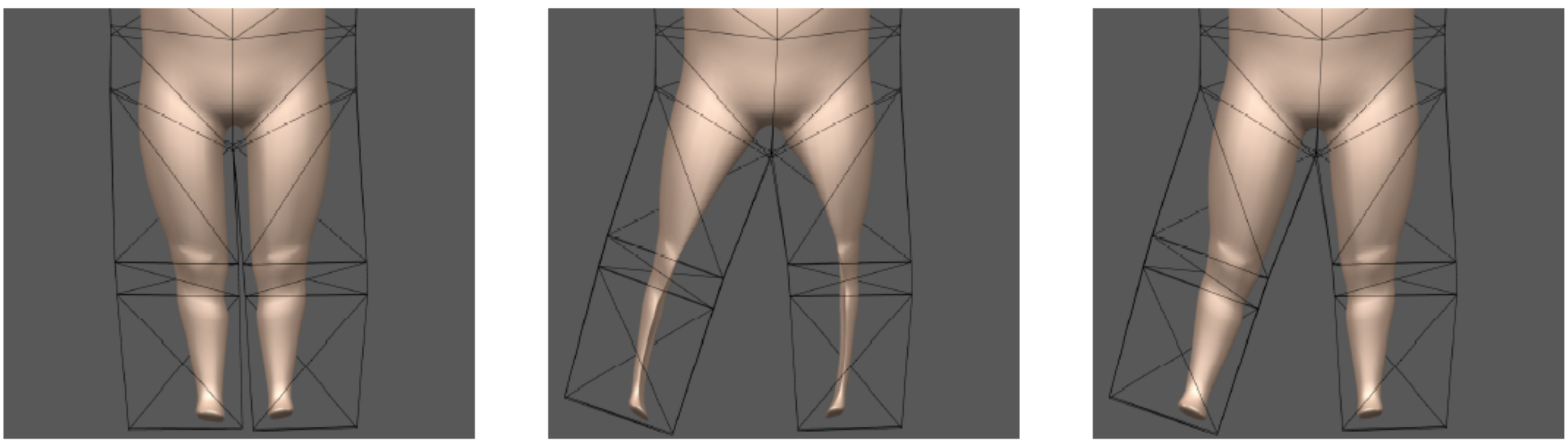
  subject to: h_i linear on the boundary s.t. $h_i(\mathbf{p}_j) = \delta_{ij}$
- Volumetric Laplace equation
- Discretization, no closed-form



Coordinate Functions

- Harmonic coordinates (Joshi et al. 2007)

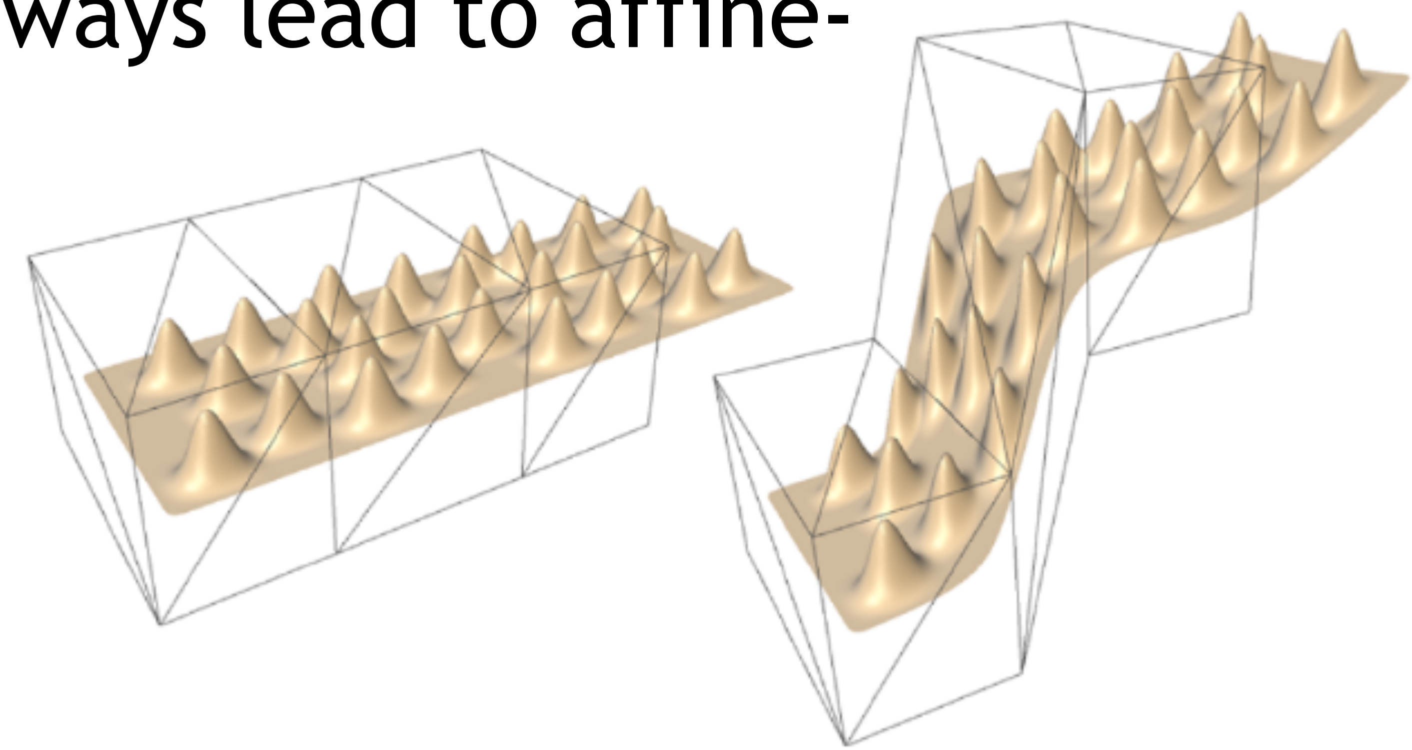




MVC HC

Coordinate Functions

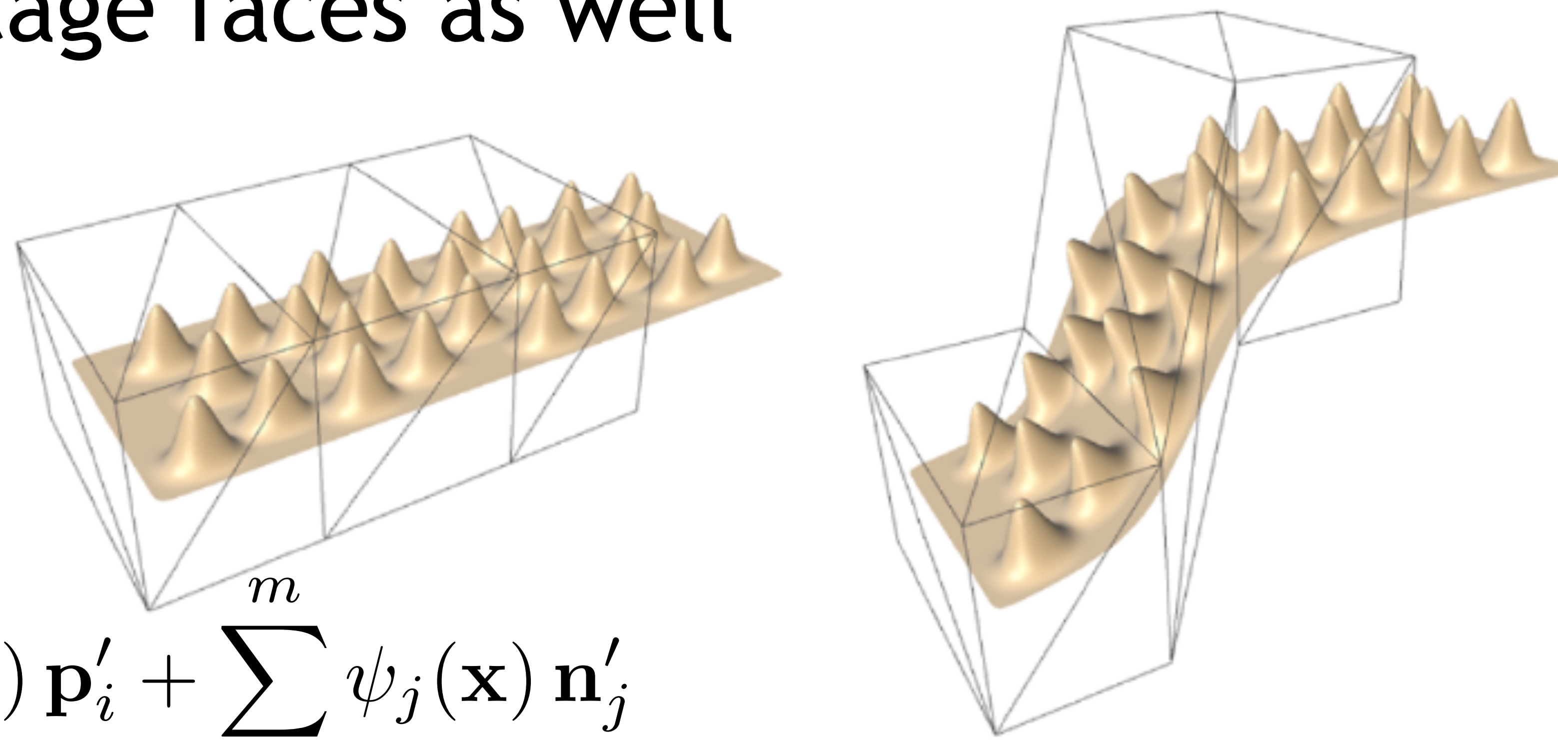
- Green coordinates (Lipman et al. 2008)
- Observation: previous vertex-based basis functions always lead to affine-invariance!



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

Coordinate Functions

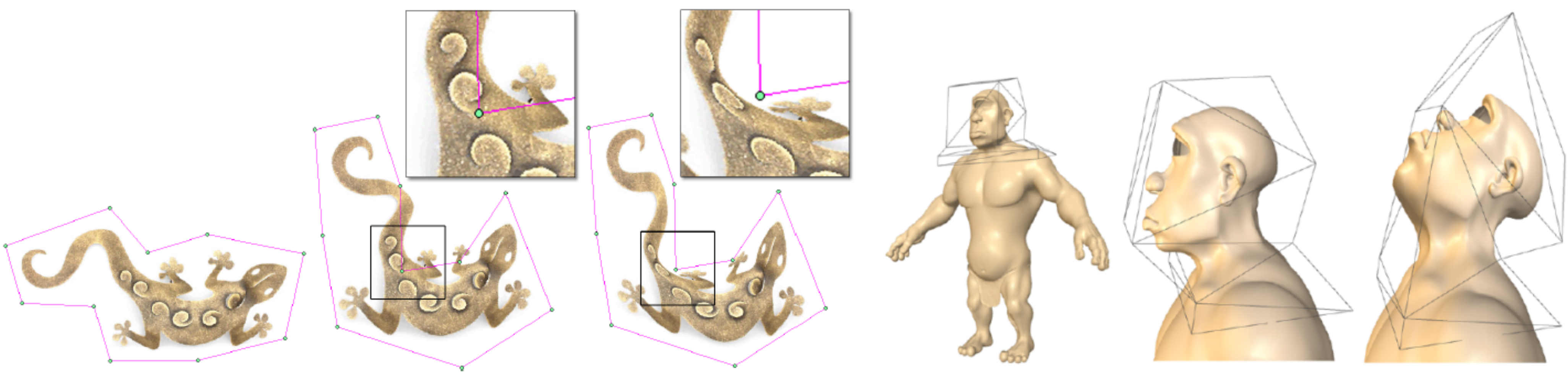
- Green coordinates (Lipman et al. 2008)
- Correction: Make the coordinates depend on the cage faces as well



$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}'_i + \sum_{j=1}^{m} \psi_j(\mathbf{x})\,\mathbf{n}'_j$$

Coordinate Functions

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

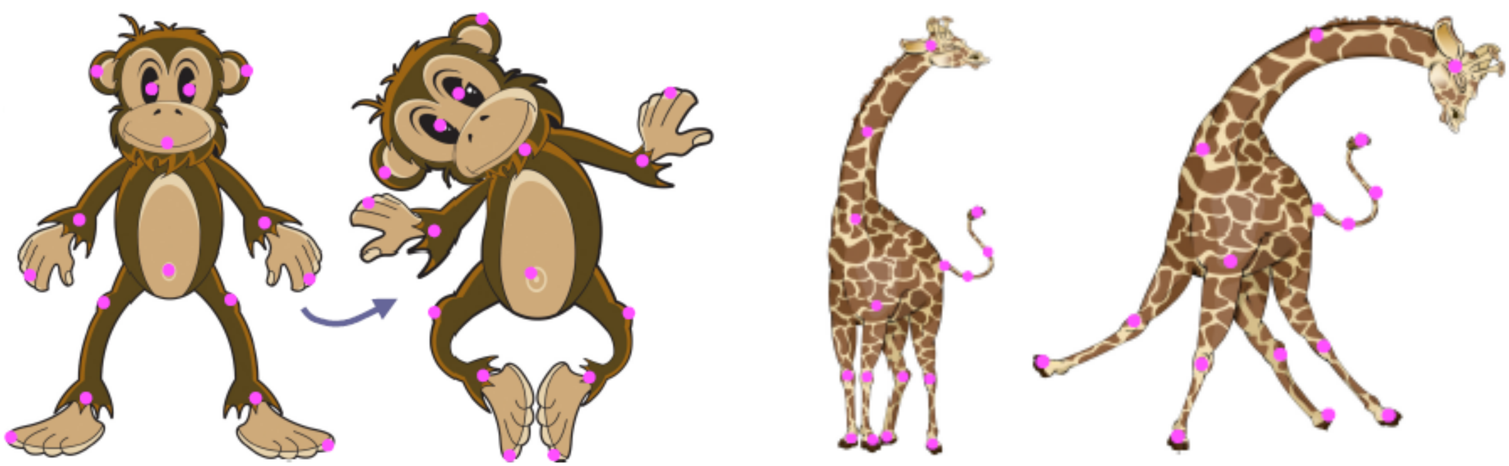


GC MVC GC

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



Cage-Based Methods: Summary

Pros:

