Modelado y Animación por Computador

Tema 2: Modelado

Dr. Miguel Davia Aracil

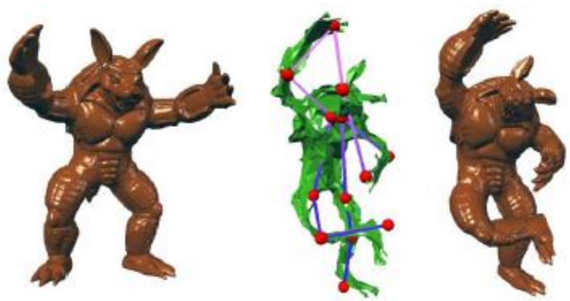
Tema 2: Modelado

Deformadores

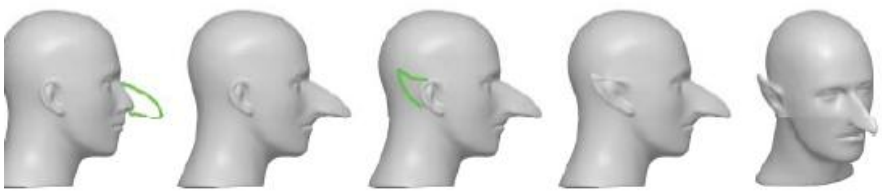
Deformaciones

¿Por qué deformar?

- ❑ **Animación**



- ❑ **Edición**

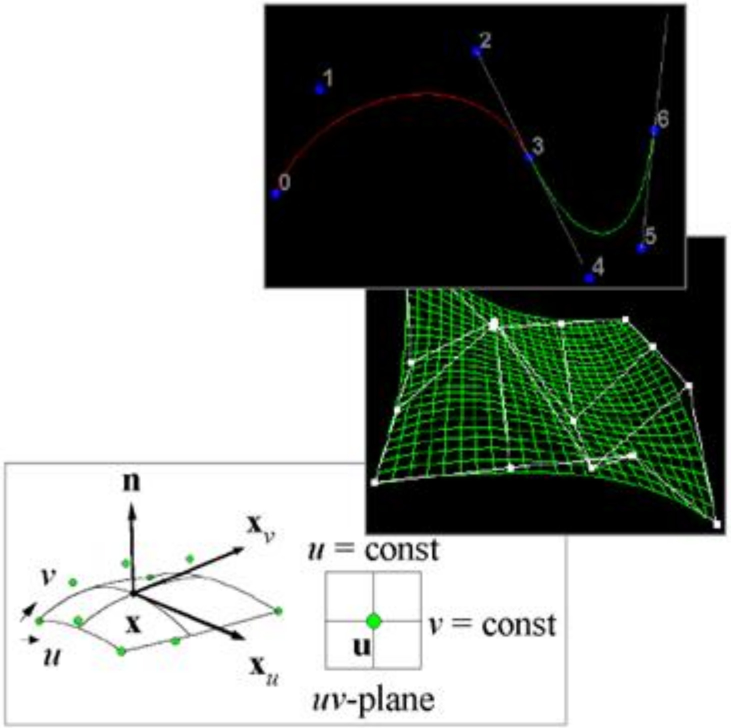


- ❑ **Simulación**

Deformadores

Curvas y superficies paramétricas

Deformación mediante manipulación de los puntos de control

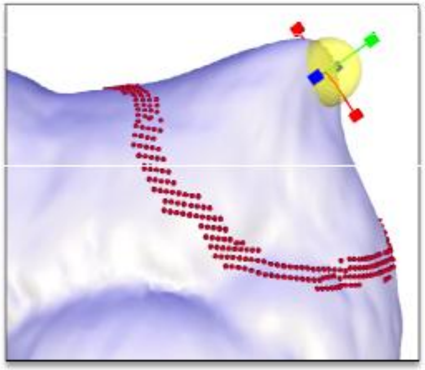


Deformadores

Deformación de meshes/shapes

Idea básica

- ❑ **Técnica básica:** mover vértices **individualmente**
- ❑ **Técnica avanzada:**
  - Crear un conjunto de **parámetros de control** de la deformación
  - Introducir un pequeño conjunto de **manejadores**
    - Simplifican la edición
    - Permiten la vinculación entre grados de libertad y simplificación del trabajo de deformación

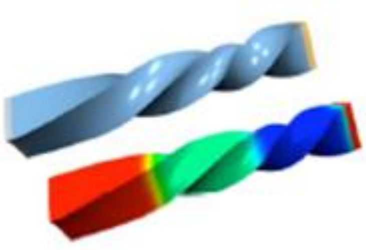


Deformadores

Deformación de meshes/shapes

Paradigmas utilizados

- **Surface based deformation**
  - Edición mediante operador laplaciano y otras aproximaciones basadas en **minimización de funciones de energía**
  - Permiten preservar la forma a través de la malla Laplaciana. Resistencia a “bending/streching”



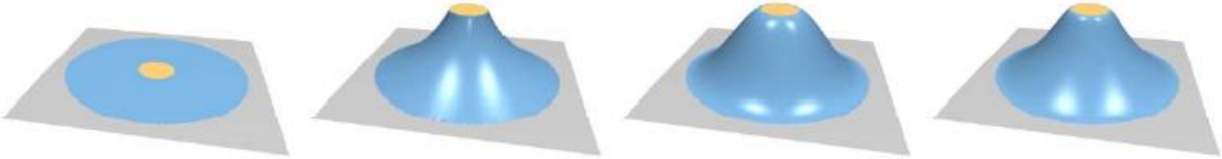
- **Space deformation**
  - Se deforma el espacio 2D/3D en el cual queda incluido el objeto. Normalmente se establece un **objeto de control** que estructura la deformación
  - La deformación se propaga a todos los puntos del espacio
  - Es independiente de la forma del objeto y su posición, así como de su representación geométrica

Deformadores

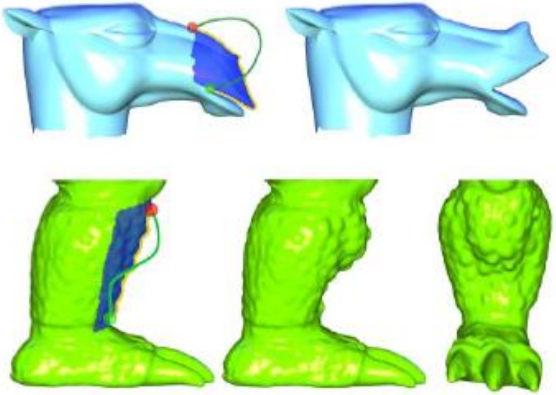
Surface-based deformation

Ejemplos

- Region of interest (ROI) + affine deformation of handle with variable boundary continuity



- Intuitive sketch-based deformation interfaces

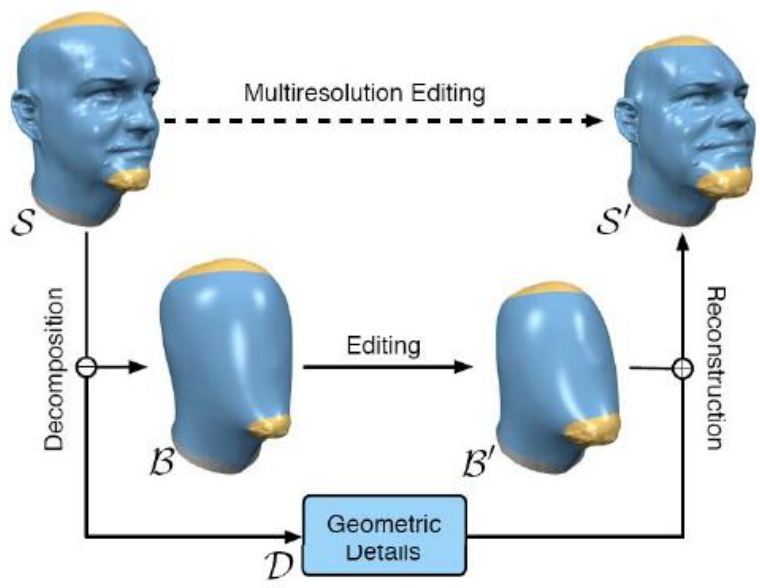


Deformadores

Surface-based deformation

Ejemplos

- Multi-resolution mesh editing



Deformadores

Surface-based deformation

Estructura general

- Encontrar una “mesh” que optimice alguna **función objetivo** y satisfaga ciertas **restricciones** de modelado

$$\mathbf{x}' = \arg\min_{\mathbf{x}'} F(\mathbf{x}') \quad \text{s.t. } \mathbf{x}'_i = \mathbf{c}_i$$

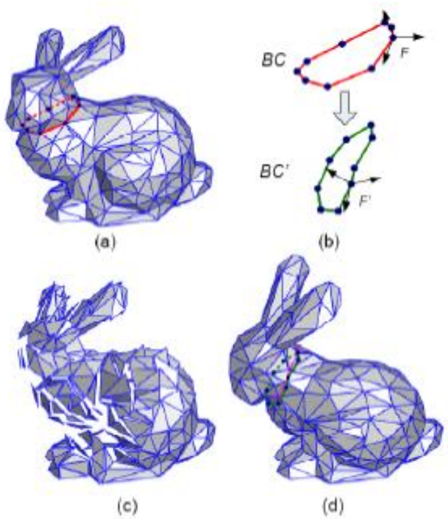
Deformadores

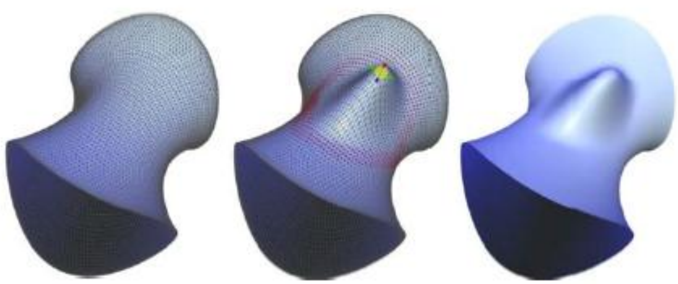
Surface-based deformation

Métodos lineales

- (2D) As rigid as possible shape manipulation

- Triangle gradient methods
- Laplacian surface editing



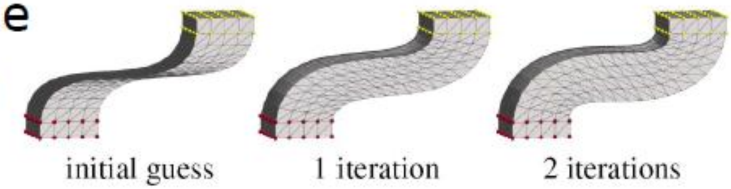


Deformadores

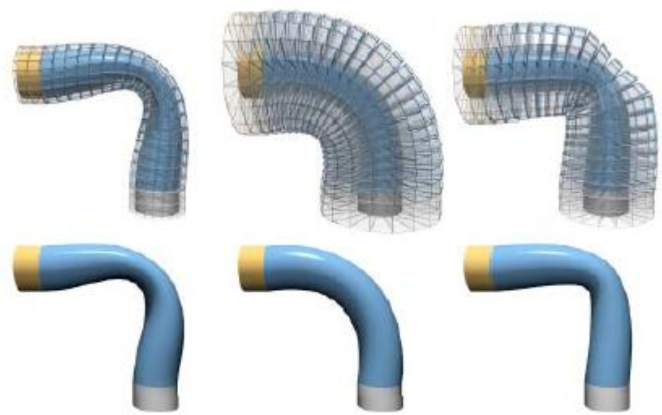
Surface-based deformation

Métodos no lineales

- As rigid as possible surface modeling



- PriMo



- Mesh Puppetry

Deformadores

Surface-based deformation

Resumen

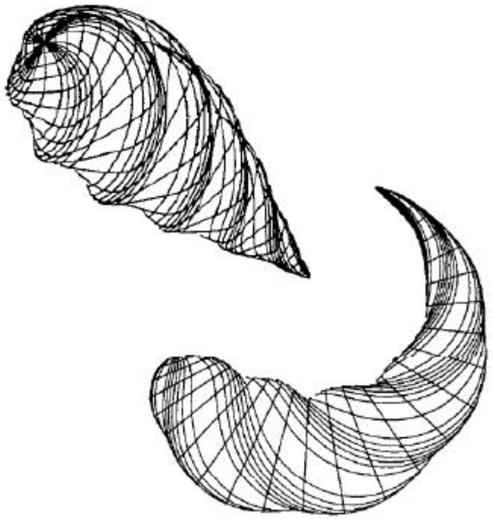
- La **función objetivo** opera sobre los vértices de la malla (geometría a deformar)
- La complejidad depende del tamaño de la malla
- **Métodos lineales:**
  - Resuelven sistemas lineales globales sobre la malla
  - A veces se producen comportamientos erróneos en la deformación
- **Métodos no lineales:**
  - Menos errores y problemas, pero lento en el proceso de deformación y complejos de implementar

