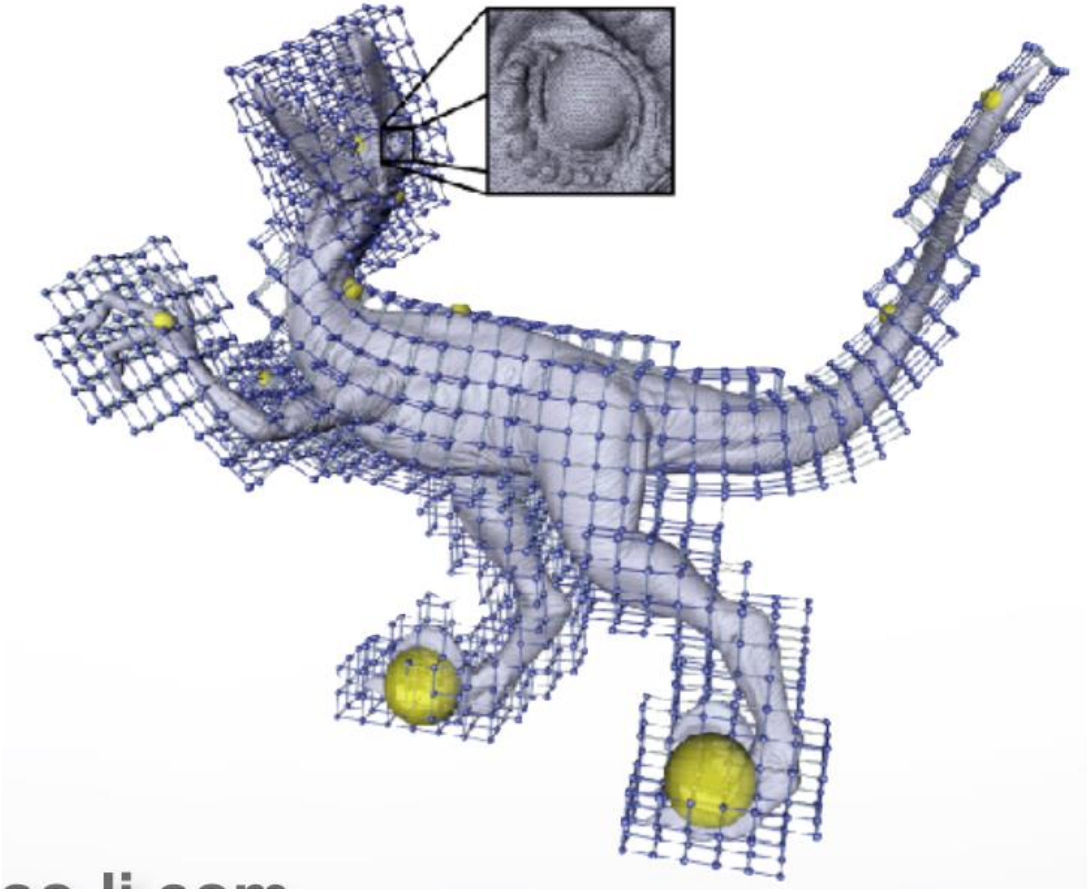
Digital Geometry - Space Deformations



Junjie Cao @ DLUT Spring 2018

http://jjcao.github.io/DigitalGeometry/

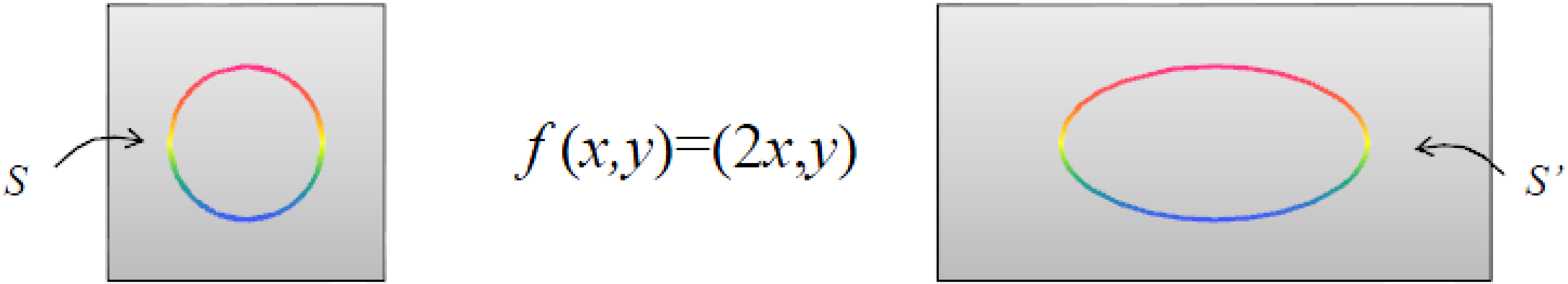
Space Deformation

- **Deformation function on ambient space**

$$f: R^n \to R^n$$

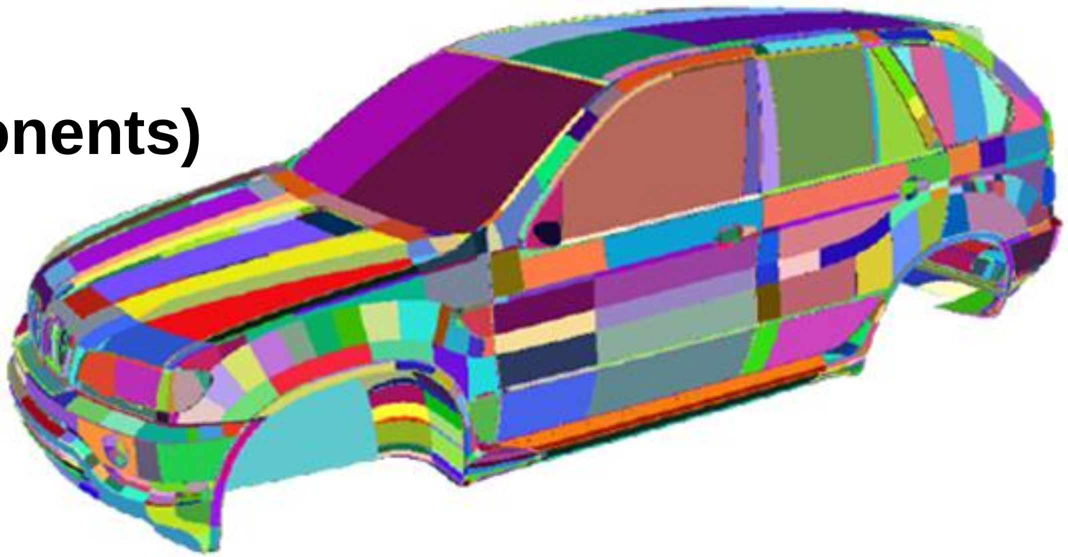
- **Shape *S* deformed by applying *f* to points of *S***

$$S' = f(S)$$



Motivation

- **Can be applied to any geometry**
  - **Meshes (= non-manifold, multiple components)**
  - **Polygon soups**
  - **Point clouds**
  - **Volumetric data**



- **Complexity decoupled from geometry complexity**
  - **Can pick the best complexity for required deformation**

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

Required Properties

- Smooth

- Efficient to compute

- “Intuitive deformation” ?
    - Can pose constraints as in surface deformation

Types of Space Deformations

- Function-based deformation
- Free form deformation
- Cage deformation
- Skeleton deformation

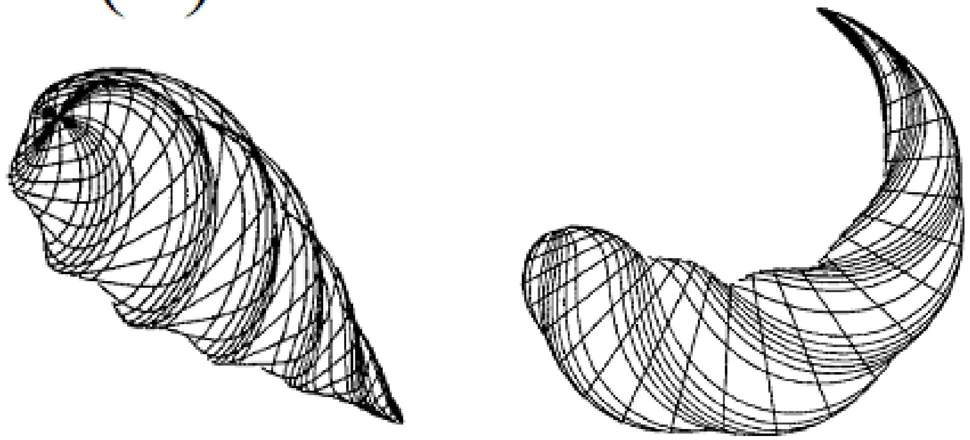
Space Deformations

- Displacement function defined on the ambient space

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

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

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

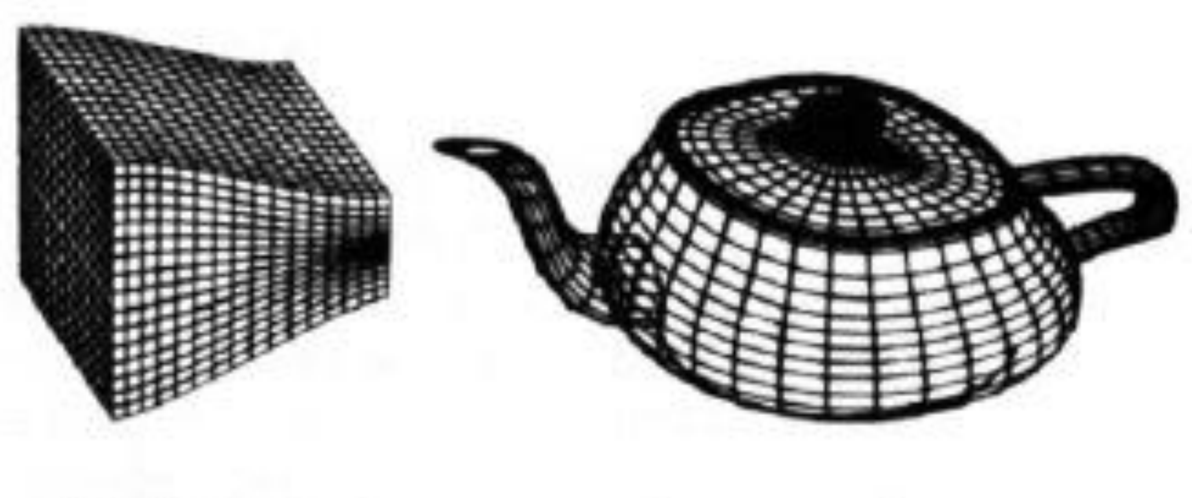


Twist warp
Global and local deformation of solids
[A. Barr, SIGGRAPH 84]

Taper

$$s(x) = \begin{cases} 1, & x \le x_0 \\ 1 - .5\dfrac{x - x_0}{x_1 - x_0}, & x_0 < x < x_1 \\ .5, & x_1 \le x \end{cases}$$

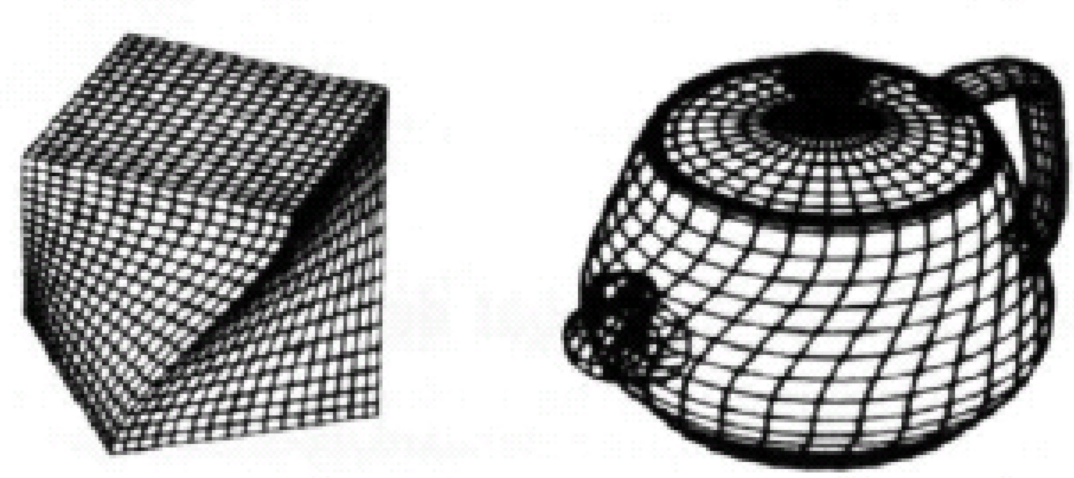
$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & s(p_x) & 0 \\ 0 & 0 & s(p_x) \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$



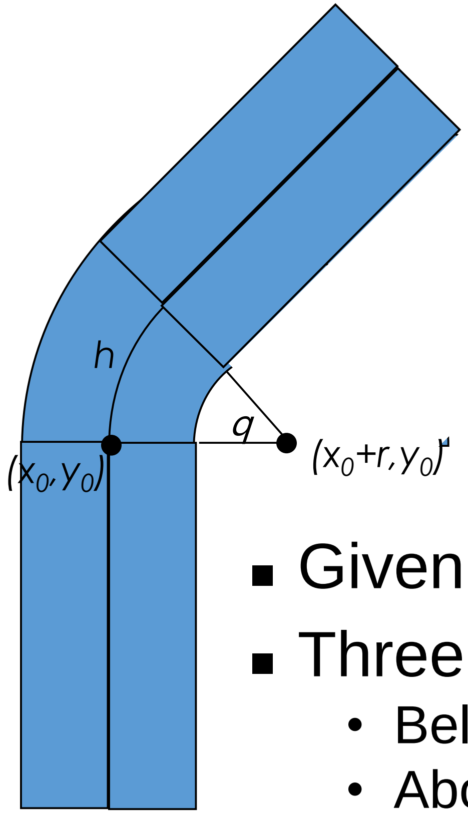
Twist

$$r(z) = \begin{cases} 0, & z \le z_0 \\ \dfrac{z - z_0}{z_1 - z_0}\theta_{\max}, & z_0 < z < z_1 \\ \theta_{\max}, & z_1 \le z \end{cases}$$

$$P' = \begin{bmatrix} \cos(r(p_z)) & -\sin(r(p_z)) & 0 \\ \sin(r(p_z)) & \cos(r(p_z)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

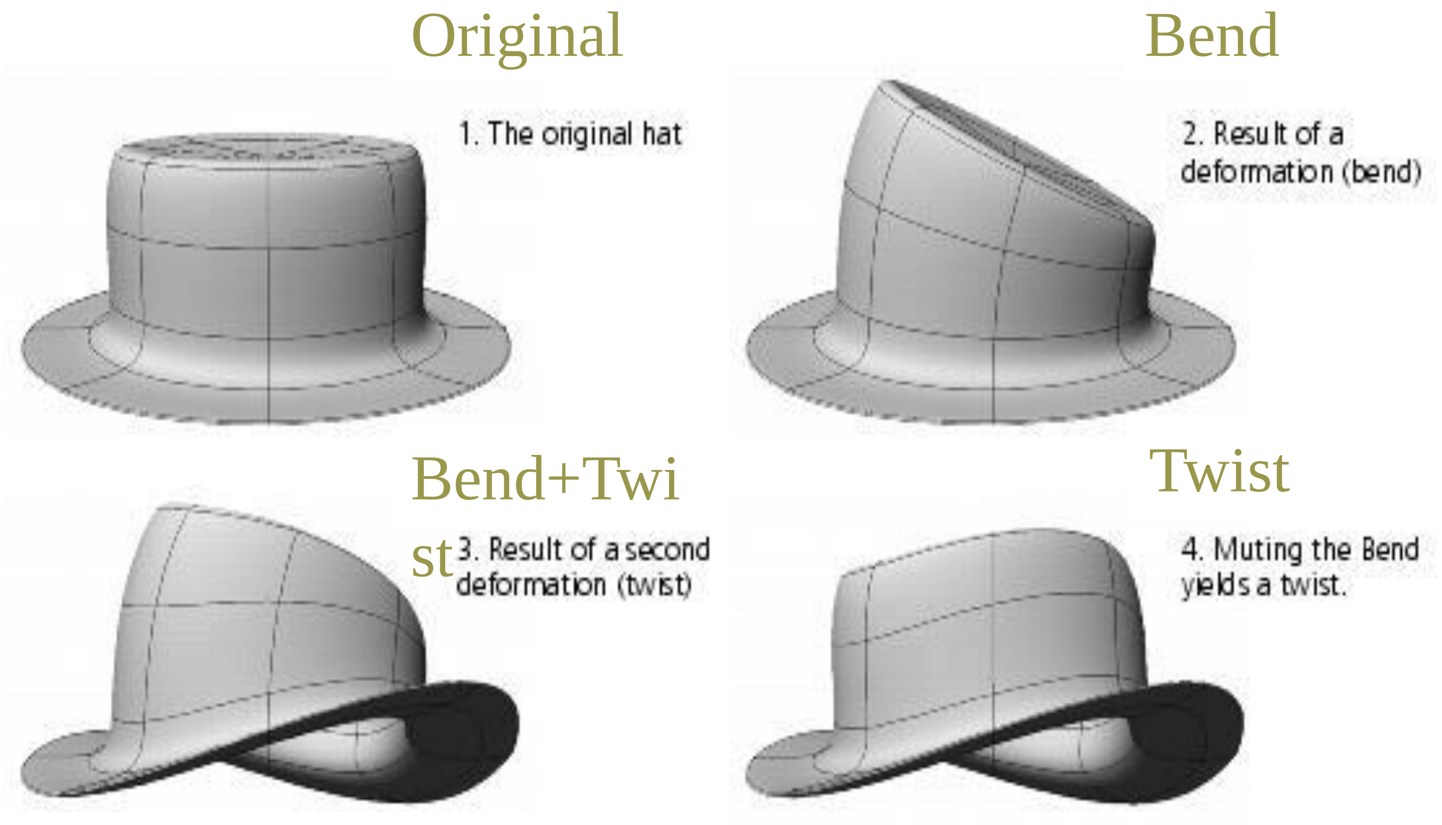


Bend



- Given x_0, y_0, h, q, r=h/q
- Three regions:
  - Below y_0: unaffected
  - Above y_0+h
    - translate down by h
    - rotate by -q about (x_0+r, y_0)
  - Between y_0 and y_0+h:
    - interpolate translation
    - interpolate rotation angle

Combinations of Deformations



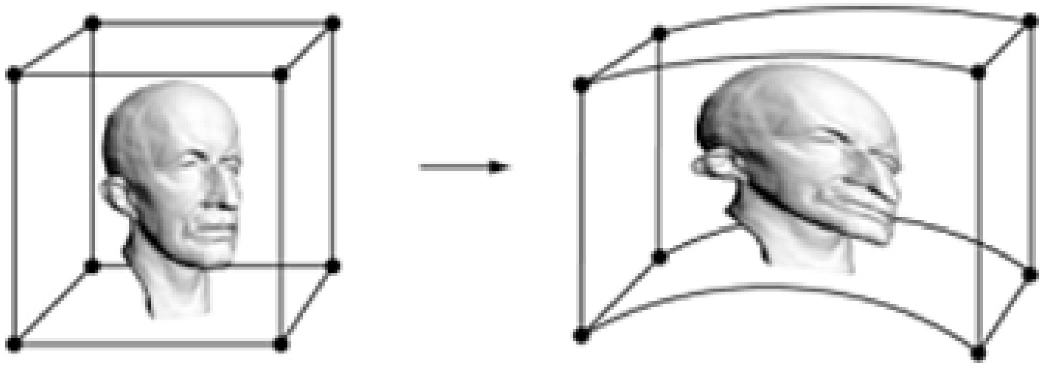
Types of Deformation

- Space deformation
  - Function-based deformation
  - **Free form deformation**
  - Radial Basis Functions
  - Cage deformation
  - Skeleton deformation
- Surface deformation

Freeform Deformations – basic ideas

- **Control object**: composed by vertices $\{p_i\}$

$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x}) \cdot \mathbf{p}_i \quad \mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x}) \cdot \mathbf{p}'_i \quad \mathrm{d}(\mathbf{x}) = \sum_{i=1}^{k} \mathrm{d}_i\, B_i(\mathbf{x})$$

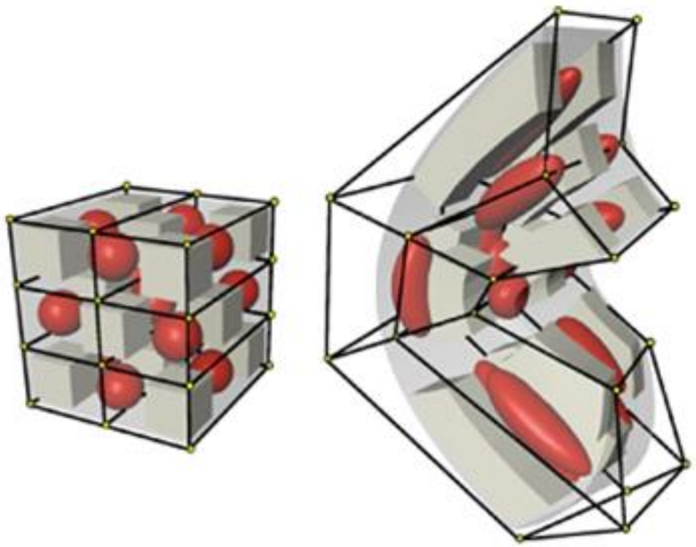


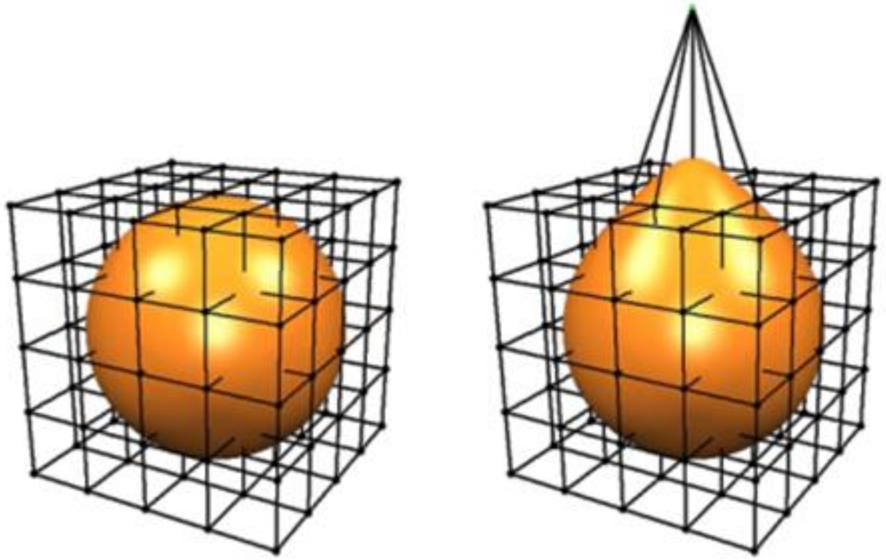
- When control object changes, the weights/coords stay the same, and x changes