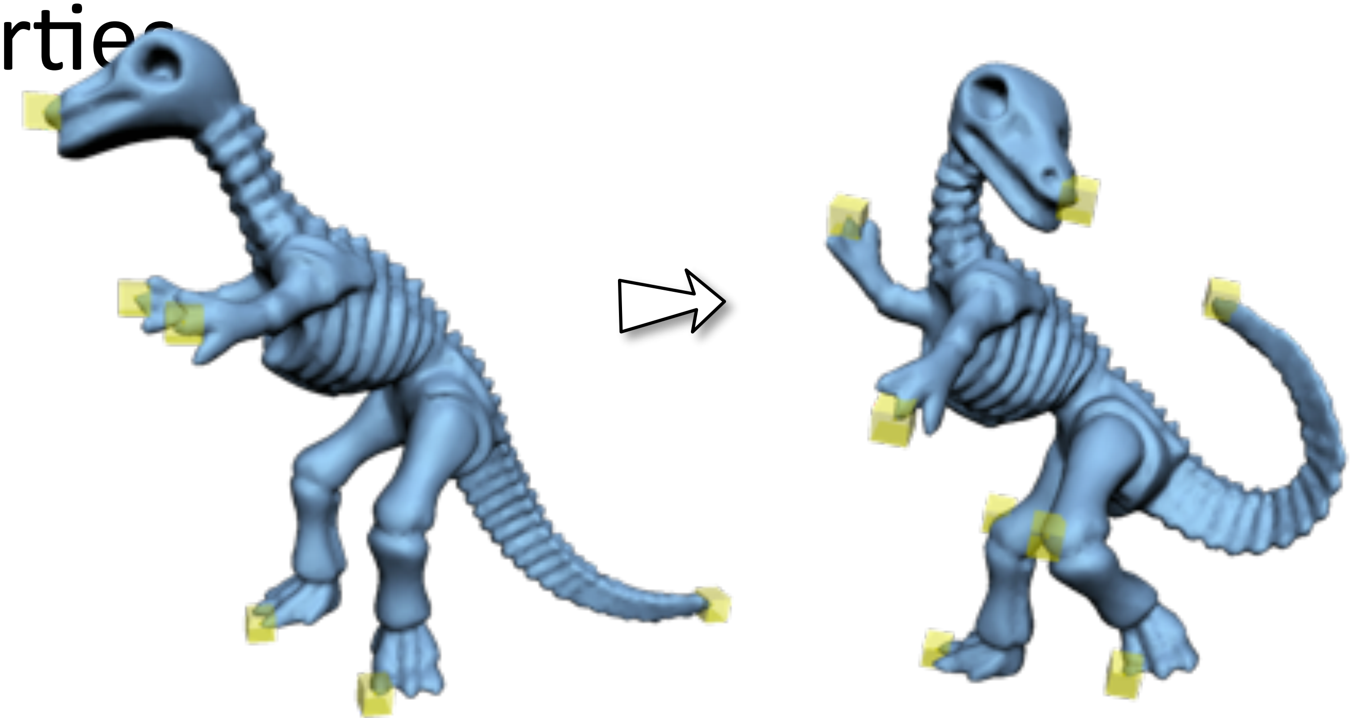
- Nice control over volume
  - Squish/stretch

Cons:

- Hard to control details of embedded surface

Non-Linear Space Deformation

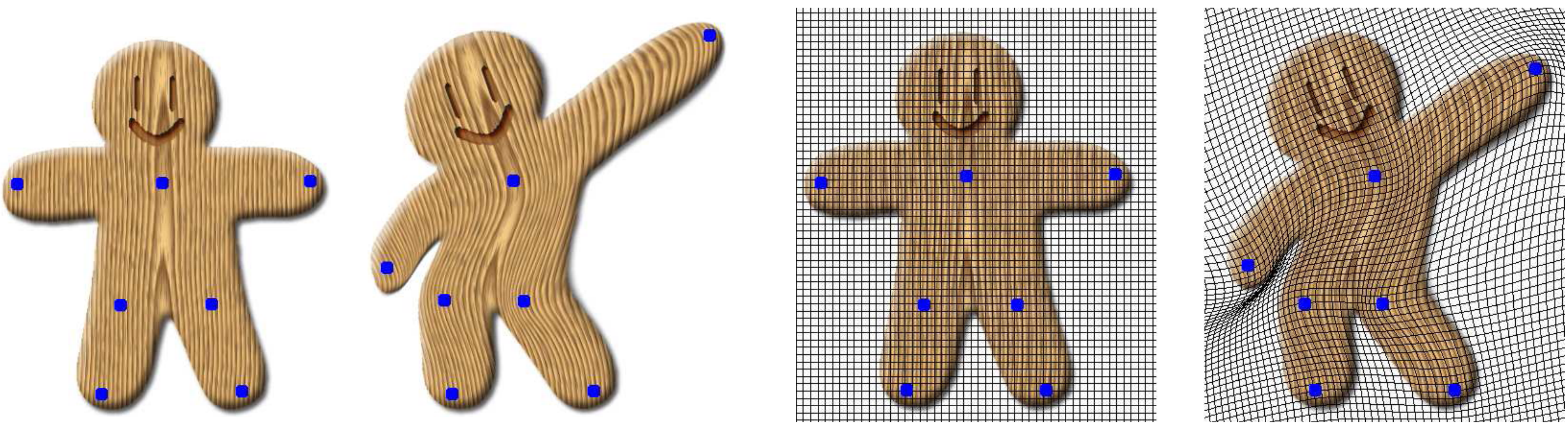
- Involve nonlinear optimization
- Enjoy the advantages of space warps
- Additionally, have shape-preserving properties



As-Rigid-As-Possible Deformation

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)

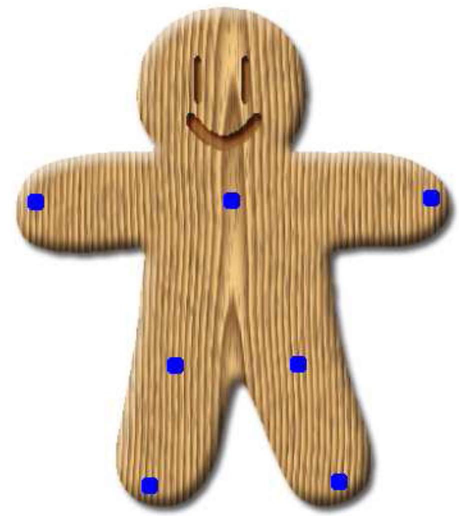


As-Rigid-As-Possible Deformation

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

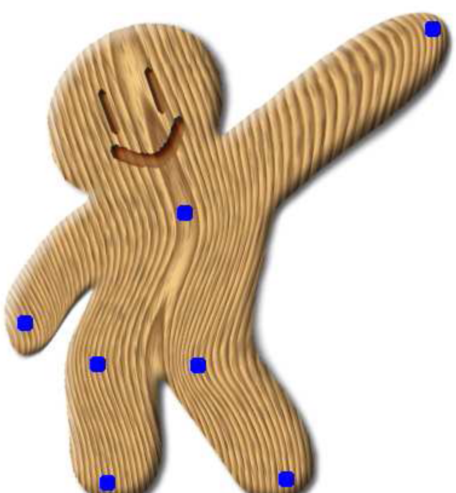
- Attach an affine transformation to each point $\mathbf{x} \in \mathrm{R}^3$:

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



- The space warp:

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



As-Rigid-As-Possible Deformation

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

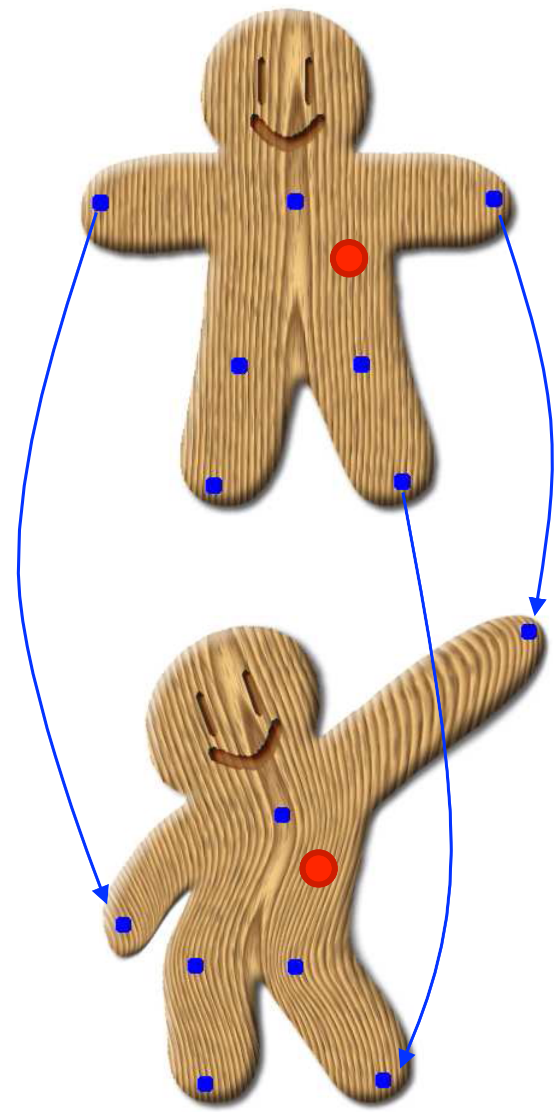
- Handles $\mathbf{p}_i$ are displaced to $\mathbf{q}_i$
- The local transformation at $\mathbf{x}$:

$$\mathrm{A}_{\mathbf{x}}(\mathbf{p}) = \mathrm{M}_{\mathbf{x}}\mathbf{p} + \mathbf{t}_{\mathbf{x}} \quad \text{s.t.}$$

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

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

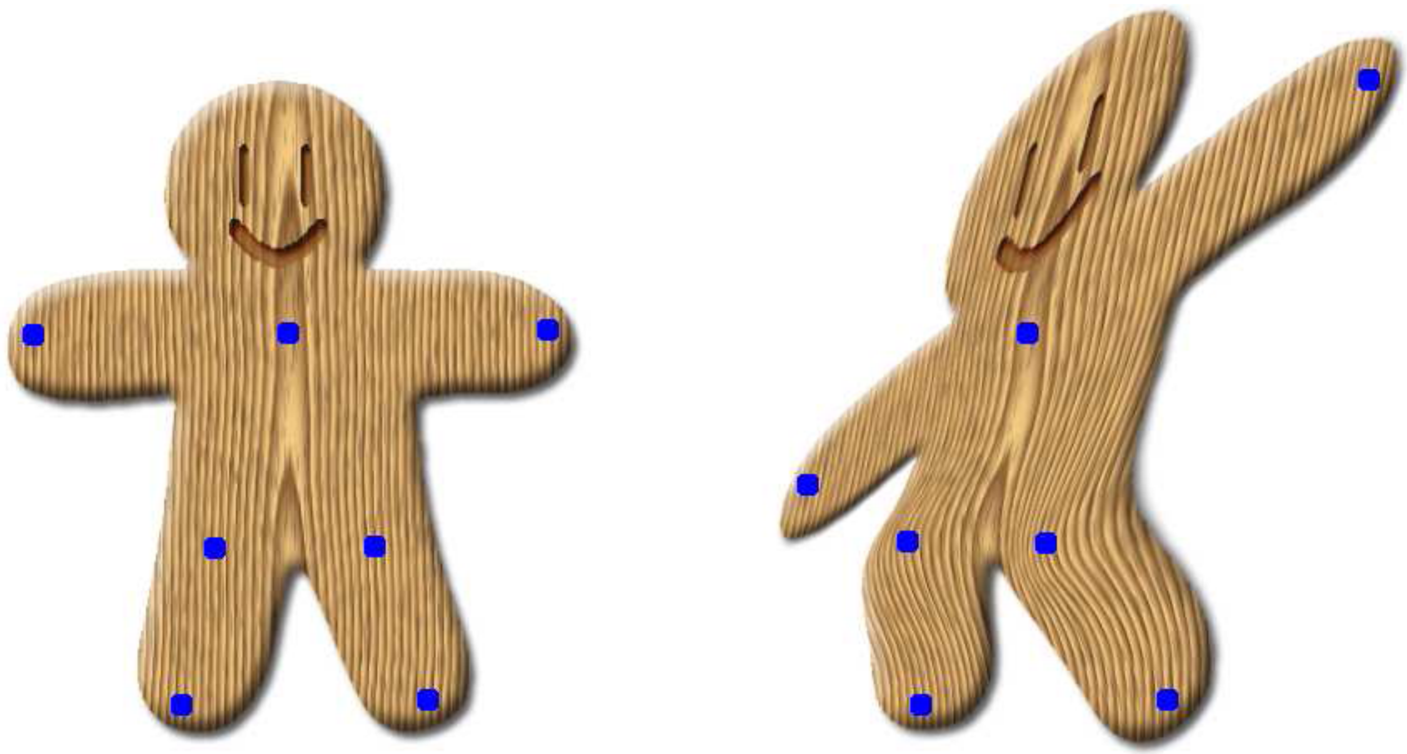
$$w_i(\mathbf{x}) = \|\mathbf{p}_i - \mathbf{x}\|^{-2\alpha}$$



