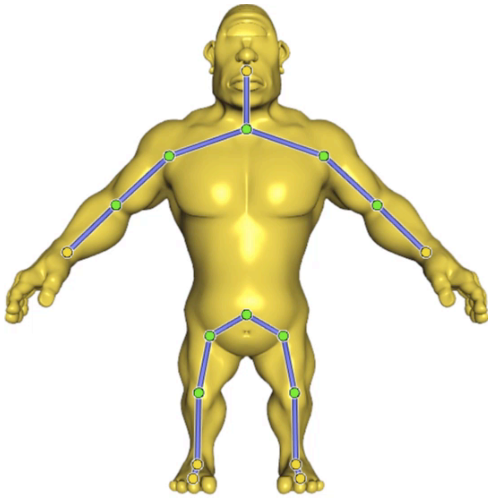
CS 6501

2D/3D Shape Manipulation, 3D Printing

Linear Blend Skinning

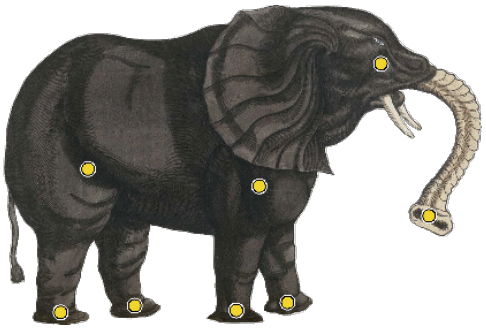
Slides from Olga Sorkine

LBS generalizes to different handle types

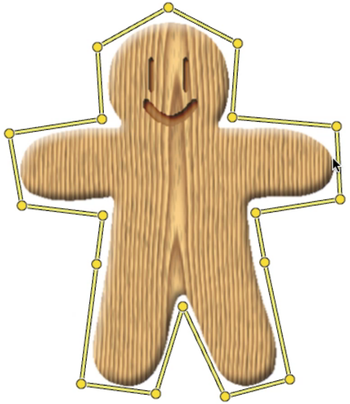


skeletons

regions

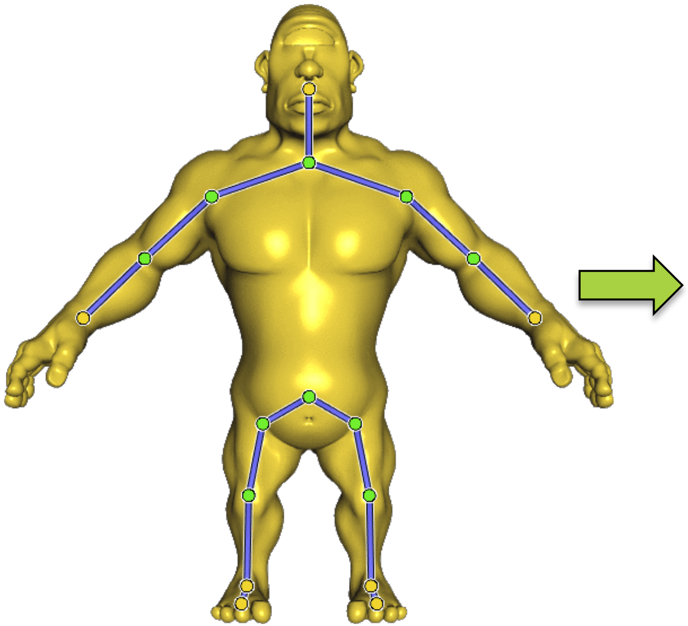


points



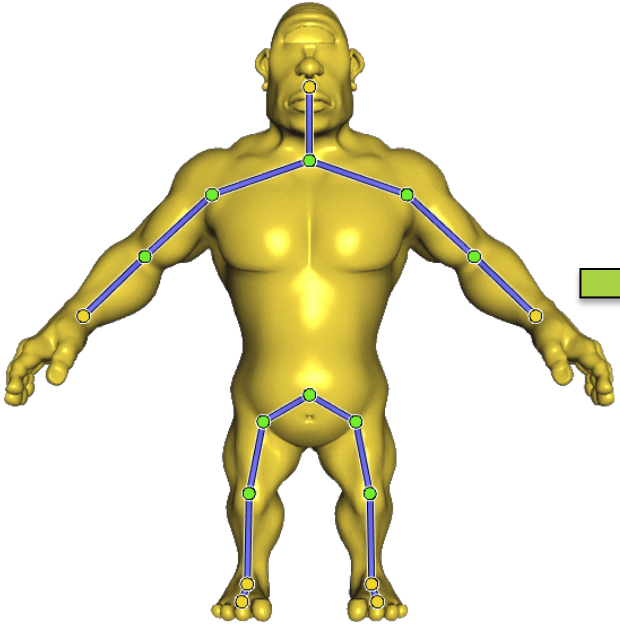
cages

Linear Blend Skinning rigging preferred for its real-time performance

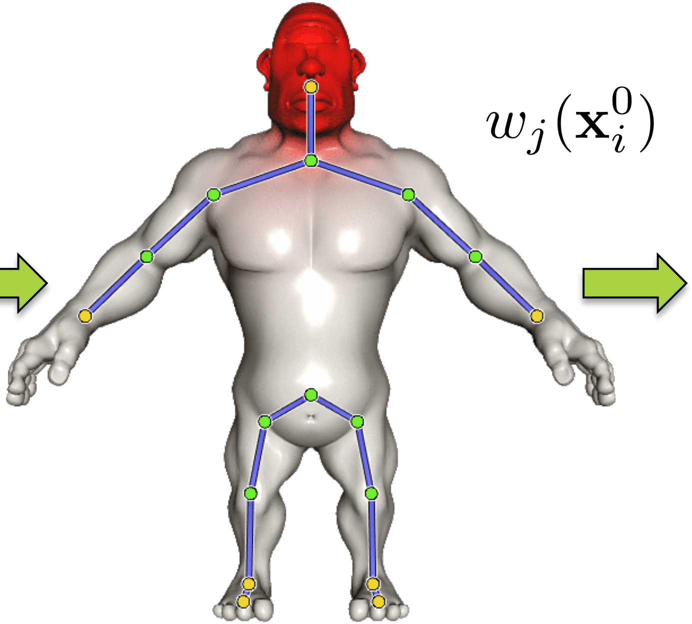


place handles in shape

Linear Blend Skinning rigging preferred for its real-time performance



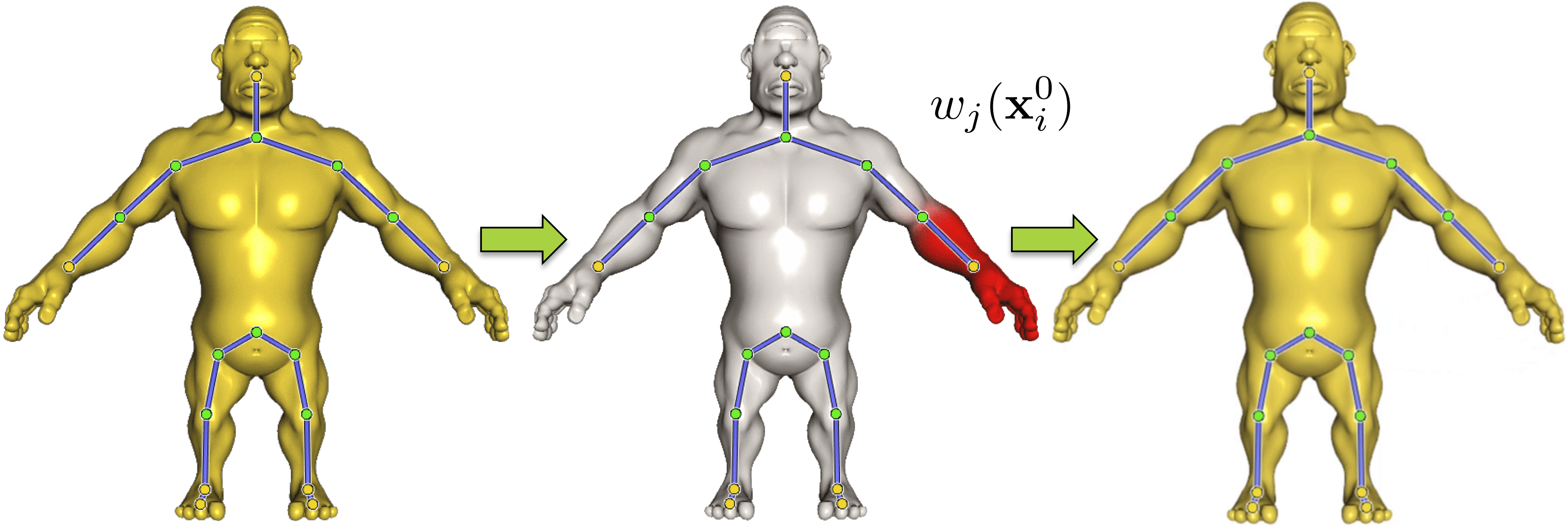
place handles in shape



$w_j(\mathbf{x}_i^0)$

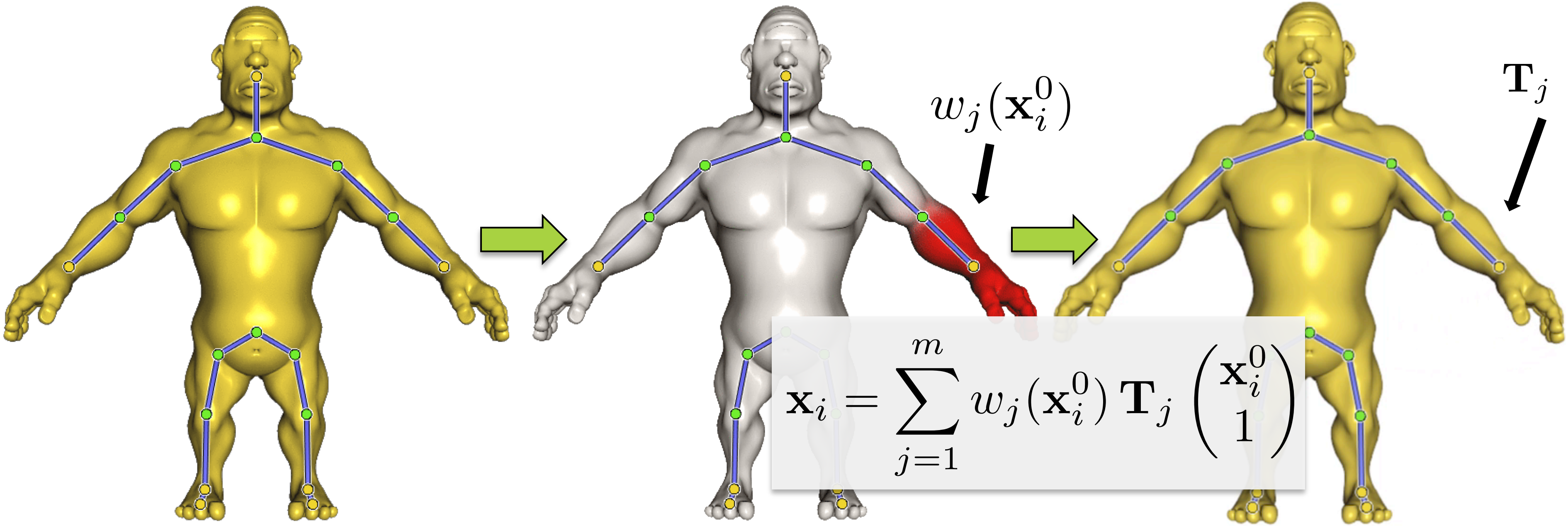
paint weights

Linear Blend Skinning rigging preferred for its real-time performance



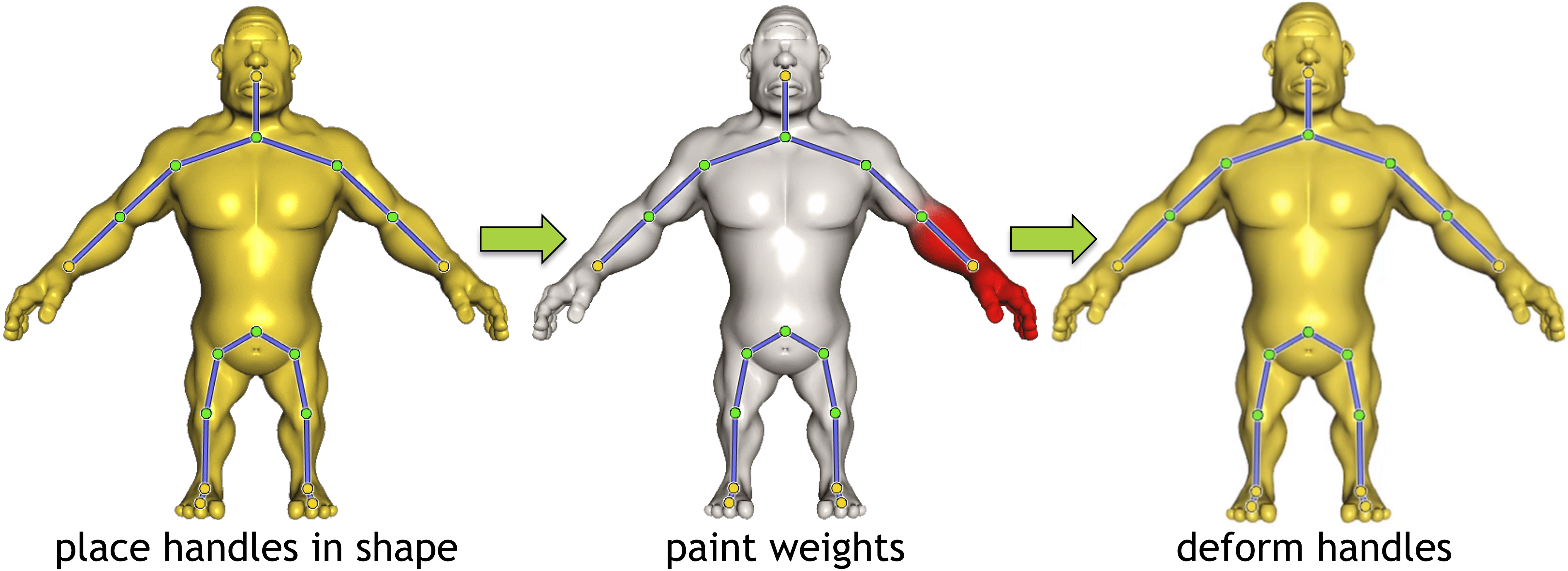
place handles in shape | paint weights | deform handles