Trivariate Tensor Product Bases [Sederberg and Parry 86]

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

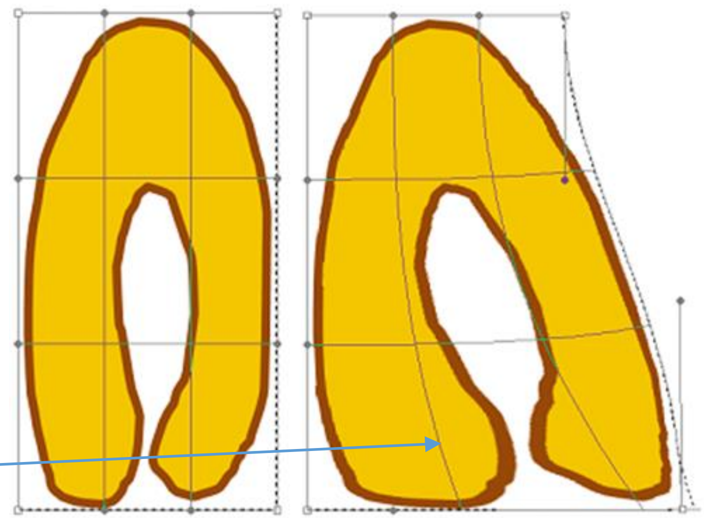
$$\mathbf{d}(x,y,z) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} \mathbf{d}_{ijk}\, N_i(x) N_j(y) N_i(z)$$





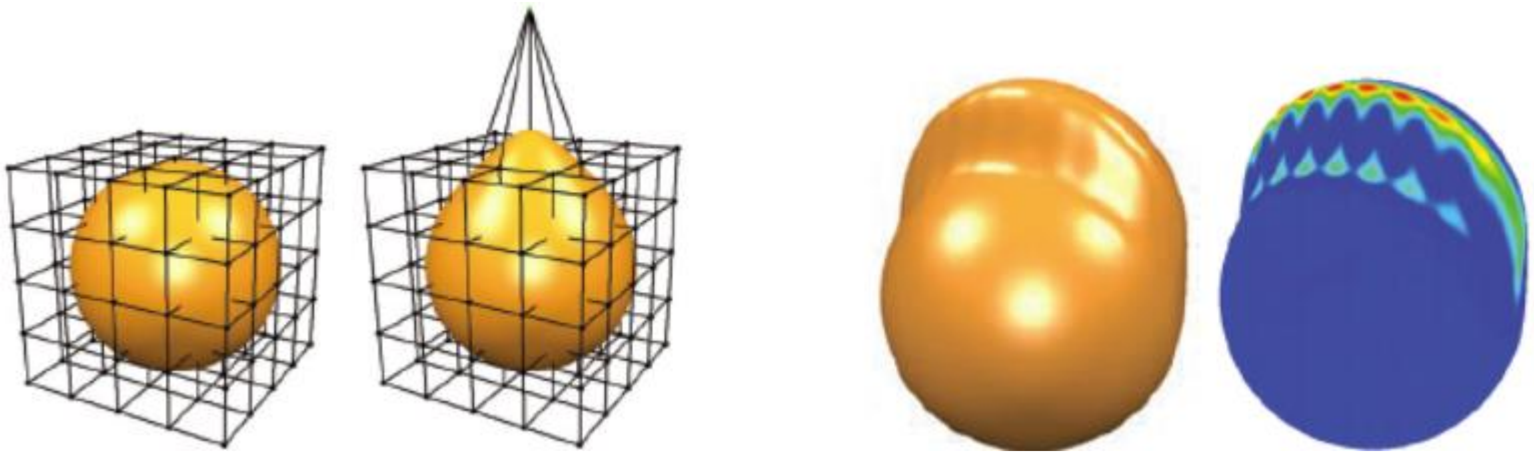
Limitations of Lattices as Control Objects

- **Difficult to manipulate**
- The control object is not related to the shape of the edited object
  - The deformation is only **loosely aware** of the shape that is being edited



- Small Euclidean distance → **similar deformation**

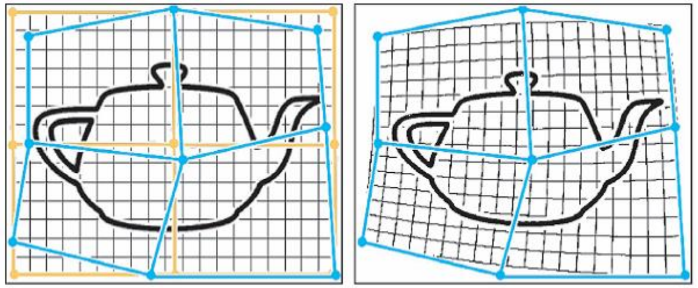
even if geodesically far

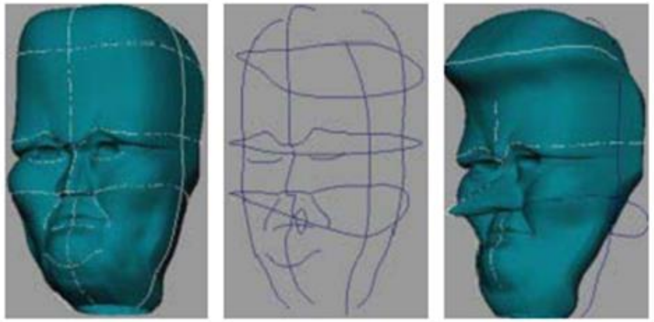


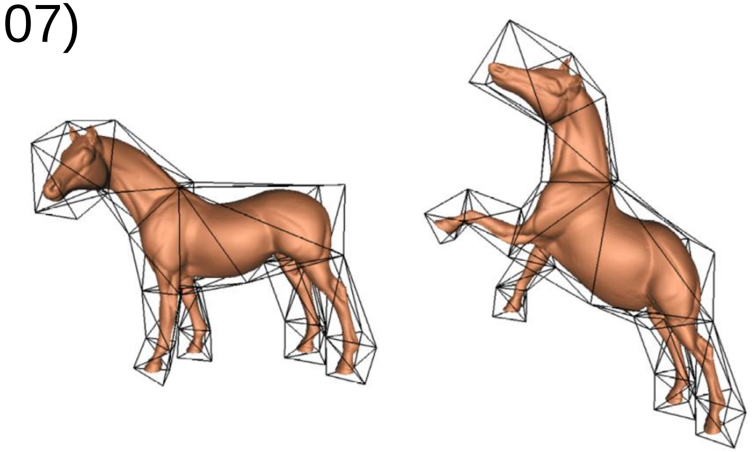
- Local surface detail may be distorted

Free-Form Deformation

- FFD (space deformation)
  - Lattice-based (Sederberg & Parry 86, Coquillart 90, …)
  - Curve-/handle-based (Singh & Fiume 98, Botsch et al. 05, …)
  - Cage-based (Ju et al. 05, Joshi et al. 07, Kopf et al. 07)



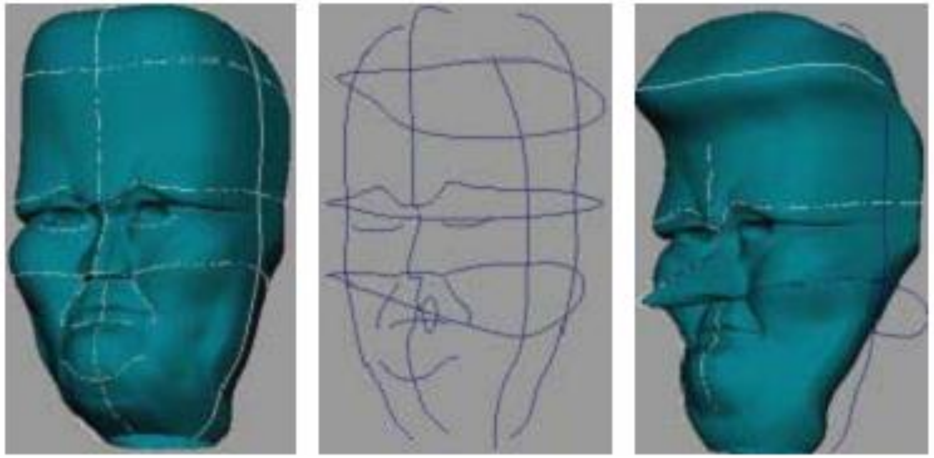


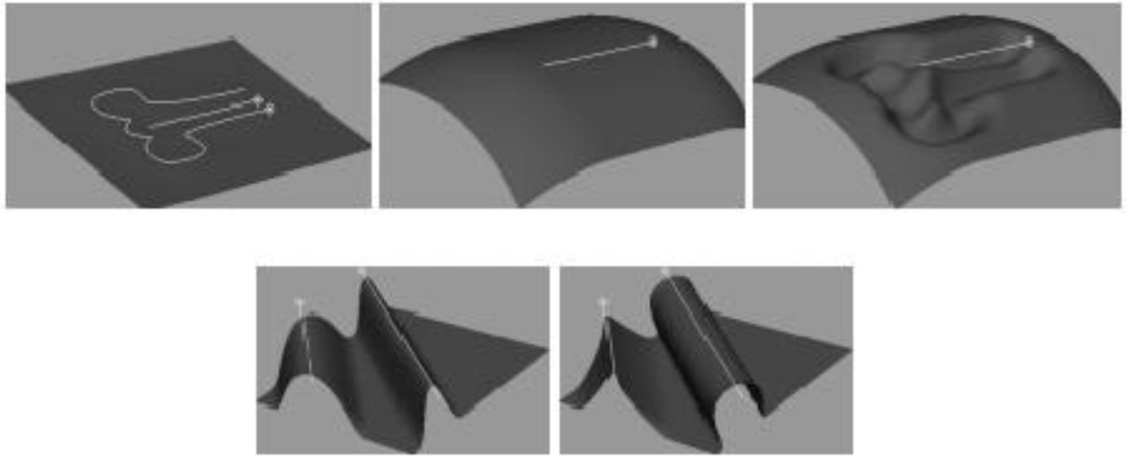


- Pros:
  - efficiency almost independent of the surface resolution
  - possible reuse
- Cons:
  - space warp, so can't precisely control surface properties

Control Object -- Wires [Singh and 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

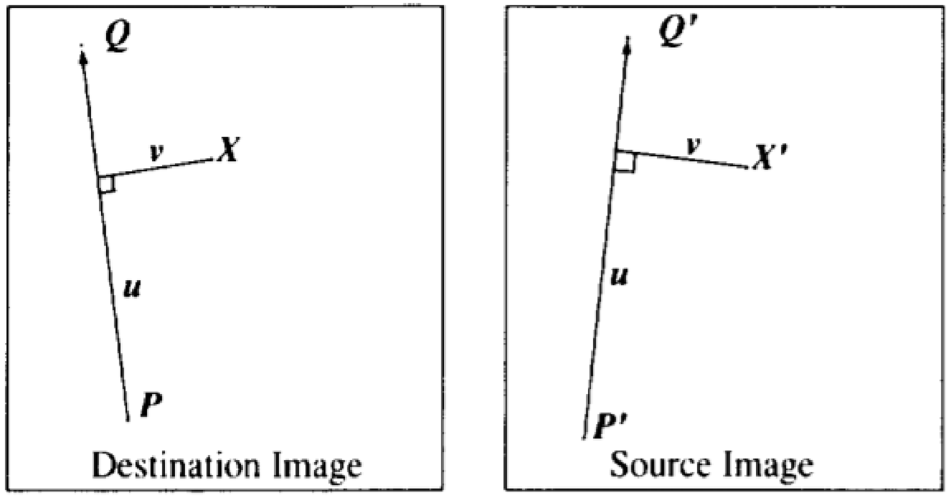


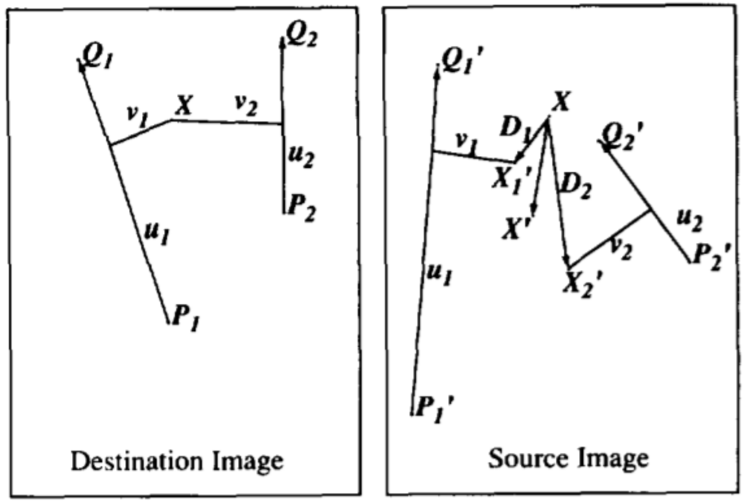


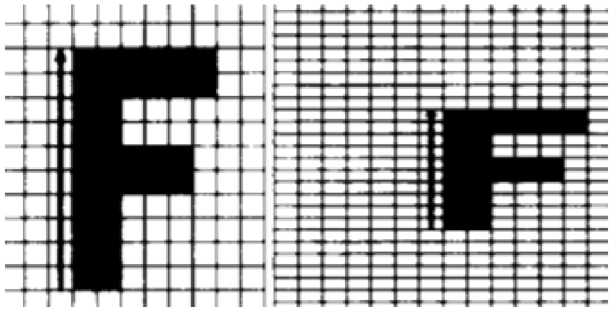
Feature-based image metamorphosis

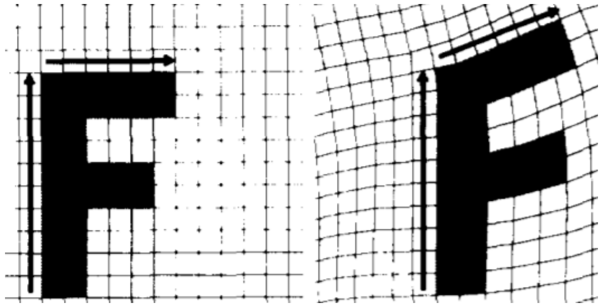
Beier and Neely @ SIGGRAPH 92

- Transformation with one pair of lines: Keep u & v



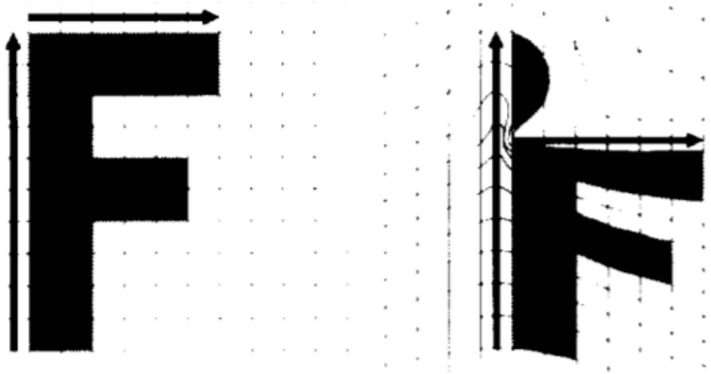






- Transformation with multiple pairs of lines
  - Weighted average of X' determined by each pair of line

- Undesirable folding



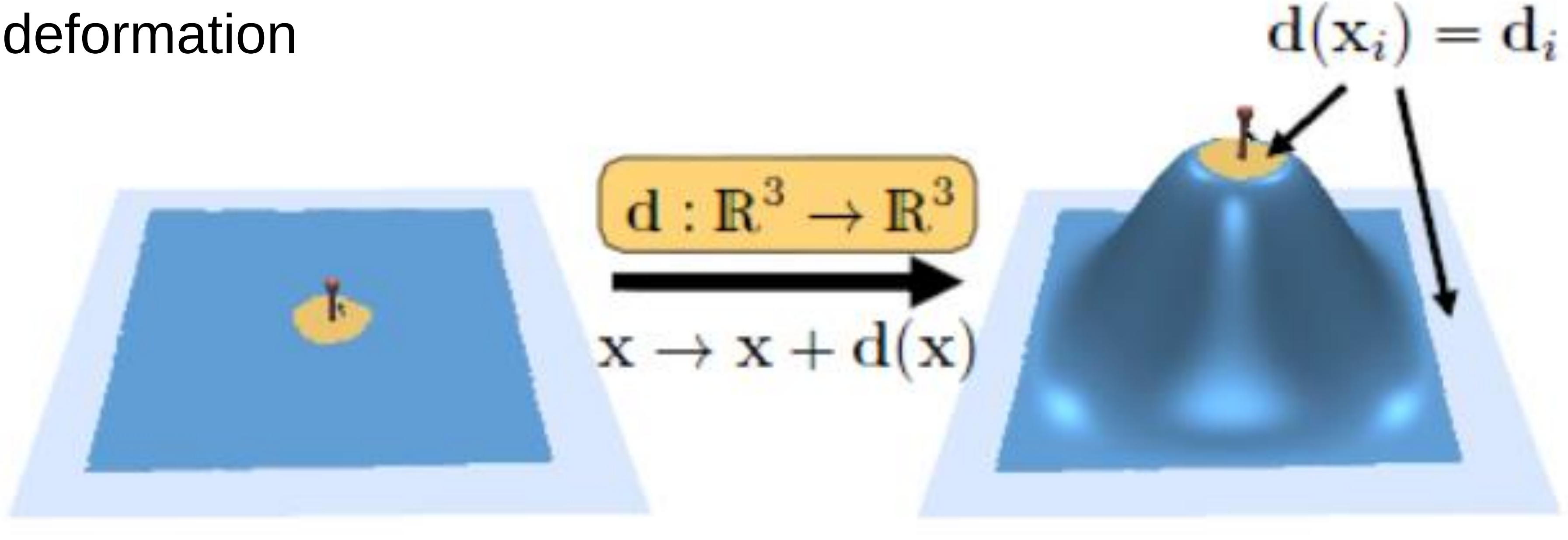
ghostbusting

Types of Deformation

- Space deformation
  - Function-based deformation
  - Free form deformation
  - **Radial Basis Functions**
  - Cage deformation
  - Skeleton deformation
- Surface deformation

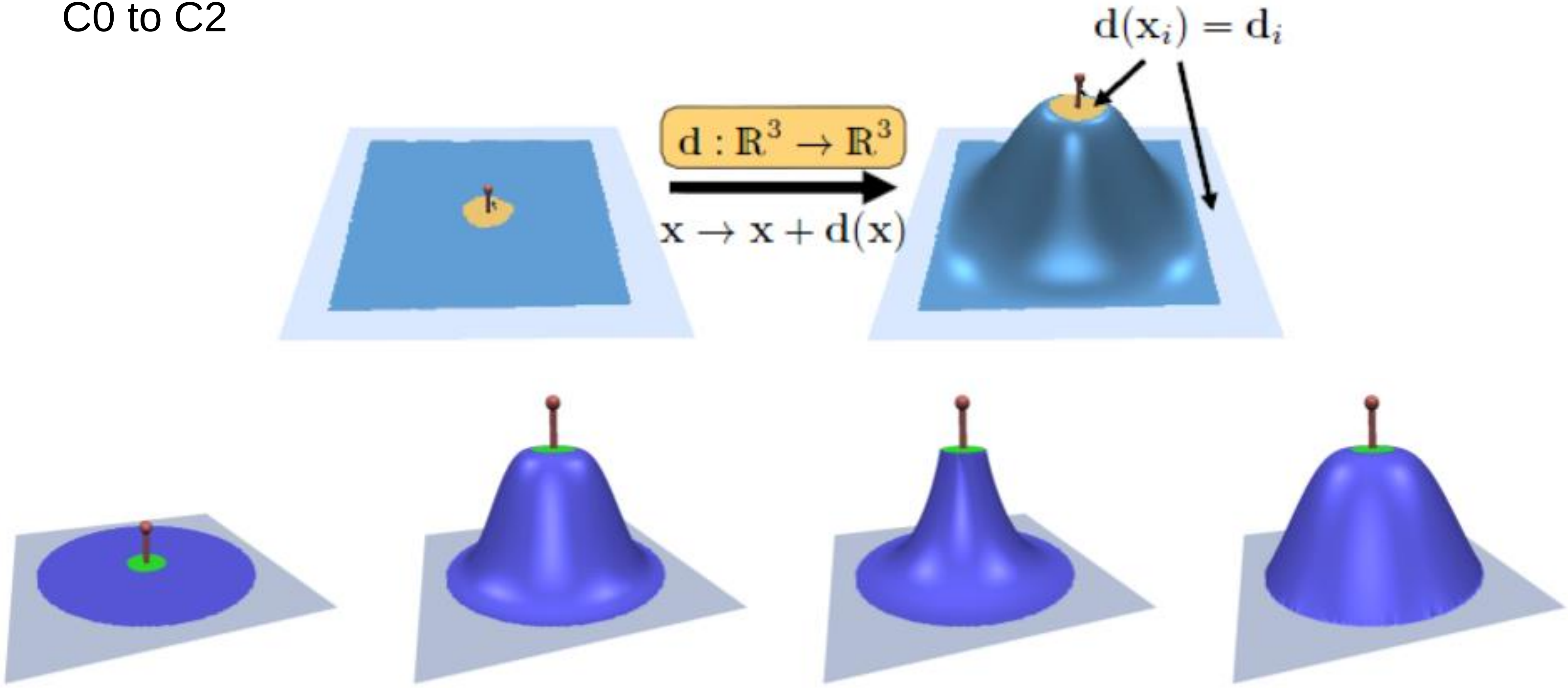
Handle Metaphor [RBF, Botsch & Kobbelt 05]

- FFD: specify displacement at control points
  - not precise, difficult for complex deformation
- Inverse FFD [HHK92]: manipulate the object directly, instead of the lattice
  - minimizing control points' movements, does not necessarily imply a fair deformation of low curvature energy.
- RBF: solve a displacement function **d**(**x**)
  - Interpolate prescribed displacements at vertices of the object directly:
  - Smooth, intuitive deformation



The modeling metaphor

- Support (blue), handle (green), and fixed (gray) region: **p, h, f**
  - Given displacement of **h** & **f**, solve continuously blended **p** between them, from C0 to C2



Volumetric Energy Minimization

- Minimize similar energies to surface case in [Botsch & Kobbelt, SIGGRAPH 04]

$$E_k(d) = \int_{\Omega} \left\|\nabla^k d\right\|^2 du dv$$

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

$$\min E_k(d(x)) \Rightarrow \Delta^k d(x) = 0, x \in \Omega \backslash \delta\Omega$$

s.t. $\Delta^j d(x) = b_j(x), x \in \delta\Omega, j < k$

Minimizing E_k => C_{k-1} blending

membrane surface (k = 1), thin-plate surface (k = 2), minimal curvature variation (k = 3).