As-Rigid-As-Possible Deformation

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

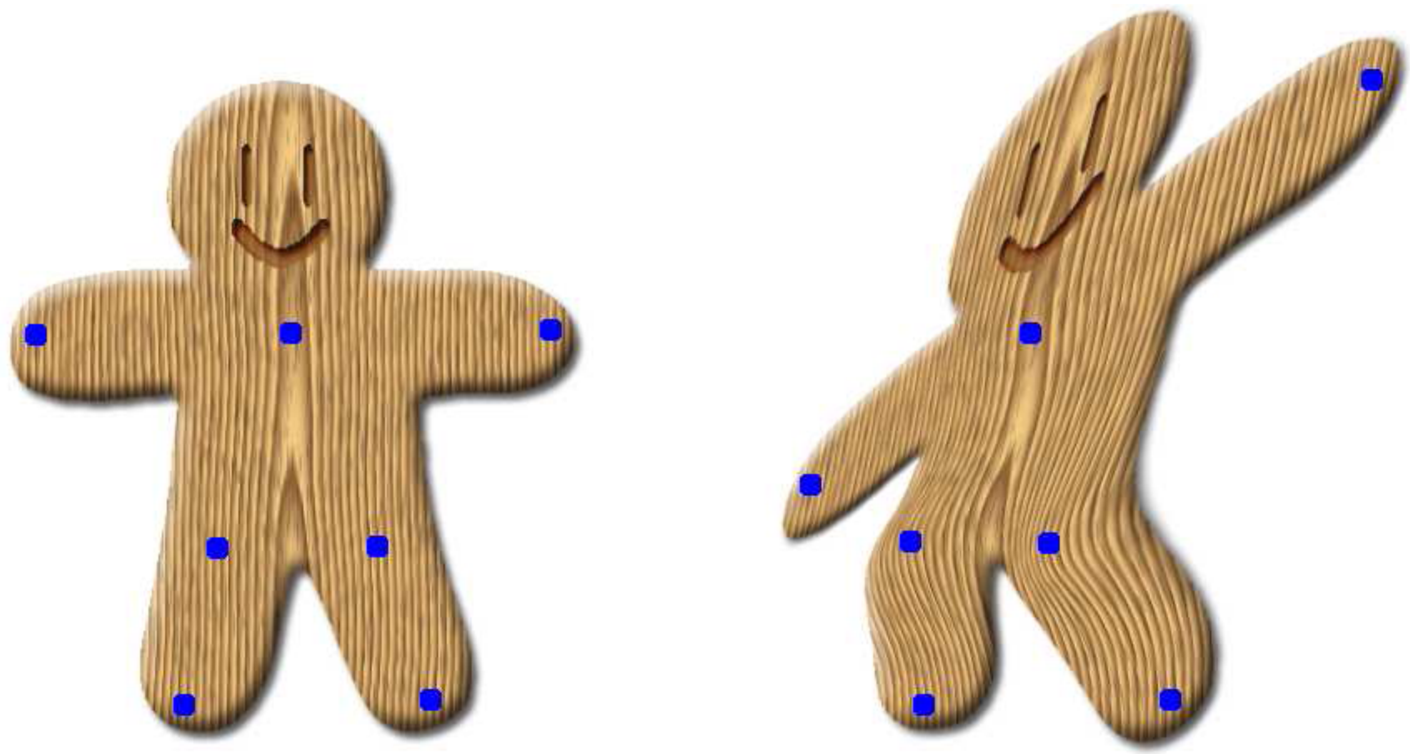
- No additional restriction on $A_{\mathbf{x}}(\cdot)$ – affine local transformations



As-Rigid-As-Possible Deformation

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

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

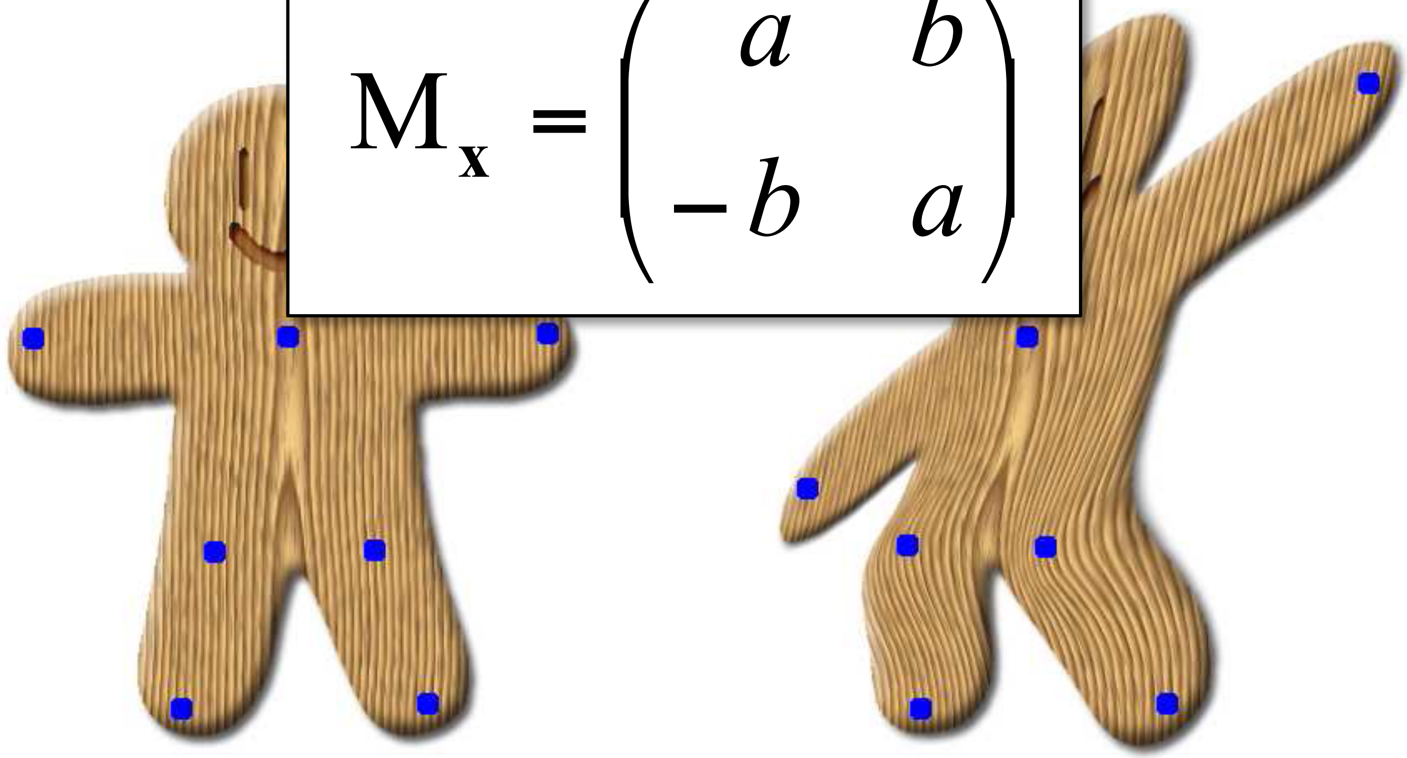


As-Rigid-As-Possible Deformation

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

- Restrict $\mathrm{A}_{\mathbf{x}}(\cdot)$ to similarity

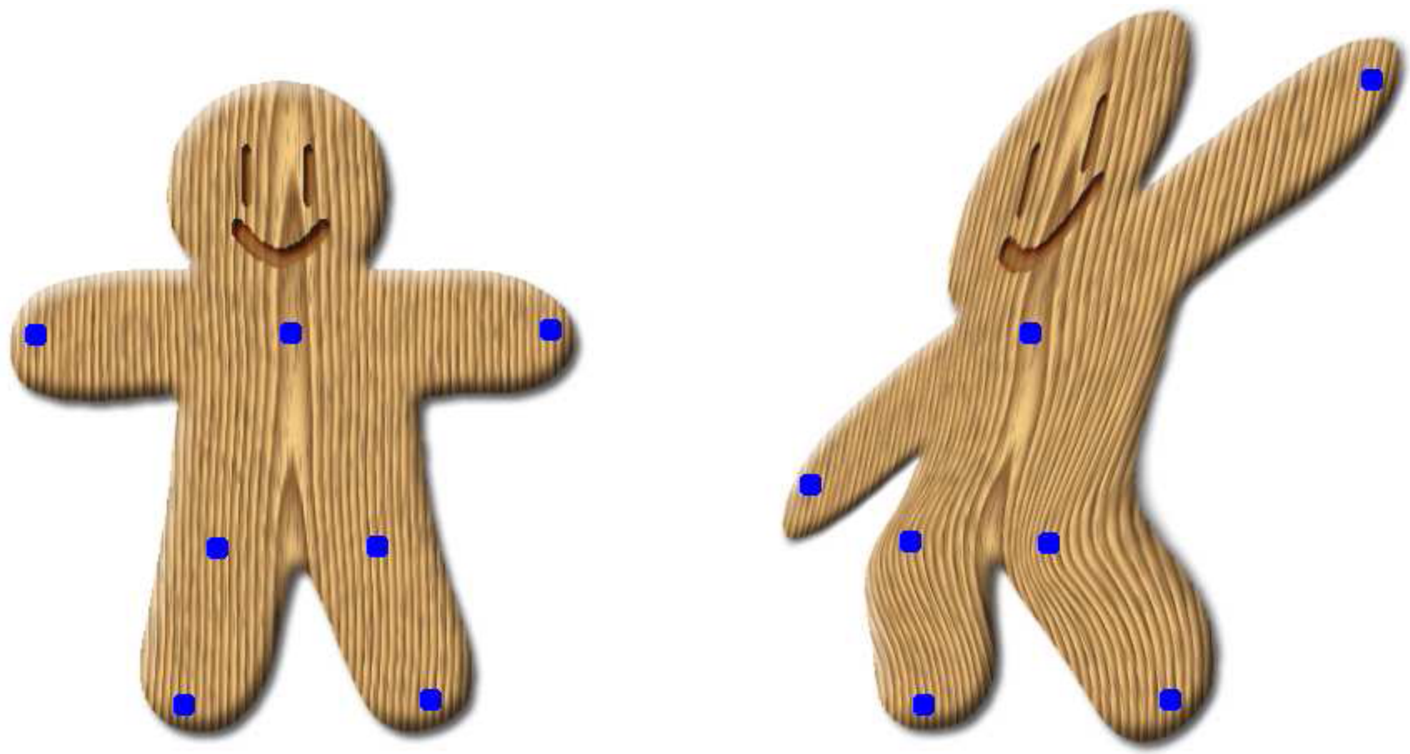
$$\mathrm{M}_{\mathbf{x}} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$



As-Rigid-As-Possible Deformation

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

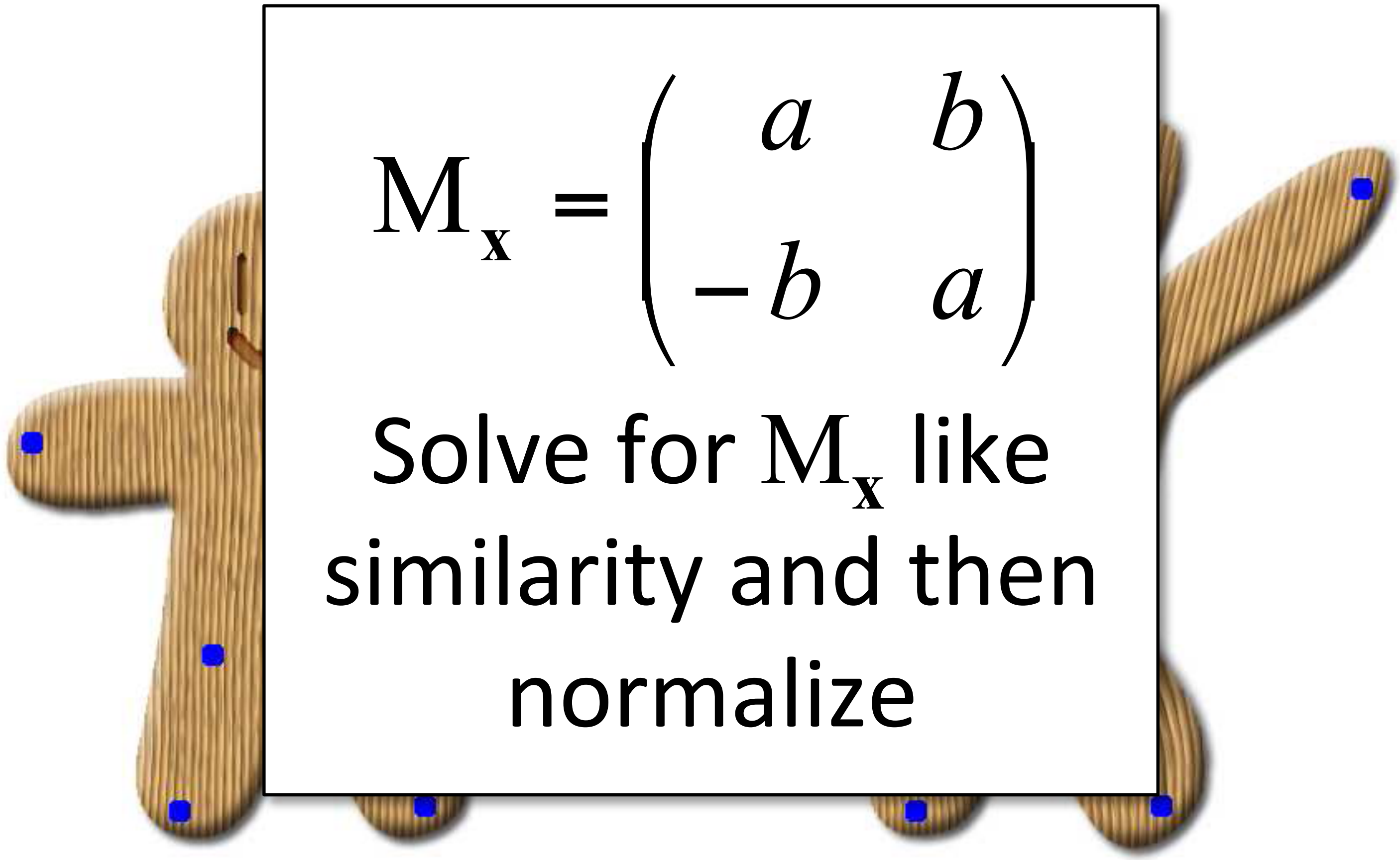
- Restrict $\mathrm{A}_{\mathbf{x}}(\cdot)$ to rigid