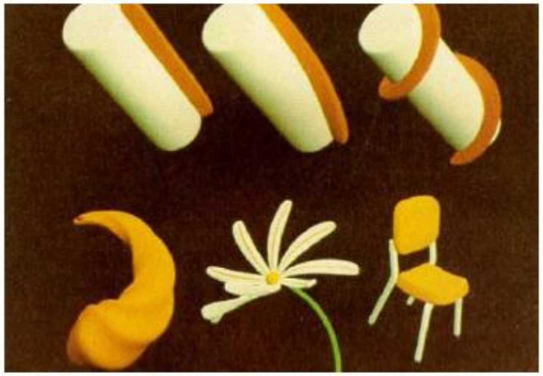
Deformadores

Space deformation

Primeros trabajos en deformación de formas mediante computador

- Global and local deformation of solids [Barr 1984]



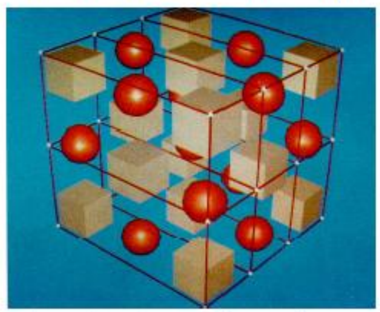


Deformadores

Space deformation

Primeros trabajos en deformación de formas mediante computador

- Free form deformations [Sederberg and Parry 1986]
  - Uses trivariate tensor product polynomial basis




  - Can be designed to be volume preserving

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

then the Jacobian is the determinant

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

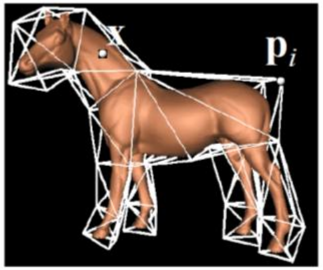
Deformadores

Space deformation

Idea básica

- Se determina un conjunto de **coordenadas** ($\mathbf{w}_i$) para todos los puntos del espacio R^d en base a los vértices de la **estructura de control**
  - Cada punto **x** de la geometría se representa como la suma ponderada **($\mathbf{w}_i$)** de los vértices $\mathbf{p}_i$ de la estructura de control

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



  - Cuando la **estructura de control cambia**, sin alterar las coordenadas, se obtiene el valor para los nuevos puntos de la geometría original **x'** (deformación)

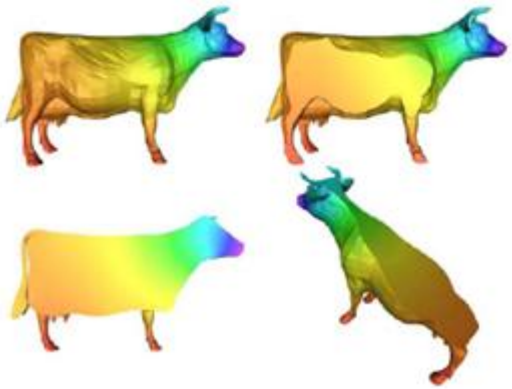
$$\mathbf{x}' = \sum_{i=1}^{k} w_i(\mathbf{x}) \cdot \mathbf{p}'_i$$

Deformadores

Space deformation

Idea básica

- Se determina un conjunto de coordenadas ($\mathbf{w}_i$) para todos los puntos del espacio R^d en base a los vértices de la **estructura de control**
  - Las coordenadas varían de forma suave y garantizan **suavidad** en el interior del volumen

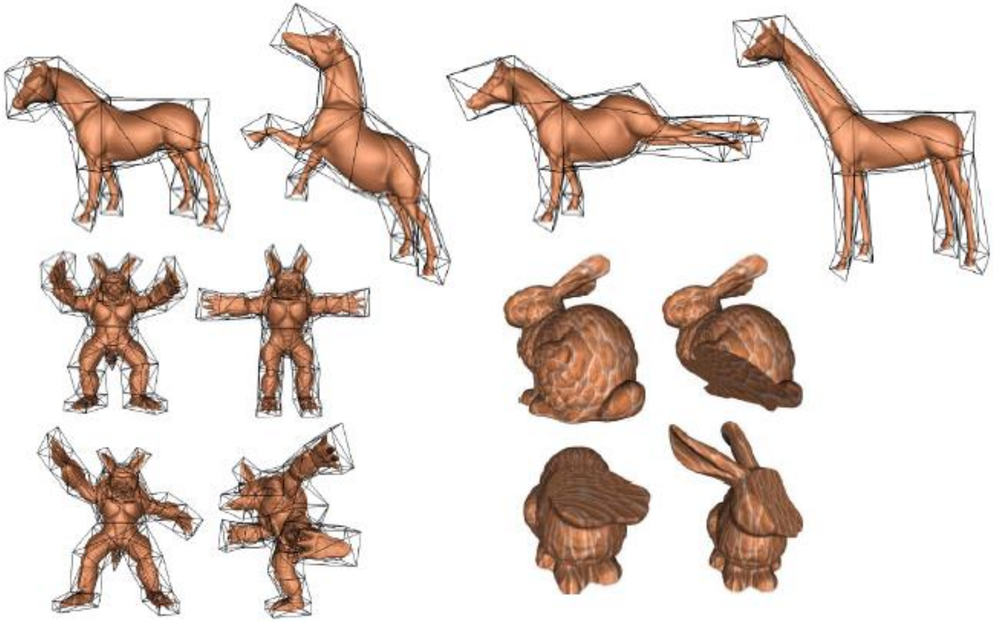


Deformadores

Space deformation

Ejemplos

- Mean value coordinates for closed tri meshes

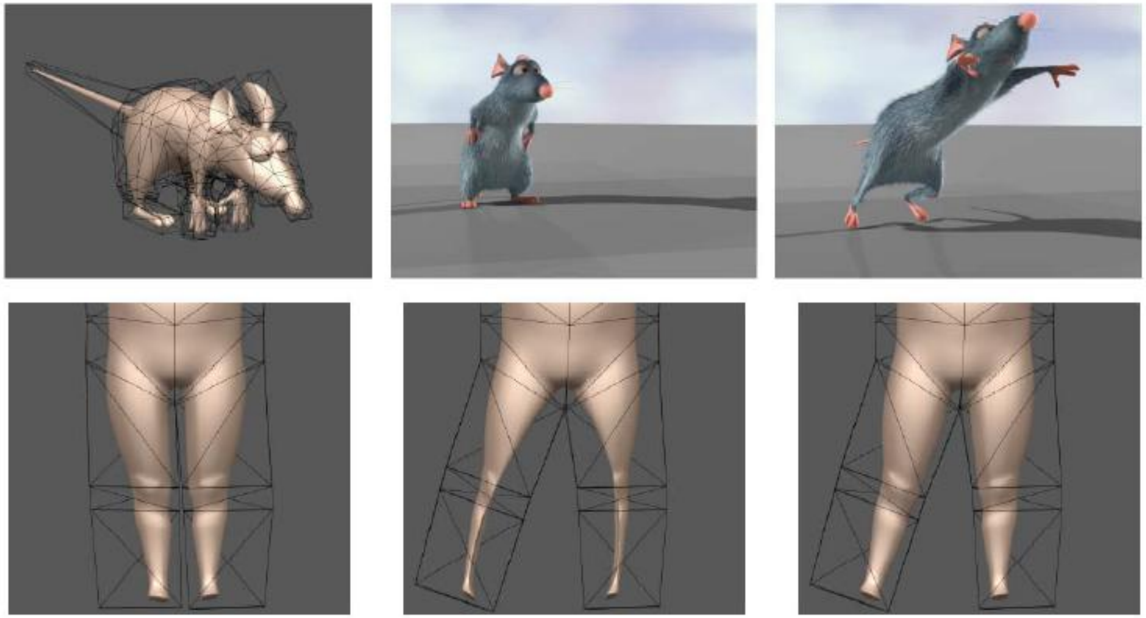


Deformadores

Space deformation

Ejemplos

- Harmonic coordinates

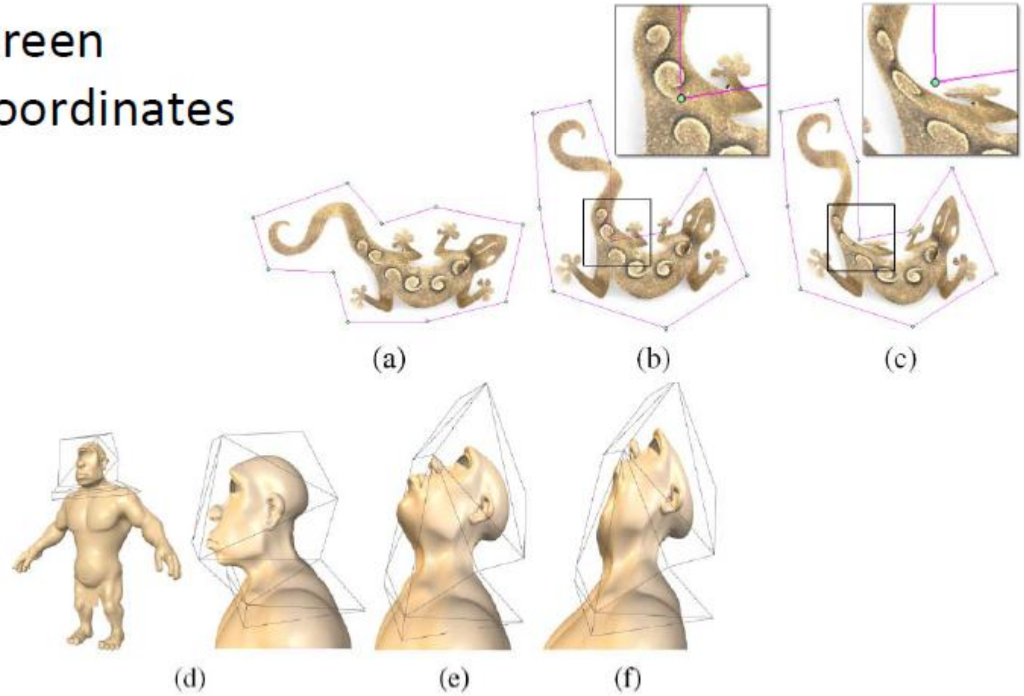


Deformadores

Space deformation

Ejemplos

- Green coordinates



Deformadores

Space deformation

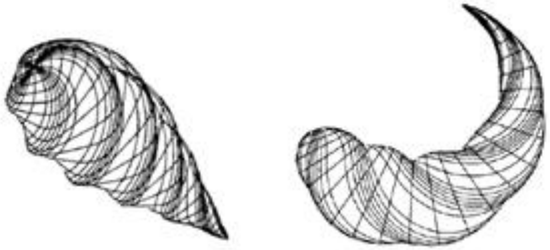
Conceptos

- Función de desplazamiento definida en el espacio

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

- Se evalúa la función para cada punto de la geometría y se calcula su nueva posición