Linear Blend Skinning rigging preferred for its real-time performance



place handles in shape | paint weights | deform handles

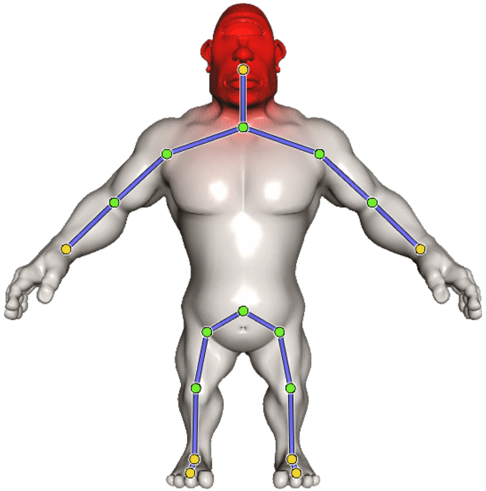
Linear Blend Skinning rigging preferred for its real-time performance

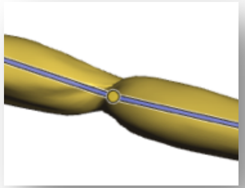


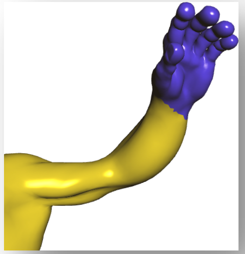
Challenges with LBS

- Weight functions w_j
  - Need intuitive, general and automatic weights
- Degrees of freedom $\mathbf{T}_j$
  - Let the energy decide!
- Richness of achievable deformations
  - Want to avoid common LBS pitfalls – candy wrapper, collapses

$$\mathbf{x}_i = \sum_{j=1}^{m} \underline{w_j(\mathbf{x}_i^0)}\, \underline{\mathbf{T}_j} \begin{pmatrix} \mathbf{x}_i^0 \\ 1 \end{pmatrix}$$



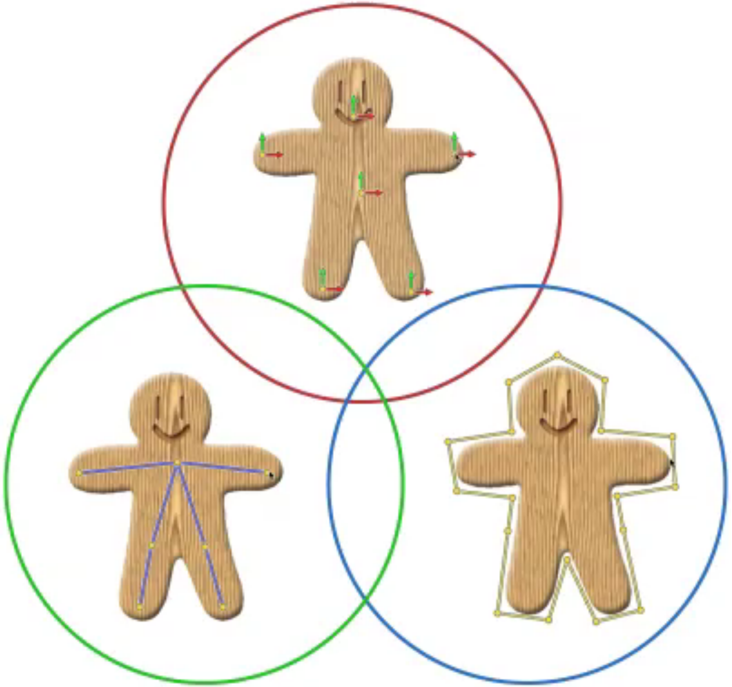




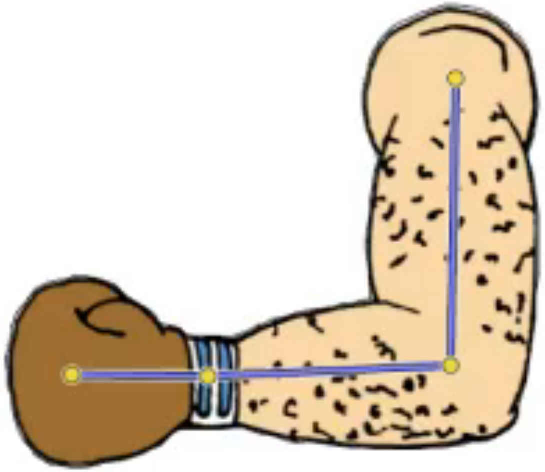
Alec Jacobson, Ilya Baran, Jovan Popović, S
ACM SIGGRAPH 2011; selected for Research Highlights in CACM (2013)