As-Rigid-As-Possible Deformation

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

- Restrict $\mathrm{A}_{\mathbf{x}}(\cdot)$ to rigid



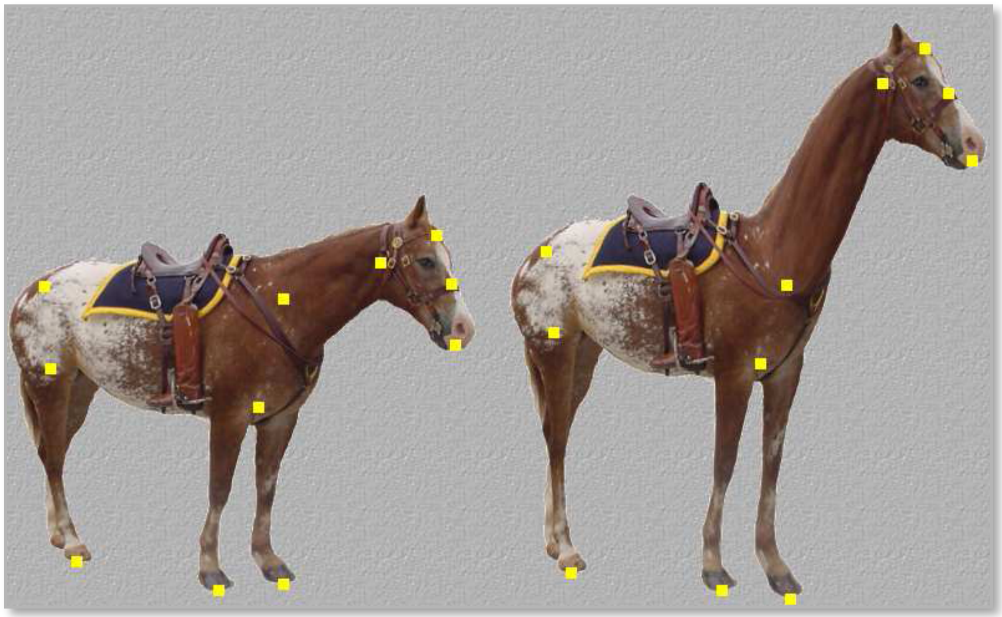
$$\mathrm{M}_{\mathbf{x}} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

Solve for $\mathrm{M}_{\mathbf{x}}$ like similarity and then normalize

As-Rigid-As-Possible Deformation

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

- Examples



As-Rigid-As-Possible Deformation

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

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

$$\arg\min_{\mathrm{R}\in SO(3)} \sum_{i=1}^{k} w_i(\mathbf{x}) \left\| \mathrm{R}\mathbf{p}_i - \mathbf{q}_i \right\|^2$$

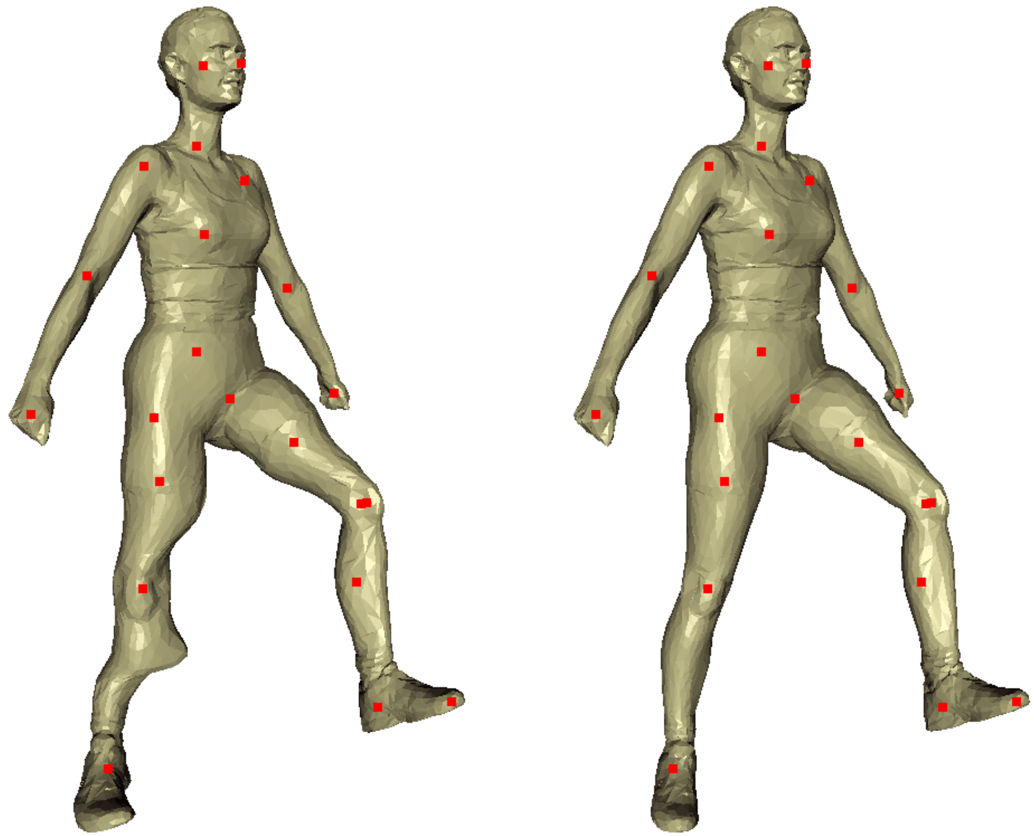
by polar decomposition of the 3×3 covariance matrix

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"

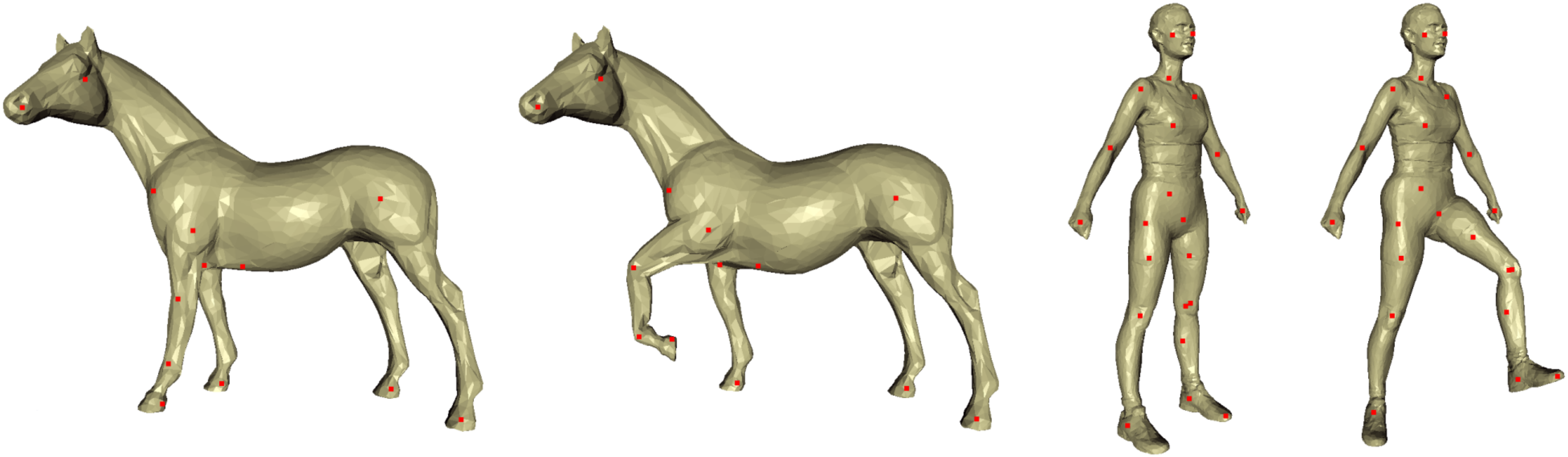
$$w_i(\mathbf{x}) = d(\mathbf{p}_i\,,\ \mathbf{x})^{-2\alpha}$$

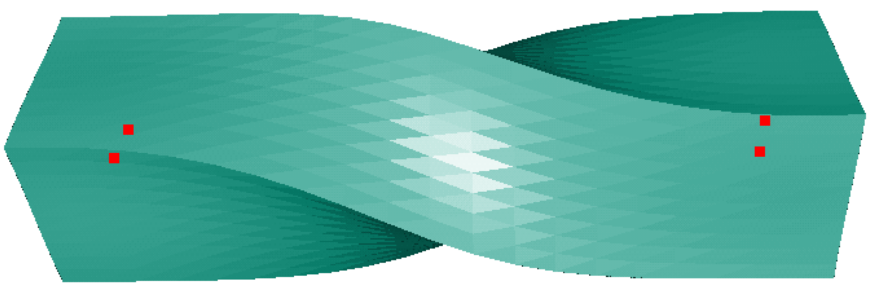


As-Rigid-As-Possible Deformation

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

- More results

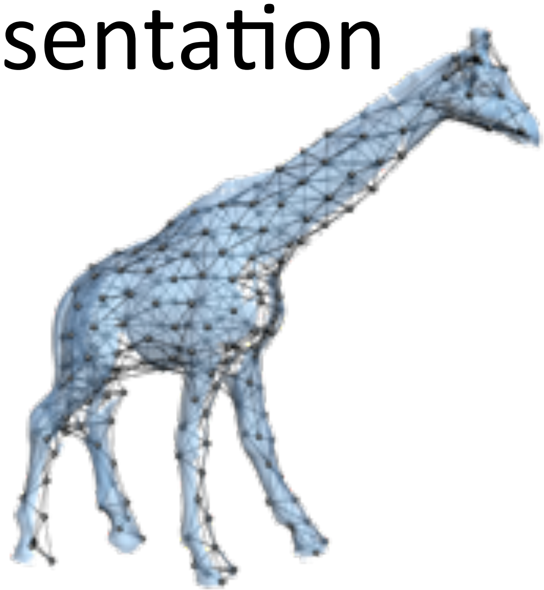




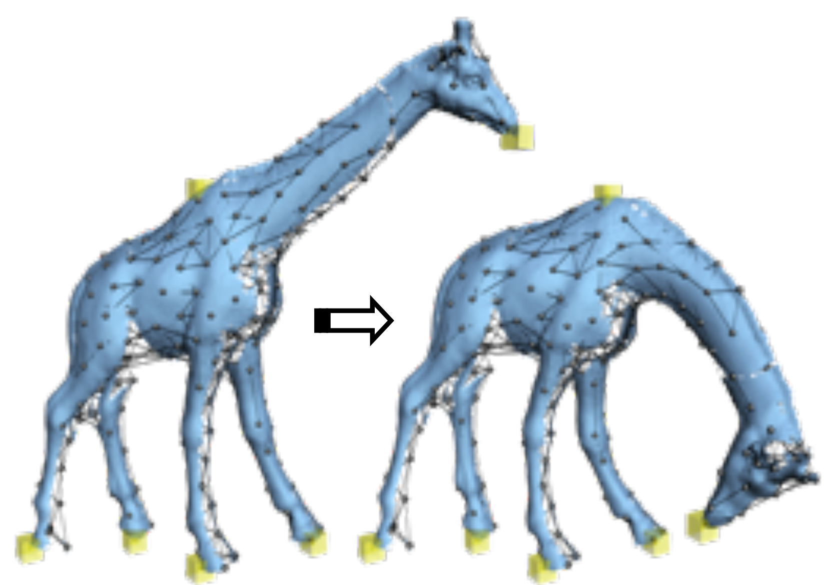
As-Rigid-As-Possible Deformation

Embedded Deformation [Sumner et al. 07]

- Surface handles as interface
- Underlying graph to represent the deformation; nodes store rigid transformations
- Decoupling of handles from def. representation