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



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

Deformadores

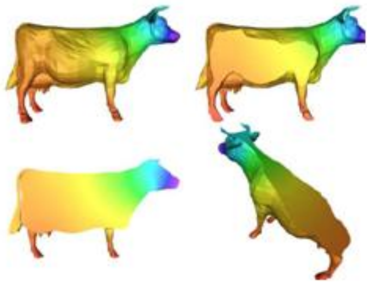
Space deformation

FFD (freeform deformations)

- Se especifica un **objeto de control** que deforma el espacio
- El usuario modifica el **objeto de control** y genera desplazamientos $\mathbf{d}_i$ para cada **punto de control** del mismo
- Los desplazamientos son interpolados en el espacio usando “basis functions”

$$B_i(\mathbf{x}) : \mathrm{R}^3 \rightarrow \mathrm{R}$$

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



- Las “basis functions” deben ser suaves para garantizar resultados estéticamente correctos

Deformadores

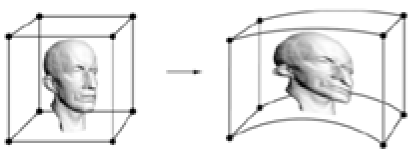
Space deformation

FFD (freeform deformations)

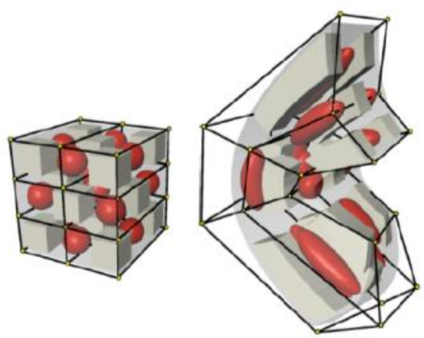
Trivariate Tensor Product Bases

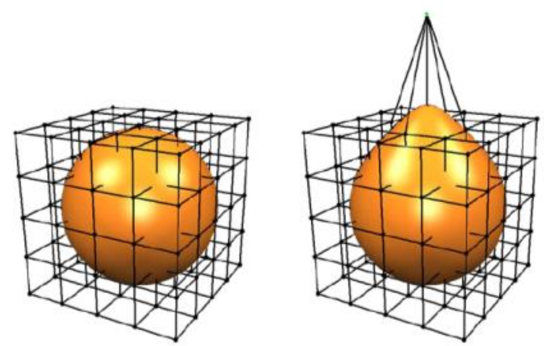
[Sederberg and Parrry 86]

- Objeto de control = **rejilla** (**lattice**)
- Las “basis functions” $B_i(\boldsymbol{x})$ son “trivariate tensor-product splines”:



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





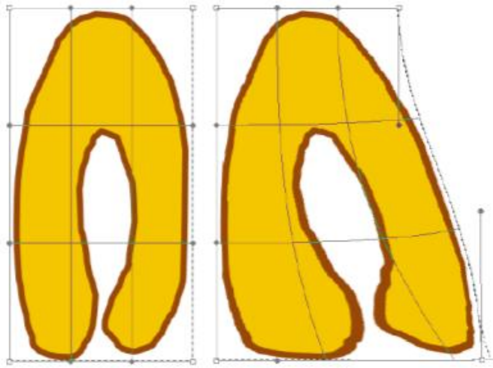
Deformadores

Space deformation

FFD (freeform deformations)

“Lattice” como objeto de control

- **Difícil** de manipular
- El objeto de control **no tiene relación directa** con la forma de la geometría a editar (deformar)
- Existen **vértices interiores** del objeto de control que dificultan la deformación de la geometría

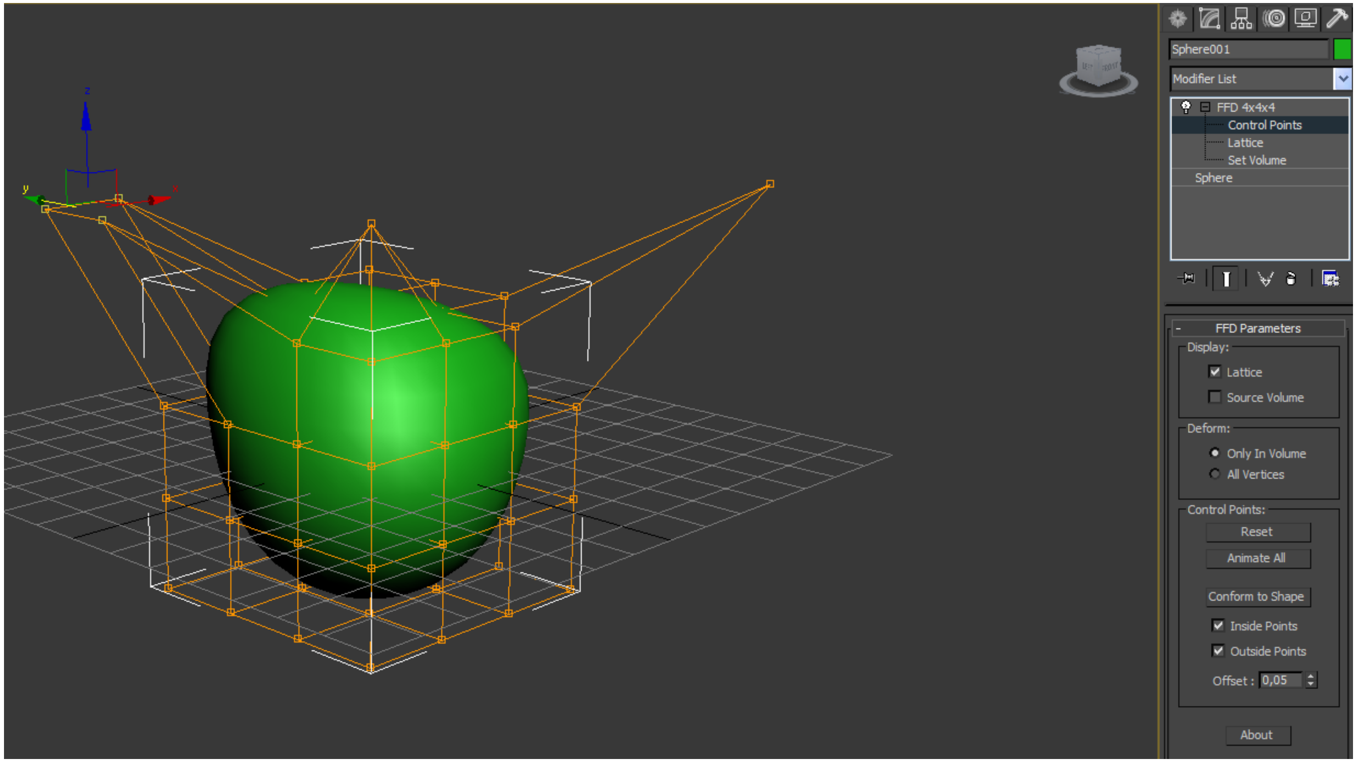


Transformaciones geométricas

Modifiers

3D Studio Max. Object-space modifiers

- FFD 2x2x2
- FFD 3x3x3
- FFD 4x4x4
- FFD (box)
- FFD (cyl)



Deformadores

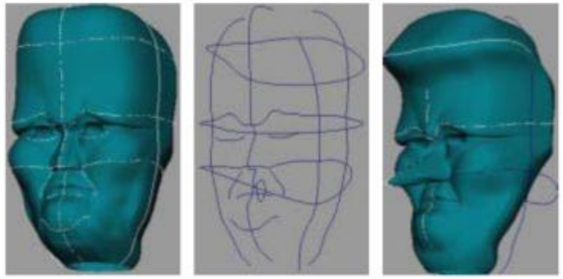
Space deformation

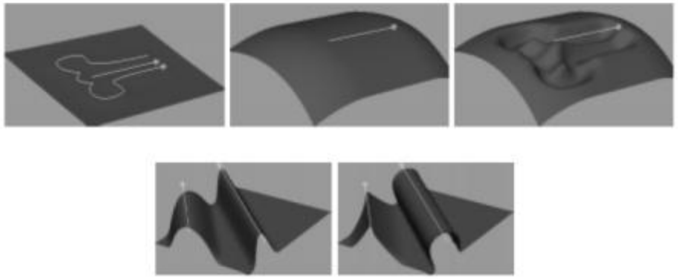
Wires

Wires

[Singh and Fiume 98]

- Los **objetos de control** son **curvas** 3D arbitrarias
- Se pueden situar las curvas de control en lugares determinados para editar (deformar) el objeto en cuestión
- Se produce una deformación suave en torno a la curva de control en función de la **influencia** de ésta sobre el objeto (decrece con respecto a la distancia)





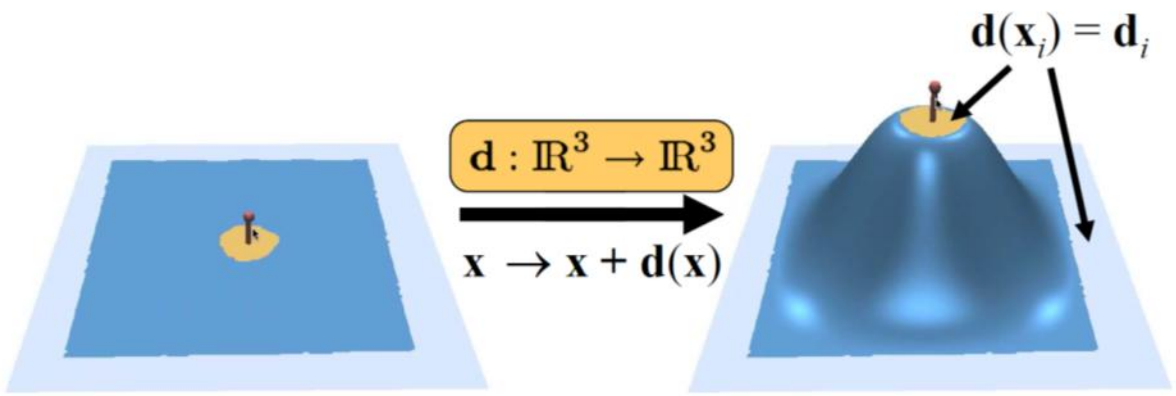
Deformadores

Space deformation

Handle Metaphor

Handle Metaphor
[RBF, Botsch and Kobbelt 05]

- Función desplazamiento **d(x)**
- Interpolación en función de determinadas restricciones
- Suavidad y deformación intuitiva



Deformadores

Space deformation

Handle Metaphor

Radial Basis Functions

[RBF, Botsch and Kobbelt 05]

