Bounded Biharmonic Weights for Real-Time Deformation

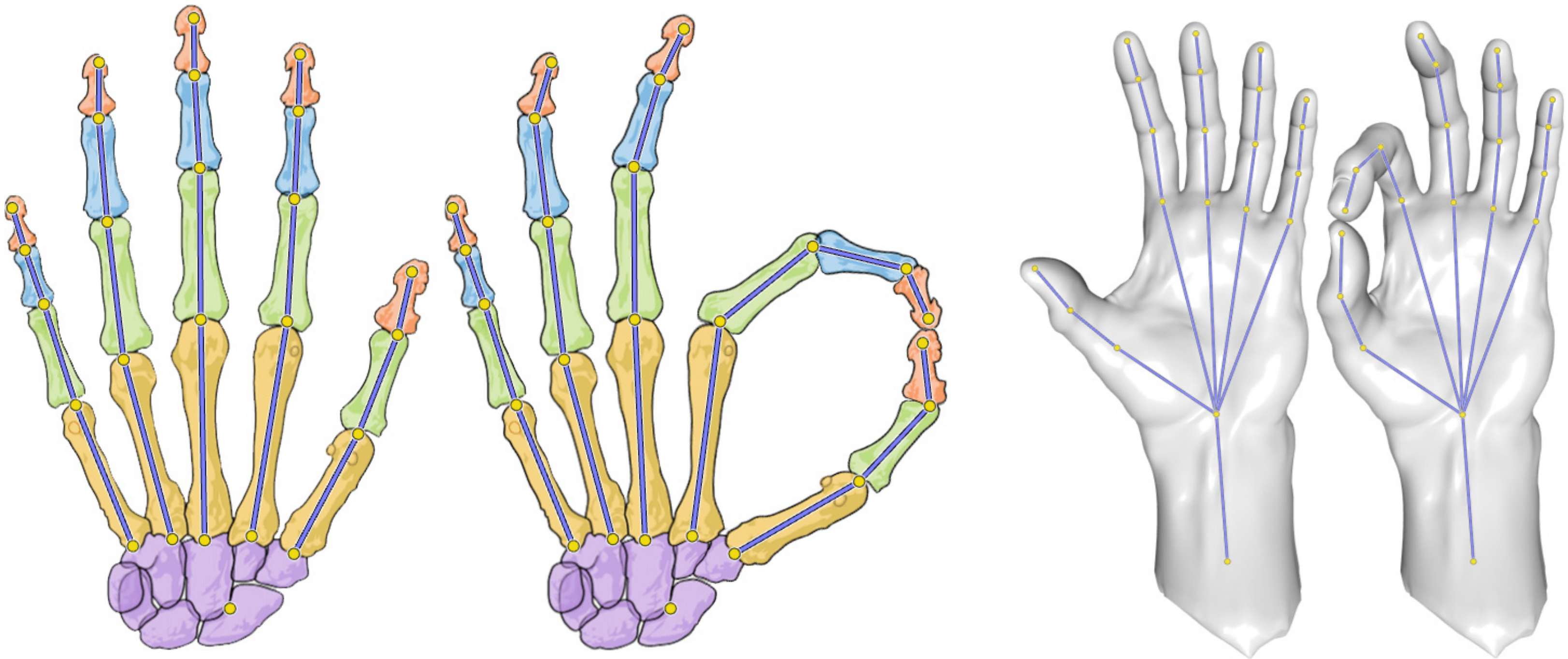
Alec Jacobson — NYU, ETH Zurich
Ilya Baran — Disney Research, Zurich
Jovan Popović — Adobe Systems, Inc.
Olga Sorkine — NYU, ETH Zurich