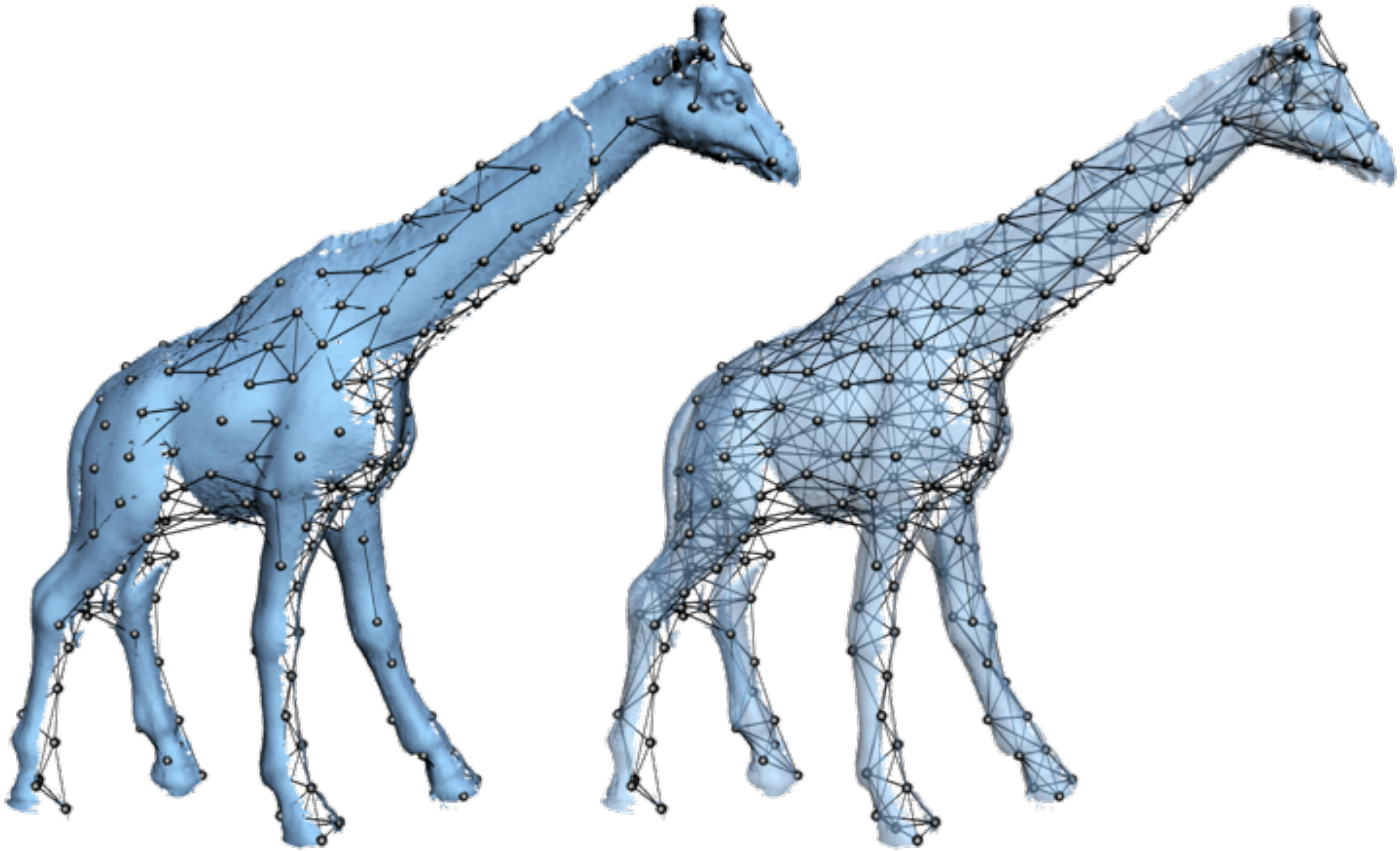
Deformation Graph



Optimization Procedure

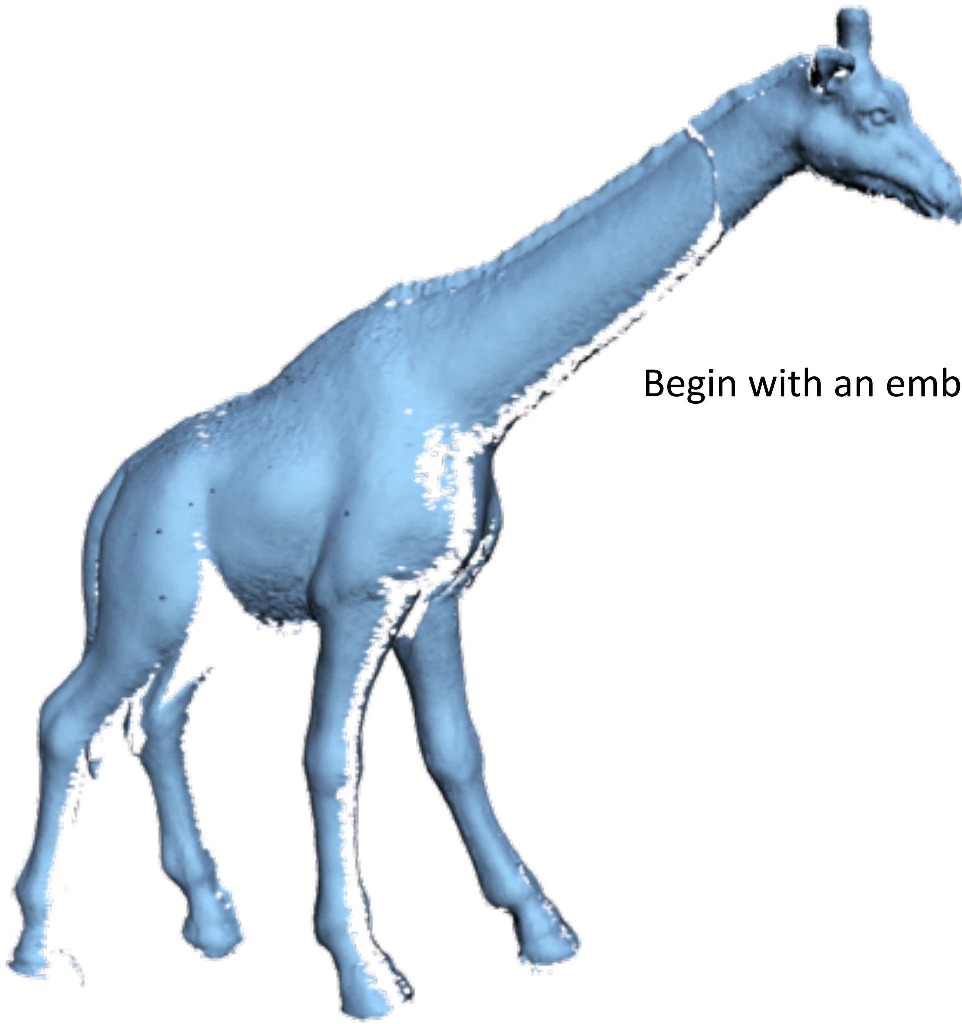
Deformation Graph

Embedded Deformation [Sumner et al. 07]



Deformation Graph

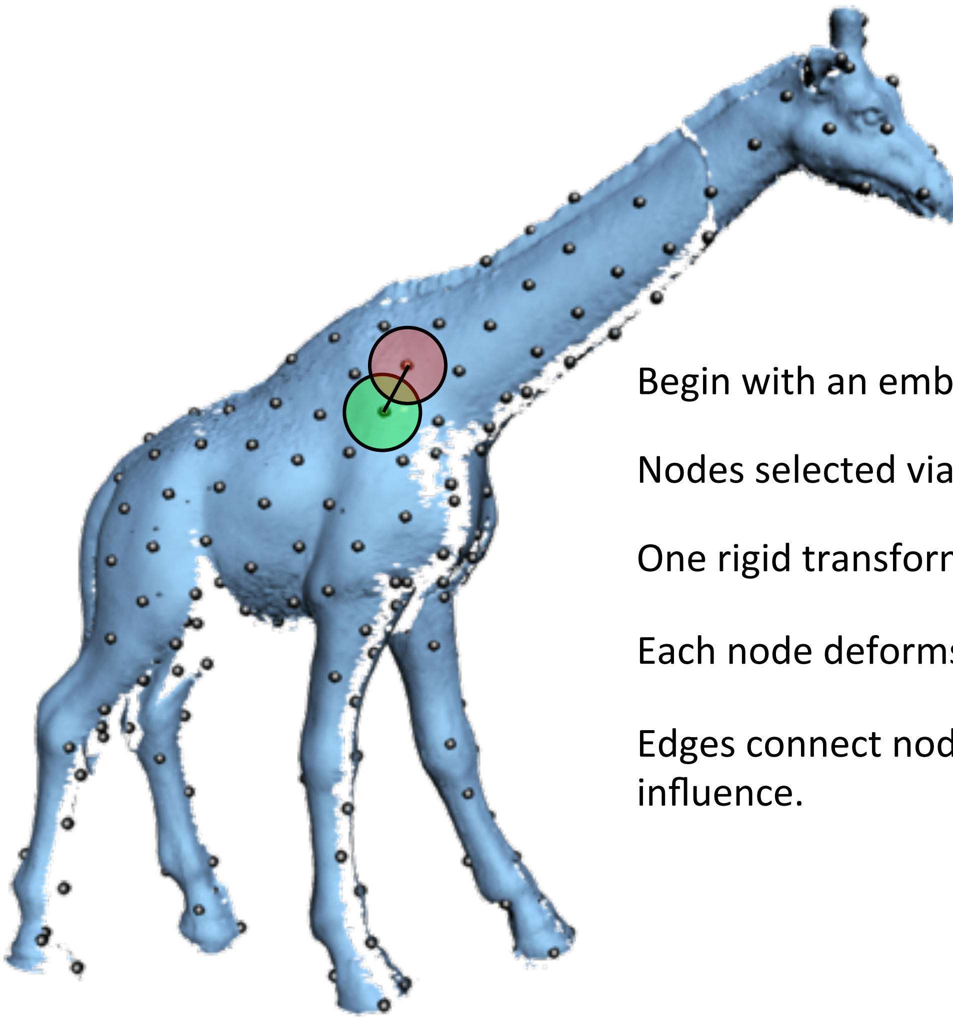
Embedded Deformation [Sumner et al. 07]



Begin with an embedded object.

Deformation Graph

Embedded Deformation [Sumner et al. 07]



Begin with an embedded object.

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

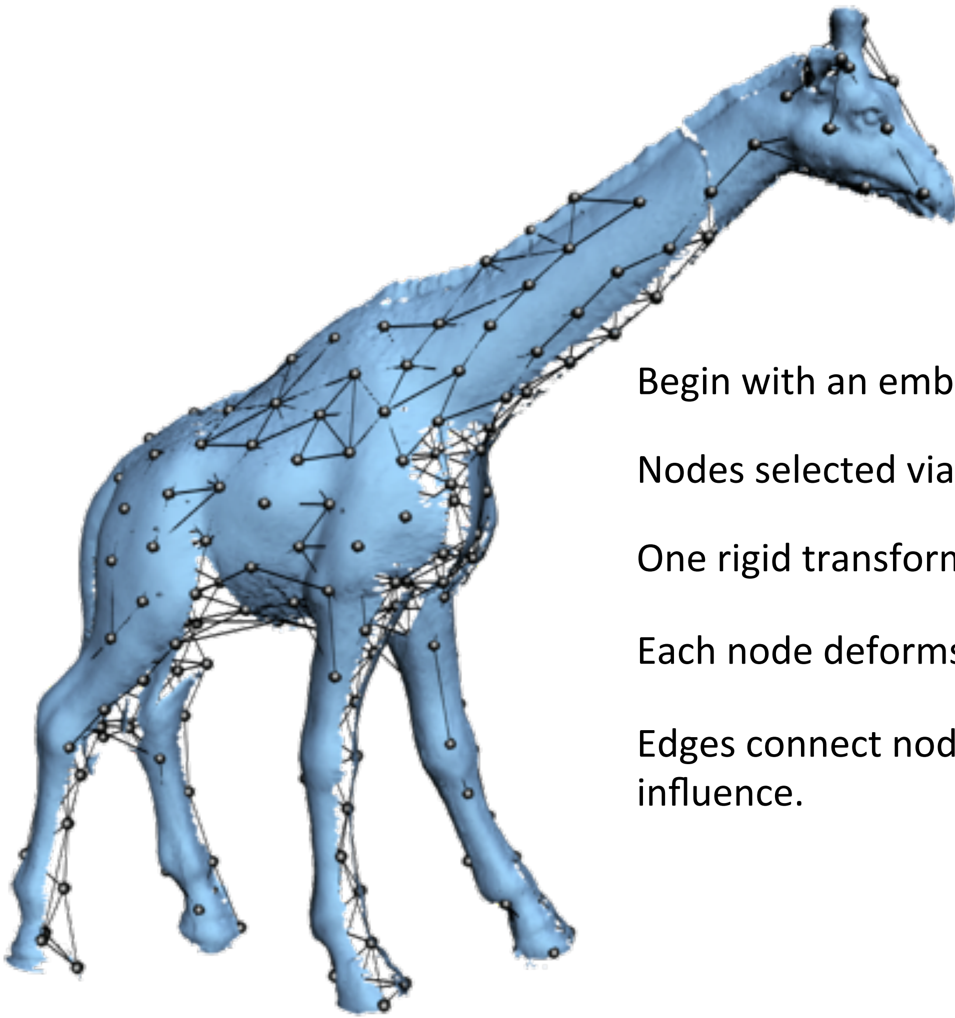
One rigid transformation for each node: R_j , $\mathbf{t}_j$

Each node deforms nearby space.

Edges connect nodes of overlapping influence.

Deformation Graph

Embedded Deformation [Sumner et al. 07]



Begin with an embedded object.