- La deformación se representa por **RBFs**

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

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

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

- Se resuelven *sistemas lineales* para $\mathbf{w}_j$ y $\mathbf{p}$

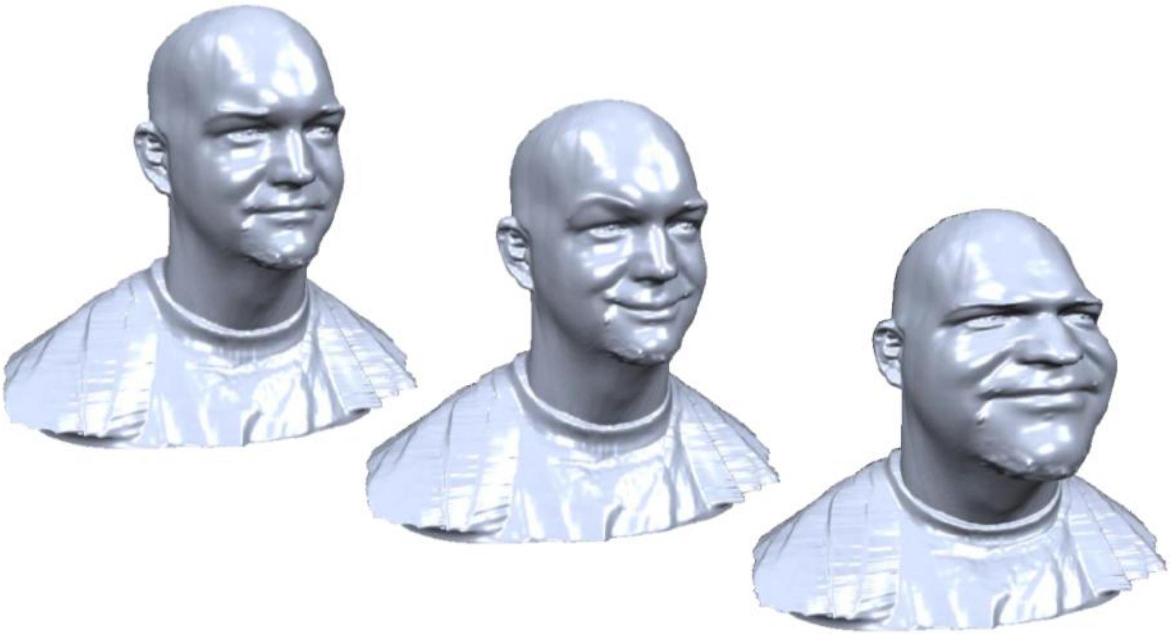
Deformadores

Space deformation

Handle Metaphor

Local & Global Deformations

[RBF, Botsch and Kobbelt 05]



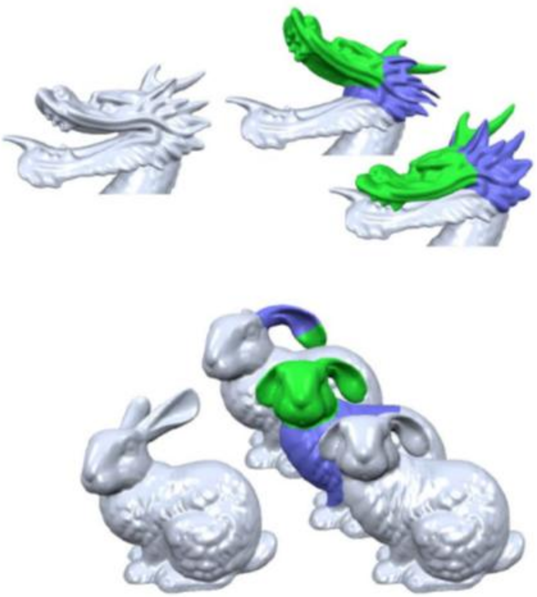
Deformadores

Space deformation

Handle Metaphor

Local & Global Deformations

[RBF, Botsch and Kobbelt 05]

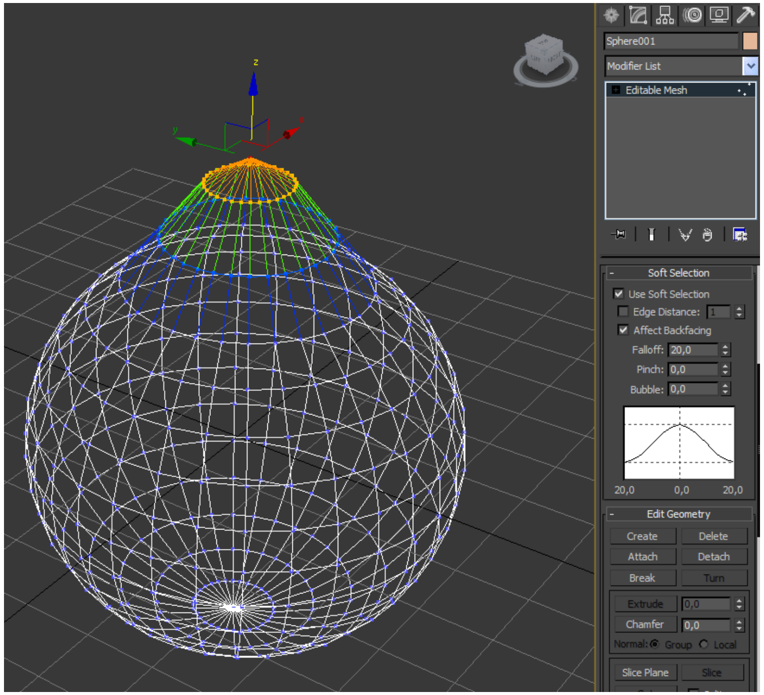


1M vertices
movie

Deformadores

Space deformation

3D Studio Max. Editable Mesh – Soft Selection

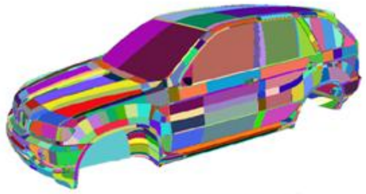


Deformadores

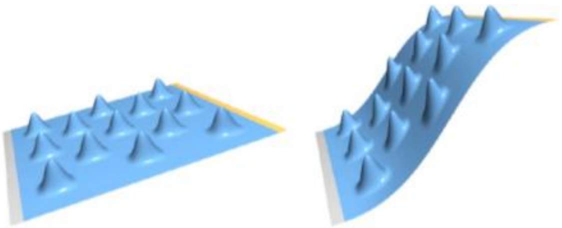
Space deformation

Resumen

- Trabaja con **cualquier tipo de geometría** (tipo representación)
  - Meshes (también non-manifold)
  - Point sets
  - Polygonal soup
  - …
- La **complejidad** depende directamente del **objeto de control**, no del objeto a deformar
- Los detalles locales de la superficie deformada pueden ser alterados. **No preserva detalle local.**



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



Deformadores

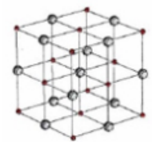
Space deformation

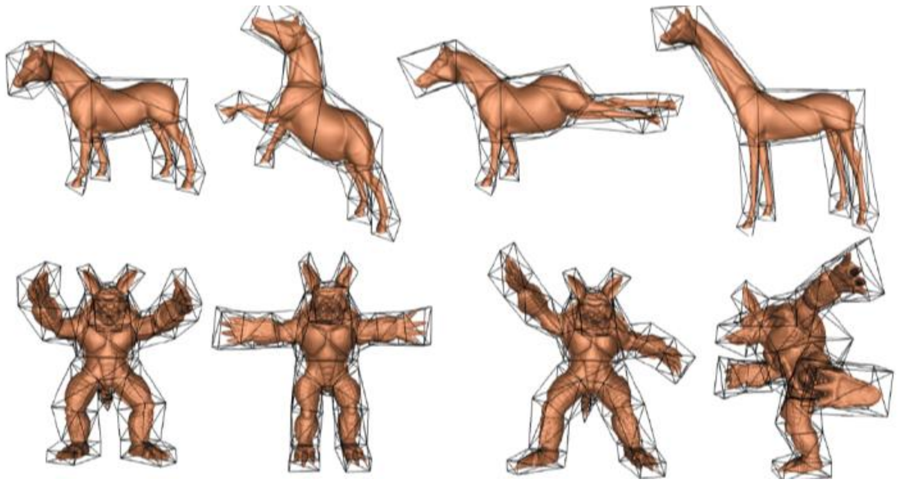
Cage-based

Cage-based deformations

[MVC, Floater 05]

- **Cage** = jaula, suele ser una **representación primaria** del objeto a deformar
- Es un **polihedro**, no una **rejilla** (lattice)

 ≠ 



Deformadores

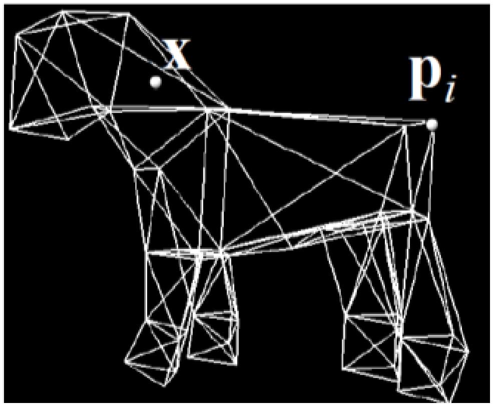
Space deformation

Cage-based

Cage-based deformations

[MVC, Floater 05]

- Cada punto **x** en el espacio es representado a partir de la suma ponderada de los elementos de la jaula usando para ello funciones de coordenadas (**coordinate functions**)



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

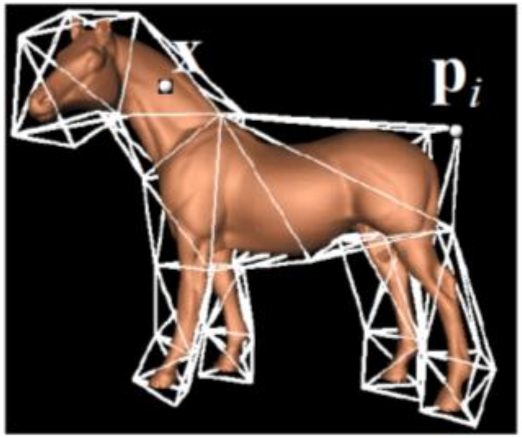
Deformadores

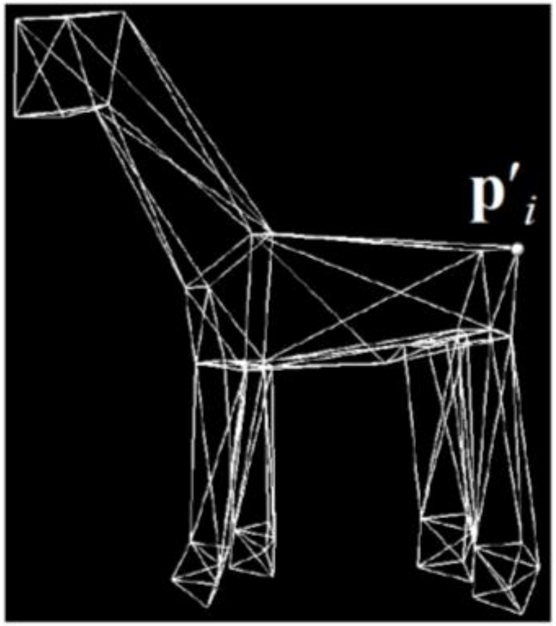
Space deformation

Cage-based

Cage-based deformations
[MVC, Floater 05]