INTERACTIVE GEOMETRY LAB

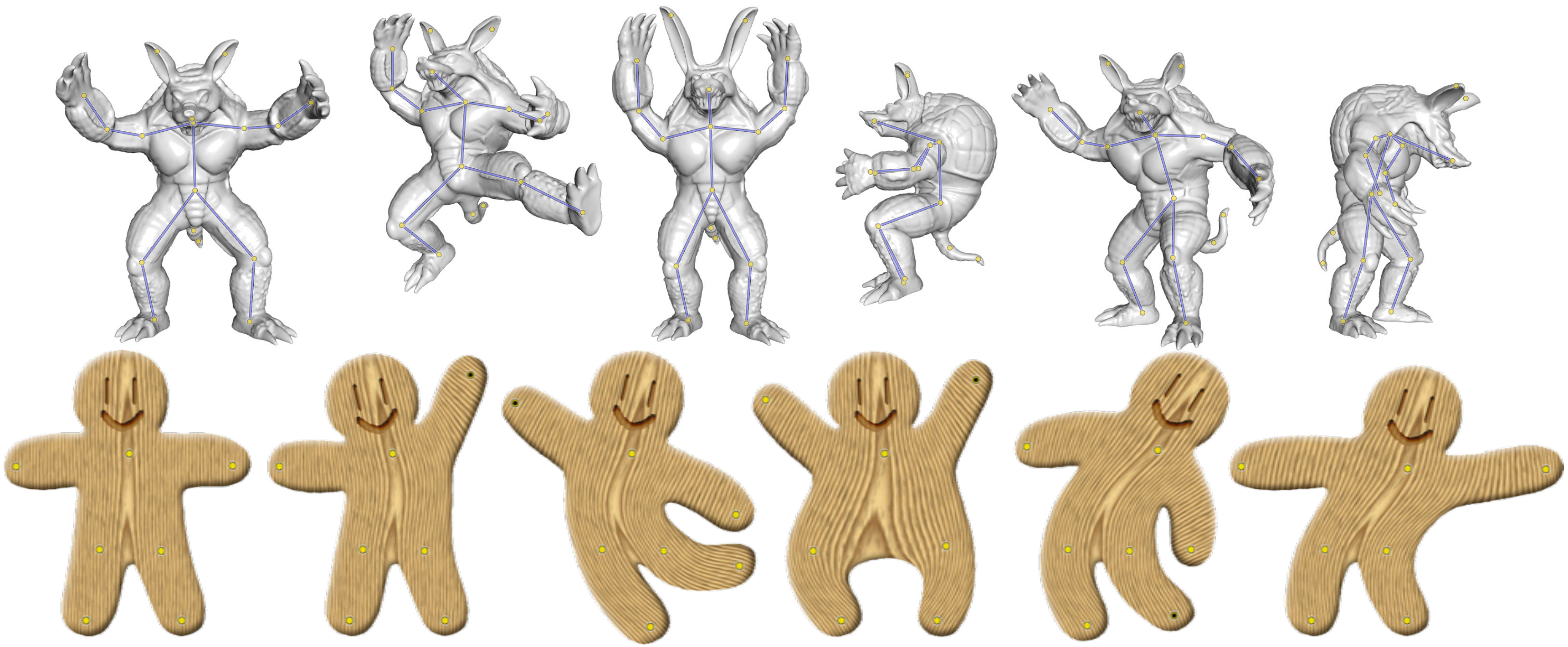
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

September 14, 2011

Real-time performance critical for interactive design and animation



Real-time performance critical for interactive design and animation



Recent works provide high quality results at cost of pose time performance

Typical stress test of high quality deformation system require pose-time optimization [Botsch et al. 2006]

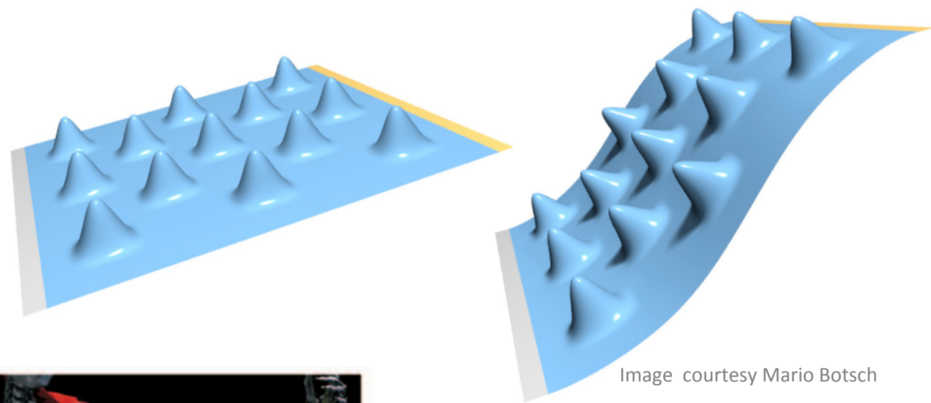


Image courtesy Mario Botsch

Physically accurate muscle systems require off-line simulation [Teran et al. 2005]