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

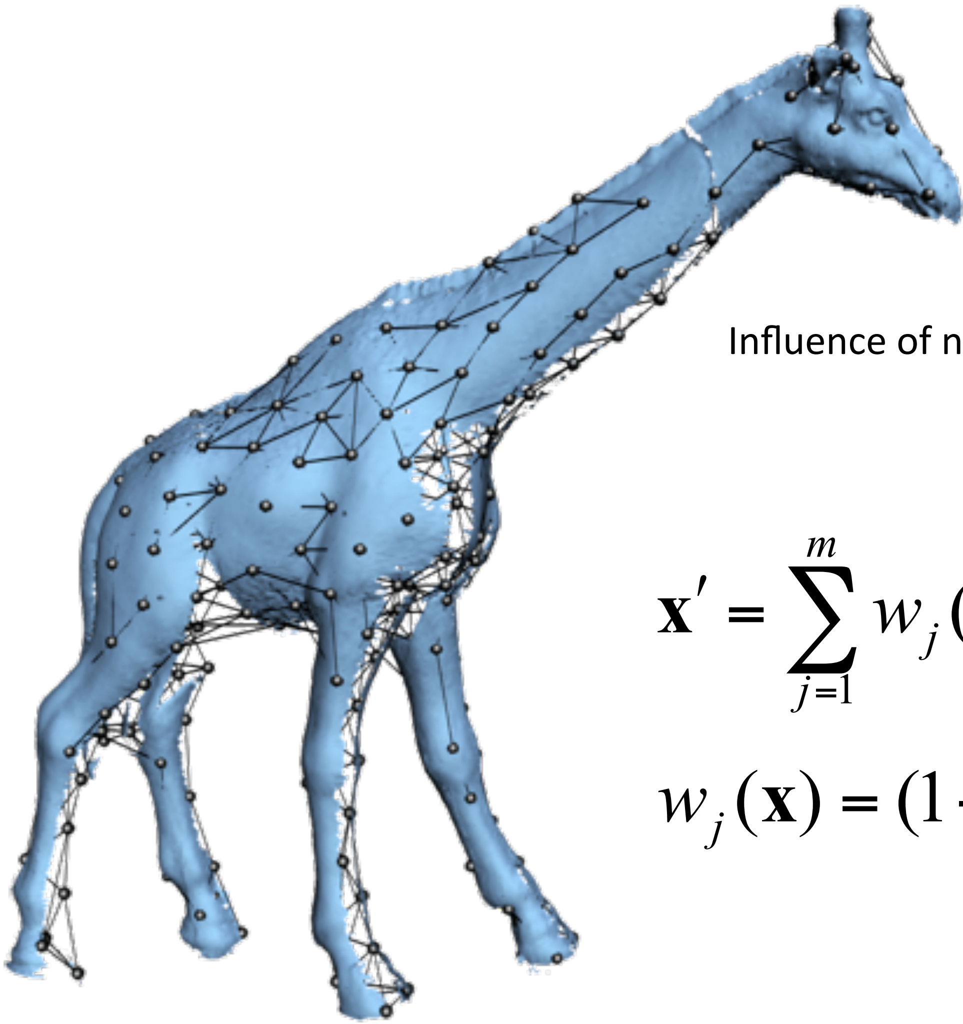
One rigid transformation for each node: R_j , $\mathbf{t}_j$

Each node deforms nearby space.

Edges connect nodes of overlapping influence.

Deformation Graph

Embedded Deformation [Sumner et al. 07]



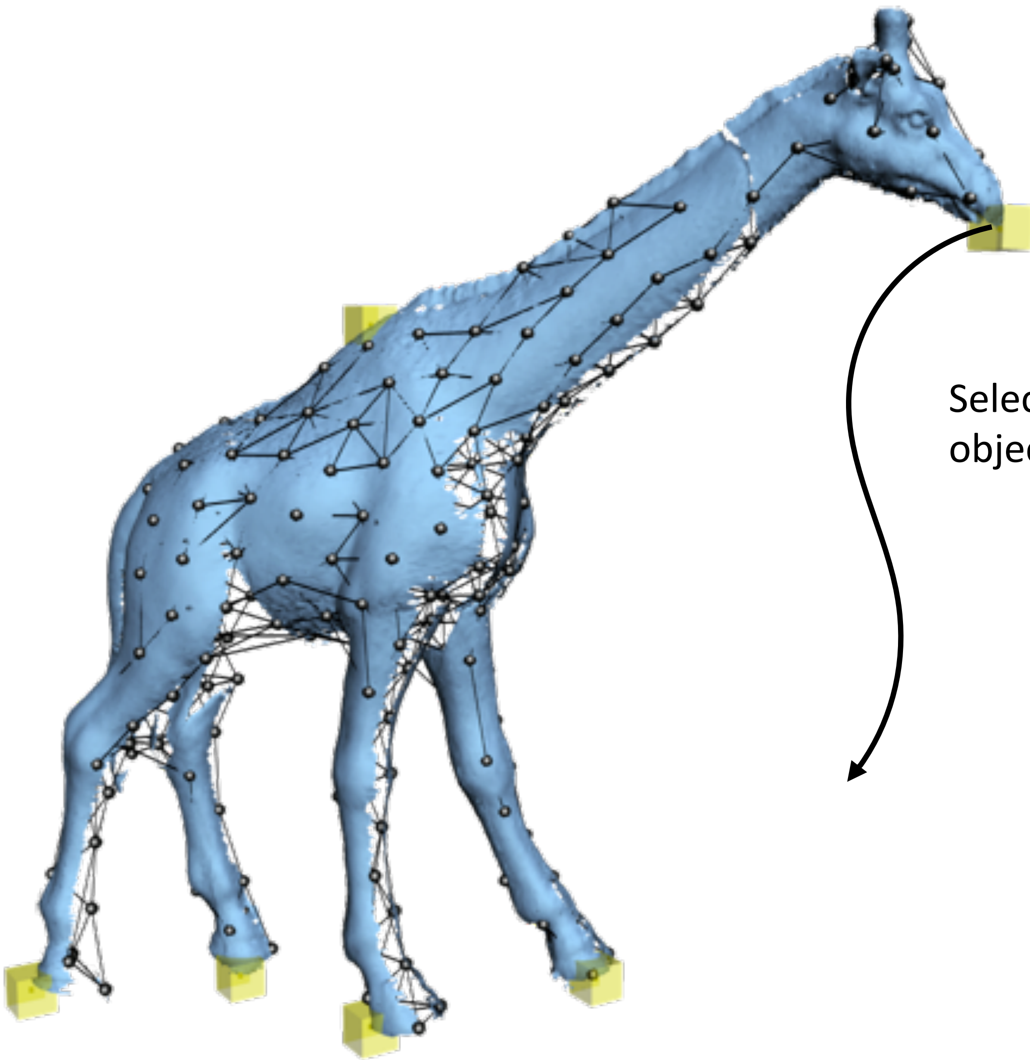
Influence of nearby transformations is blended.

$$\mathbf{x}' = \sum_{j=1}^{m} w_j(\mathbf{x}) \overbrace{\left[\mathrm{R}_j(\mathbf{x}-\mathbf{g}_j)+\mathbf{g}_j+\mathbf{t}_j\right]}^{\text{point } \mathbf{x} \text{ transformed by node } j}$$

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

Optimization

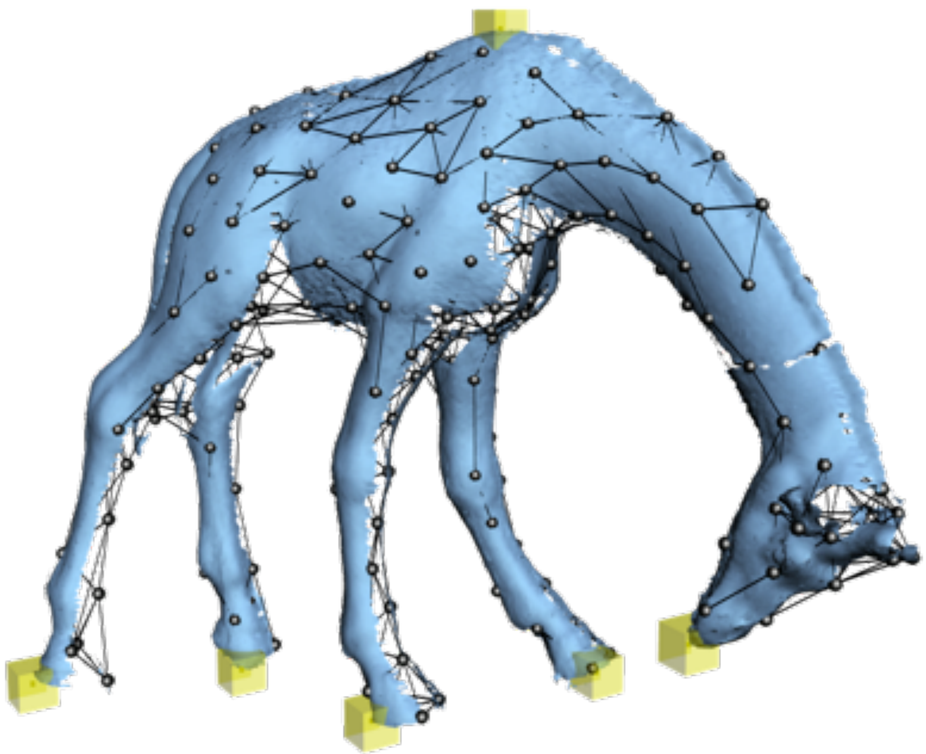
Embedded Deformation [Sumner et al. 07]



Select & drag vertices of embedded object.

Optimization

Embedded Deformation [Sumner et al. 07]



Select & drag vertices of embedded object.

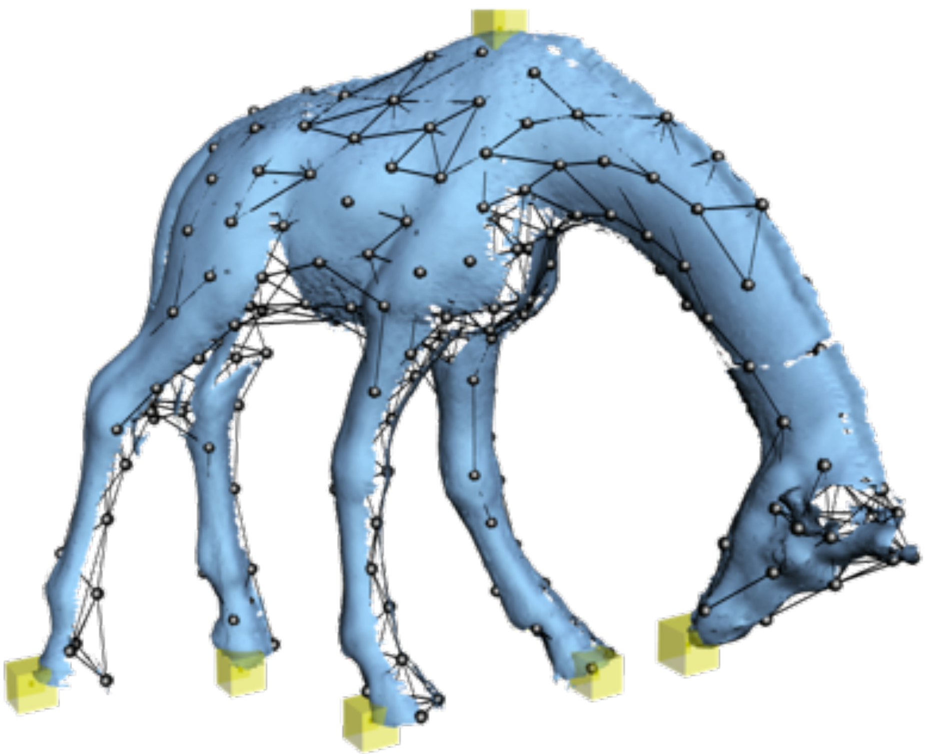
Optimization finds deformation parameters R_j , $\mathbf{t}_j$.

Optimization

Embedded Deformation [Sumner et al. 07]

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

Graph parameters	Rotation term	Regularization term	Constraint term



Select & drag vertices of embedded object.

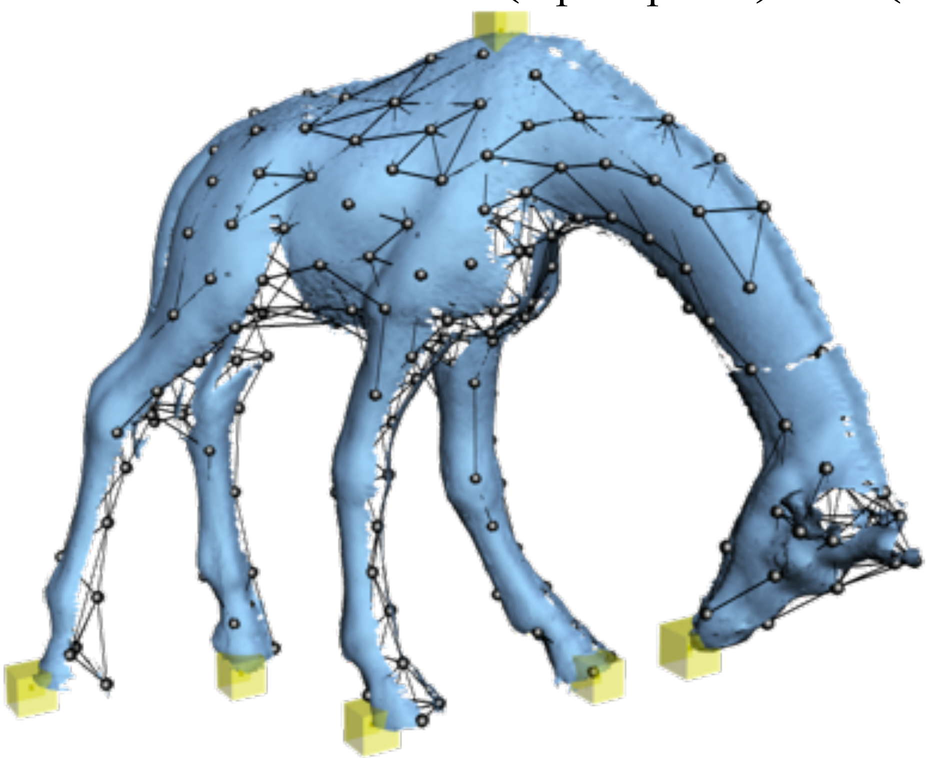
Optimization finds deformation parameters R_j , $\mathbf{t}_j$.