- Cuando se modifican los puntos $\mathbf{p}_i$ de la jaula a $\mathbf{p'}_i$ …





Deformadores

Space deformation

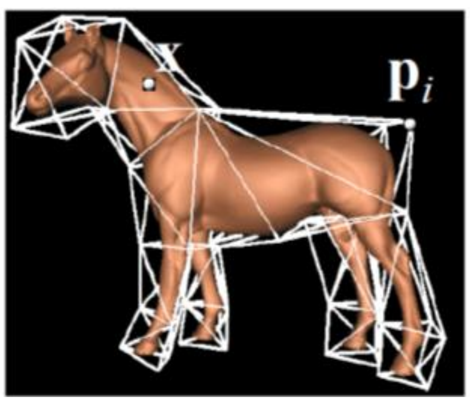
Cage-based

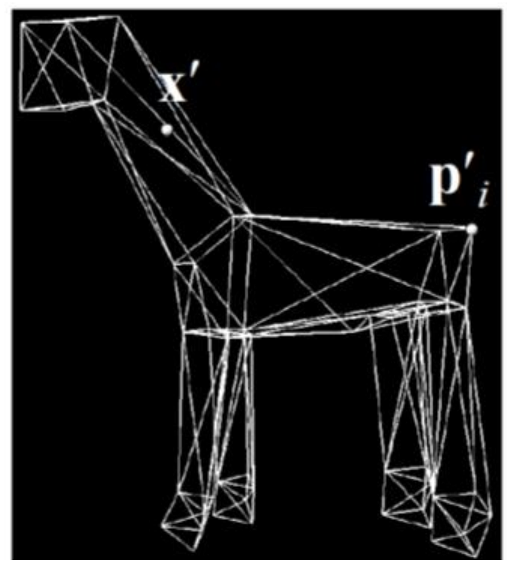
Cage-based deformations

[MVC, Floater 05]

- Cada punto **x** de la geometría a deformar **transforma** en **x'**
- **k** es el número de vértices de la jaula

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





Deformadores

Space deformation

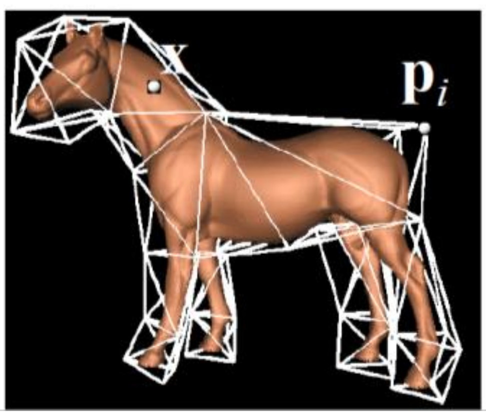
Cage-based

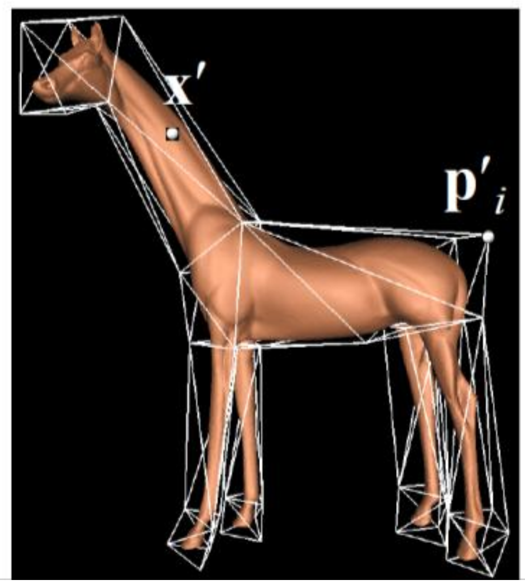
Cage-based deformations

[MVC, Floater 05]

- Dando como resultado la deformación de la geometría…

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



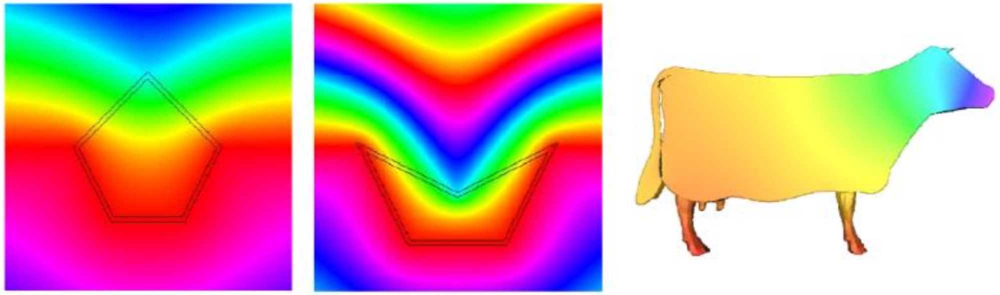


Deformadores

Space deformation

Coordinate Functions (ponderación)

- **Mean-value coordinates** (Floater, Ju et al. 2005)
  - Generalización de las coordenadas baricéntricas
  - Obtención de las coordenadas $\mathbf{w_i(x)}$

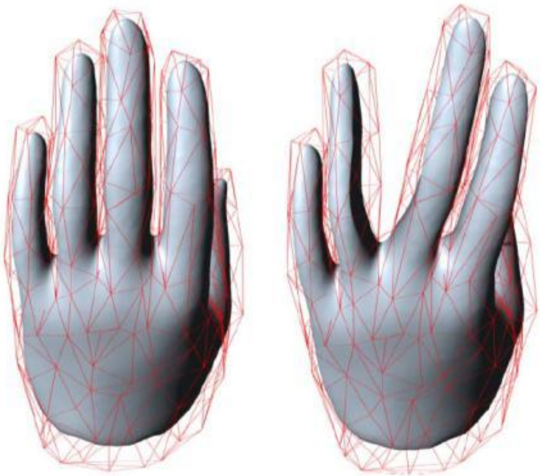


Deformadores

Space deformation

Coordinate Functions

- **Mean-value coordinates** (Floater, Ju et al. 2005)
  - No necesariamente positivas en dominios no-convexos



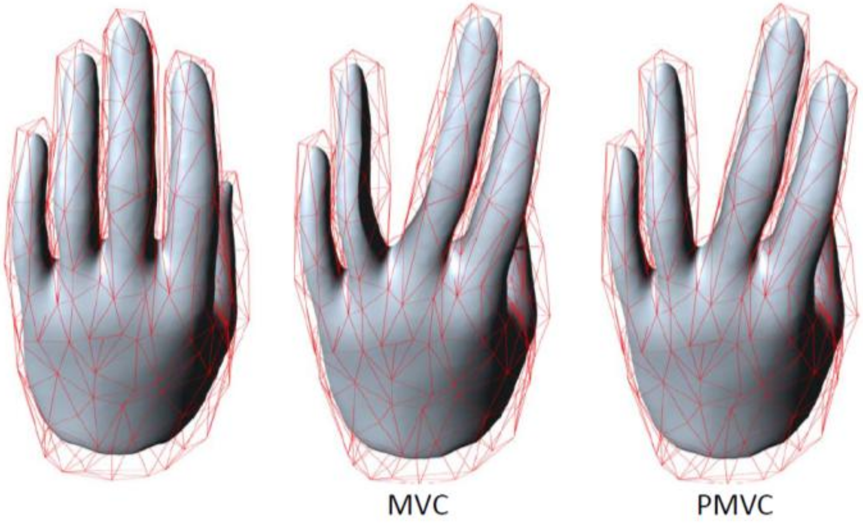
MVC

Deformadores

Space deformation

Coordinate Functions

- **Positive Mean-value coordinates PMVC** (Lipman et al. 2007)
  - Garantiza coordenadas positivas (**$w_i > 0$**)

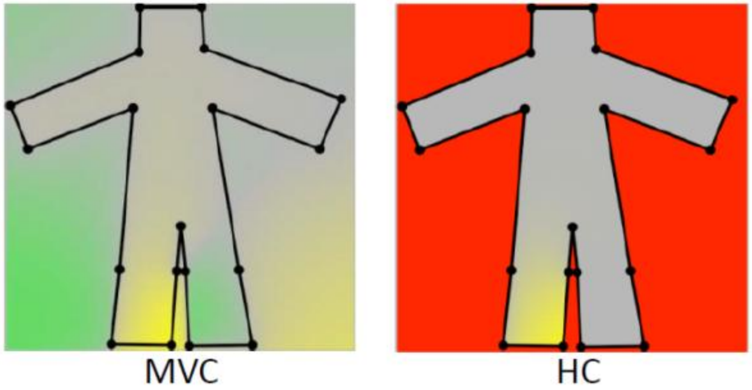


Deformadores

Space deformation

Coordinate Functions

- **Harmonic coordinates** (Joshi et al. 2007)
  - Funciones harmónicas $h_i(\boldsymbol{x})$ para cada vértice $\mathbf{p_i}$ de la jaula
  - Resolver $\Delta\, \boldsymbol{h=0}$; restricción: h_i lineal en los contornos $h_i(\mathbf{p_j})=\delta_{ij}$

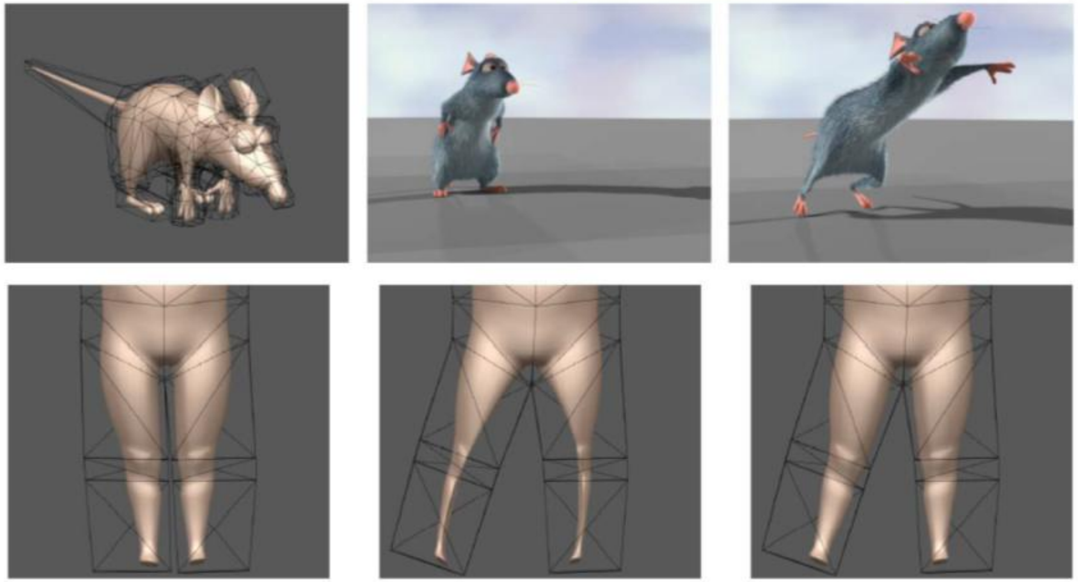


Deformadores

Space deformation

Coordinate Functions

- Harmonic coordinates (Joshi et al. 2007)