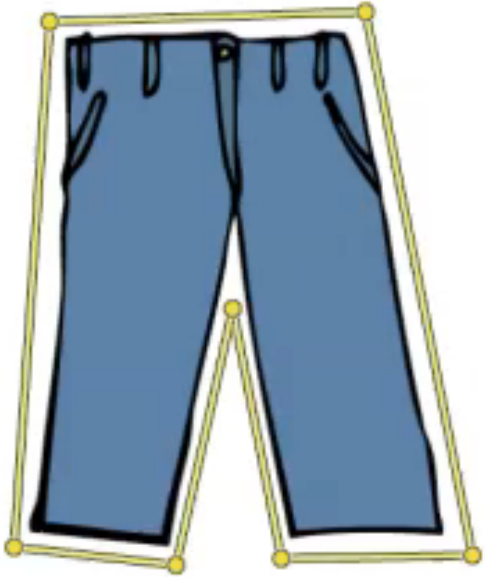
Bounded Biharmonic Weights

Automatic weights that unify points, skeletons and cages



Weights should be smooth, shape-aware, positive and *intuitive*



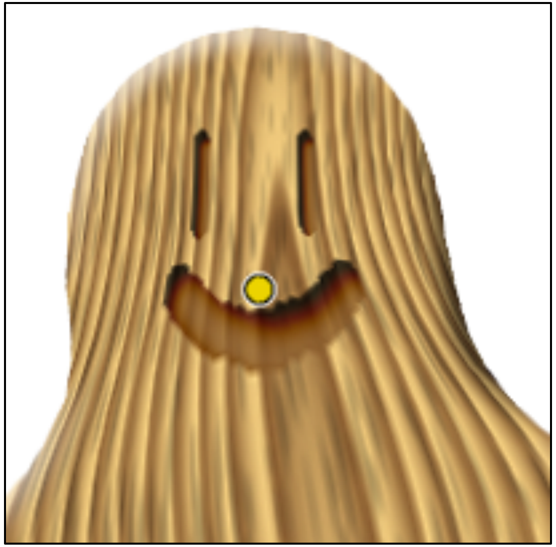


Weights must be smooth everywhere, *especially* at handles

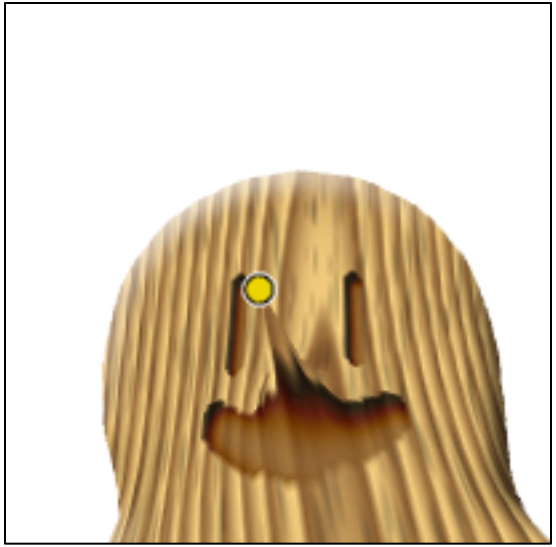
Bounded Biharmonic Weights

Extension of Harmonic Coordinates
[Joshi et al. 2005]

Weights must be smooth everywhere, *especially* at handles

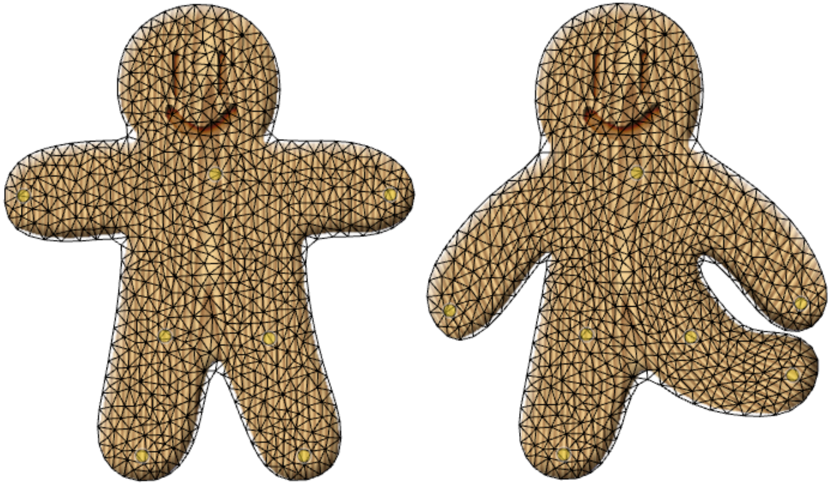


Bounded Biharmonic Weights

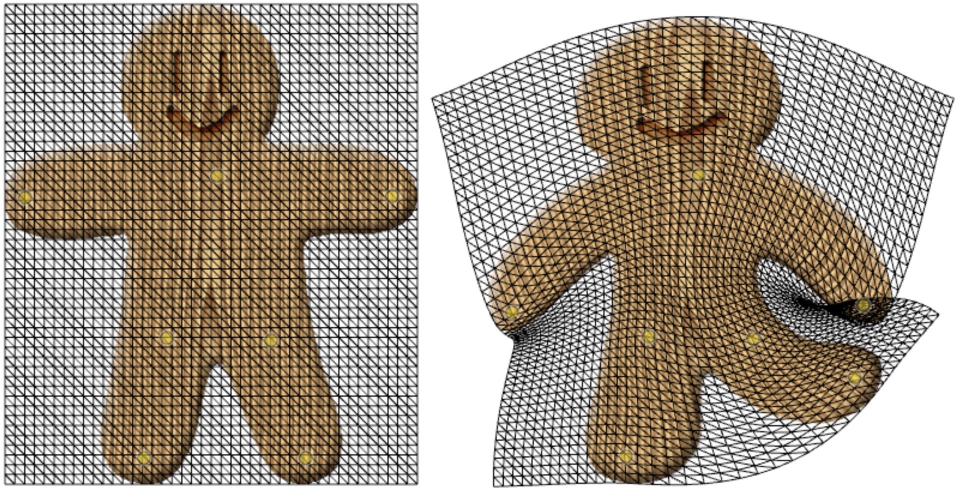


Extension of Harmonic Coordinates
[Joshi et al. 2005]

Shape-awareness ensures respect of domain's features



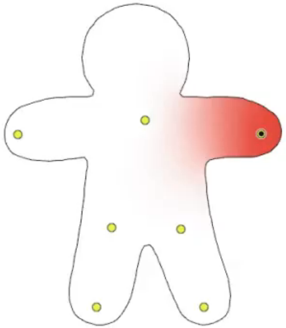
Bounded Biharmonic Weights



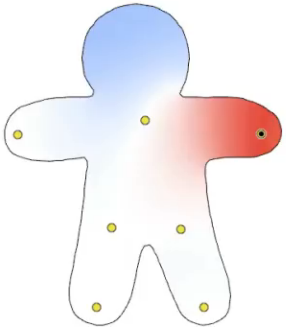
Non-shape-aware methods
e.g. [Schaefer et al. 2006]

Non-negative weights are necessary for intuitive response

Bounded Biharmonic Weights



Unconstrained biharmonic
[Botsch and Kobbelt 2004]



Weights must maintain other simple, but important properties

$$\sum_{j \in H} w_j(\mathbf{x}^0) = 1$$

Partition of unity

Handle vertices

$$w_j \Big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

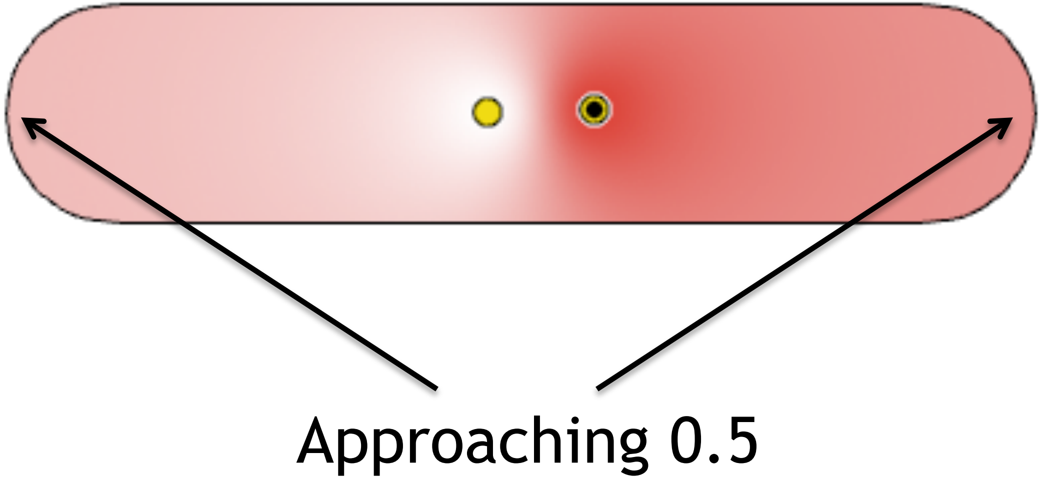
Interpolation of handles

How about $w_j(\mathbf{x}^0) = d(\mathbf{x}^0, H_j)^{-1}$?

Inverse distance methods inherently suffer from *fall-off effect*

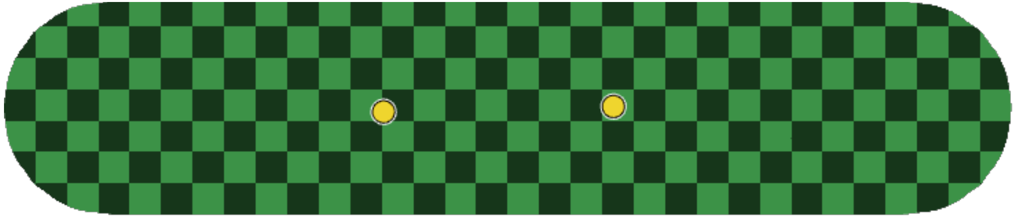
Inverse distance methods inherently suffer from *fall-off effect*

Inverse distance methods inherently suffer from *fall-off effect*

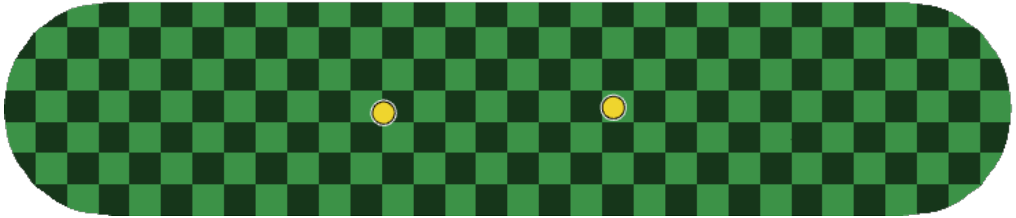


Inverse distance methods inherently suffer from *fall-off effect*

Inverse-distance weights



BBW



Bounded biharmonic weights enforce properties as constraints to minimization

$$\arg\min_{w_j} \frac{1}{2}\int_\Omega |\Delta w_j|^2 dV$$

$$w_j\Big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

Bounded biharmonic weights enforce properties as constraints to minimization

$$\arg\min_{w_j} \frac{1}{2}\int_\Omega |\Delta w_j|^2 dV$$

$$w_j\Big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

Constant inequality constraints

$$0 \le w_j(\mathbf{x}^0) \le 1$$

Partition of unity

$$\sum_{j\in H} w_j(\mathbf{x}^0) = 1$$

Bounded biharmonic weights enforce properties as constraints to minimization

$$\arg\min_{w_j} \frac{1}{2}\int_\Omega |\Delta w_j|^2 dV$$

$$w_j\Big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

Constant inequality constraints

$$0 \le w_j(\mathbf{x}^0) \le 1$$

Solve independently and normalize

$$w_j(\mathbf{x}^0) = \frac{w_j(\mathbf{x}^0)}{\sum_{i\in H} w_i(\mathbf{x}^0)}$$