Optimization

Embedded Deformation [Sumner et al. 07]

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

$$\mathrm{Rot}(\mathbf{R}) = (\mathbf{c}_1\cdot\mathbf{c}_2)^2 + (\mathbf{c}_1\cdot\mathbf{c}_3)^2 + (\mathbf{c}_2\cdot\mathbf{c}_3)^2 + (\mathbf{c}_1\cdot\mathbf{c}_1 - 1)^2 + (\mathbf{c}_2\cdot\mathbf{c}_2 - 1)^2 + (\mathbf{c}_3\cdot\mathbf{c}_3 - 1)^2$$



$$\mathrm{E}_{\mathrm{rot}} = \sum_{j=1}^{m} \mathrm{Rot}(\mathbf{R}_j)$$

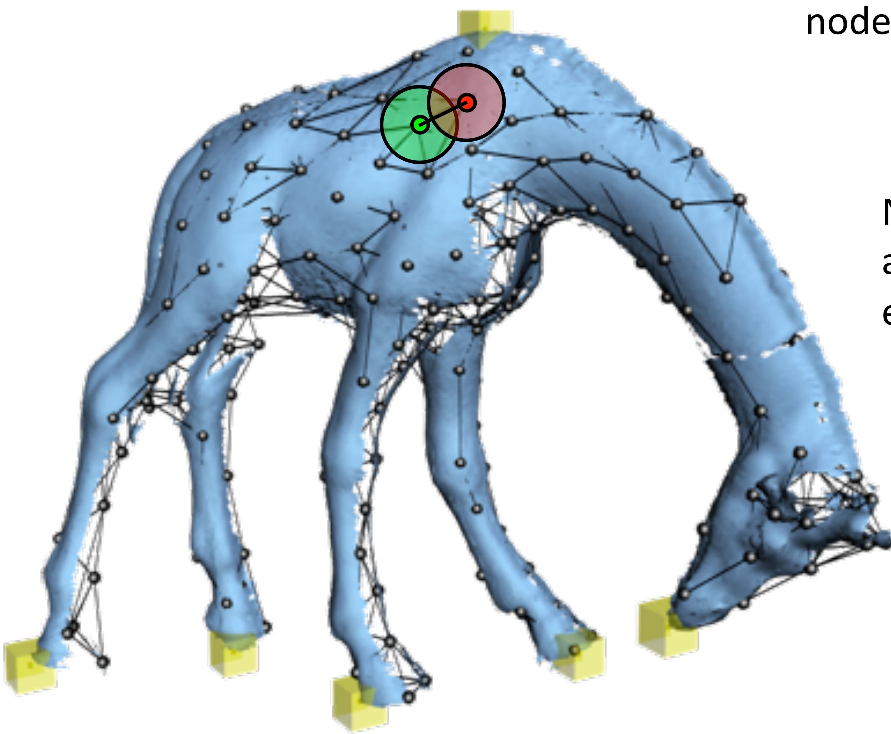
For detail preservation, features should rotate and not scale or skew.

Optimization

Embedded Deformation [Sumner et al. 07]

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

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

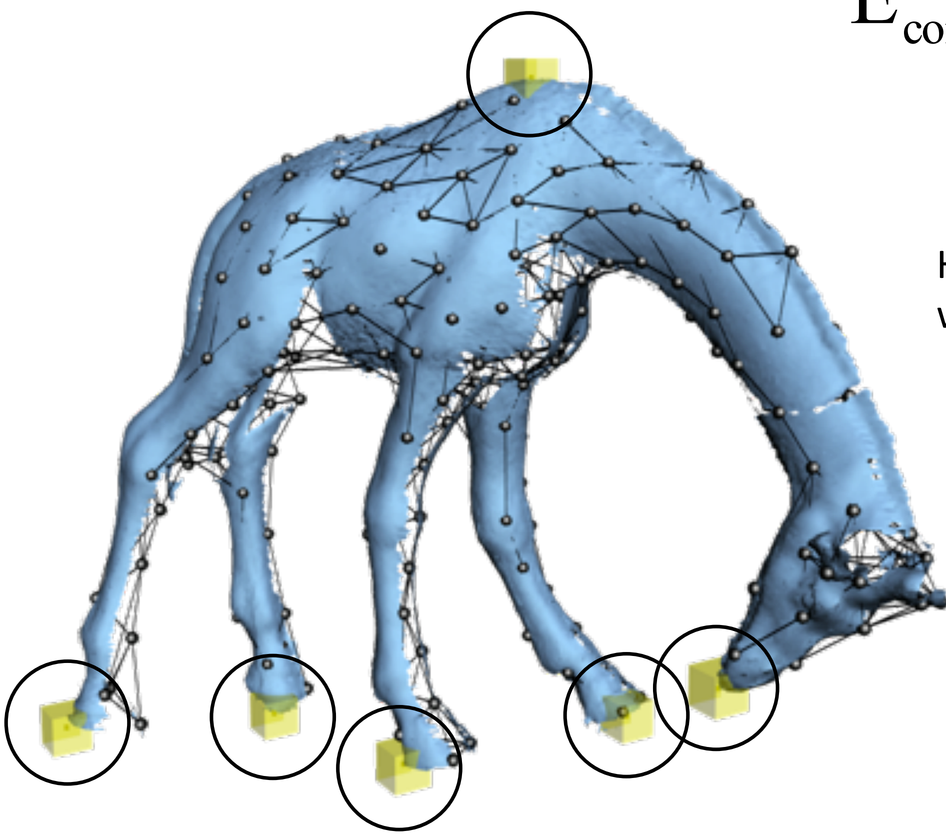


Neighboring nodes should agree on where they transform each other.

Optimization

Embedded Deformation [Sumner et al. 07]

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



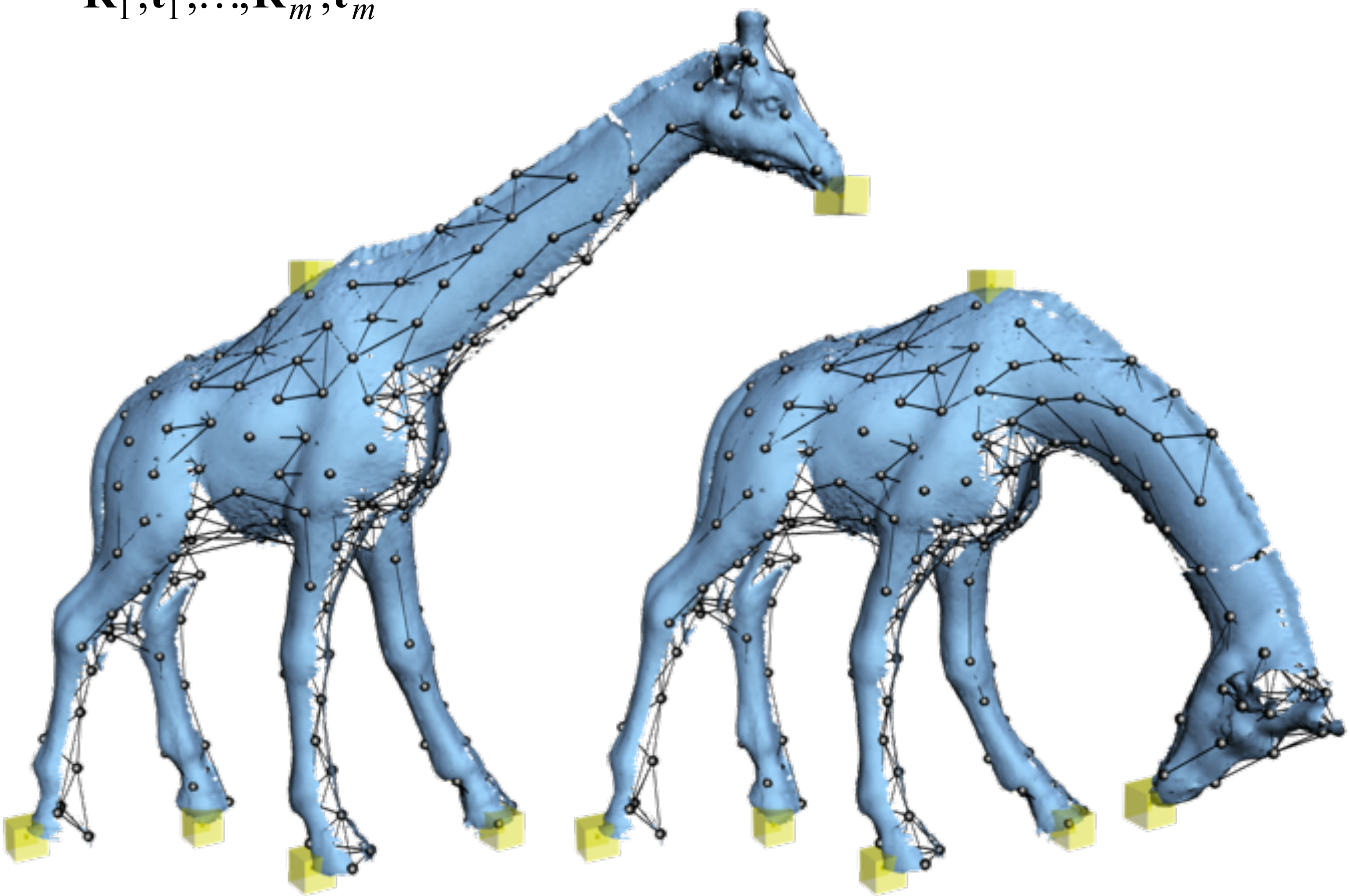
$$\mathrm{E}_{\mathrm{con}} = \sum_{l=1}^{p} \left\| \tilde{\mathbf{v}}_{\mathrm{index}(l)} - \mathbf{q}_l \right\|_2^2$$

Handle vertices should go where the user puts them.

Optimization

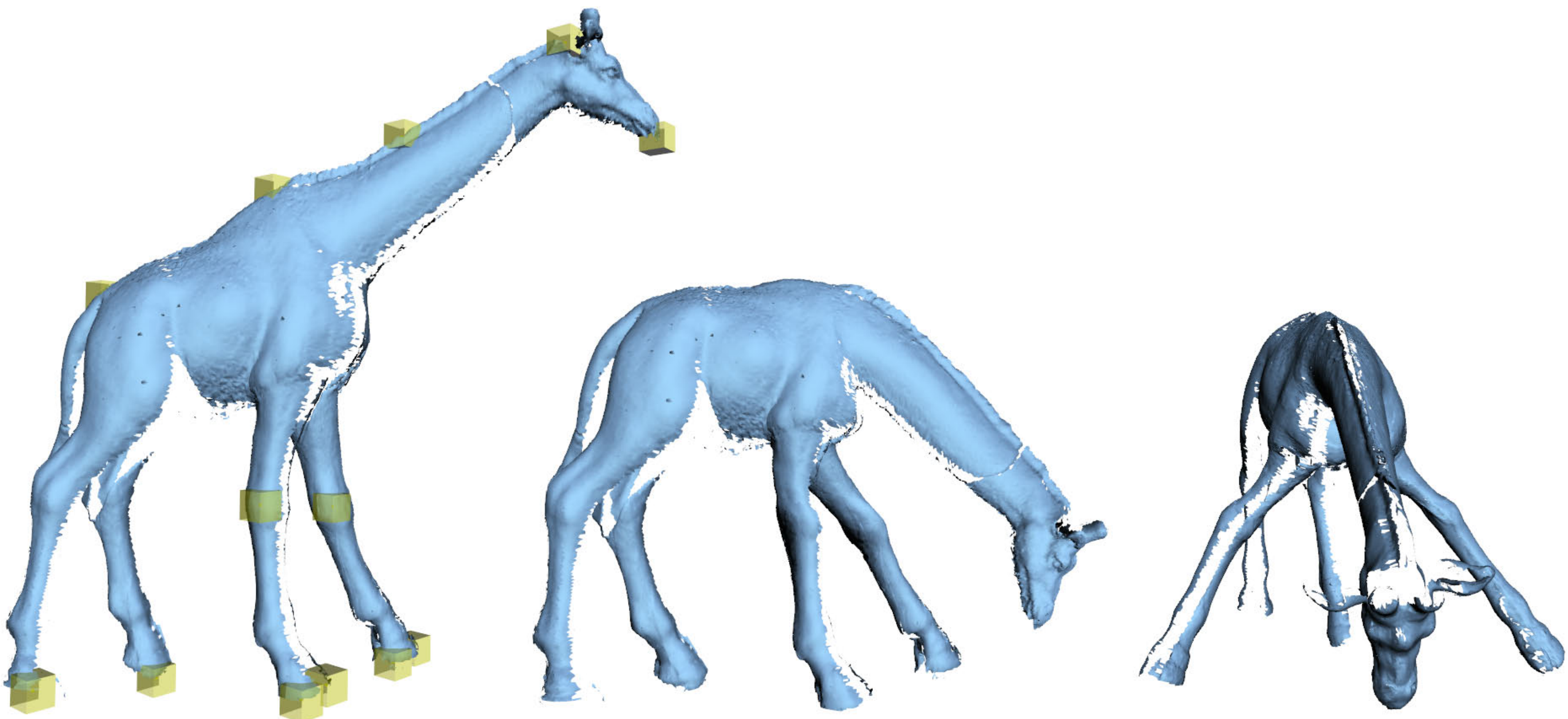
Embedded Deformation [Sumner et al. 07]