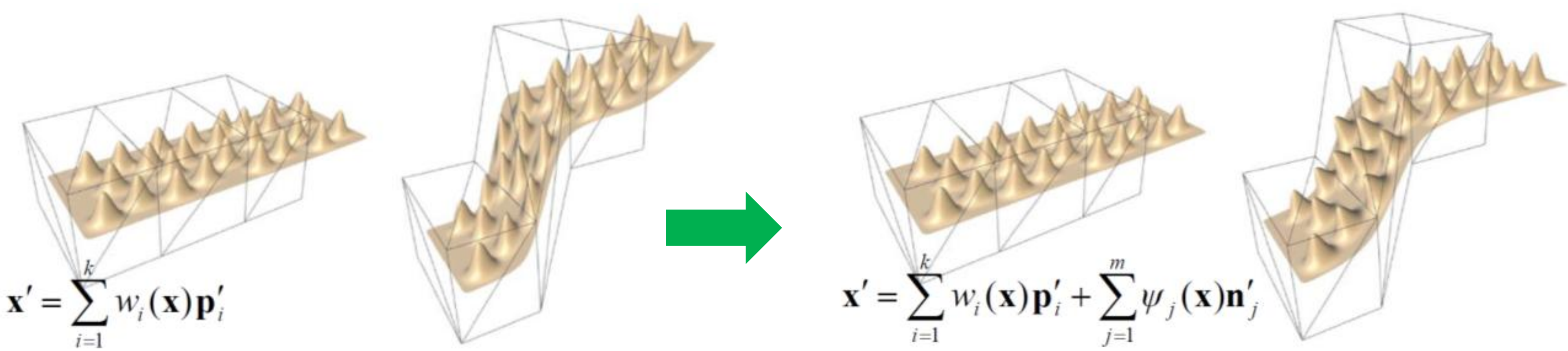


Deformadores

Space deformation

Coordinate Functions

- **Green coordinates** (Lipman et al. 2008)
  - Incluyen en el cálculo de las coordenadas $\mathbf{w_i}$ información sobre las **caras** de la jaula



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

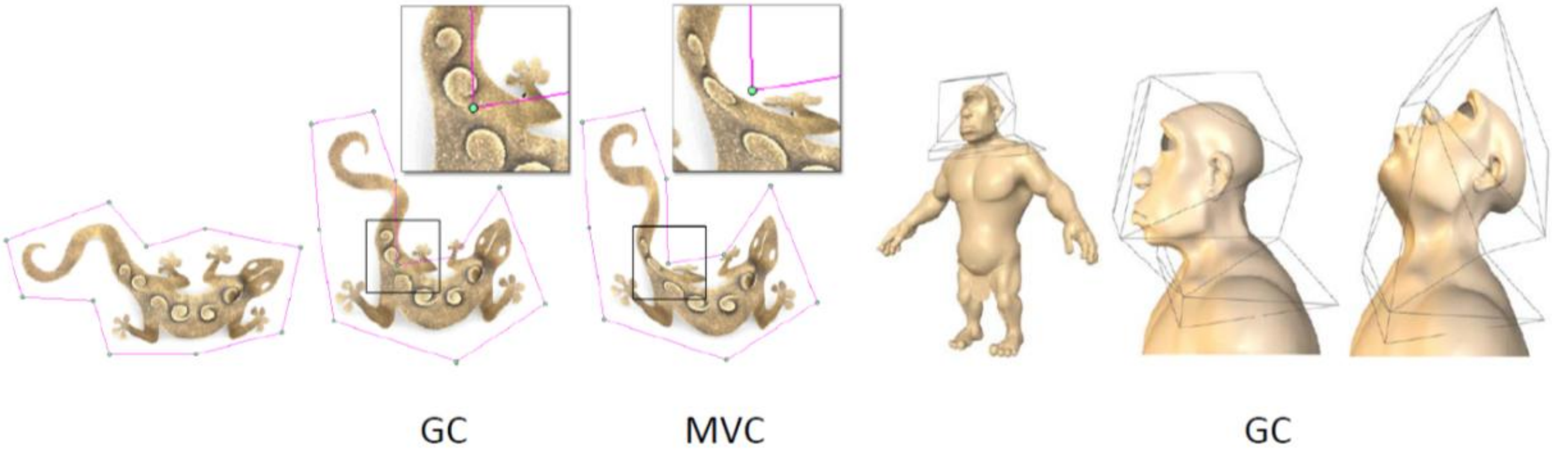
$$\mathbf{x}' = \underbrace{\sum_{i=1}^{k} w_i(\mathbf{x})\mathbf{p}'_i}_{\text{vértices}} + \underbrace{\sum_{j=1}^{m} \psi_j(\mathbf{x})\mathbf{n}'_j}_{\text{caras}}$$

Deformadores

Space deformation

Coordinate Functions

- **Green coordinates** (Lipman et al. 2008)
  - Aplicación 2D y 3D



Deformadores

Space deformation

Coordinate Functions

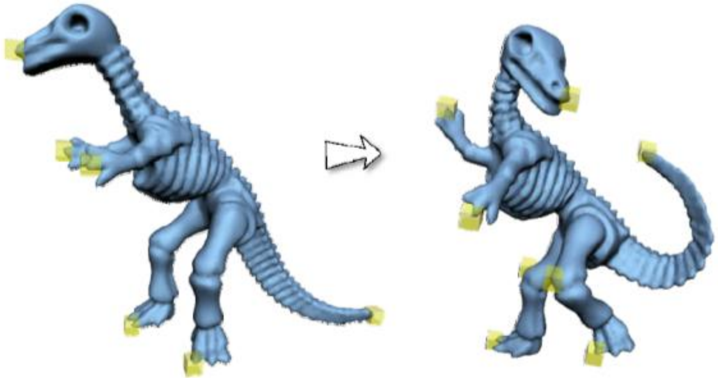
Alternative interpretation in 2D via holomorphic functions and extension to point handles : Weber et al. Eurographics 2009

Deformadores

Space deformation

Nonlinear Space Deformations

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



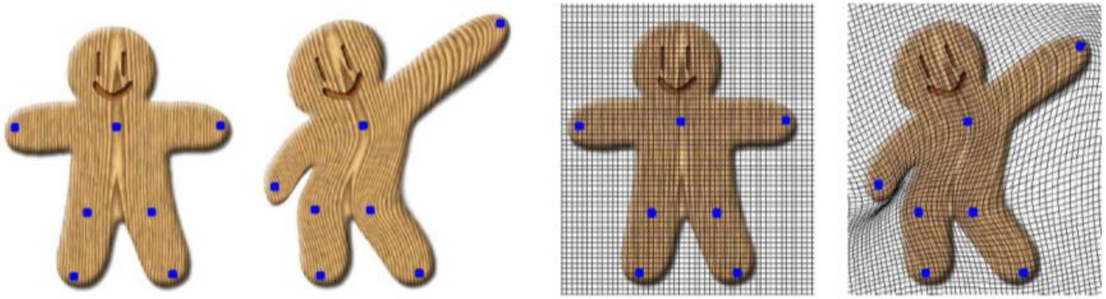
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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)



Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

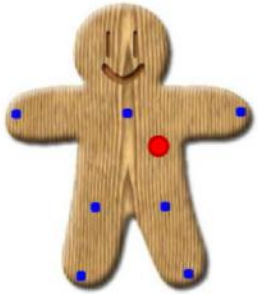
Moving-Least-Squares (MLS) approach [Schaefer et al. 2006]

- 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:

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



Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

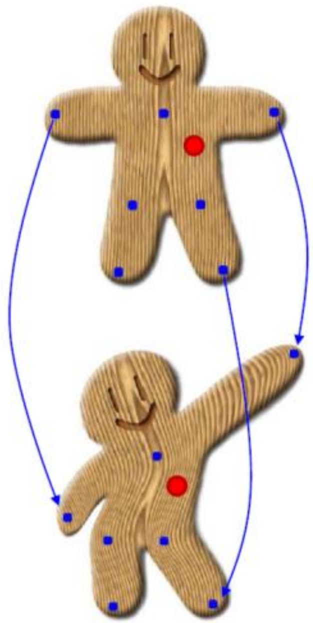
- Handles $\mathbf{p}_i$ are displaced to $\mathbf{q}_i$
- The local transformation at $\mathbf{x}$:

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

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

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

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



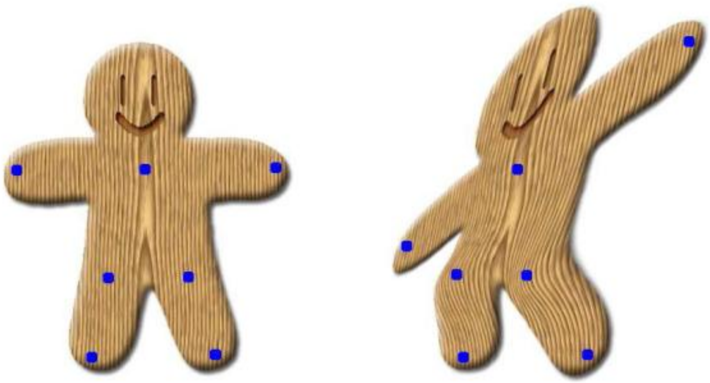
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

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



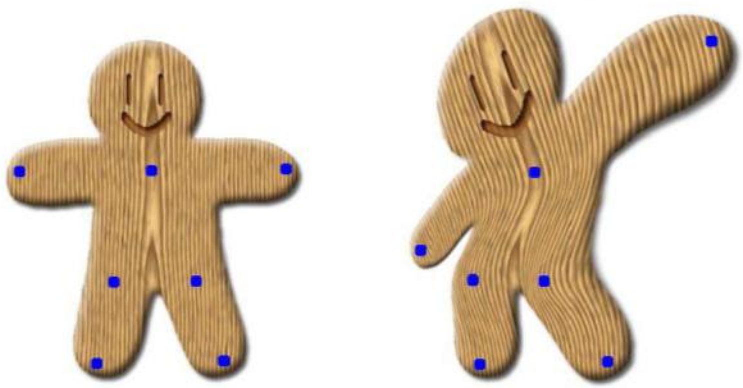
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

- Restrict $A_x(\cdot)$ to similarity



Deformadores

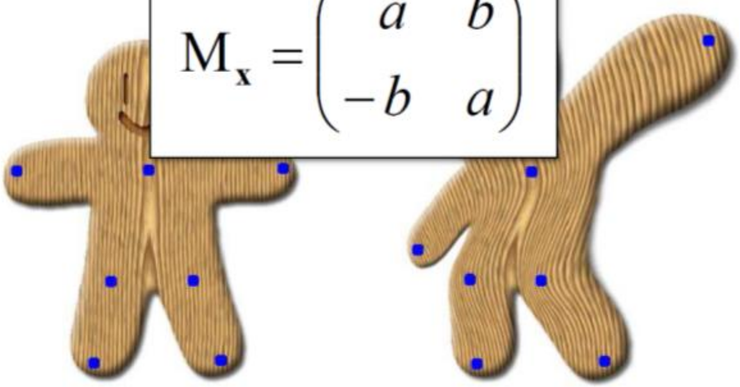
Space deformation

AS-Rigid-As-Possible Deformations

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

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

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



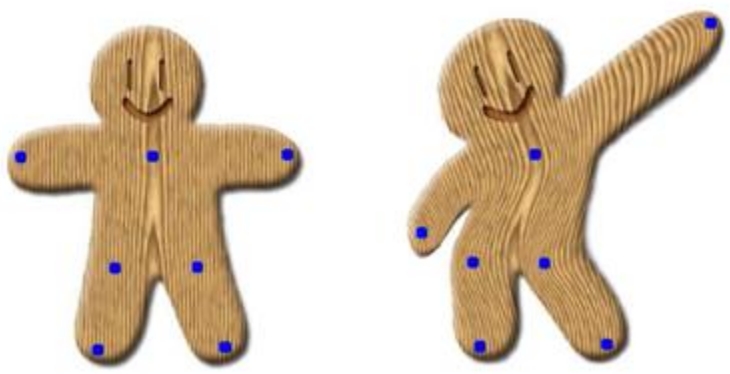
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