Weights optimized as precomputation at bind-time

$$\arg\min_{w_j} \frac{1}{2}\int_\Omega |\Delta w_j|^2 dV$$

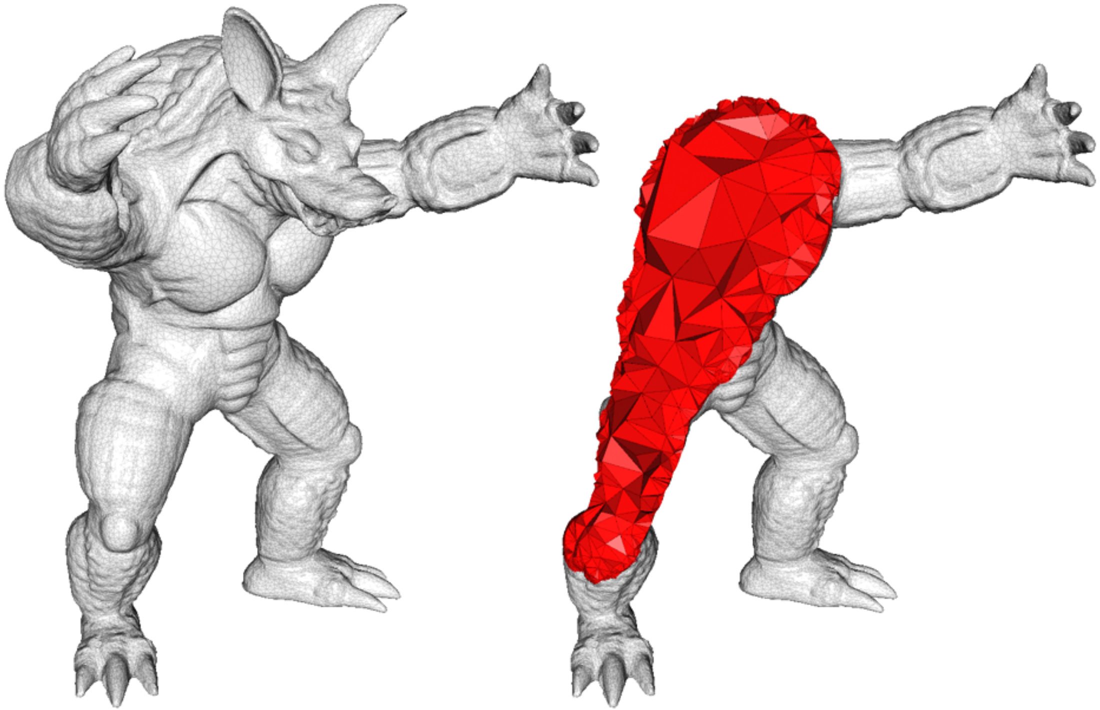
$$w_j\big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

$$0 \le w_j(\mathbf{x}^0) \le 1$$

FEM discretization

- 2D → Triangle mesh
- 3D → Tet mesh



Weights optimized as precomputation at bind-time

$$\arg\min_{w_j} \frac{1}{2}\int_{\Omega} |\Delta w_j|^2 dV$$

$$w_j\big|_{H_k} = \delta_{jk}$$

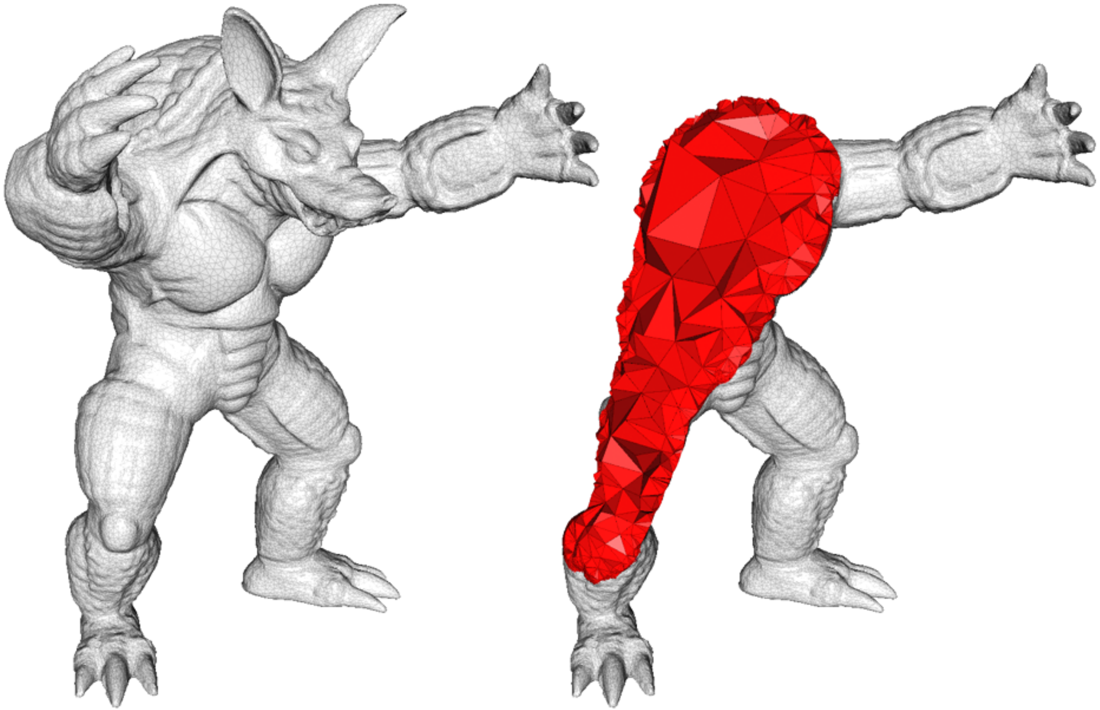
w_j is linear along cage faces

$$0 \le w_j(\mathbf{x}^0) \le 1$$

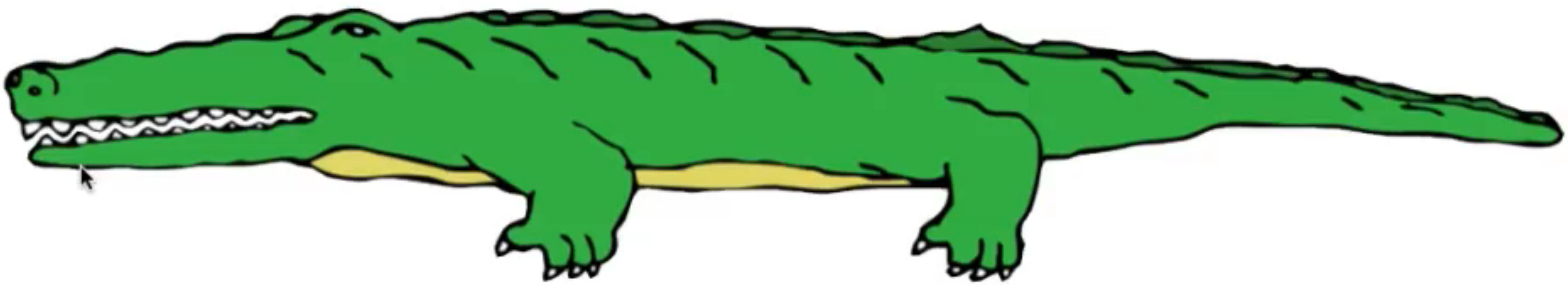
Sparse quadratic programming with constant inequality constraints