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

$$\left(\begin{array}{c|c} \multicolumn{2}{c}{\bar{\Delta}^k} \\ \hline 0 & I_{F+H} \end{array}\right) \begin{pmatrix} \mathbf{p} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix}$$

Real-Time Shape Editing using Radial Basis Functions

[RBF, Botsch and Kobbelt 05]

- Represent deformation by RBFs

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

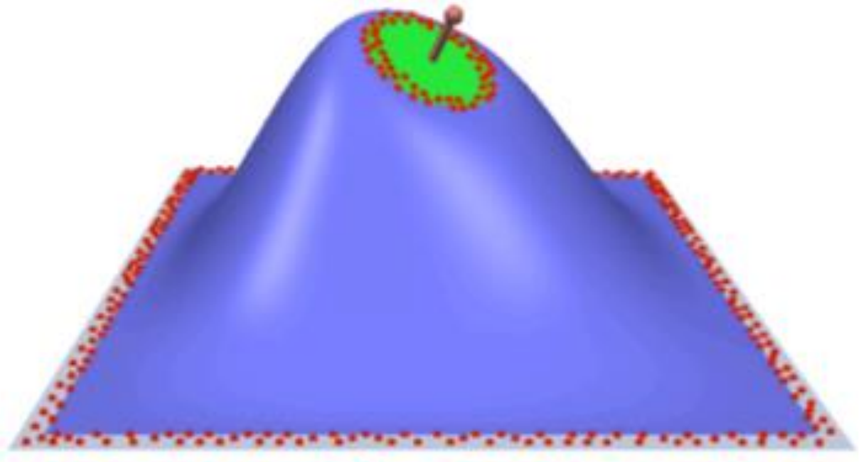
- Duchon [Duc77] showed that d(x) using $\varphi(r) = r^3$
- is triharmonic ($\Delta^3 d=0$) & hence minimizes energy

$$\int_{\Re^3} \left\|\mathbf{d}_{xxx}\right\|^2 + \left\|\mathbf{d}_{xyy}\right\|^2 + \ldots + \left\|\mathbf{d}_{zzz}\right\|^2 \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z \;\rightarrow\; \min$$

- => Highly smooth/fair interpolation: C^2

Real-Time Shape Editing using Radial Basis Functions

[RBF, Botsch and Kobbelt 05]



- Represent deformation by RBFs

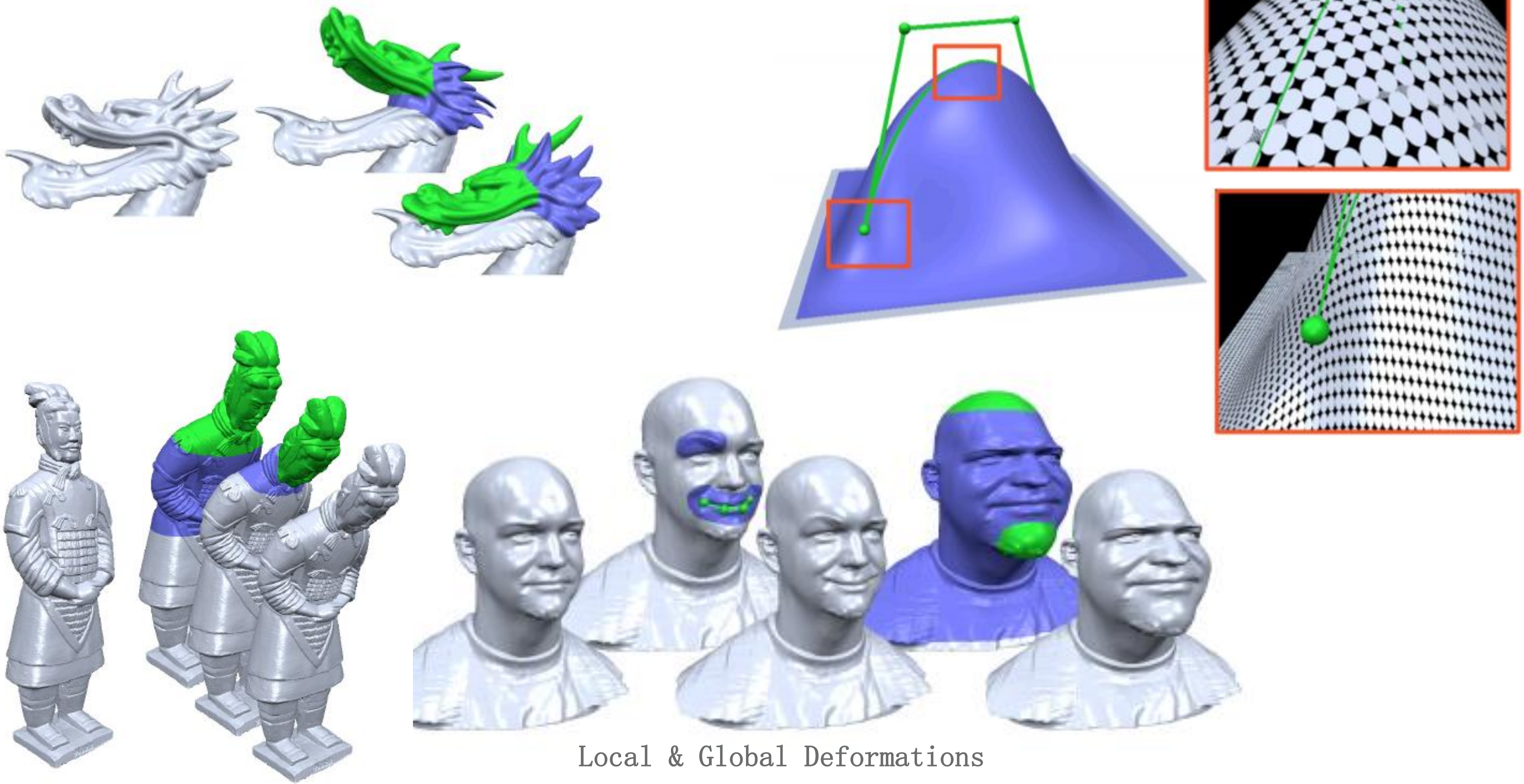
$$\mathbf{d}(\mathbf{x}) = \sum_j \mathbf{w}_j \cdot \varphi(\| \mathbf{c}_j - \mathbf{x} \|)$$

- RBF fitting
  - Interpolate displacement constraints
  - Input constraints: d(fi)=fi & d(hi)=hi=0

$$\varphi(r) = r^3 \qquad \begin{bmatrix} \varphi(\|\mathbf{p}_1 - \mathbf{p}_1\|) & \cdots & \varphi(\|\mathbf{p}_n - \mathbf{p}_1\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|\mathbf{p}_1 - \mathbf{p}_n\|) & \cdots & \varphi(\|\mathbf{p}_n - \mathbf{p}_n\|) \end{bmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{d}}_1 \\ \vdots \\ \bar{\mathbf{d}}_n \end{pmatrix}$$

  - is globally supported => dense sys.
  - Hence using a incremental solver …
  - Evaluate on the GPU!

Results

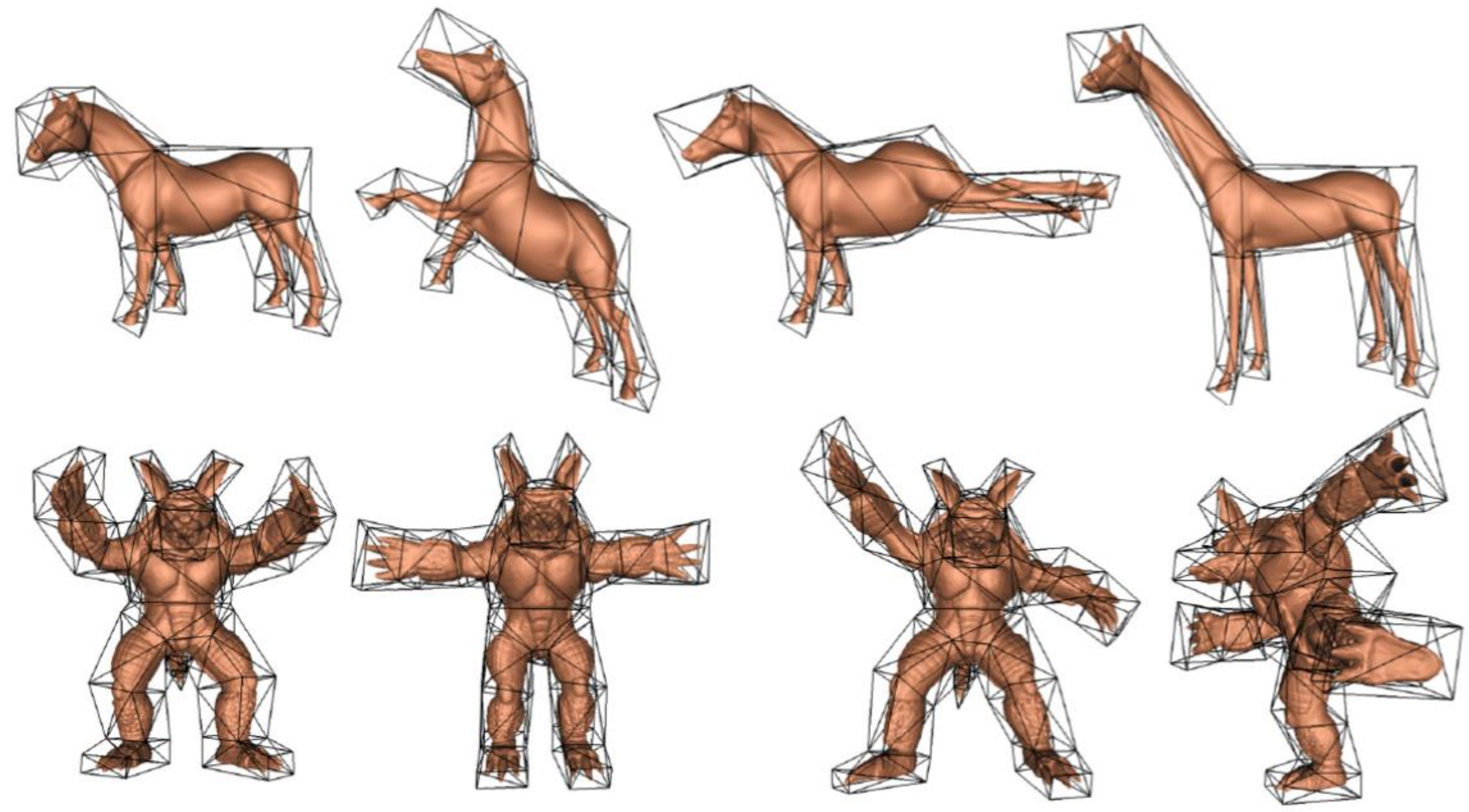


Local & Global Deformations

Cages

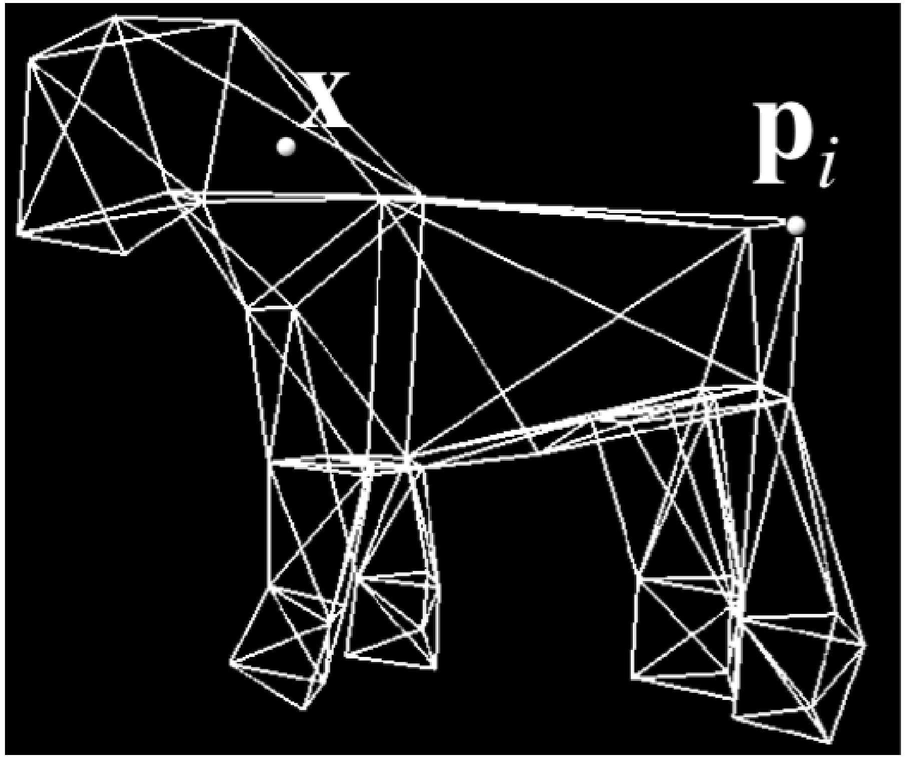
Case-based Deformation [Ju et al. 2005]

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



Case-based Deformation [Ju et al. 2005]

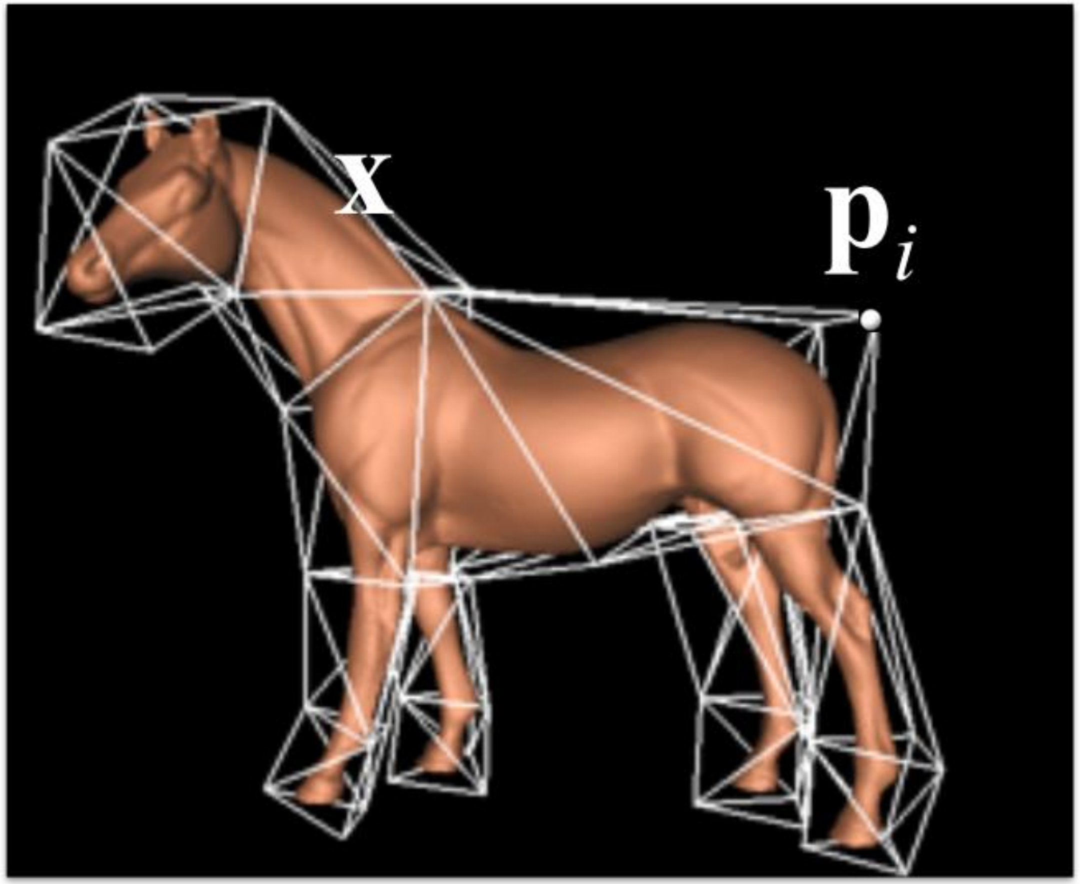
- Each point x in space is represented w.r.t. to the cage elements using coordinate functions
- Some kind of inner distance

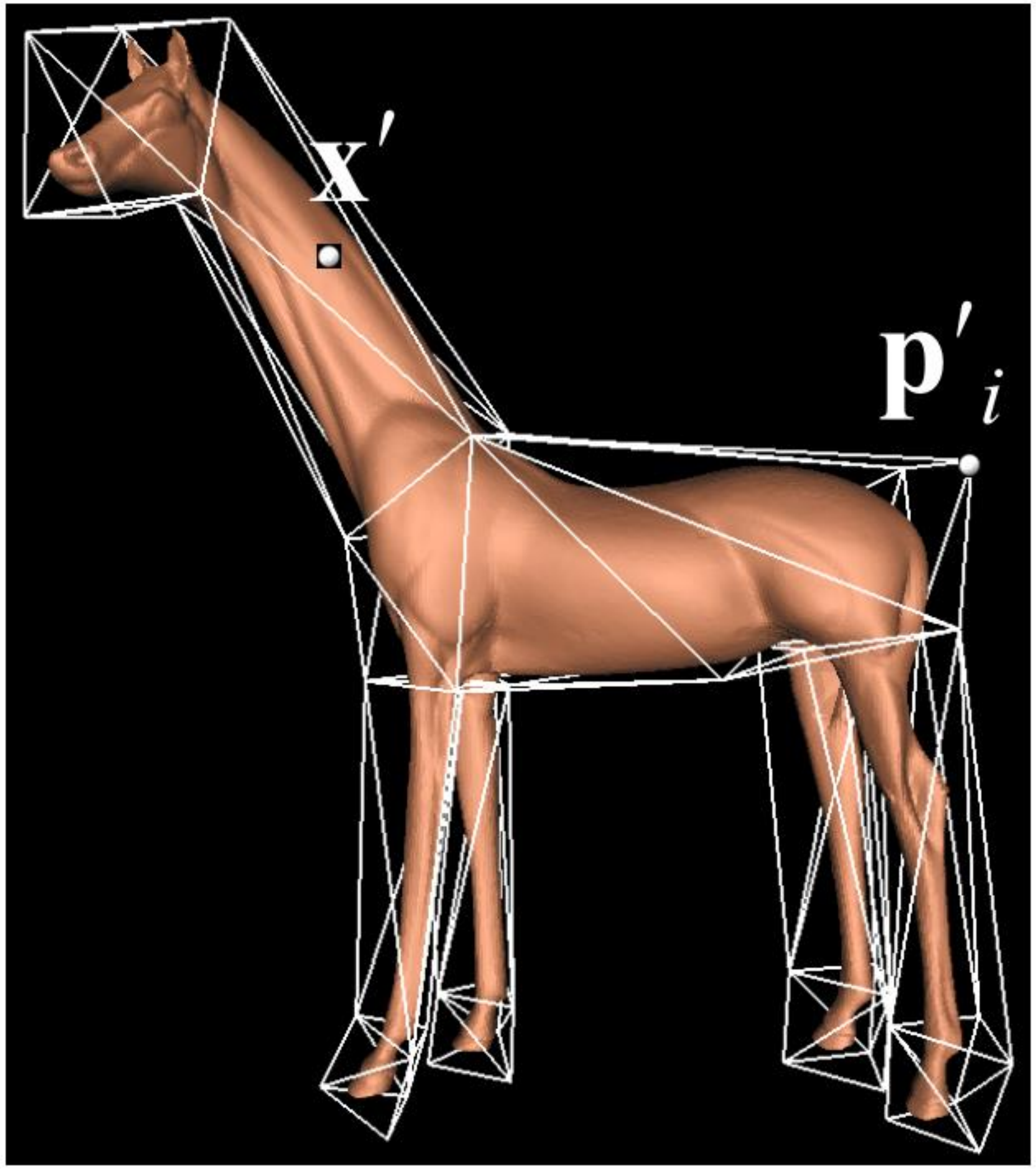


$$\mathbf{x} = \sum_{i=1}^{k} w_i(\mathbf{x})\, \mathbf{p}_i$$

Case-based Deformation [Ju et al. 2005]