- Restrict $A_x(\cdot)$ to rigid



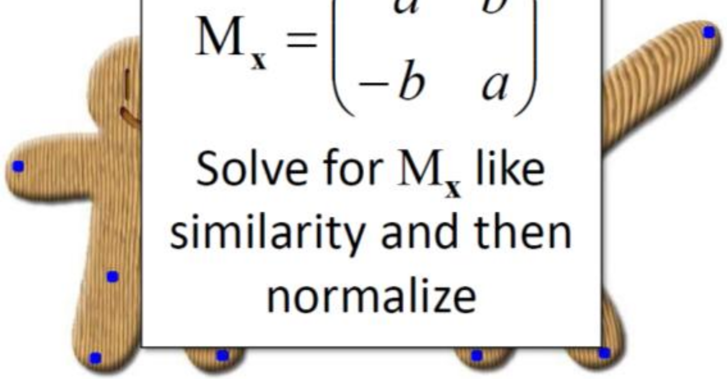
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

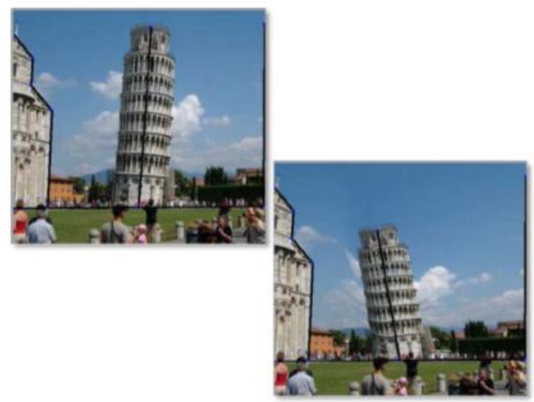
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

- Examples



Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

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

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

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

by polar decomposition of the 3×3 covariance matrix

Deformadores

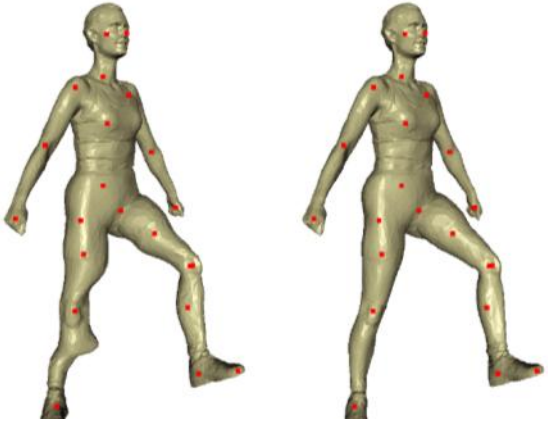
Space deformation

AS-Rigid-As-Possible Deformations

MLS approach – extension to 3D [Zhu & Gortler 2007]

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

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



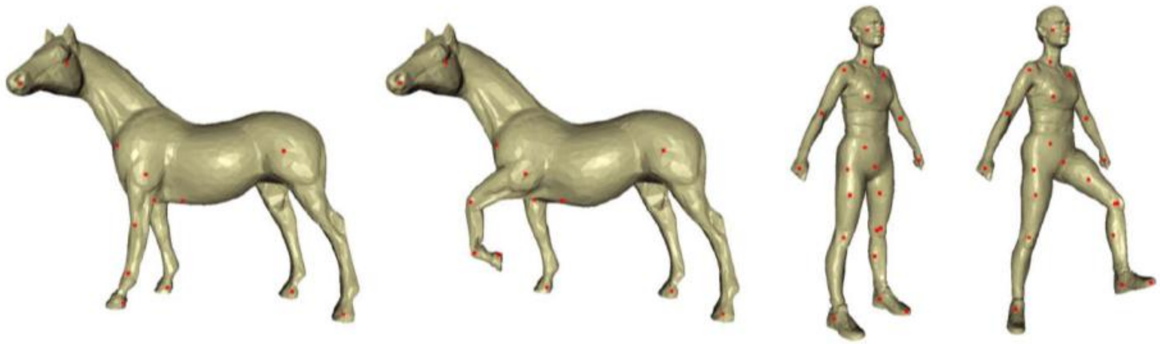
Deformadores

Space deformation

AS-Rigid-As-Possible Deformations

MLS approach – extension to 3D [Zhu & Gortler 2007]

- More results

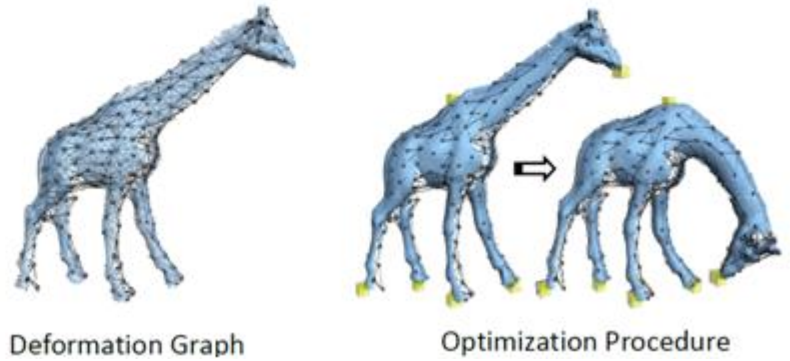


Space deformation

AS-Rigid-As-Possible Deformations

Deformation Graph approach [Sumner et al. 2007]

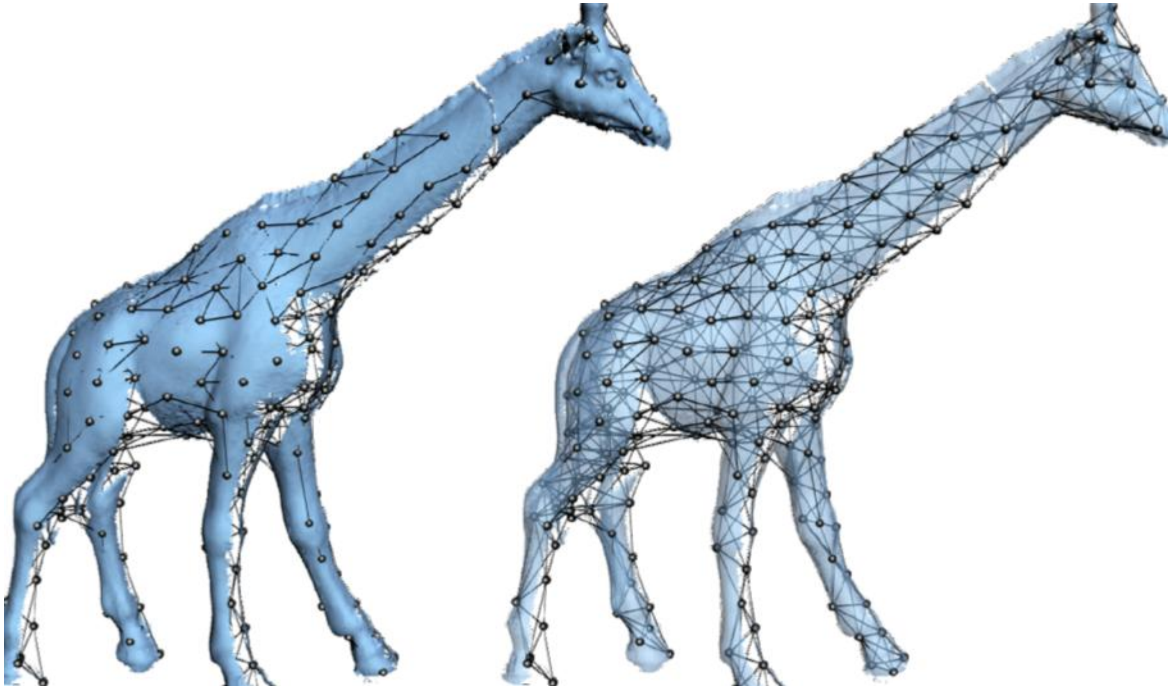
- Surface handles as interface
- Underlying graph to represent the deformation; nodes store rigid transformations
- Decoupling of handles from def. representation



Deformadores

Space deformation

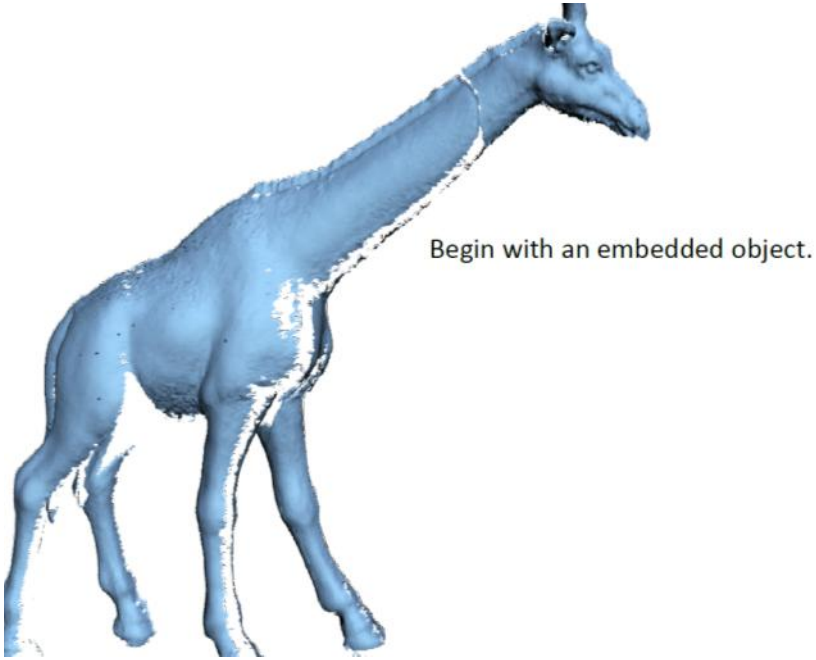
Deformation Graph
[Sumner et al. 2007]



Deformadores

Space deformation

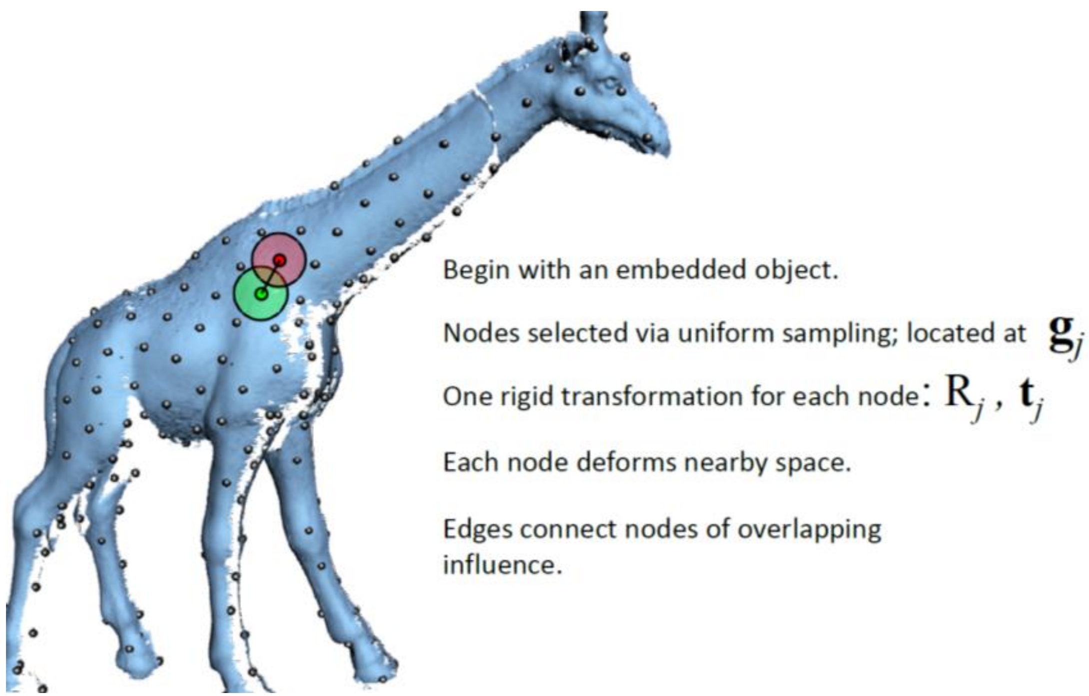
Deformation Graph
[Sumner et al. 2007]