Image courtesy Joseph Teran

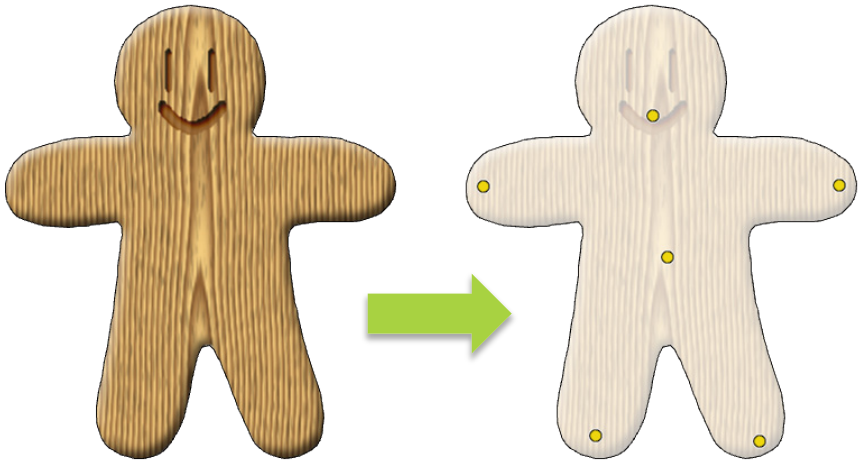
Linear Blend Skinning can’t match quality but makes up in real-time performance

load shape

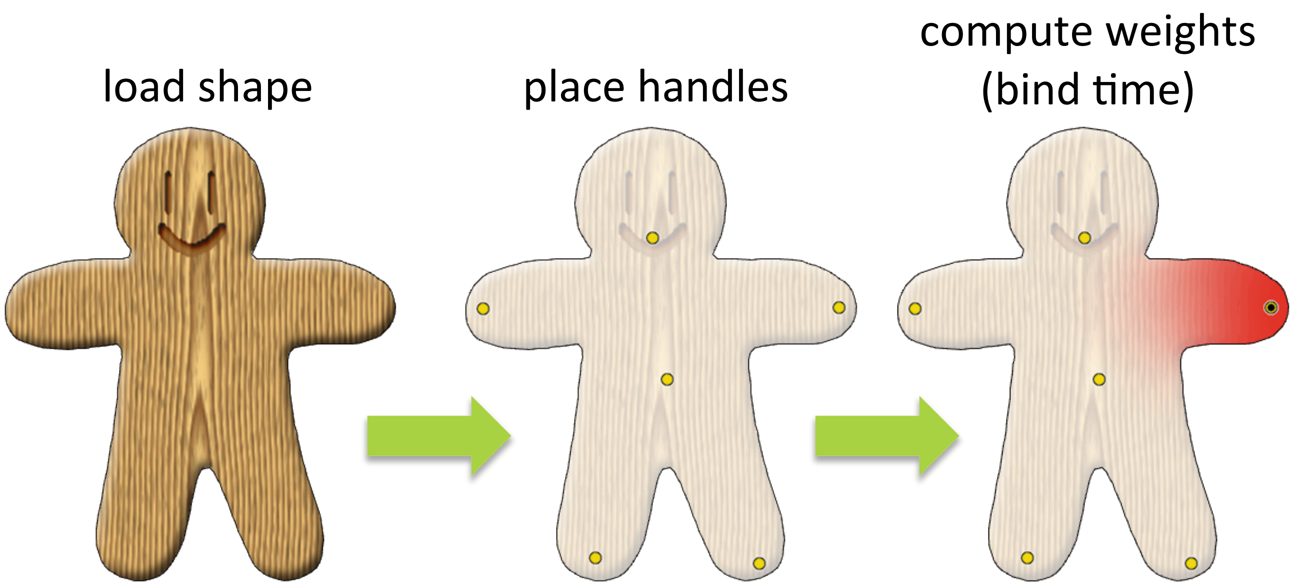
Linear Blend Skinning can't match quality but makes up in real-time performance