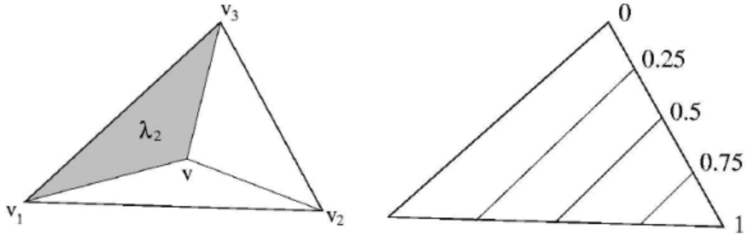
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}'_i$$



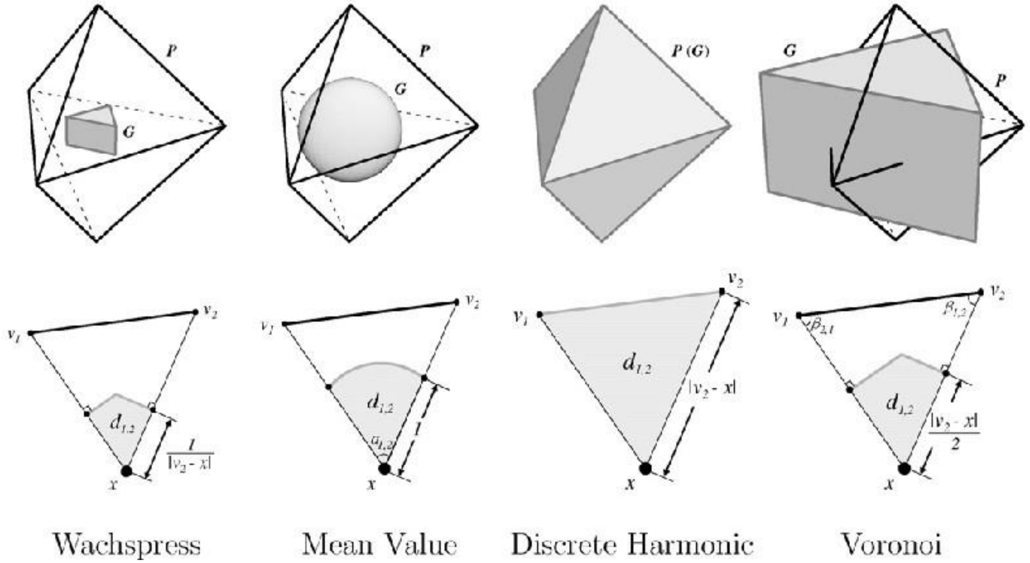


GBC

1. Basic



2. Algebra construction[FHK06]: 3-point coords and 5-point coords
3. Geometric construction [JLW07] :



4. For smooth shapes [SJW06] :

Constructing a function that interpolates values along a continuous curve using linear coordinates

five frameworks to GBC

1. Algebraic construction: by analysis the bounds of BC. **[FHK06]**
2. Algebraic construction: **[MD05], [MLD05], [SM06]**
3. Geometric construction: by Polar Duals. **[JSWD05], [JLW07]**
4. Interpolation: by Dual Shephard interpolant. **[SJW06]**
5. Transfinite Interpolation: by Gordon and Wixom scheme. **[Alexander06]**

Various of barycentric coordinates

- Mean value coordinates(MVC)
- Positive mean value coordinates(PMVC)
- Harmonic coordinates(HC)
- Boundary element method of HC(BEM)
- Maximum entropy coordinates(MEC)
- Higher Order Barycentric Coordinates(HOBC)
- Green coordinates(GC)
- Complex barycentric coordinates(CBC)
- Others…

Generalized Barycentric Coordinates

- Lagrange property

$$w_i(\mathbf{p}_j) = \delta_{ij}$$

- Reproduction

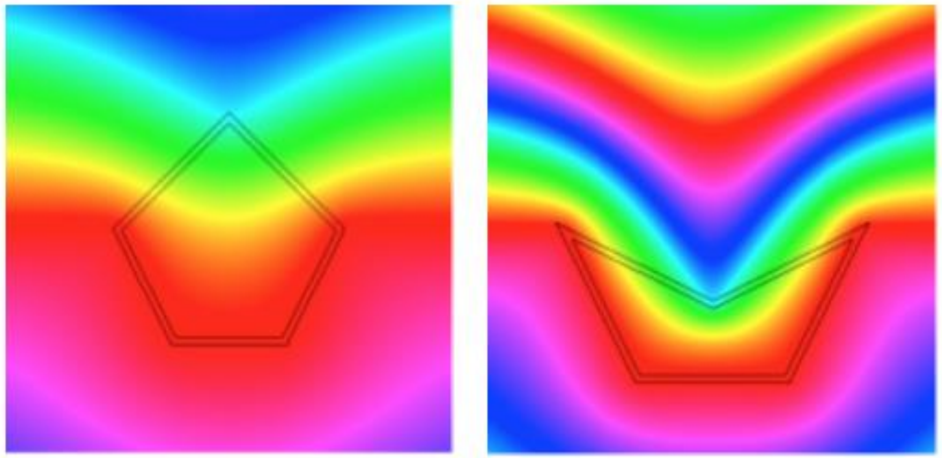
$$\forall \mathbf{x},\ \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}_i = \mathbf{x}$$

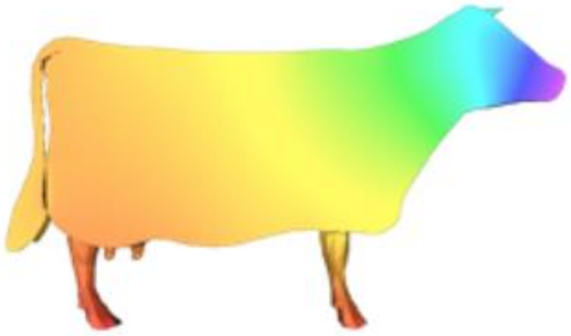
- Partition of unity

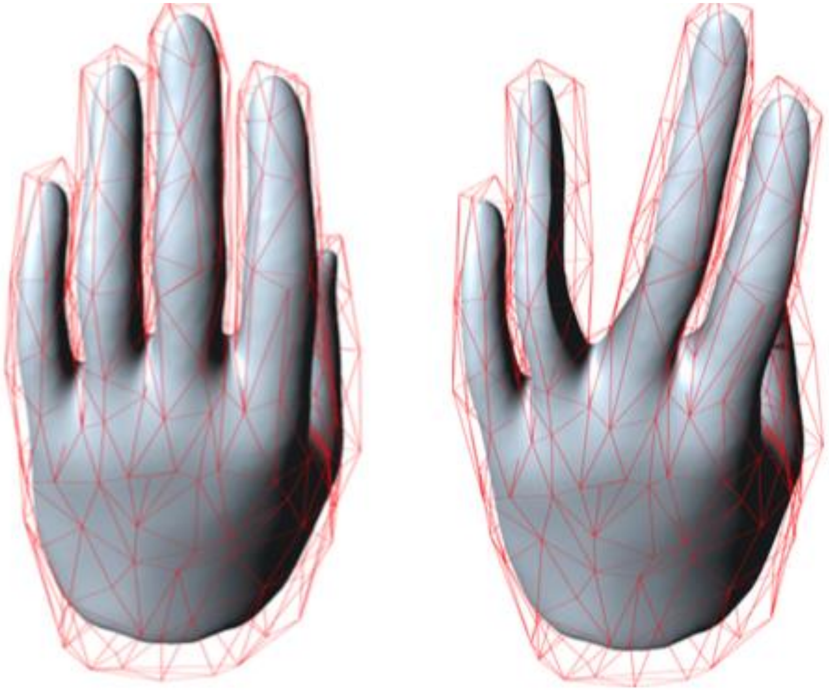
$$\forall \mathbf{x},\ \sum_{i=1}^{k} w_i(\mathbf{x}) = 1$$

Mean-value coordinates [Floater 2003, Ju et al. 2005]

- Closed-form solution for *wi* (**x**)
- Not necessarily positive on non-convex domains



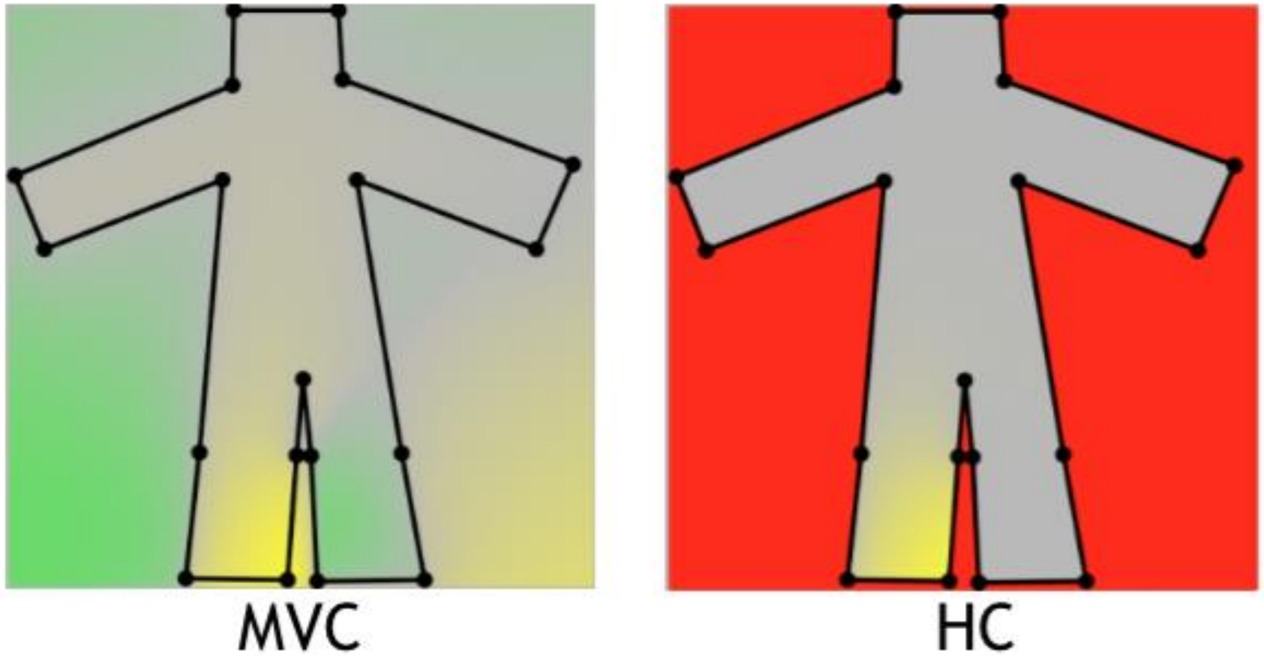




Harmonic coordinates (Joshi et al. 2007)

- 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. $h_i(\mathbf{p}_j) = \delta_{ij}$

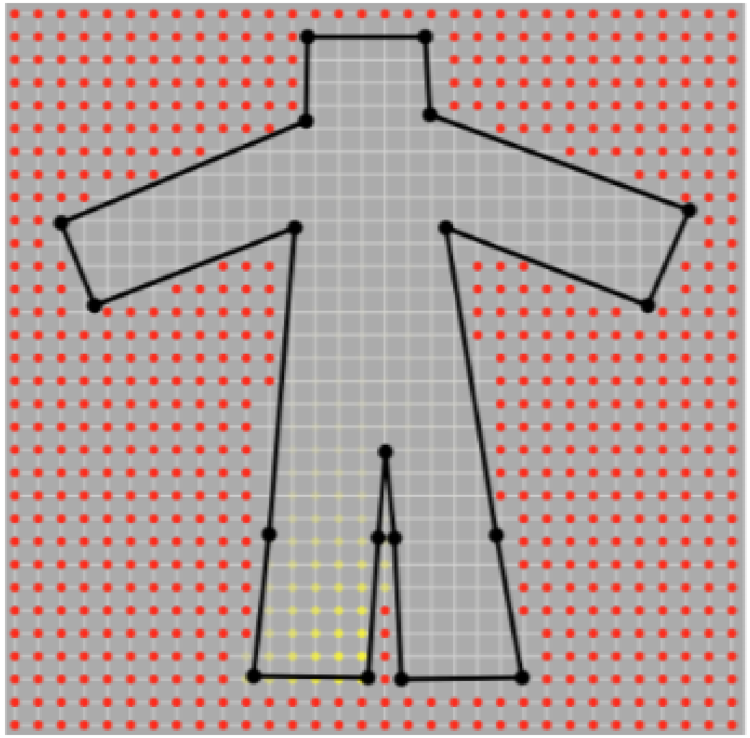


Harmonic coordinates (Joshi et al. 2007)

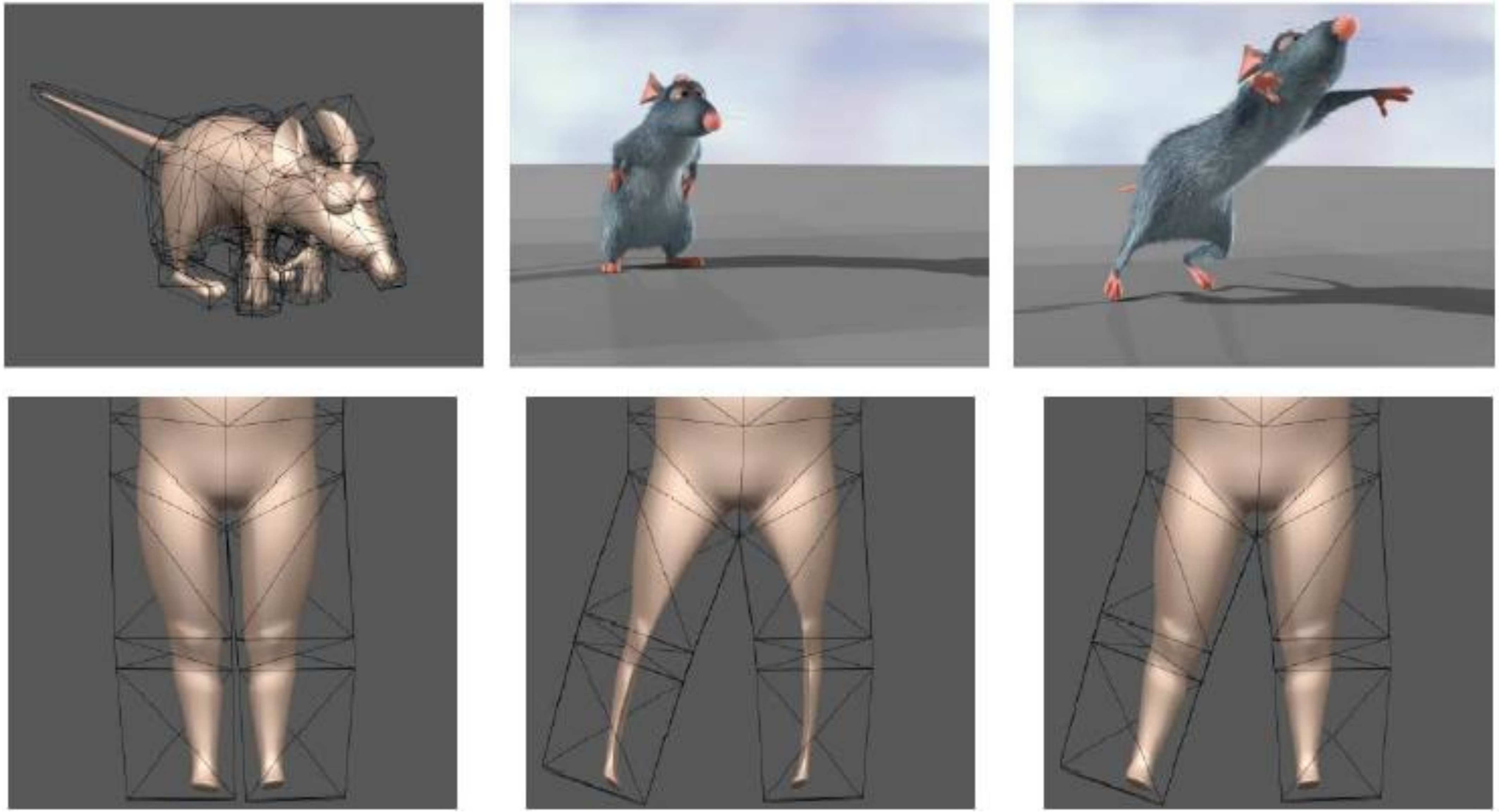
- Harmonic functions $h_i(\mathbf{x})$ for each cage vertex $\mathbf{p}_i$
- Solve $\Delta\, h = 0$

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

- Volumetric Laplace equation
- Discretization, no closed-form

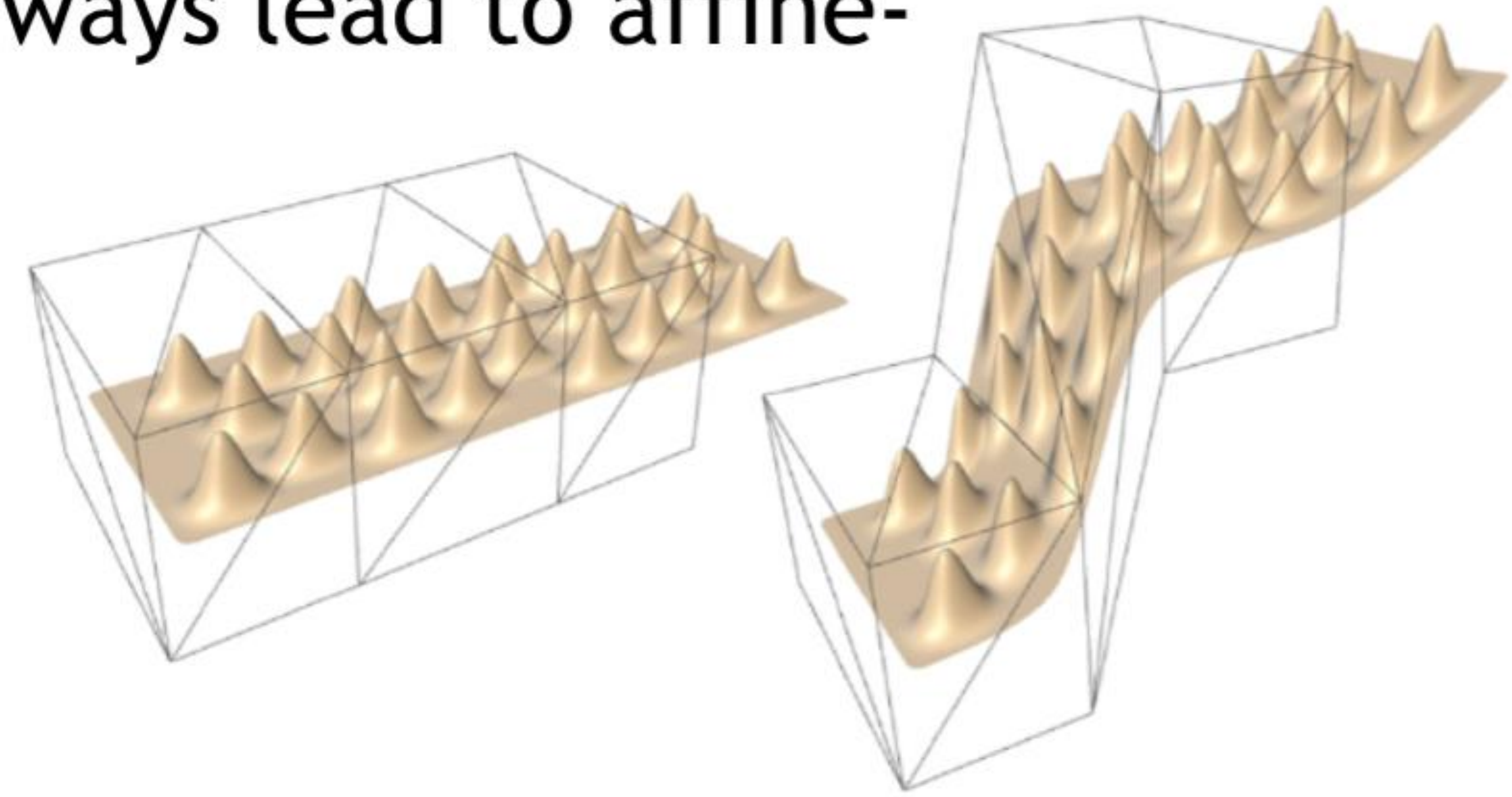


Harmonic coordinates (Joshi et al. 2007)



Green coordinates (Lipman et al. 2008)

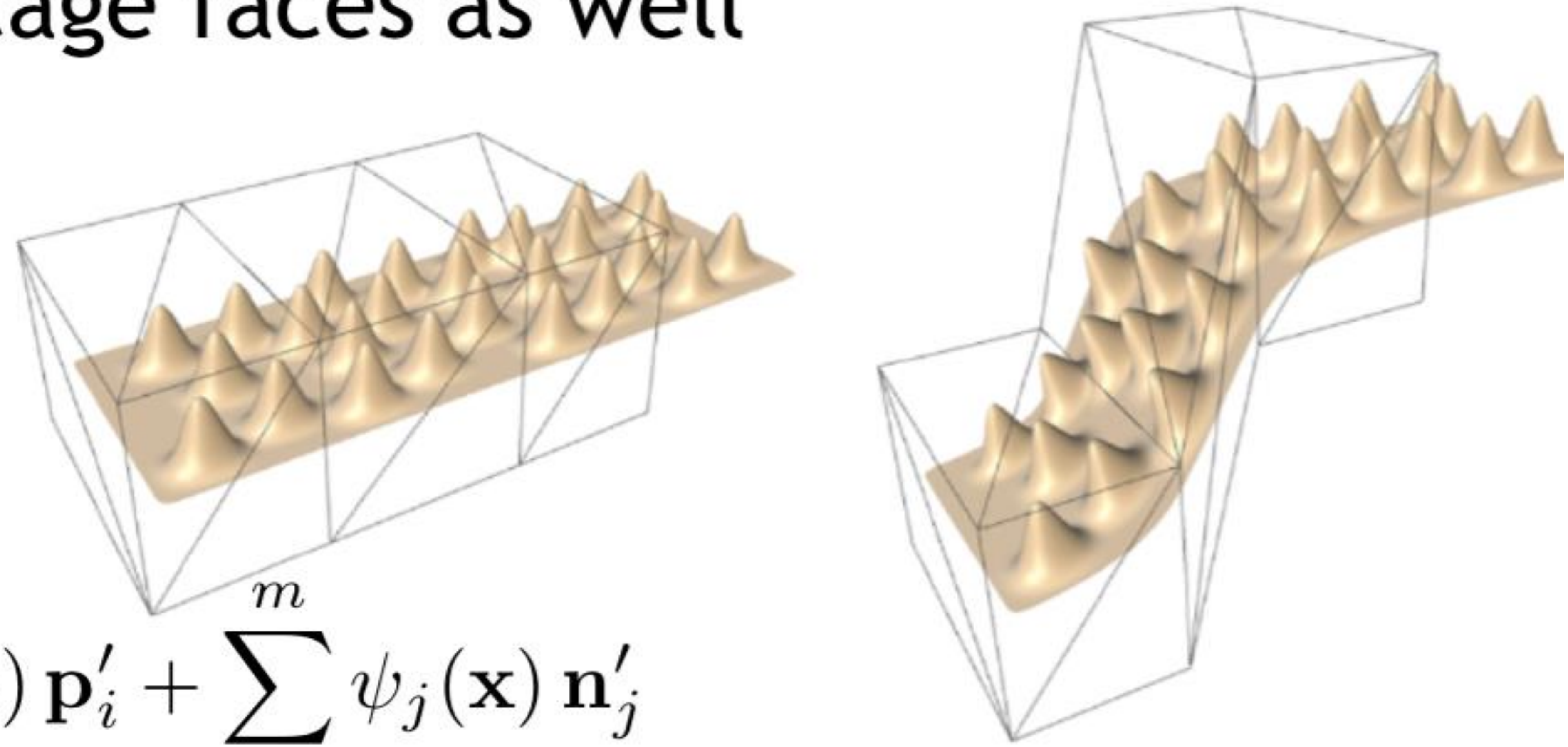
- Observation: previous vertex-based basis functions always lead to affine-invariance!