Deformadores

Space deformation

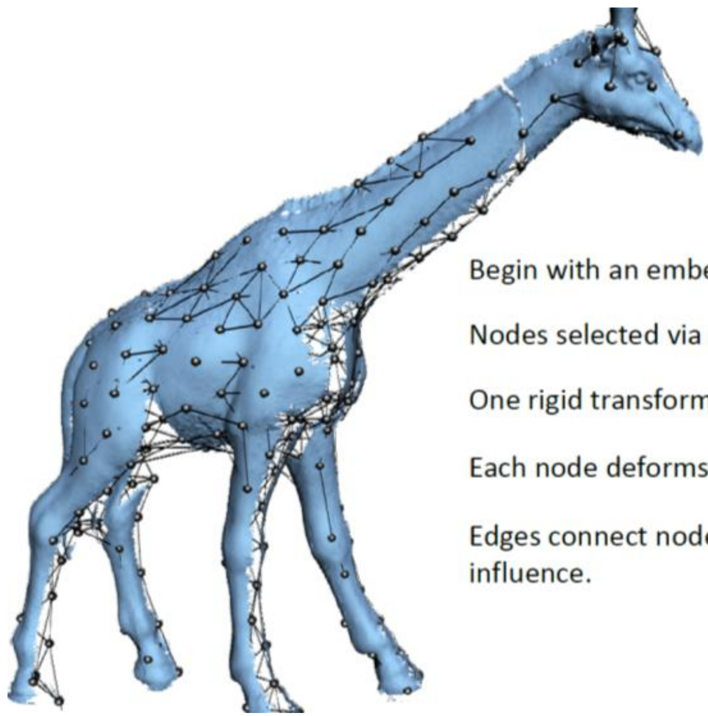
Deformation Graph
[Sumner et al. 2007]



Deformadores

Space deformation

Deformation Graph
[Sumner et al. 2007]



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.

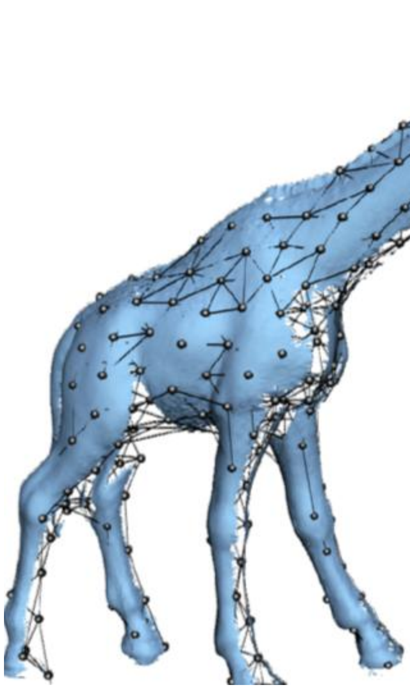
Edges connect nodes of overlapping influence.

Deformadores

Space deformation

Deformation Graph

[Sumner et al. 2007]



Influence of nearby transformations is blended.

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

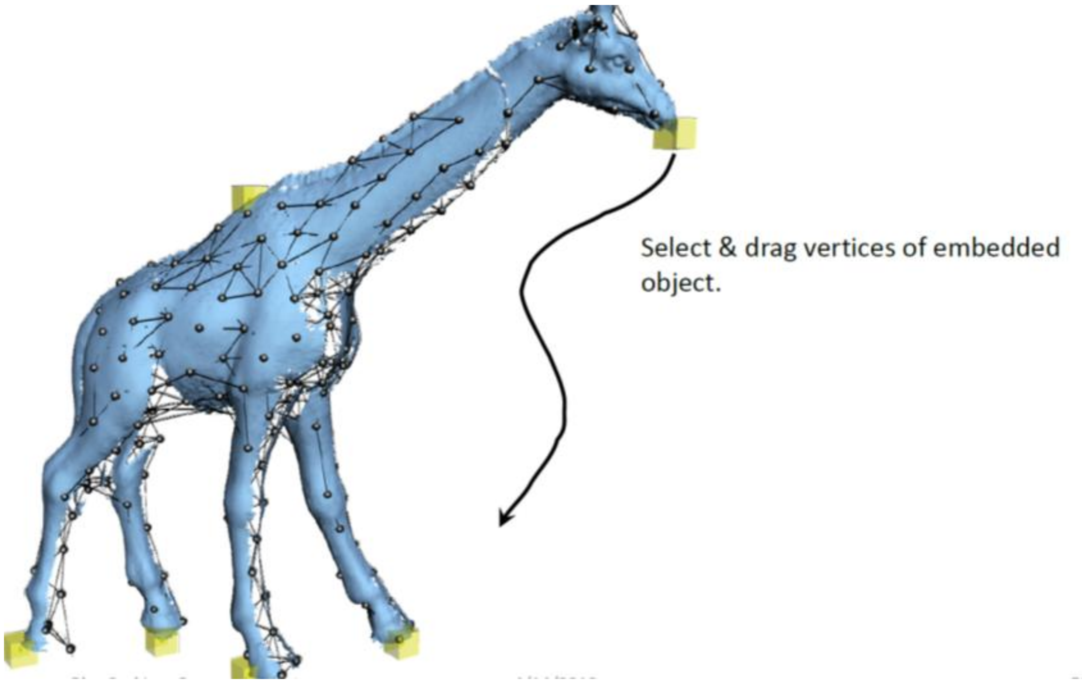
$$w_j(\mathbf{x}) = (1 - \|\mathbf{x} - \mathbf{g}_j\| / d_{\max})^2$$

Deformadores

Space deformation

Deformation Graph

[Sumner et al. 2007]

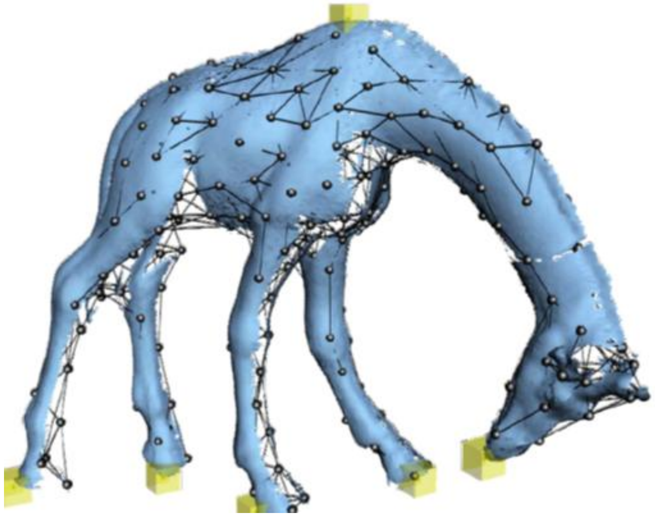


Deformadores

Space deformation

Deformation Graph

[Sumner et al. 2007]



Select & drag vertices of embedded object.

Optimization finds

deformation parameters $\mathbf{R}_j$, $\mathbf{t}_j$.

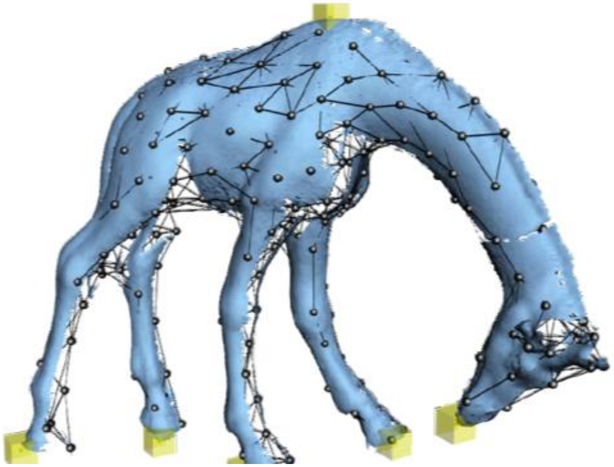
Deformadores

Space deformation

Deformation Graph

[Sumner et al. 2007]

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



Select & drag vertices of embedded object.

Optimization finds

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

Deformadores

Space deformation

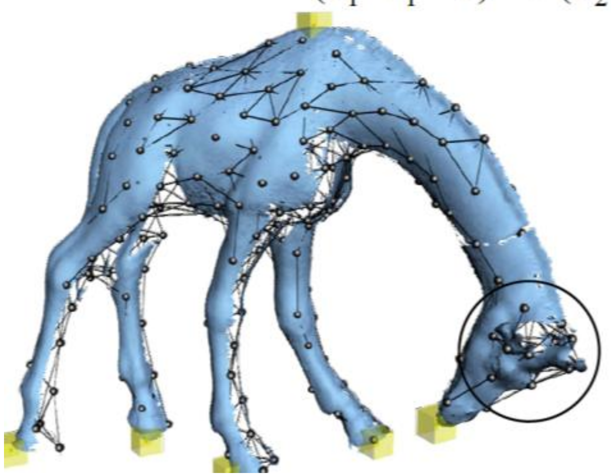
Deformation Graph

[Sumner et al. 2007]

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

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

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



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

Deformadores

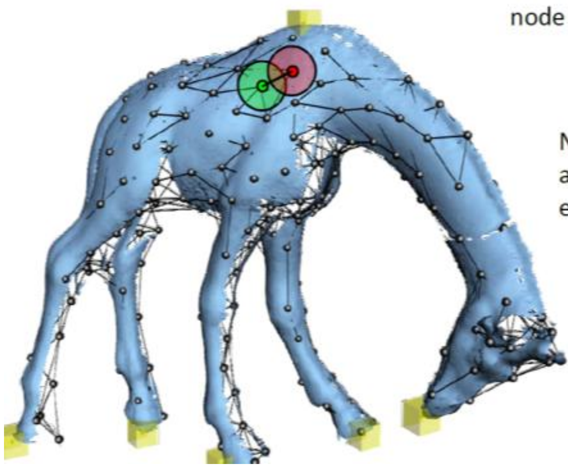
Space deformation

Deformation Graph

[Sumner et al. 2007]

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

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



Neighboring nodes should agree on where they transform each other.

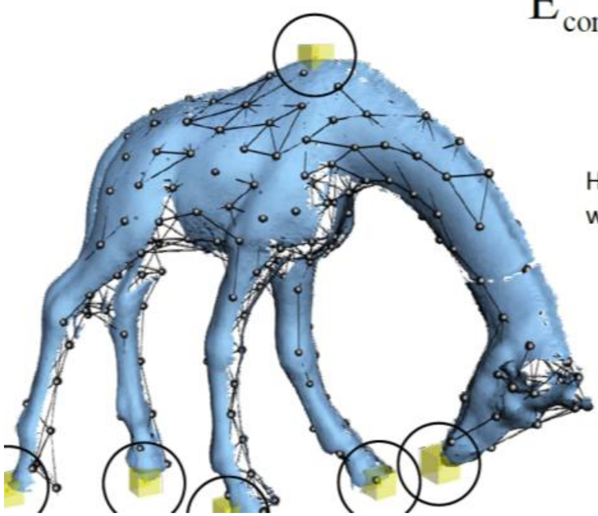
Deformadores

Space deformation

Deformation Graph
[Sumner et al. 2007]

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

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



Handle vertices should go where the user puts them.