2D → less than second per handle

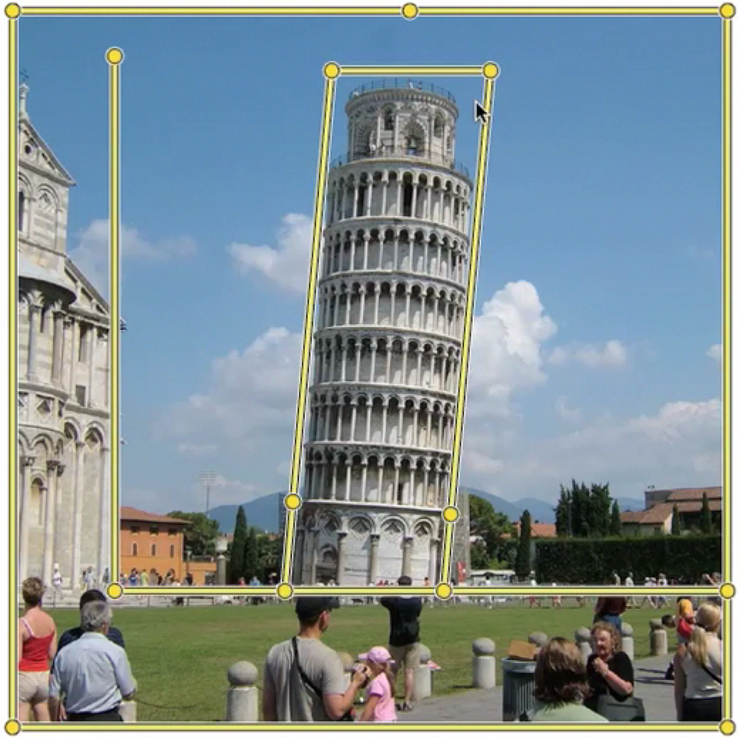
3D → tens of seconds per handle



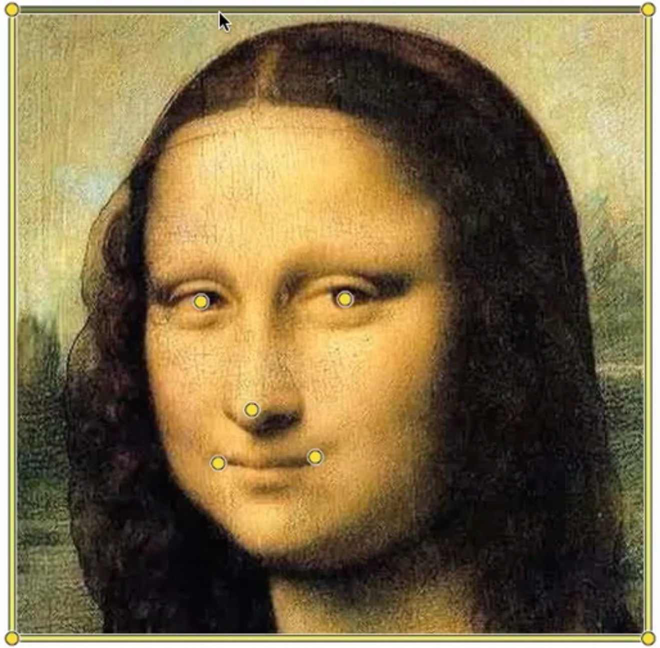
Some examples of BBW in action



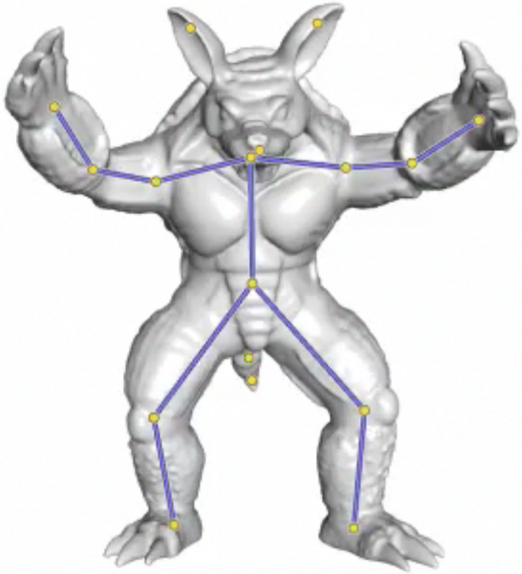
Some examples of BBW in action



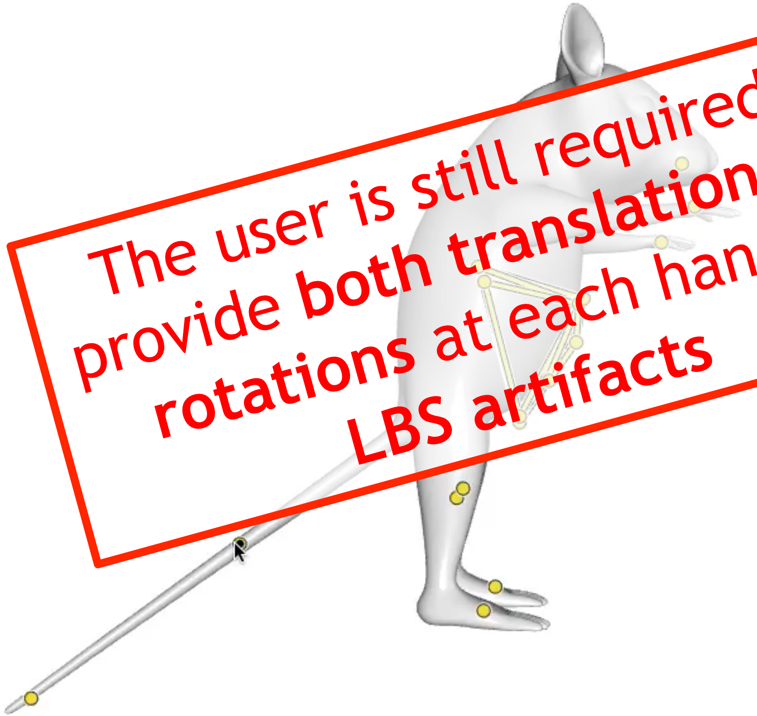
Some examples of BBW in action



3D Characters



Mixing different handle types



The user is still required to provide **both translations** and **rotations** at each handle; **LBS artifacts**

Alec Jacobson, Ilya Baran, Ladislav Kavan, Jovan Popović, S
ACM SIGGRAPH 2012

Fast Automatic Skinning Transformations

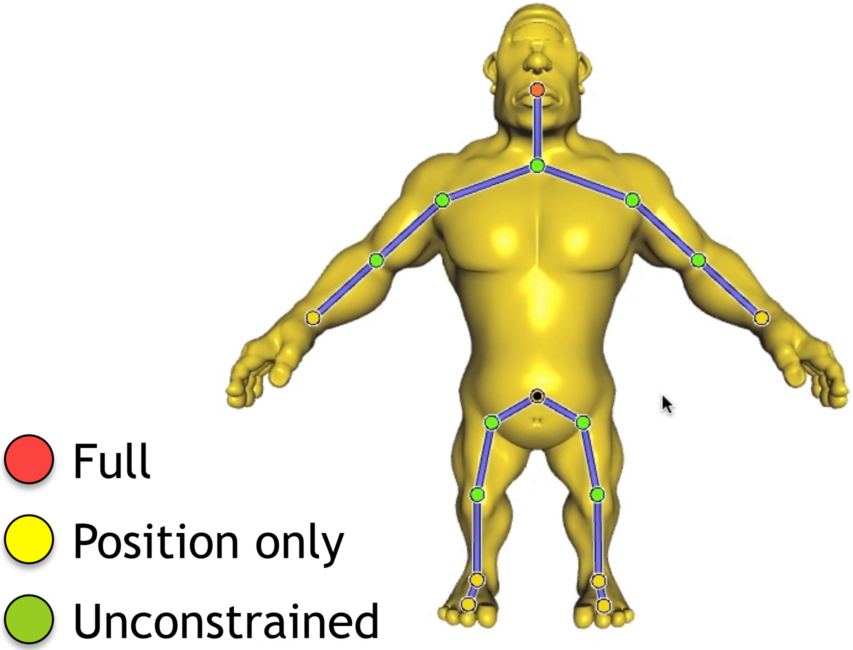
User specifies subset of parameters, automatically optimize remaining ones

Reduced optimization

$$\arg\min_{\mathbf{T}} \; E(\mathbf{M}\mathbf{T})$$

User constraints

$$\underbrace{\begin{bmatrix} \mathbf{I}_{\text{full}} \\ \mathbf{M}_{\text{pos}} \end{bmatrix}}_{\mathbf{M}_{\text{eq}}} \mathbf{T} = \underbrace{\begin{bmatrix} \mathbf{T}_{\text{full}} \\ \mathbf{P}_{\text{pos}} \end{bmatrix}}_{\mathbf{P}_{\text{eq}}}$$



User specifies subset of parameters, automatically optimize remaining ones

Full optimization

$$\arg\min_{\mathbf{x}} \; E(\mathbf{x})$$

Nonlinear elastic energy of the shape

Reduced model

$$\mathbf{x}_i = \sum_{j=1}^{m} w_j(\mathbf{x}_i^0)\, \mathbf{T}_j \begin{pmatrix} \mathbf{x}_i^0 \\ 1 \end{pmatrix}$$

Matrix form

$$\mathbf{x} = \mathbf{M}\mathbf{T}$$

Reduced optimization

$$\arg\min_{\mathbf{T}} \; E(\mathbf{M}\mathbf{T})$$

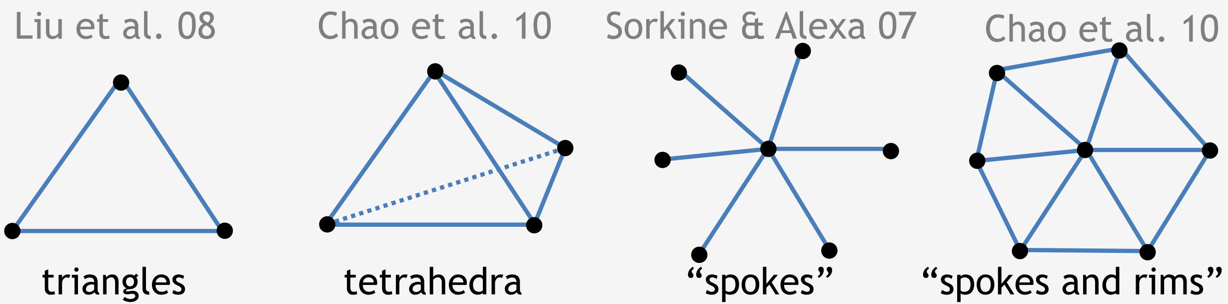
$\mathbf{E}$ is nonlinear; evaluation still expensive

We reduce any *as-rigid-as-possible* energies

Full energy $$E(\mathbf{x}) = \sum_{k \in \text{Cells}} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \left\| (\mathbf{x}_i - \mathbf{x}_j) - \mathbf{R}_k(\mathbf{x})(\mathbf{x}_i^0 - \mathbf{x}_j^0) \right\|^2$$

Best rotation that aligns deformed configuration $\mathcal{E}_k$ to rest-pose

ARAP edge-sets



We reduce any *as-rigid-as-possible* energies

Full energy $$E(\mathbf{x}) = \sum_{k \in \text{Cells}} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|(\mathbf{x}_i - \mathbf{x}_j) - \mathbf{R}_k(\mathbf{x})(\mathbf{x}_i^0 - \mathbf{x}_j^0)\|^2$$

Best rotation that aligns deformed configuration $\mathcal{E}_k$ to rest-pose

Reduction to LBS subspace: $$\mathbf{x} = \mathbf{M}\mathbf{T}$$

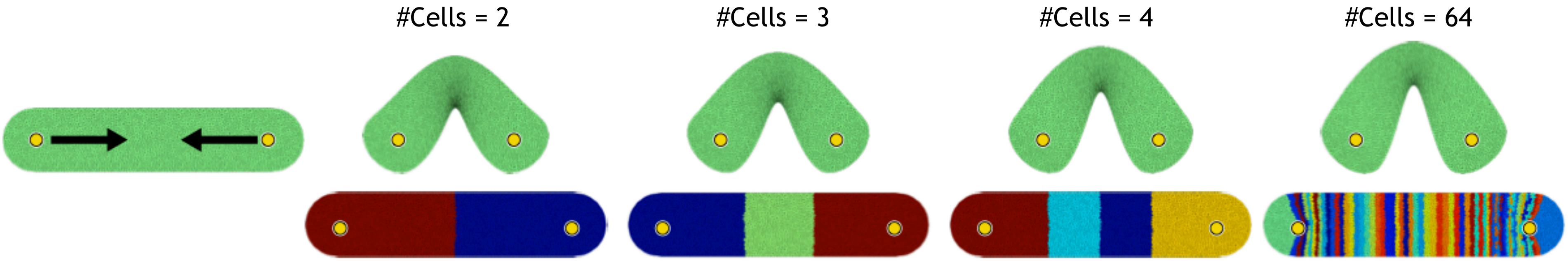
Problem: still many SVDs to do! $$\#\{\mathbf{R}_k(\mathbf{x})\} = O(\#\text{vertices})$$