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

Green coordinates (Lipman et al. 2008)

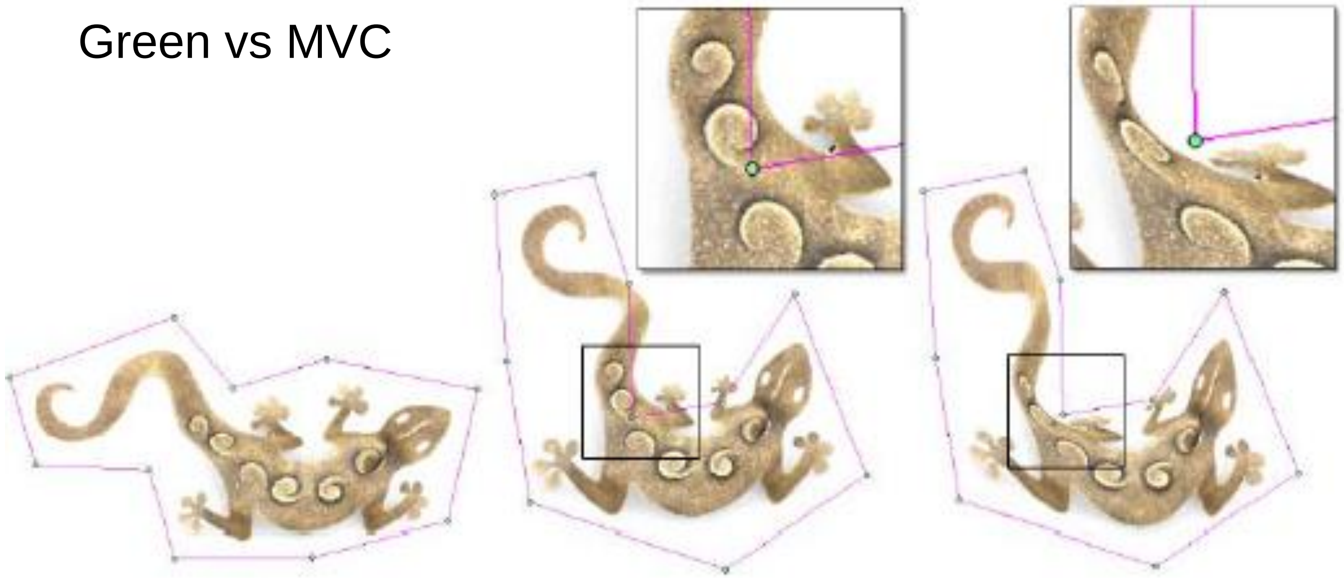
- Correction: Make the coordinates depend on the cage faces as well



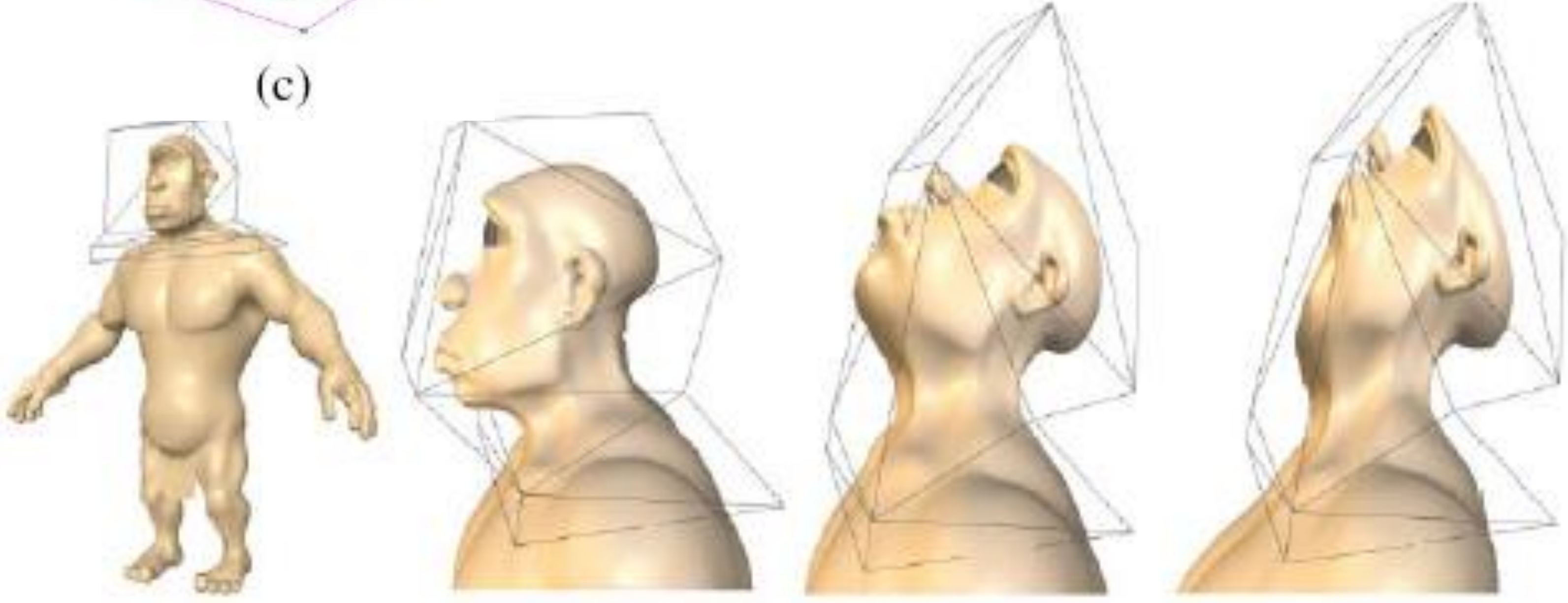
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x})\,\mathbf{p}'_i + \sum_{j=1}^{m} \psi_j(\mathbf{x})\,\mathbf{n}'_j$$

Green coordinates (Lipman et al. 2008)

Green vs MVC



(a) (b) (c)

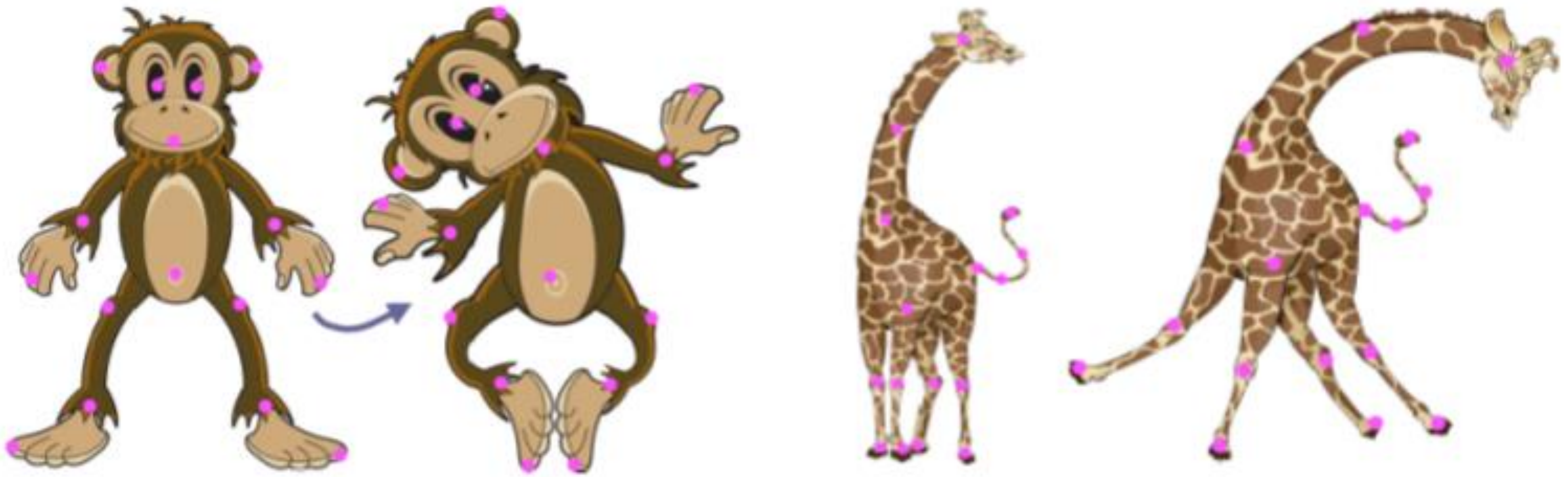


(d) (e) (f)

Coordinate Functions

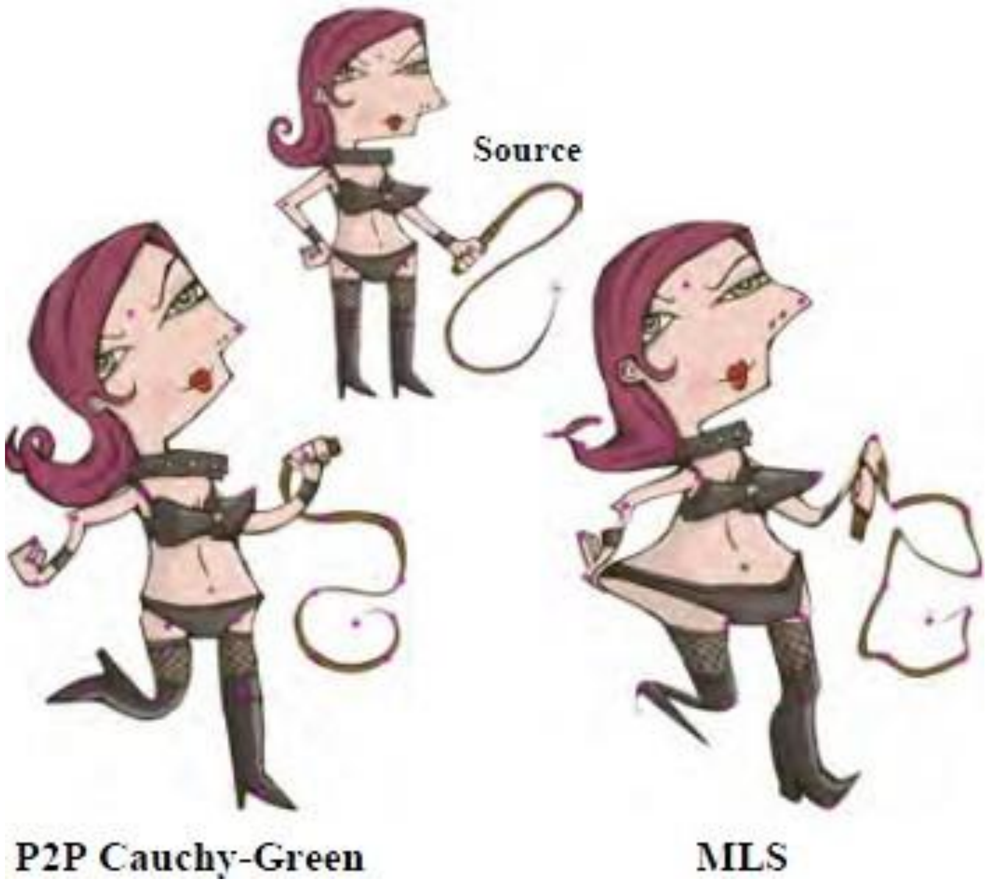
- Green coordinates (Lipman et al. 2008)
- Closed-form solution
- Conformal in 2D, quasi-conformal in 3D

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



Cauchy Coords --[WBG09]

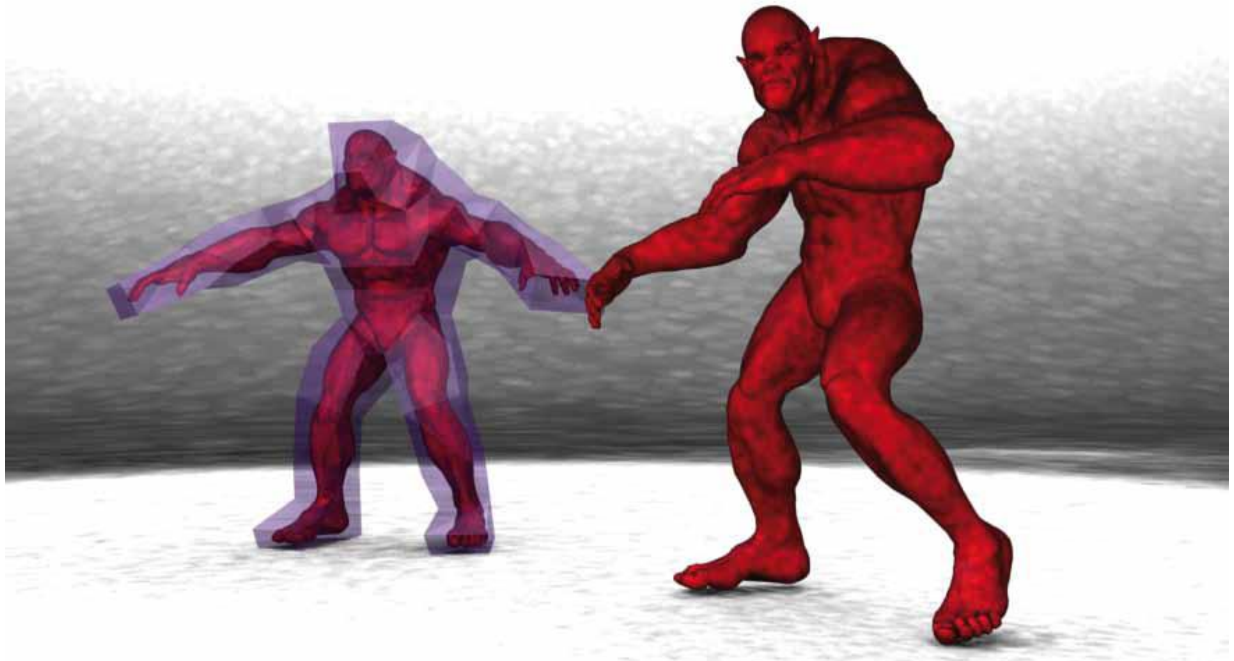
- Better control;
- Points based coordinates;



Variational Harmonic Coords--[BWG09]

Contributions:

① The mapping has closed-form expression
② Constraints defined as non-linear energy functional of the mapping
③ No explicit discretization of the domain
④ robust, provably convergent, and easy to implement



Cage-Based Methods: Summary

- Pros:
  - Nice control over volume
    - Squish/stretch
- Cons:
  - Hard to control details of embedded surface

References -- Deformation

[SDHW05]	**Barycentric Coordinates for Convex Sets, 05**
[WBG09]	Complex Barycentric Coordinates with Applications to Planar Shape Deformation_eg09
[LLC08]	Green Coordinates_sig08
[JMD07]	Harmonic Coordinates for Character Articulation_tog07
[RLF08]	Interior Distance Using Barycentric Coordinates_sgp08
[TSJ05]	Mean value coordinates for closed triangular meshes_tog05
[Cohen09]	Space Deformations, Surface Deformations and the Opportunities In-Between 09
[BWG091]	Space Deformation Transfer_sca09
[BWG09]	Variational Harmonic Maps for Space Deformation_TOG09

References -- GBC

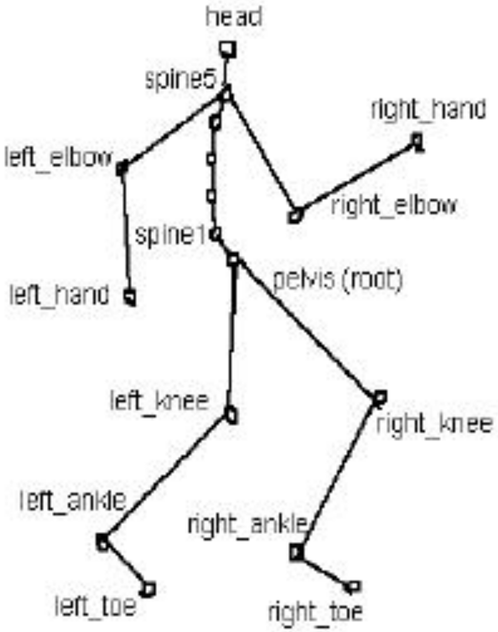
[FHK06]	A general construction of barycentric coordinates over convex polygons 06
[JLW07]	A general geometric construction of coordinates in a convex simplicial polytope, CAGD07
[SJW06]	A Unified, Integral Construction For Coordinates Over Closed Curves, CAGD06
[MD05]	Algebraic construction of smooth interpolants on polygonal domains. 05.
[MD04a]	Interpolations for temperature distributions: A method for all non-concave polygons. 04a.
[Alexander06]	On Transfinite Barycentric Coordinates, SGP06
[SM06]	Recent advances in the construction of polygonal finite element interpolants. 06.
[MLD05]	Smooth two dimensional interpolants: a recipe for all polygons. 05.
[MD04b]	Shape functions for polygonal domains with interior nodes. 04b.

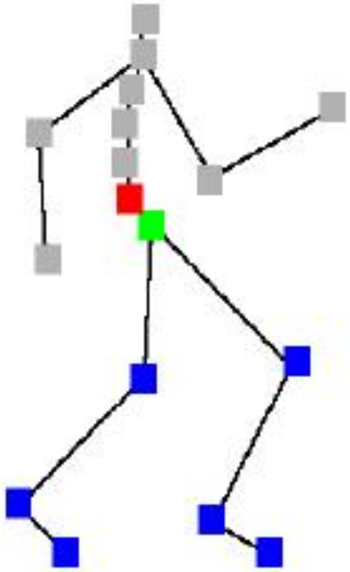
Deformation

- To deform a model, move its control points.
  - The rest is details…
- Types of deformation:
  - Function-based deformation
  - Space deformation
  - **Skeleton deformation**
  - Physically-based deformation
  - Laplacian-based deformation

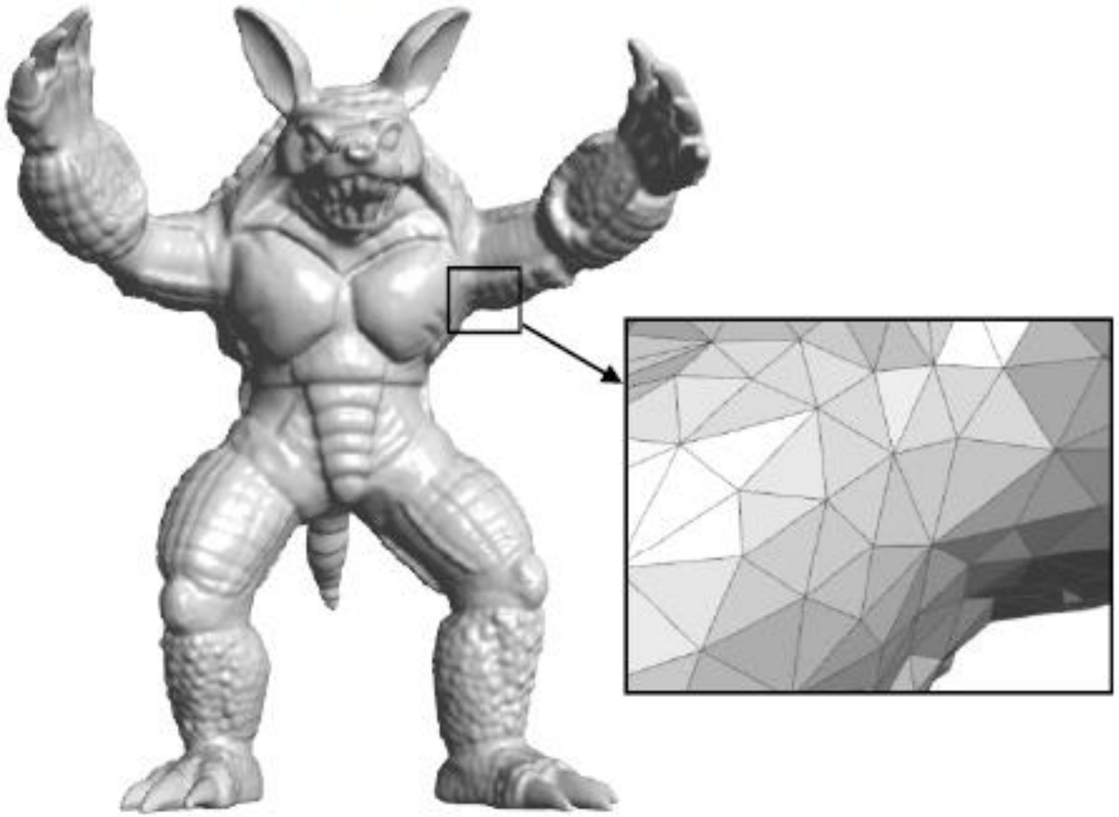
Skeletal deformation

- Skeleton:
  - collection of line segments
  - connected by joints





- Skin:
  - discrete samples of the surface
  - polygonal mesh

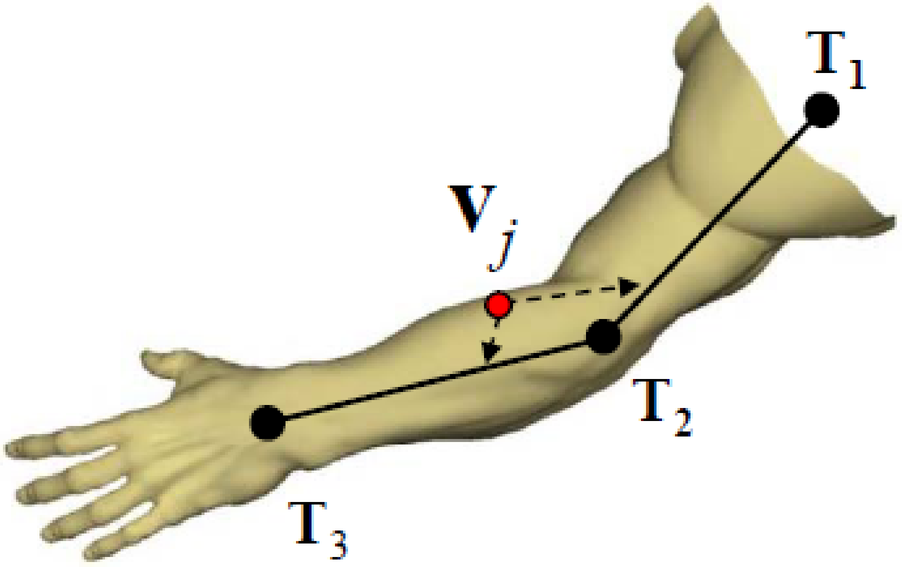


Skeletal subspace deformation

[Han-Bing Yan el at. 2008]