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



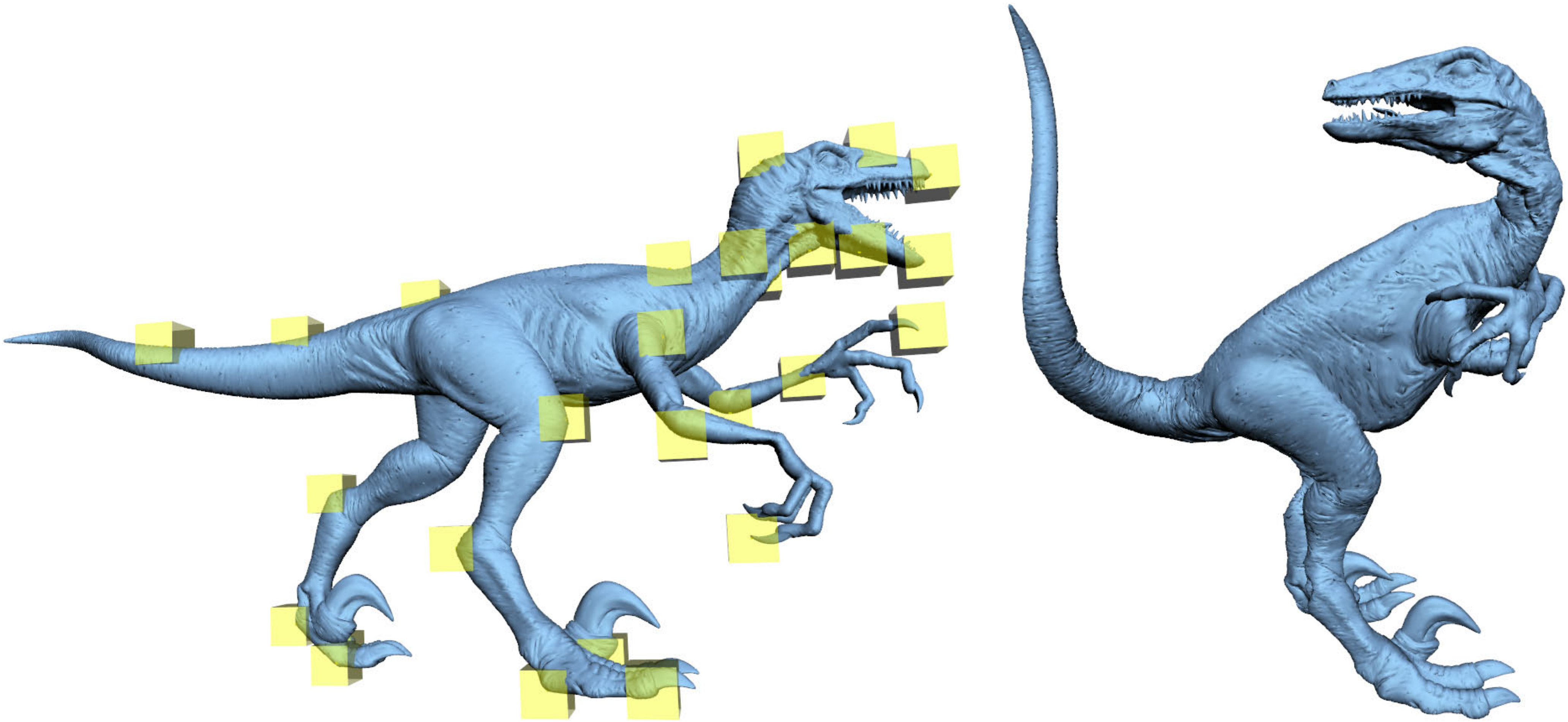
Results on Polygon Soups

Embedded Deformation [Sumner et al. 07]



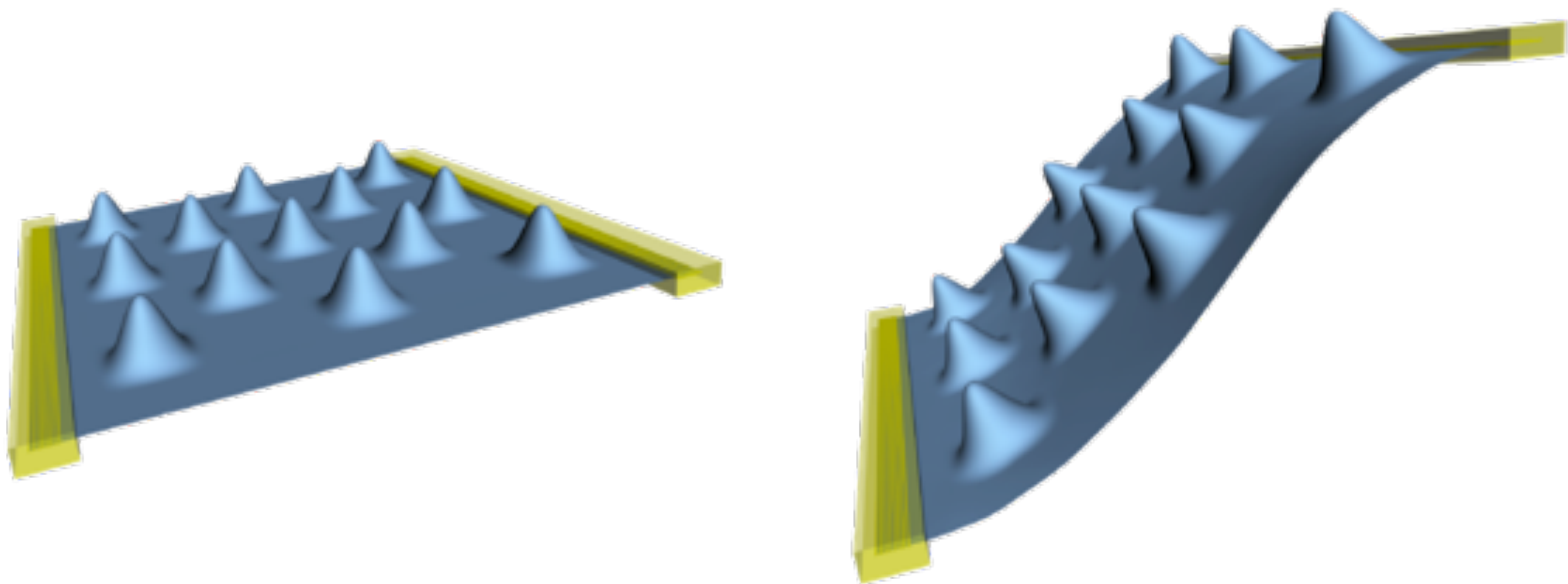
Results on Giant Mesh

Embedded Deformation [Sumner et al. 07]



Detail Preservation

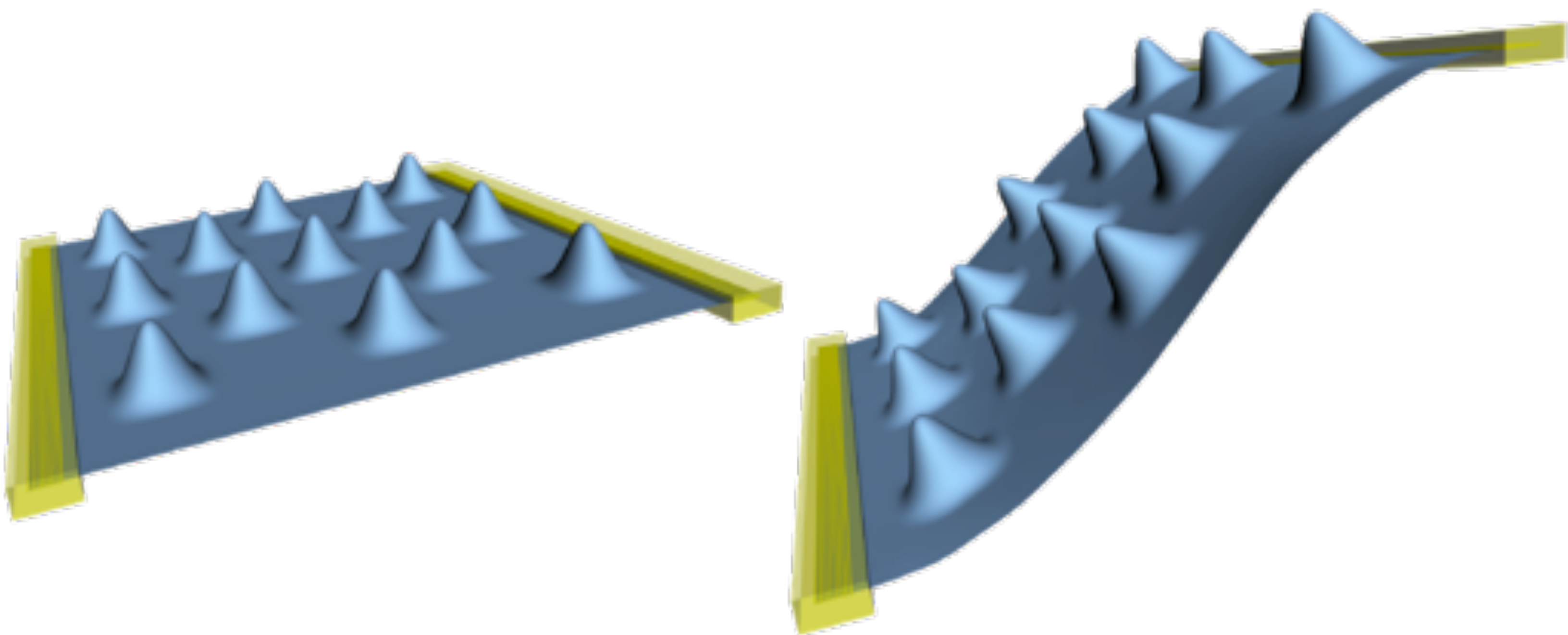
Embedded Deformation [Sumner et al. 07]



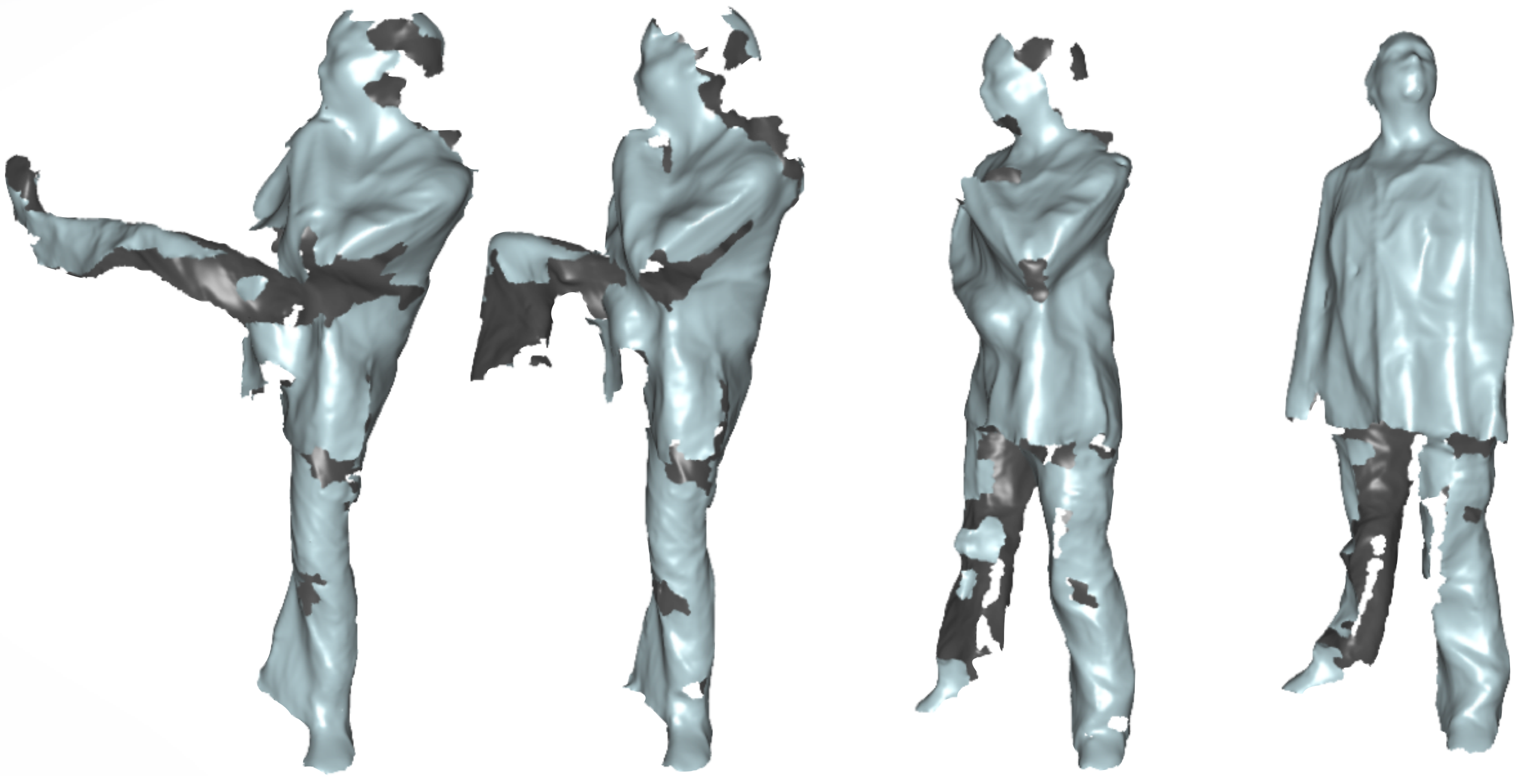
Discussion

Embedded Deformation [Sumner et al. 07]

- Decoupling of deformation complexity and model complexity
- Nonlinear energy optimization – results comparable to surface-based approaches



Next Time



Dynamic Geometry Processing

http://cs599.hao-li.com

Thanks!