Rotation evaluations may be reduced by clustering in *weight space*

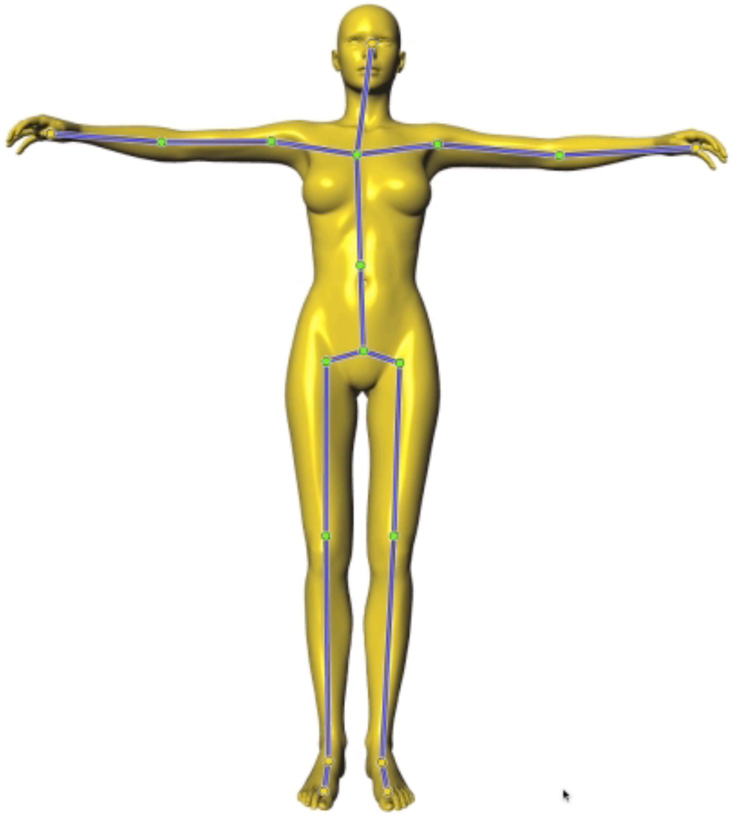
$$E(\mathbf{x}) = \sum_{k \in \text{Cells}} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \|(\mathbf{x}_i - \mathbf{x}_j) - \mathbf{R}_k(\mathbf{x})(\mathbf{x}_i^0 - \mathbf{x}_j^0)\|^2$$

Rotation evaluations may be reduced by clustering in *weight space*

$$E(\mathbf{x}) = \sum_{k \in \text{Cells}} \sum_{(i,j) \in \mathcal{E}_k} c_{ijk} \| (\mathbf{x}_i - \mathbf{x}_j) - \mathbf{R}_k(\mathbf{x})(\mathbf{x}_i^0 - \mathbf{x}_j^0) \|^2$$



Real-time automatic degrees of freedom



With more and more user constraints we fall back to standard skinning

Extra weights would expand subspace to approximate elasticity better

$$\mathbf{x}_i = \sum_{j=1}^{m} w_j(\mathbf{x}_i^0)\, \mathbf{T}_j \begin{pmatrix} \mathbf{x}_i^0 \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \mathbf{M}\mathbf{T}$$

Extra weights would expand subspace to approximate elasticity better

$$\mathbf{x}_i = \sum_{j=1}^{m} w_j(\mathbf{x}_i^0)\,\mathbf{T}_j \begin{pmatrix}\mathbf{x}_i^0 \\ 1\end{pmatrix} + \sum_{k=1}^{m_{\text{extra}}} \tilde{w}_k(\mathbf{x}_i^0)\,\mathbf{T}_k \begin{pmatrix}\mathbf{x}_i^0 \\ 1\end{pmatrix}$$

$$\mathbf{x} = \mathbf{M}\mathbf{T} + \mathbf{M}_{\text{extra}}\mathbf{T}_{\text{extra}}$$

Extra weights would expand subspace to approximate elasticity better

Need:

- smooth
- local, sparse
- respect intent of original weights

$$\mathbf{x}_i = \sum_{j=1}^{m} w_j(\mathbf{x}_i^0)\,\mathbf{T}_j \begin{pmatrix}\mathbf{x}_i^0 \\ 1\end{pmatrix} + \sum_{k=1}^{m_{\text{extra}}} \tilde{w}_k(\mathbf{x}_i^0)\,\mathbf{T}_k \begin{pmatrix}\mathbf{x}_i^0 \\ 1\end{pmatrix}$$

Do not need:

- partition of unity
- interpolation
- scale, sign, etc.

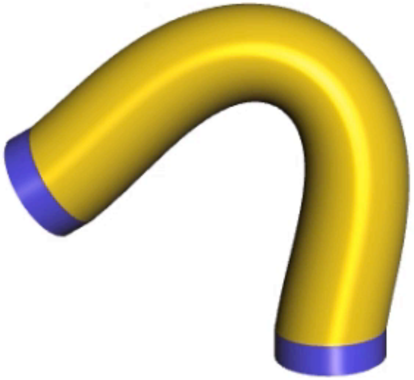
$$\mathbf{x} = \mathbf{M}\mathbf{T} + \mathbf{M}_{\text{extra}}\mathbf{T}_{\text{extra}}$$

Extra weights expand deformation subspace, while respecting user intent

Computation:

- Distribute samples in weight space
- Smooth B-Spline “bumps” around samples

Subspace now rich enough for lightning fast **variational elastic modeling**

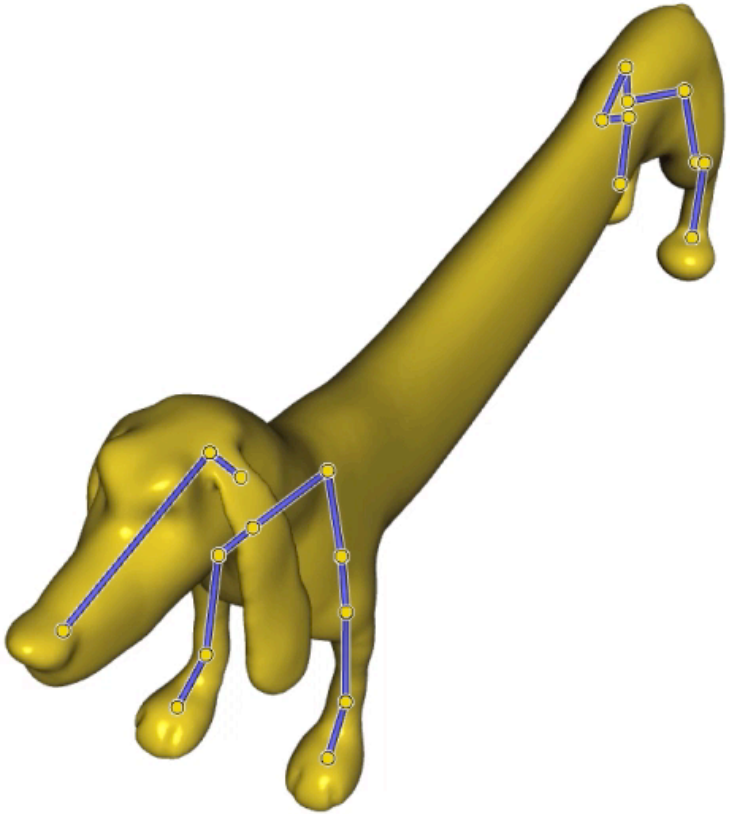


Reference nonlinear deformation:
PriMo [Botsch et al. SGP 2006]