- Affine combination of transformations

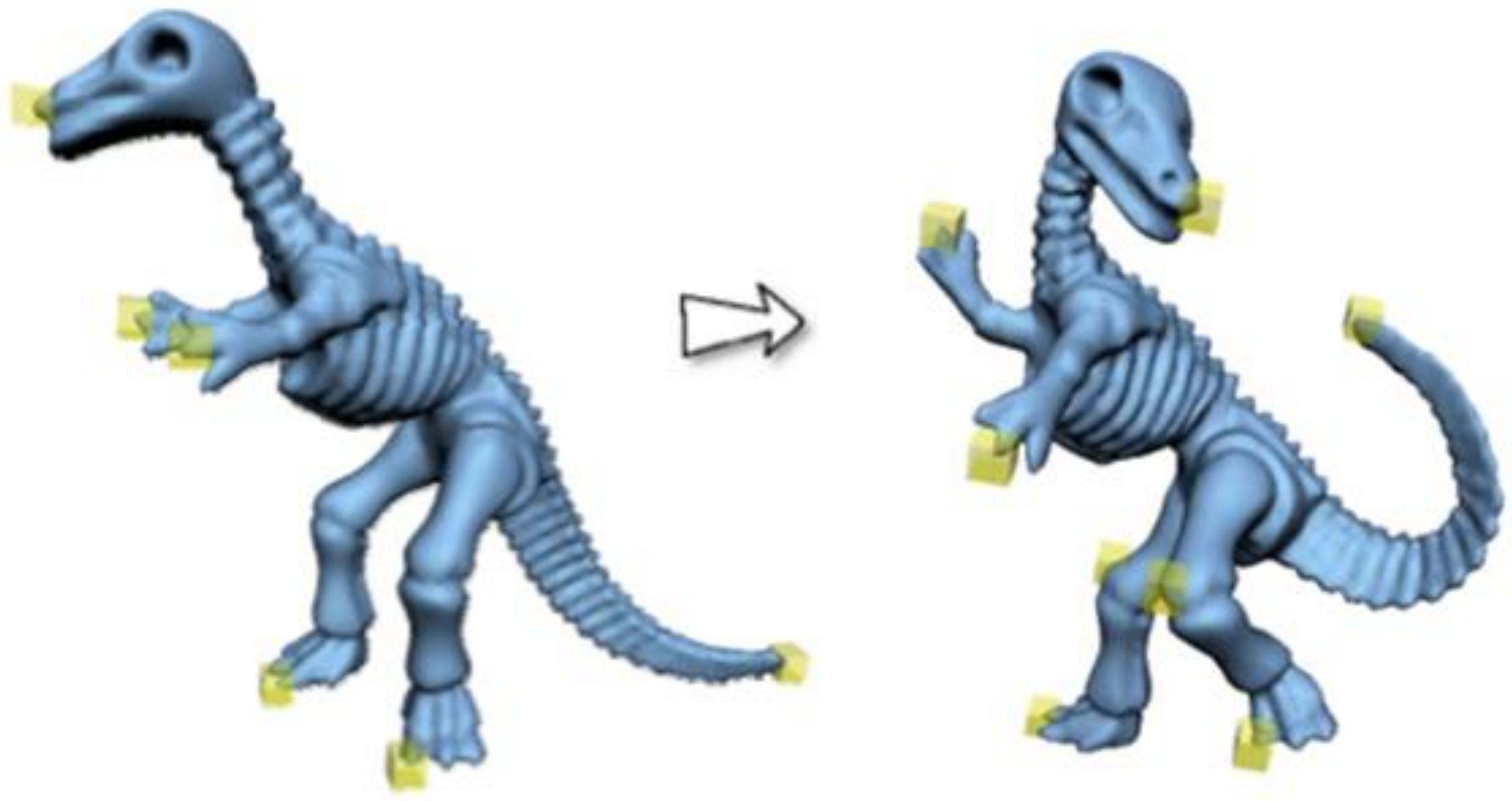
$$\mathbf{v}'_j = \sum_{k=1}^{K} w_{kj} \cdot \mathrm{T}_k \cdot \mathbf{v}_j$$



- De facto standard for interactive applications – simple + fast + works on the GPU

Nonlinear Space Deformations

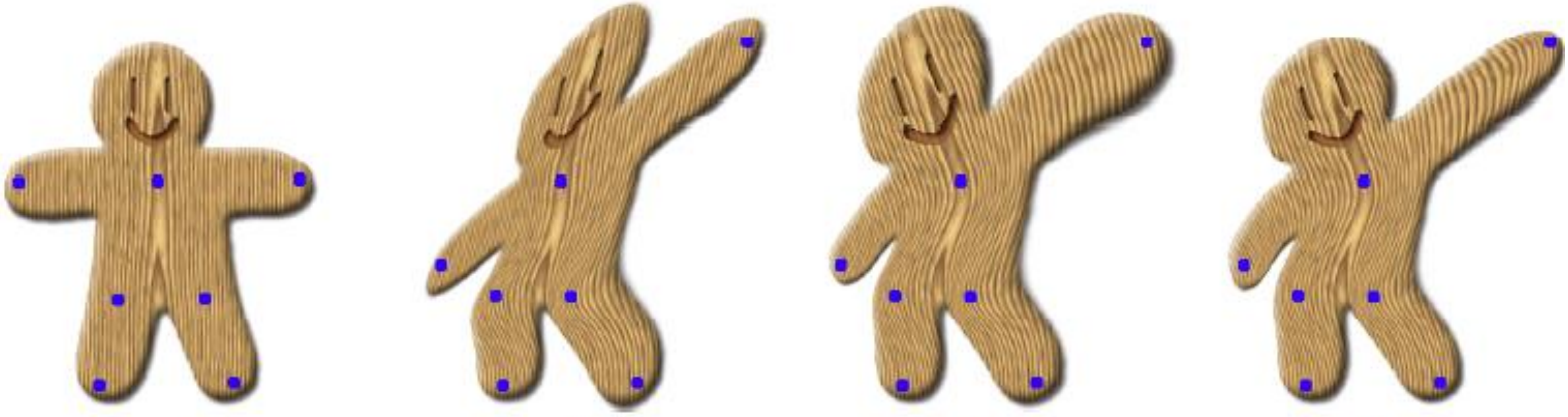
- Involve nonlinear optimization
- Enjoy the advantages of space warps
- Additionally, have shape preserving properties



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)

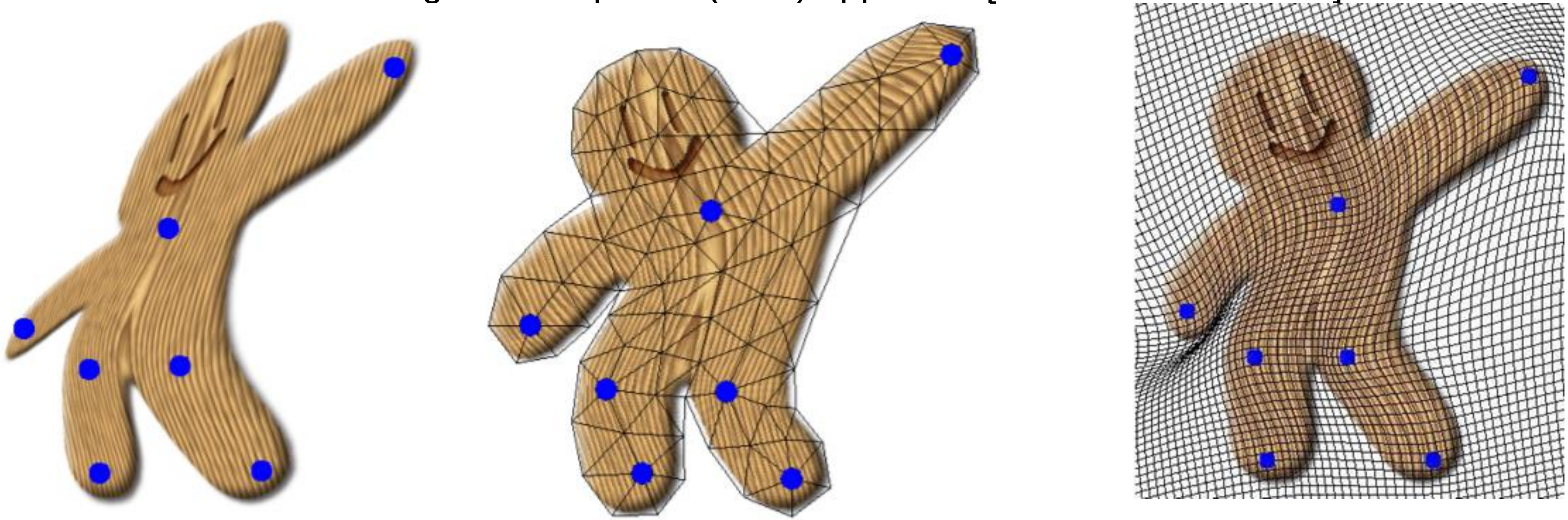


(a) (b) (c) (d)

affine transformations (b), similarity transformations (c) and rigid transformations (d).

Motivation

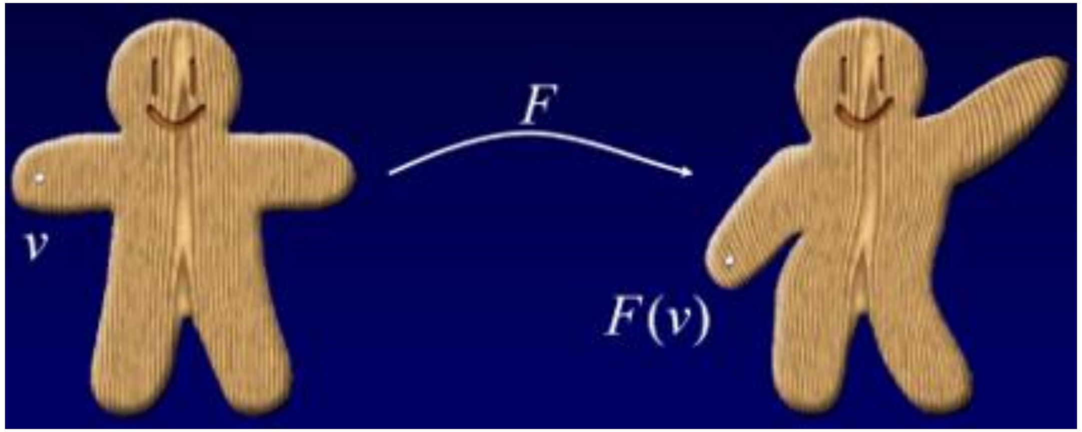
Moving Least Squares (MLS) approach [Schaefer et al. 2006]



- Left: Deformation using thin plate splines [Bookstein 1989]. The deformation is smooth but lacks realism.
- Center: method by Igarashi et al. shown with triangulation. The lack of smoothness is clearly visible in the wood grain. **Slow**
- Right: MLS solves a small linear system (2×2) at each point in a uniform grid. **Fast**

(Image) Deformation

- Controlled by handles: points, lines…
- Function f
  - Interpolation: f(pi)=qi (handles)
  - Smoothness: smooth deformations
  - Identity: qi=pi => f(x)=x
- Similar to scattered data interpolation



- RBF and MLS are both interpolation techniques!

As Rigid As Possible Deformation

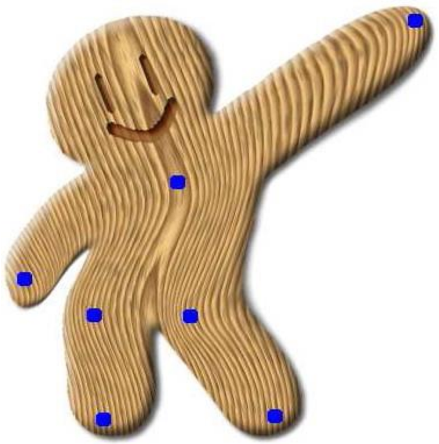
Moving Least Squares (MLS) approach [Schaefer et al. 2006]

- Attach an affine transformation to **each point x$\in$ R3, i.e. each grid point**:

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

- Satisfy: $A_{\mathbf{x}}(\mathbf{p}_{\mathbf{i}})=q_i$

- The space warp: $\mathbf{x} \rightarrow A_{\mathbf{x}}(\mathbf{x})$
  - It is different then compute a single function for the whole space
  - It compute a different Ax for each point of the space.



As Rigid As Possible Deformation

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

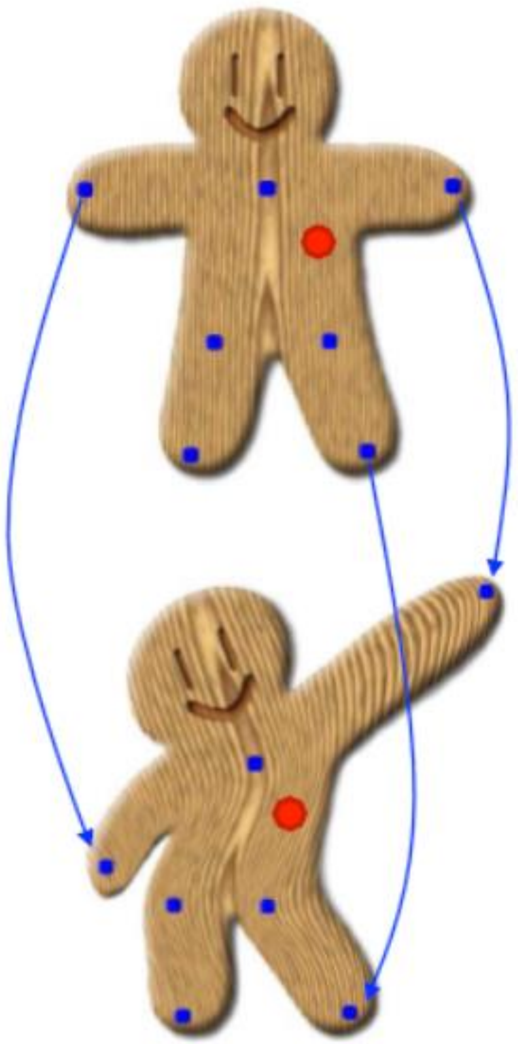
- Handles $\mathbf{p}_i$ are displaced to $\mathbf{q}_i$
- The local transformation at $\mathbf{x}$:

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

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

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

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



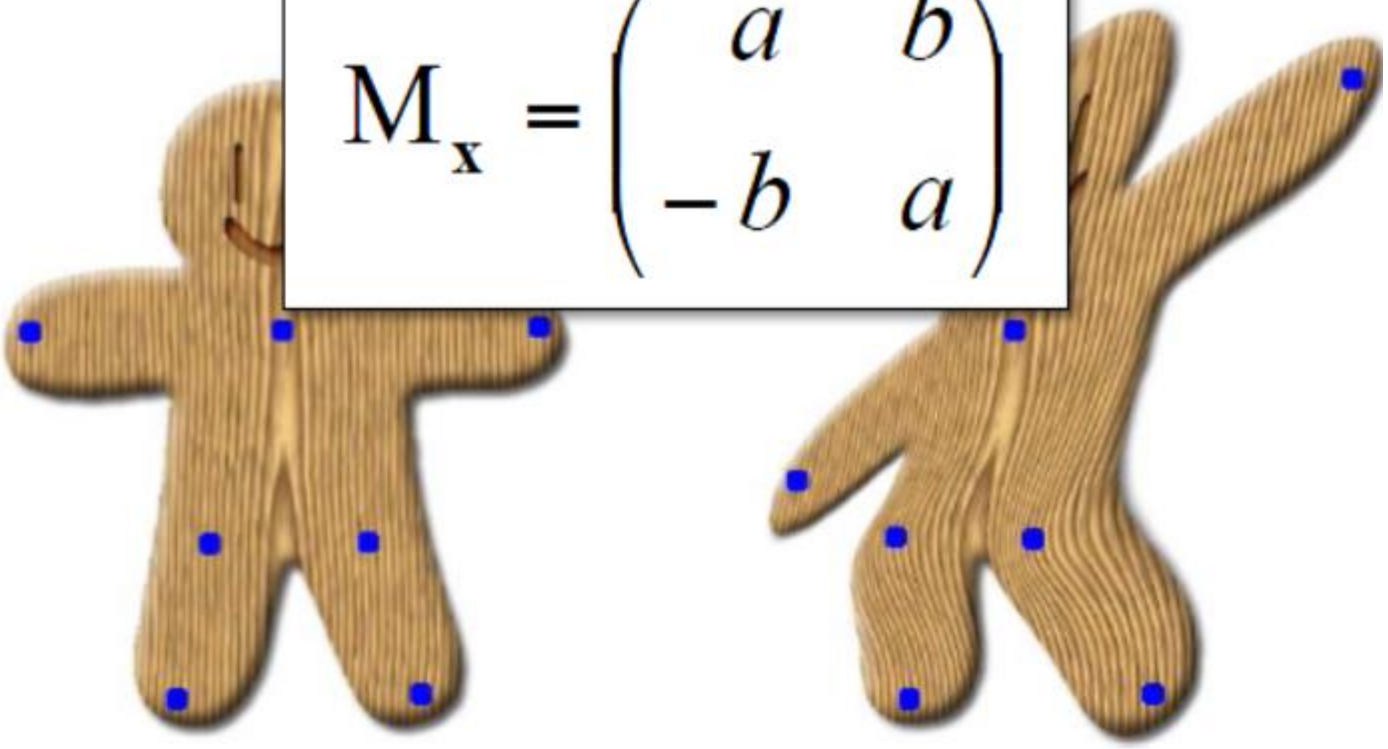
Why called MLS?

Because the weights *wi(x)* in this least squares problem are dependent on the point of evaluation *v*, we call this a *Moving Least Squares* minimization. Therefore, we obtain a different transformation *Ax(p)* for each x.

As Rigid As Possible Deformation

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

Restrict A**x**(·) to similarity transformation

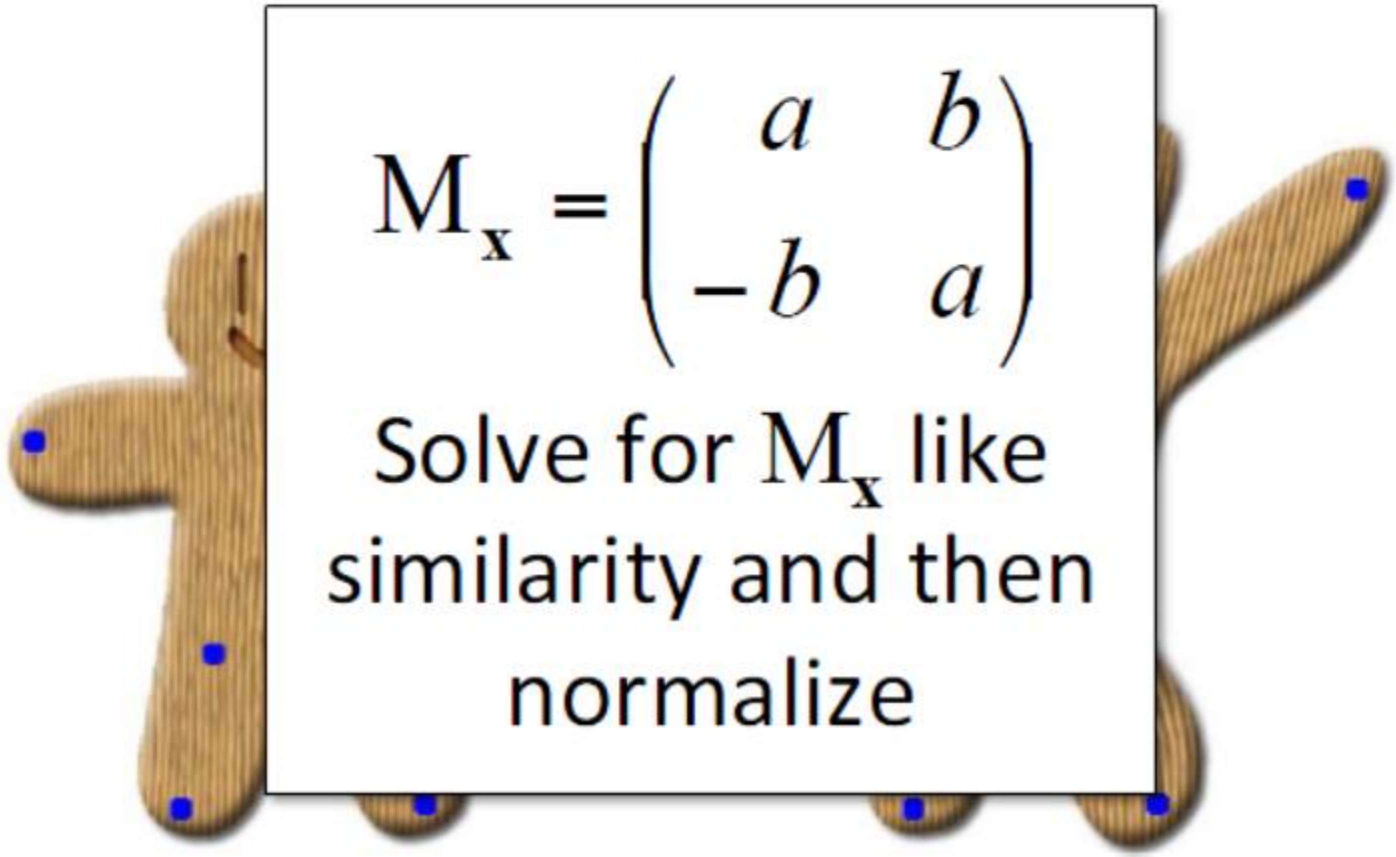


$$\mathrm{M}_{\mathbf{x}} = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}$$

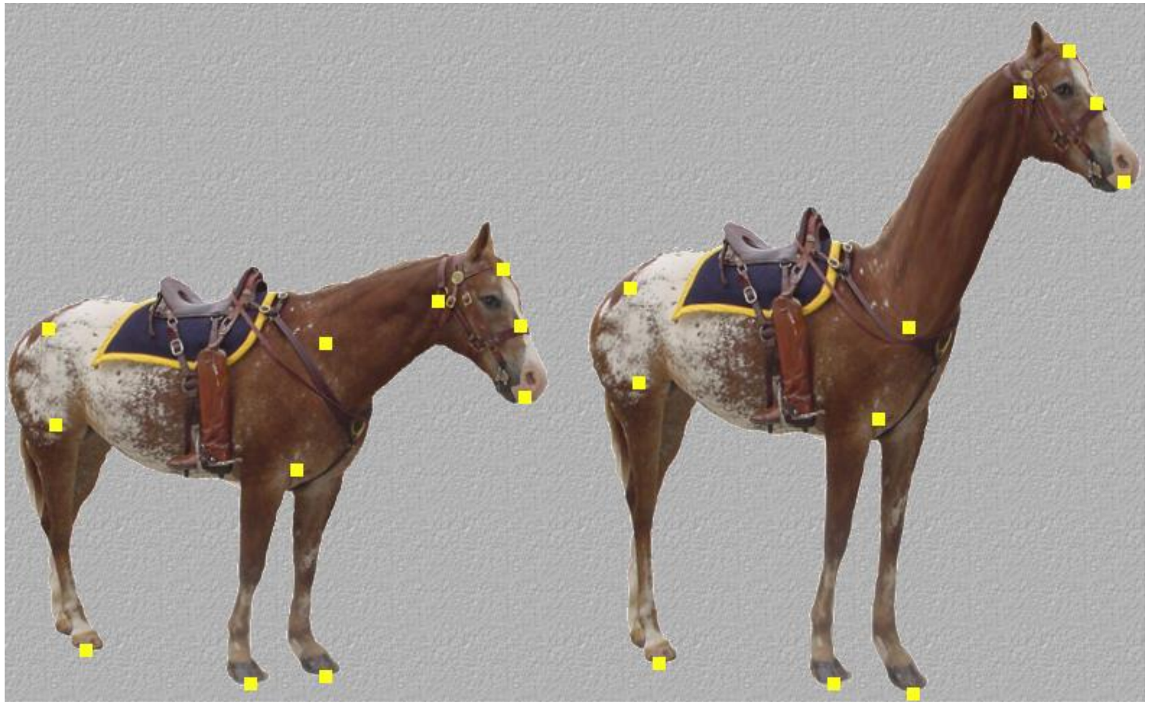
As Rigid As Possible Deformation

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

Restrict A**x**(·) to rigid transformation



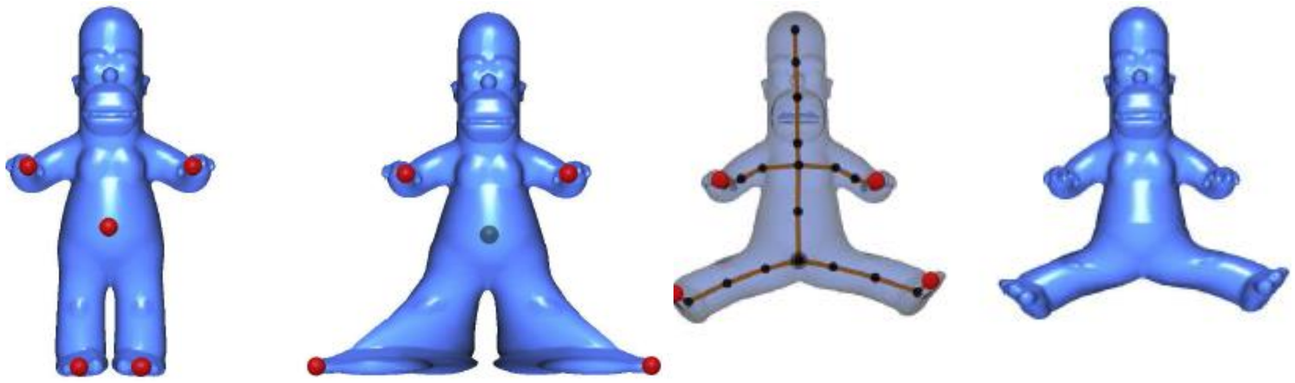
Examples



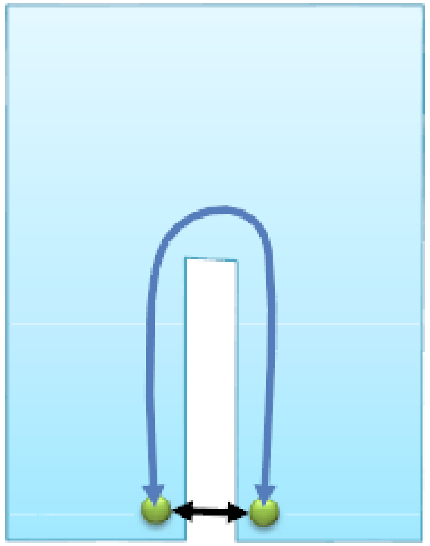
Comparison of the line deformation method of Beier etal. (left) with the Rigid MLS deformation (right).

Limitations - interpolation based space deformations

- Deforms all space - is not “shape aware”

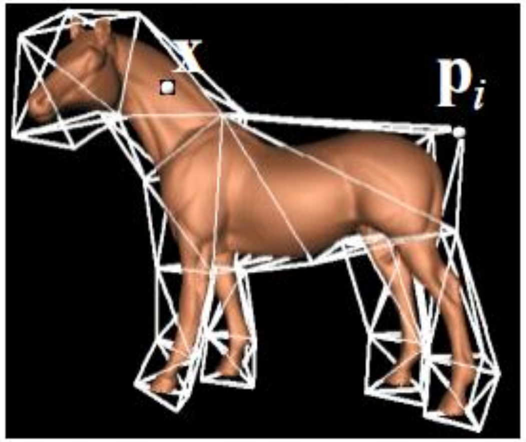


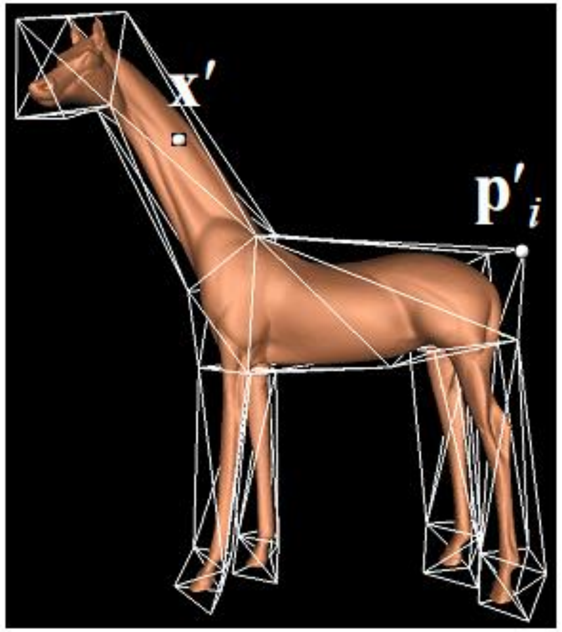
- The “Pants” Problem
  - Small Euclidean distance Large **geodesic** distance
  - Actually, we don’t care about distortion **outside** the shape



Solutions - Cages

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





- The clam and the chameleon's mouth cannot be opened using cage-based methods.

As Rigid As Possible Deformation

MLS approach –extension to 3D [Zhu & Gortler2007]

- 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 3*3 covariance matrix

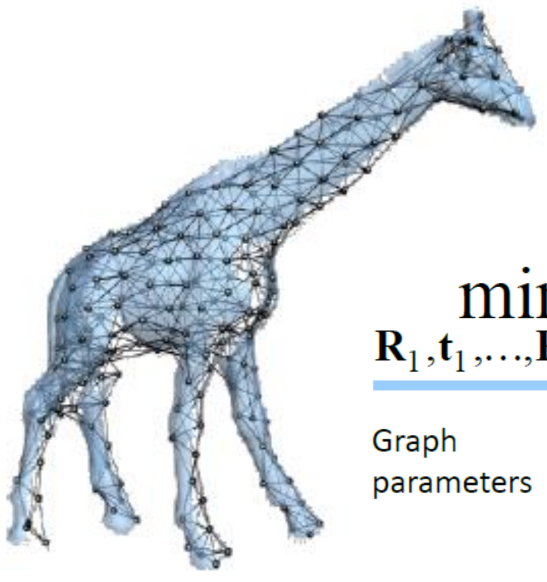
- Also replace Euclidean distance in the weights by "**distance within the shape**"

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

As Rigid As Possible Deformation

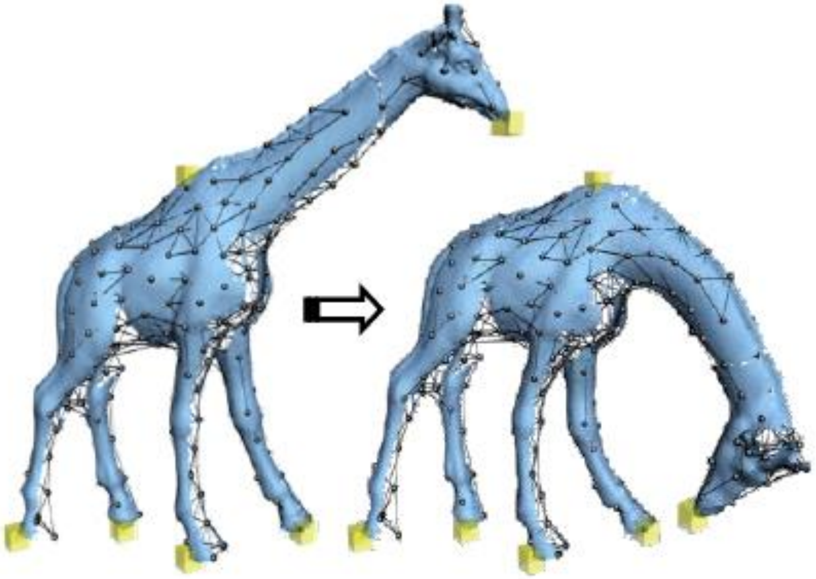
Embedded Deformation [Sumner et al. 07]

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



Deformation Graph

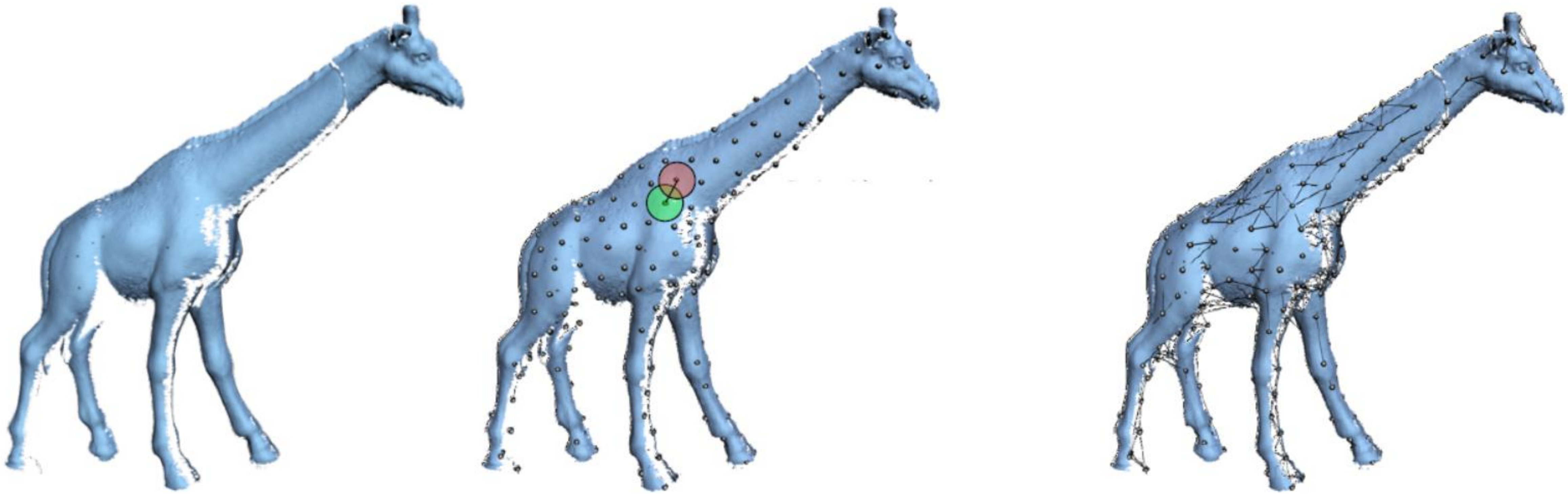
$$\underbrace{\min_{\mathbf{R}_1,\mathbf{t}_1,\dots,\mathbf{R}_m,\mathbf{t}_m}}_{\text{Graph parameters}} \underbrace{w_{\mathrm{rot}}\mathrm{E}_{\mathrm{rot}}}_{\text{Rotation term}} + \underbrace{w_{\mathrm{reg}}\mathrm{E}_{\mathrm{reg}}}_{\text{Regularization term}} + \underbrace{w_{\mathrm{con}}\mathrm{E}_{\mathrm{con}}}_{\text{Constraint term}}$$



Optimization Procedure

As Rigid As Possible Deformation

Embedded Deformation [Sumner et al. 07]

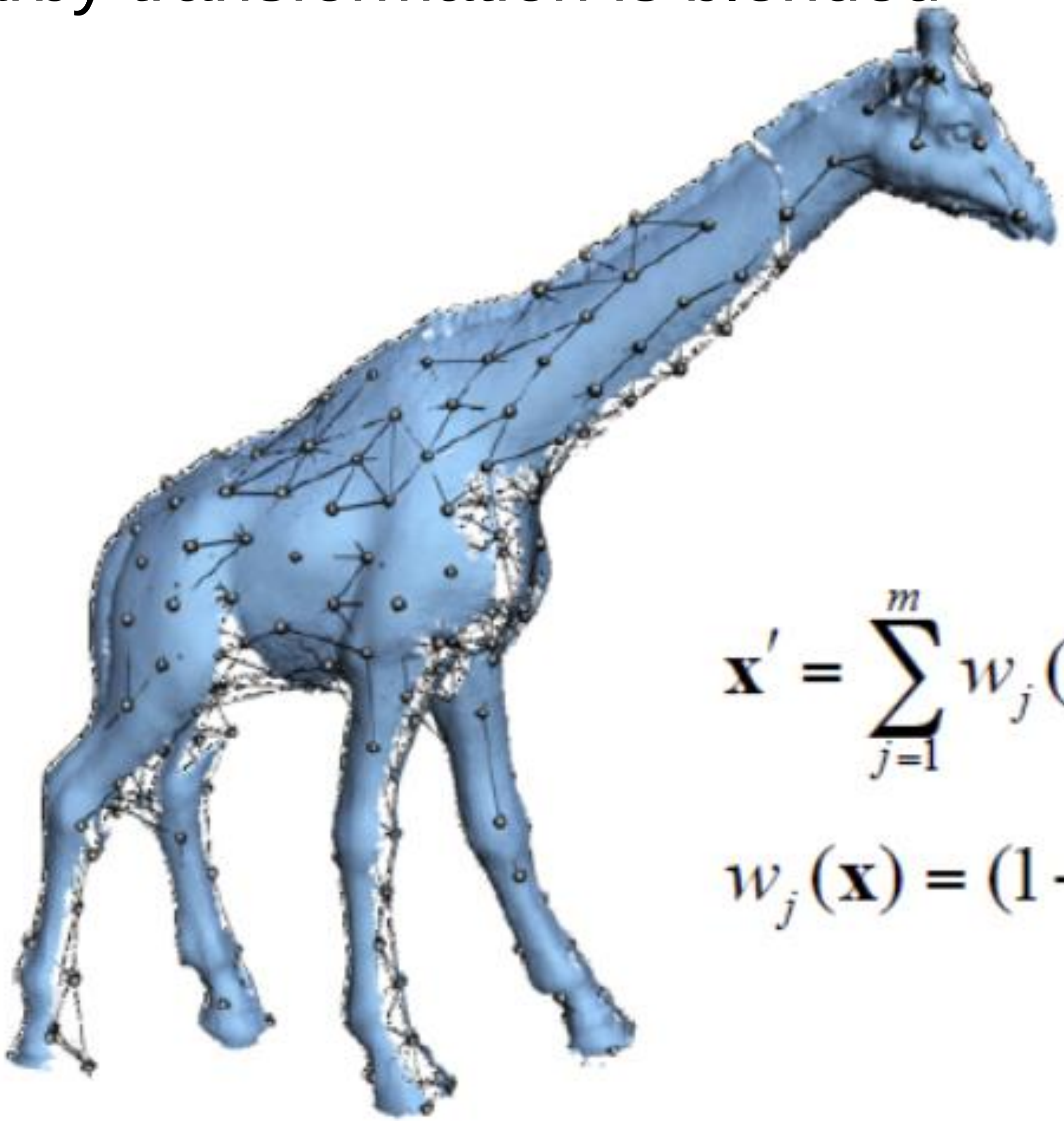


- Begin with an embedded object
- Nodes **{gi}** selected via uniform sampling
- One rigid transformation per node: Ri, ti
- Each node deforms nearby space
- Edges connect nodes of overlapping influence

As Rigid As Possible Deformation

Embedded Deformation [Sumner et al. 07]

- Influence of nearby transformation is blended



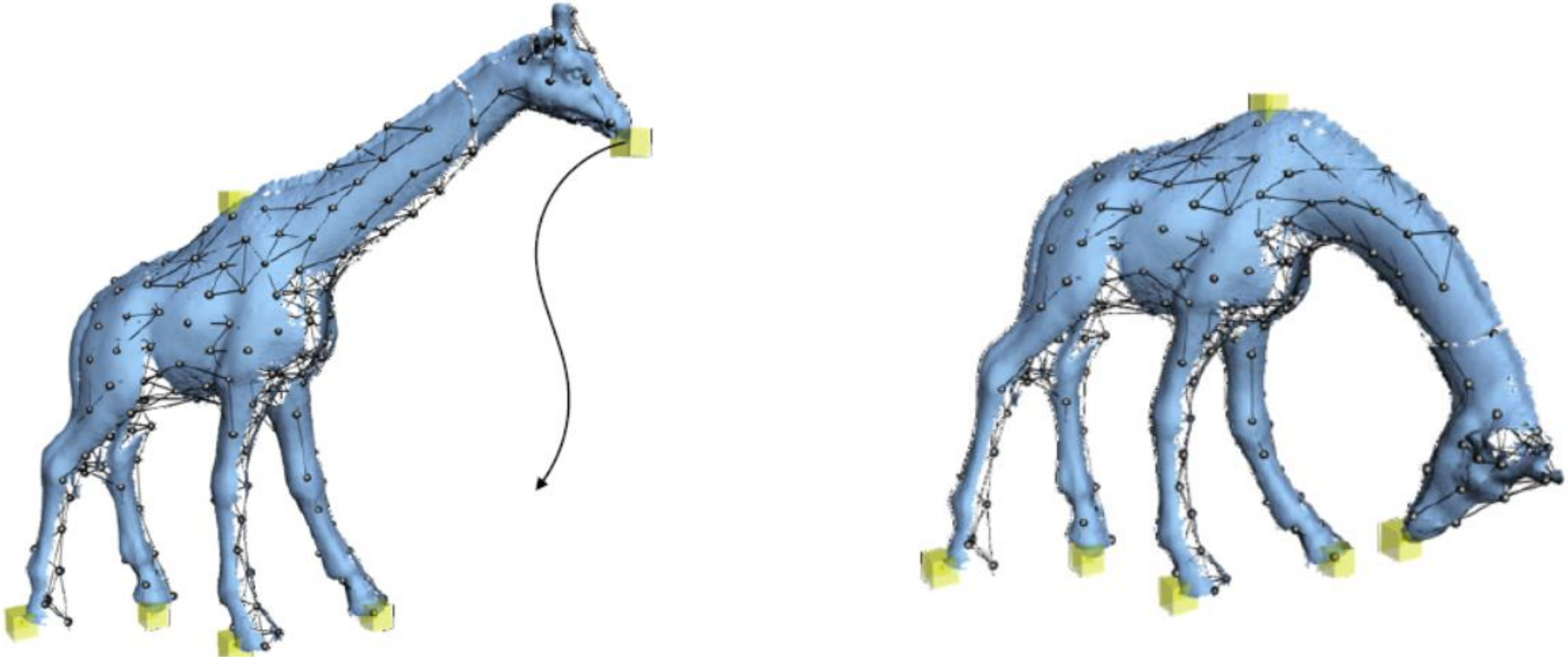
$$\mathbf{x}' = \sum_{j=1}^{m} w_j(\mathbf{x}) \overbrace{\left[\mathrm{R}_j(\mathbf{x} - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j \right]}^{\text{point } \mathbf{x} \text{ transformed by node } j}$$