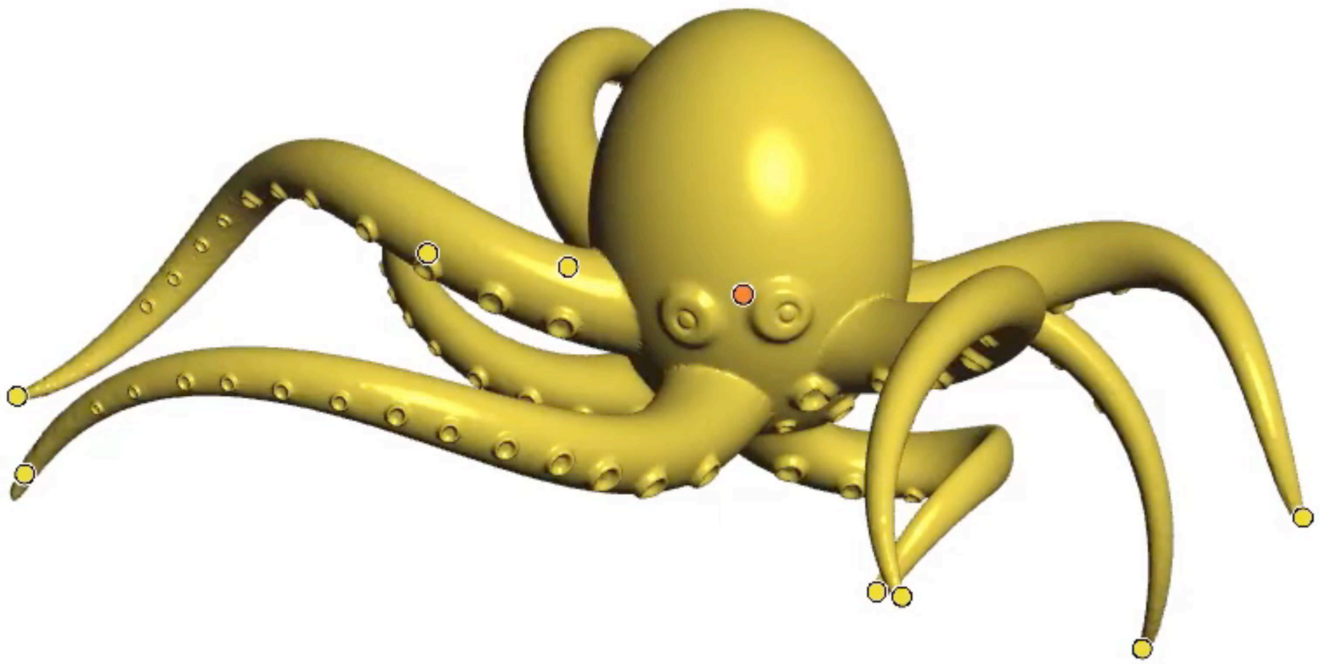
Our reduced method

Extra weights and disjoint skeletons make flexible control easy

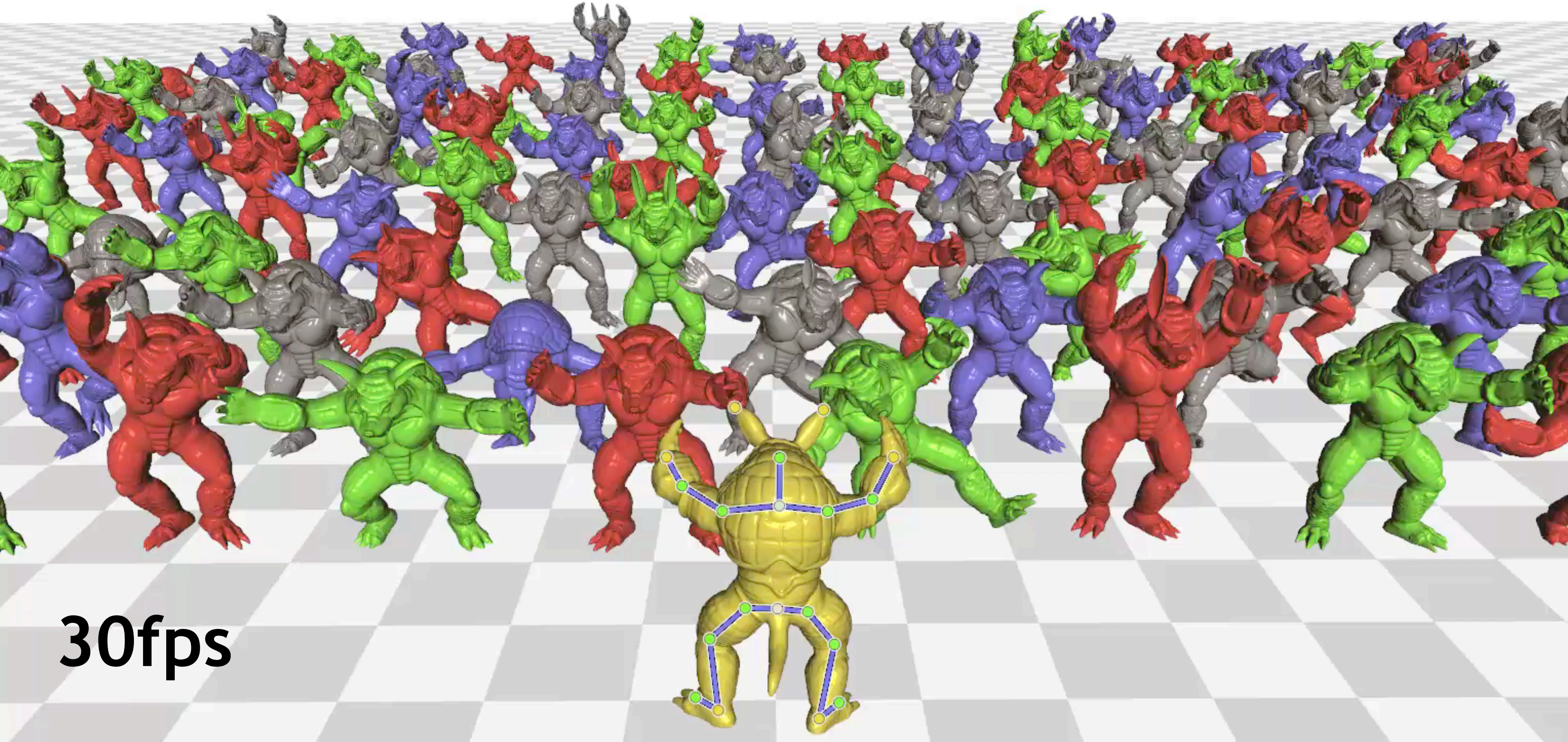
Shape-aware IK!



Simple drag-only interface for point handles



100 Armadillos, 86K triangles each



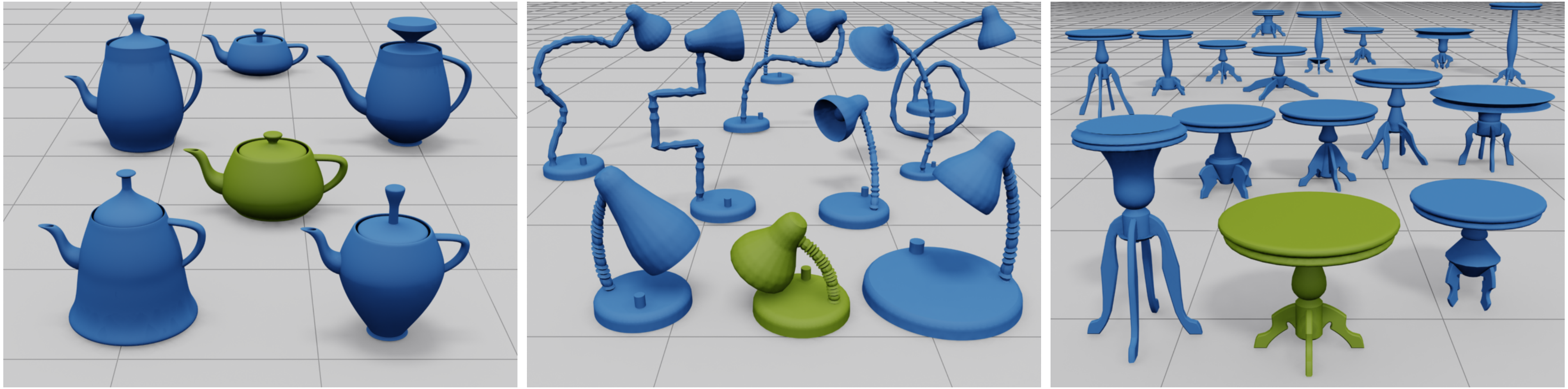
30fps

Realtime variational mesh editing solved…

What next??

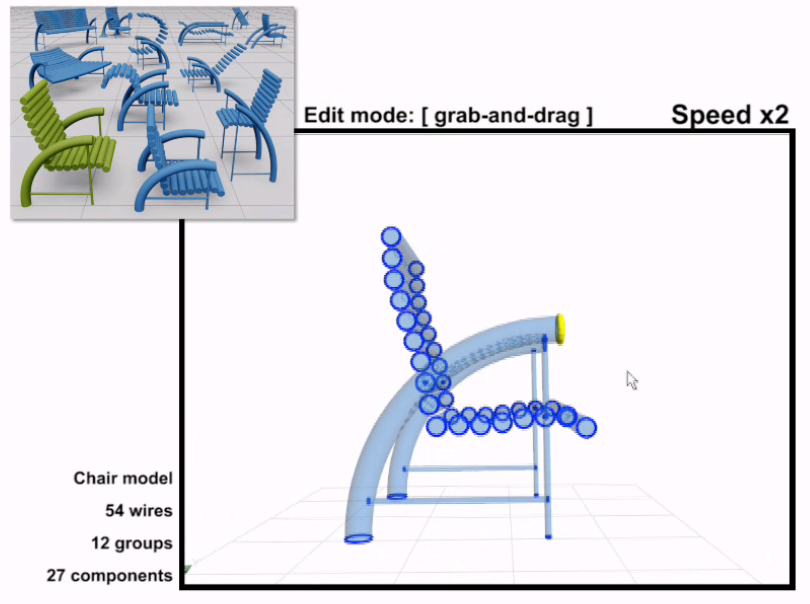
Non-Elastic Deformations

- Many shapes, e.g. man-made, are **not** made of rubber
  - Extract and preserve high-level structures while editing!

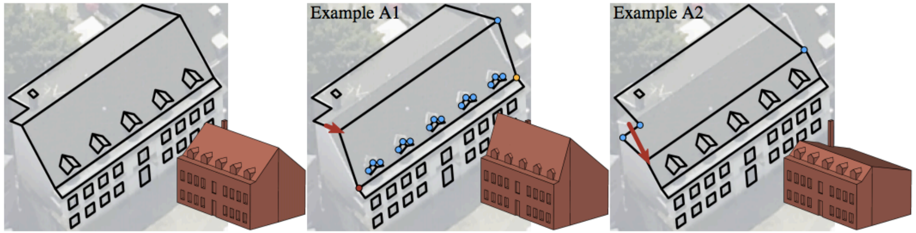


Images from **iWires** [Gal, S, Mitra, Cohen-Or, SIGGRAPH 2009]

Non-Elastic Deformations

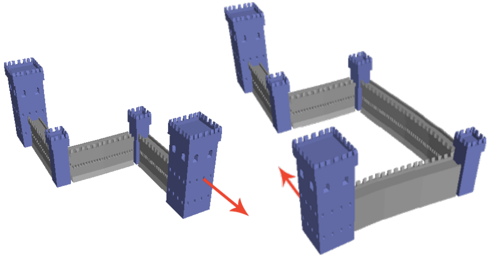


iWires [Gal, S, Mitra, Cohen-Or 2009]
Sequels, e.g. "Component-wise controllers for structure-preserving shape manipulation", EG 2011