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

As Rigid As Possible Deformation

Embedded Deformation [Sumner et al. 07]

- Select & drag vertices of embedded object, not the nodes
- Optimization finds deformations parameters Rj, tj

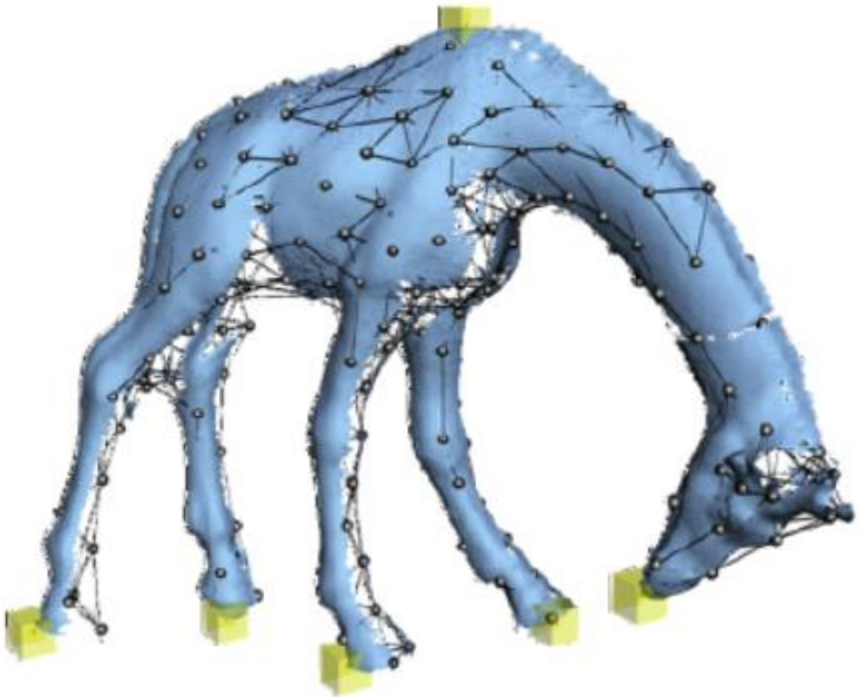


As Rigid As Possible Deformation

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

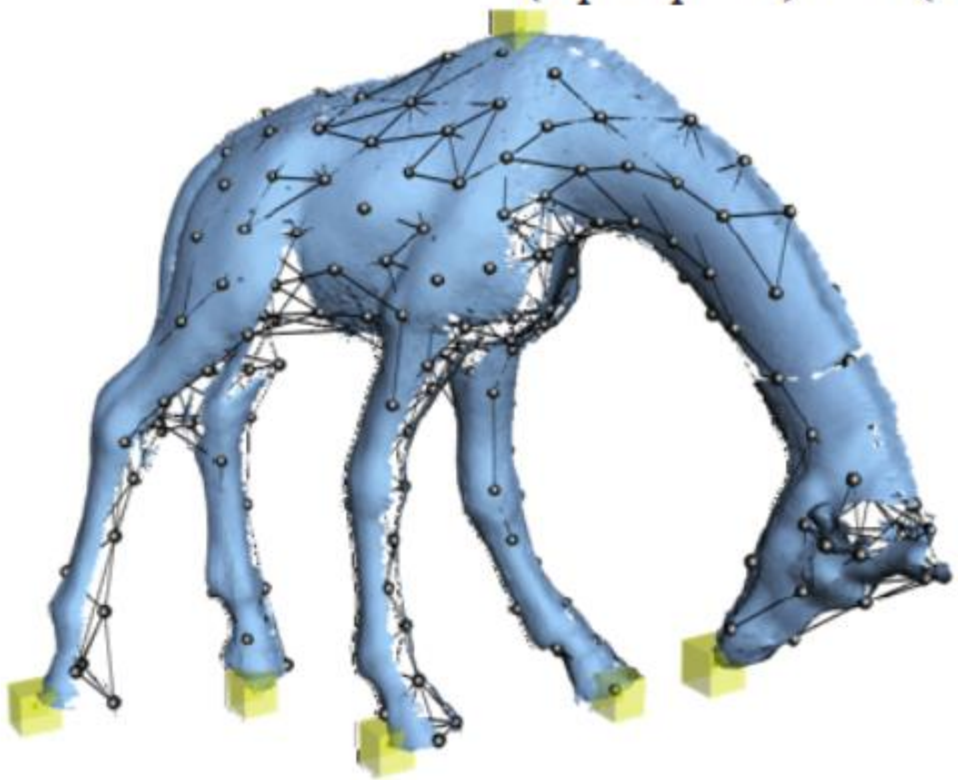


As Rigid As Possible Deformation

Embedded Deformation [Sumner et al. 07]

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

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



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

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

As Rigid As Possible Deformation

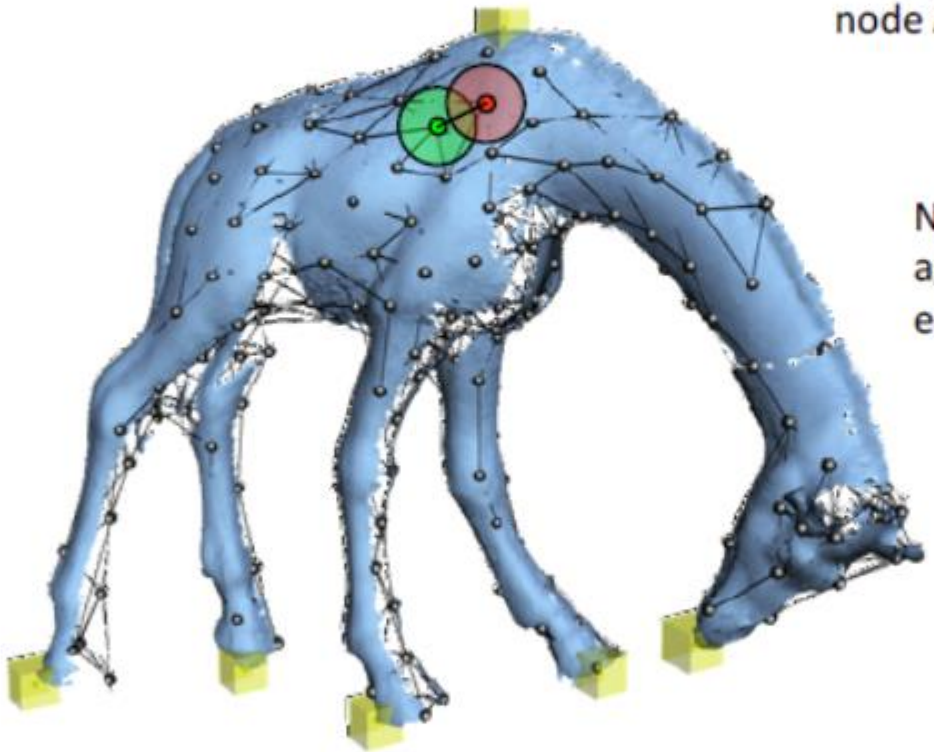
Embedded Deformation [Sumner et al. 07]

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

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

where node j thinks node k should go

where node k actually goes



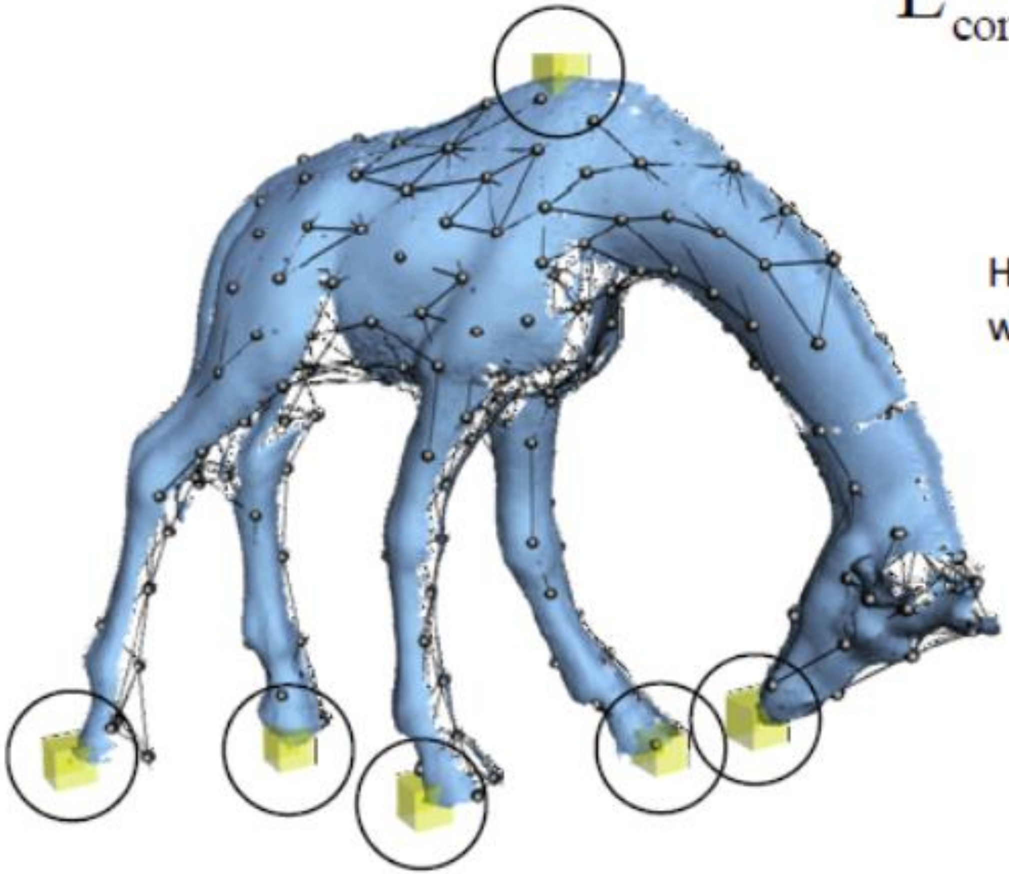
Neighboring nodes should agree on where they transform each other.

As Rigid As Possible Deformation

Embedded Deformation [Sumner et al. 07]

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

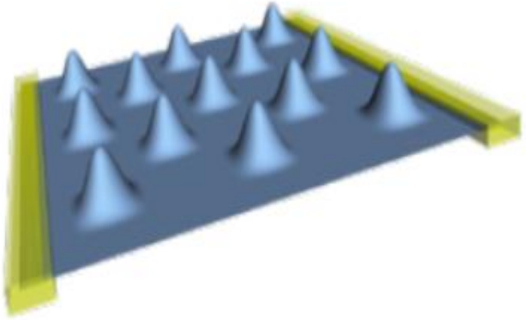
$$\mathrm{E}_{\text{con}} = \sum_{l=1}^{p} \left\| \tilde{\mathbf{v}}_{\text{index}(l)} - \mathbf{q}_l \right\|_2^2$$

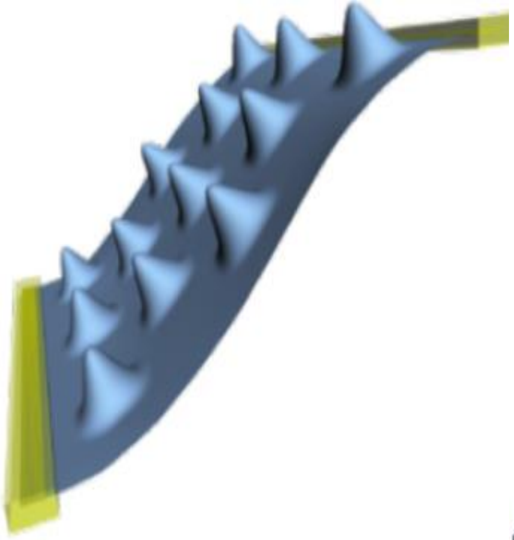


Handle vertices should go where the user puts them.

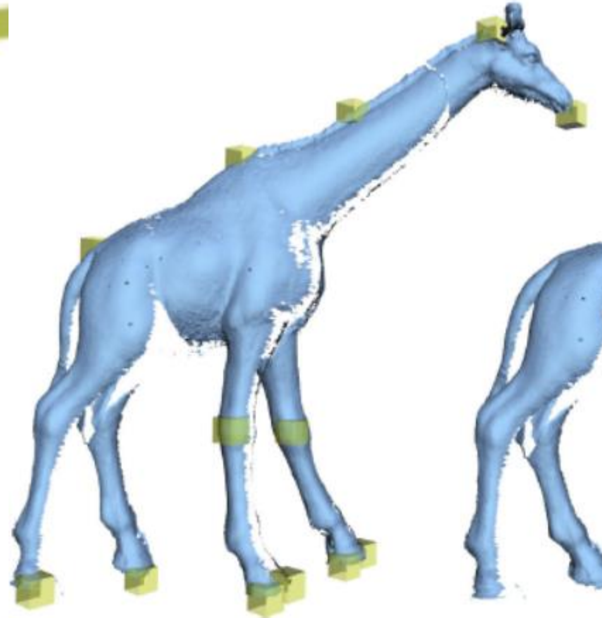
$$\tilde{\mathbf{v}}_i = \sum_{j=1}^{m} w_j(\mathbf{v}_i)\left[\mathbf{R}_j(\mathbf{v}_i - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j\right]$$

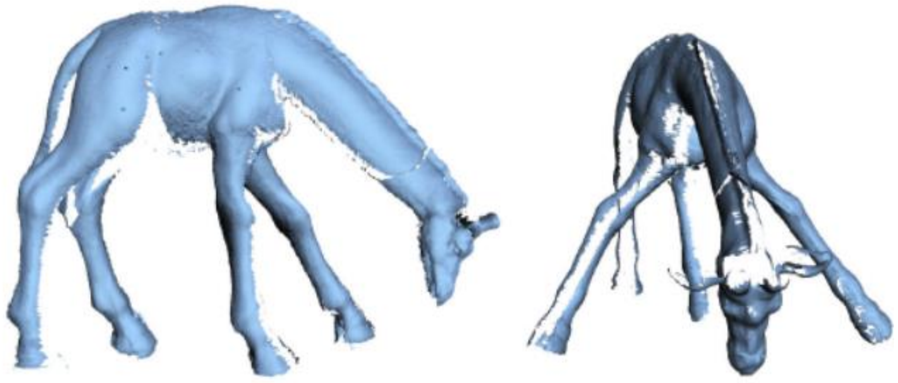
Detail Preservation

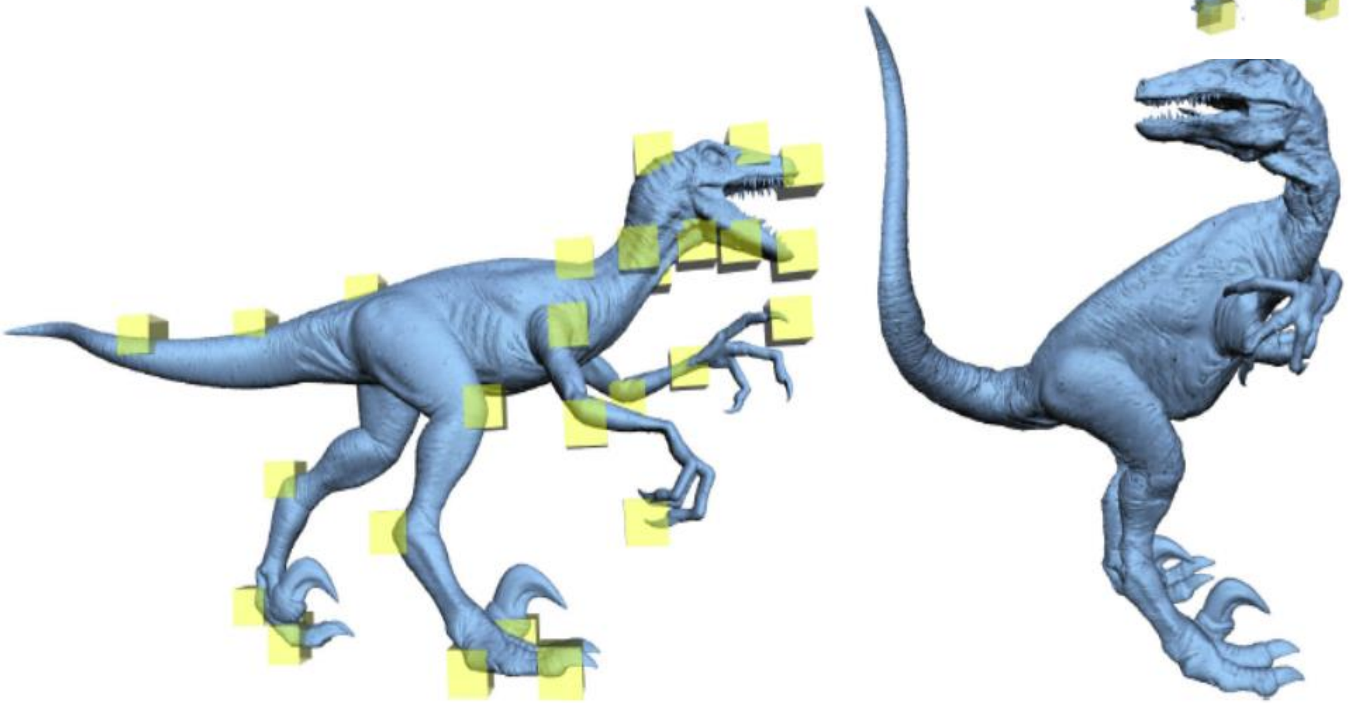




Results on Polygon Soups





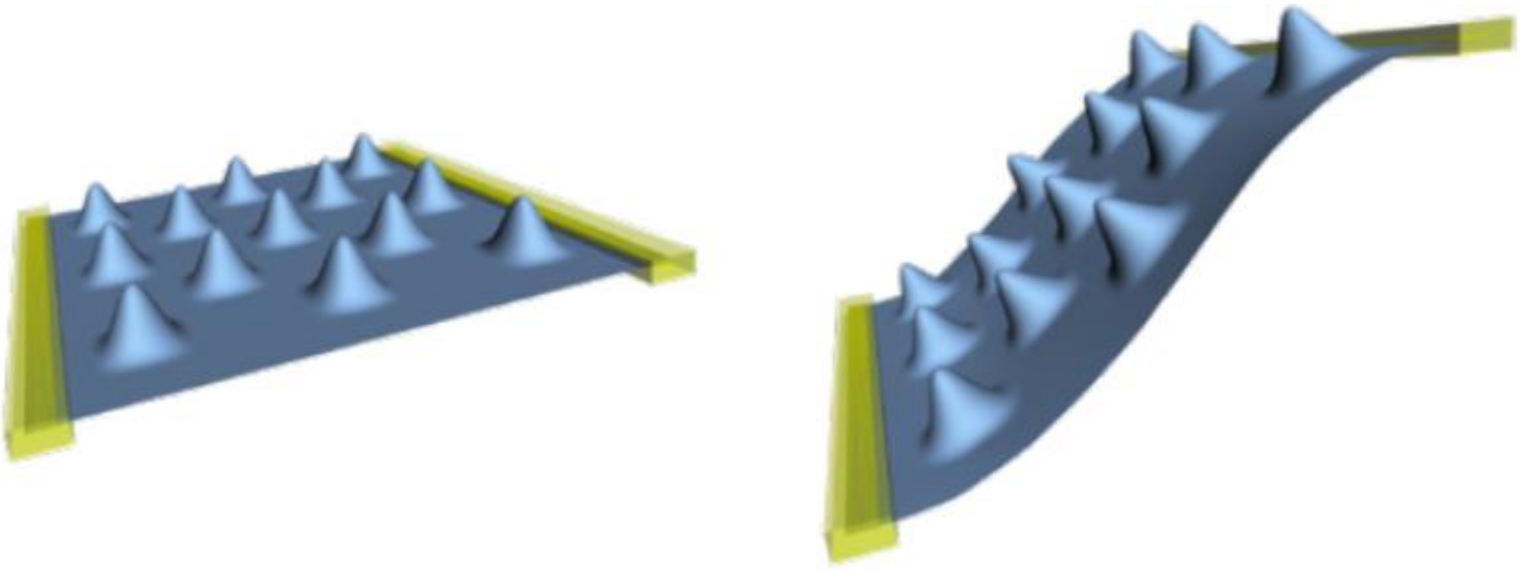


Results on Giant Mesh

Discussion

Embedded Deformation [Sumner et al. 07]

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



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	vs.	Space Deformation
+ More precise control of surface properties		– Doesn't know about embedded surface
– Depends on surface complexity & quality		+ 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

References

- Beier and Neely, "Feature-based image metamorphosis," in Proc. SIGGRAPH 92
- HSU W. M., HUGHES J. F., KAUFMAN H.: Direct manipulation of free-form deformations. SIGGRAPH 92
- Takeo Igarashi, Tomer Moscovich, John F. Hughes, "As-Rigid-As-Possible Shape Manipulation", ACM SIGGRAPH 2005,
- Mario Botsch and Leif Kobbelt. Real-time shape editing using radial basis functions. Computer Graphics Forum, 2005.
- Image Deformation Using Moving Least Squares. Scott Schaefer, Travis McPhail, Joe Warren. SIGGRAPH 2006
- Sumner, R., Schmid, J., Pauly, M. 2007. Embedded Deformation for Shape Manipulation. sig07

Thanks