Habbecke & Kobbelt, EG 2012

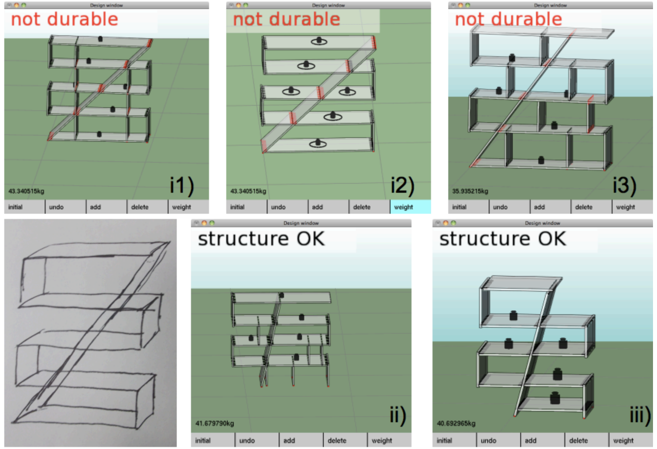
Bokeloh et al.
SIGGRAPH 2012



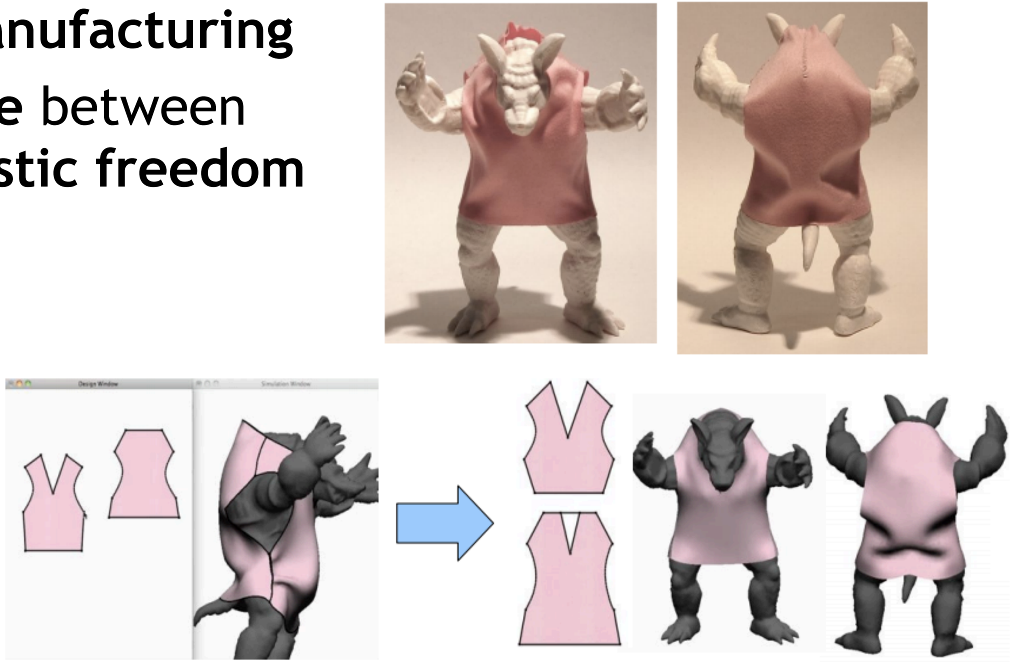
Milliez et al.
EG 2013

Modeling for the Real World

- From modeling directly to **manufacturing**
- Need to find the right **balance** between **physical constraints** and **artistic freedom**



Umetani et al. SIGGRAPH 2012



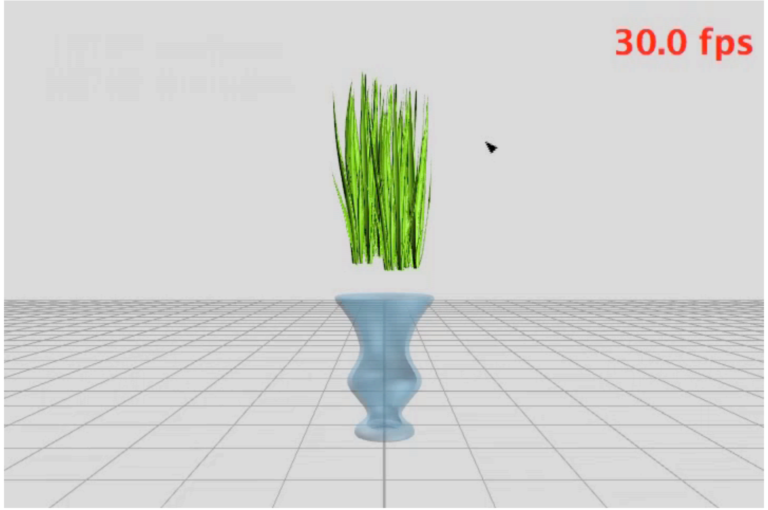
Umetani et al. SIGGRAPH 2011

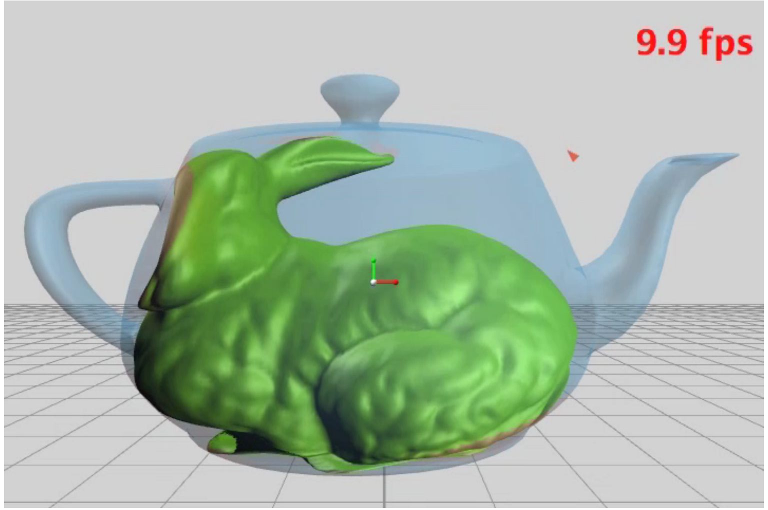
Modeling for the Real World

- Specialized systems vs. general principles?

Modeling for the Real World

- Specialized systems vs. general principles?
- Example: **Self-intersections and collisions**

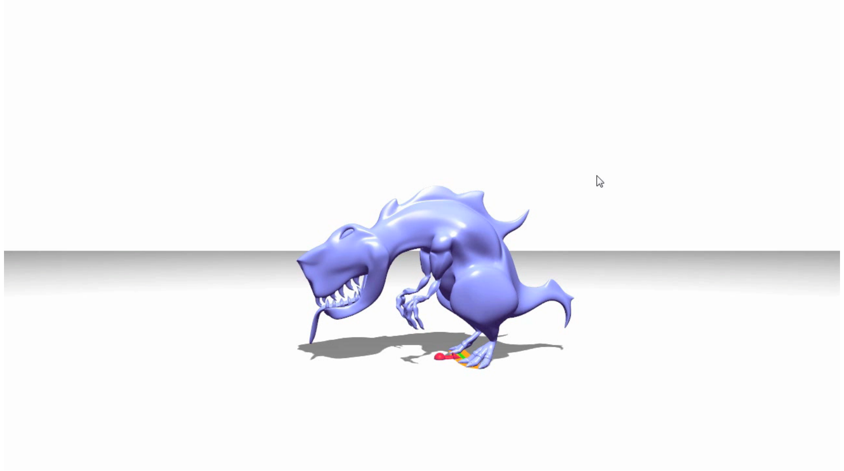




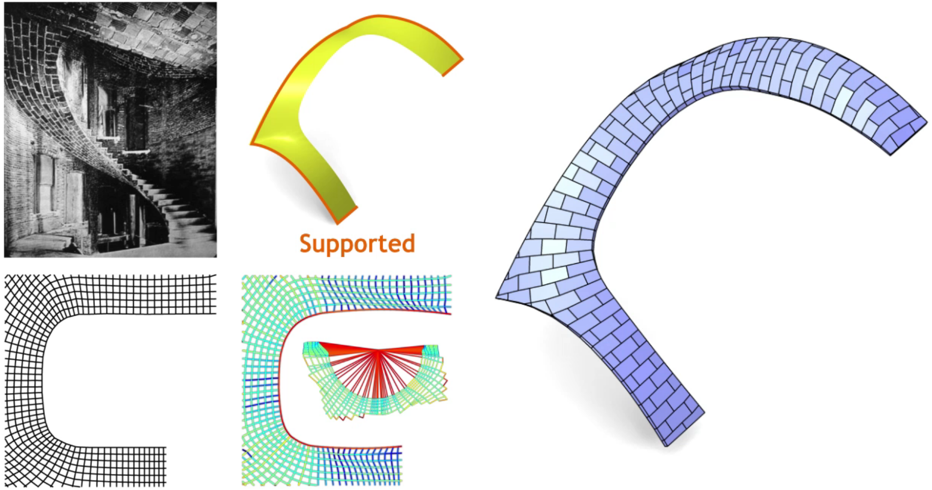
“Interface Aware Geometric Modeling”, Harmon, Panozzo, S, Zorin, SIGGRAPH ASIA 2011

Modeling for the Real World

- Specialized systems vs. general principles?
- Example: **Gravity**



“Make It Stand”, Prevost, Whiting, Lefebvre, S
SIGGRAPH 2013



“Designing Masonry Models”, Panozzo, Block, S
SIGGRAPH 2013

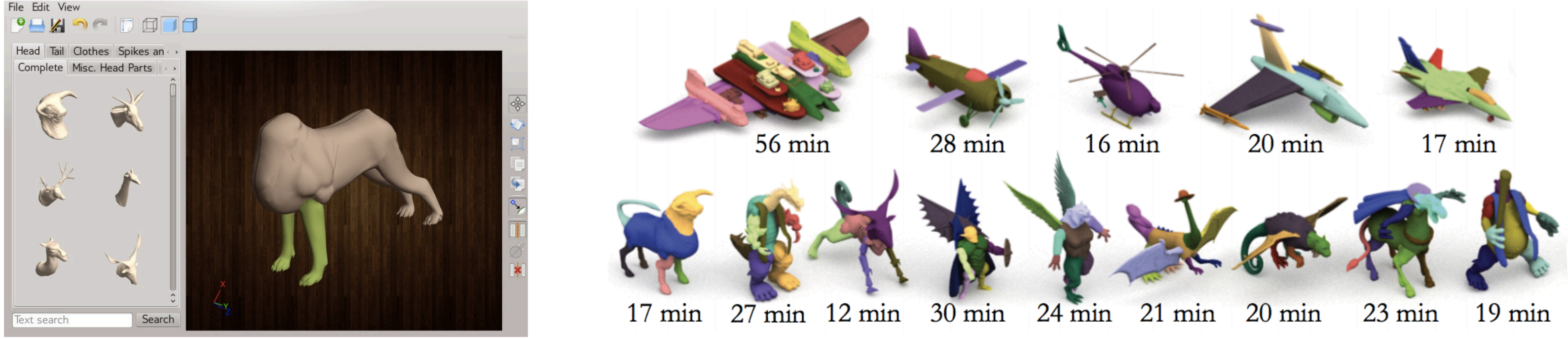
Big Data

- Large model collections: learn **model structure, semantic segmentation**, inspiration for modeling

"Co-Hierarchical Analysis of Shape Structures", van Kaick et al., SIGGRAPH 2013

Big Data

- Large model collections: learn model structure, semantic segmentation, **inspiration for modeling**



"Probabilistic Reasoning for Assembly-Based 3D Modeling", Chaudhuri et al., SIGGRAPH 2011

- Not just static data but **modeling process data?**

Thank You!