load shape

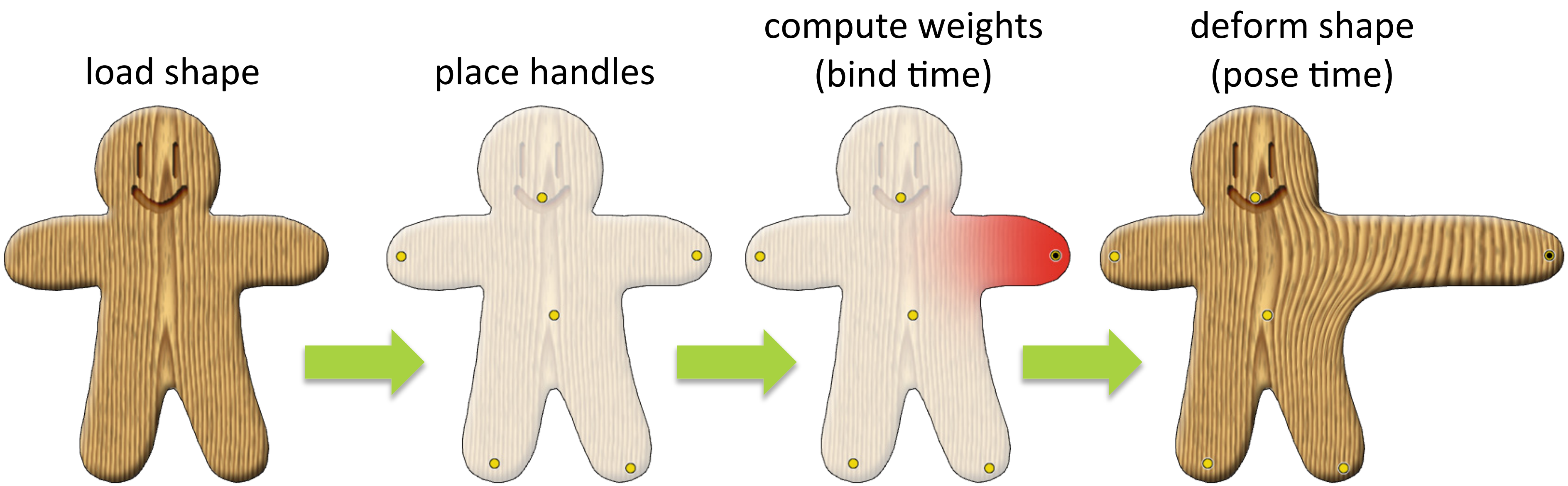
place handles



Linear Blend Skinning can't match quality but makes up in real-time performance

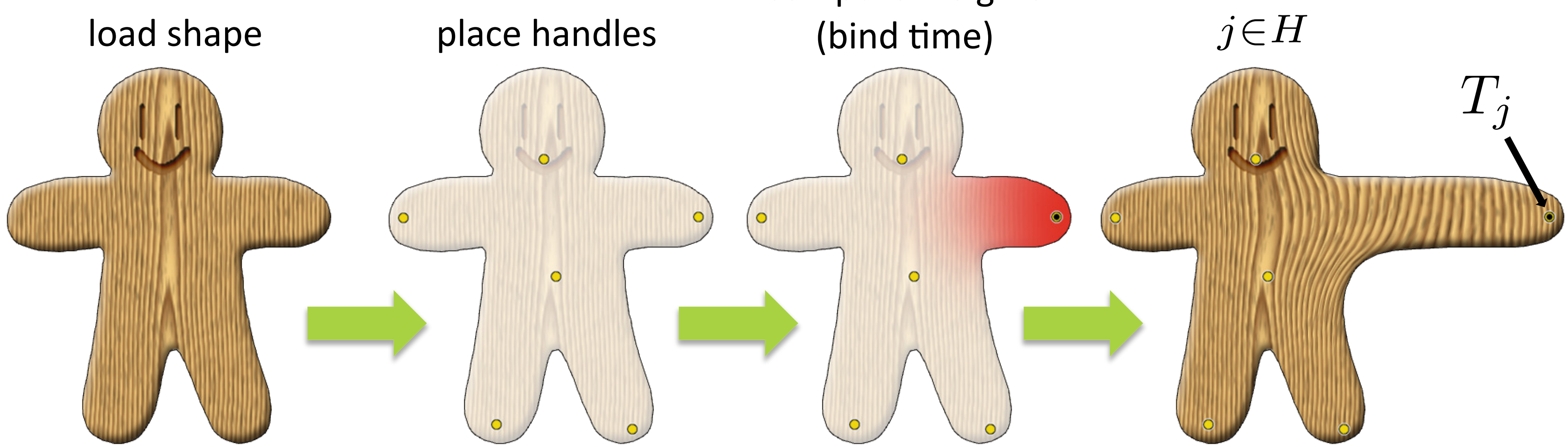


Linear Blend Skinning can’t match quality but makes up in real-time performance



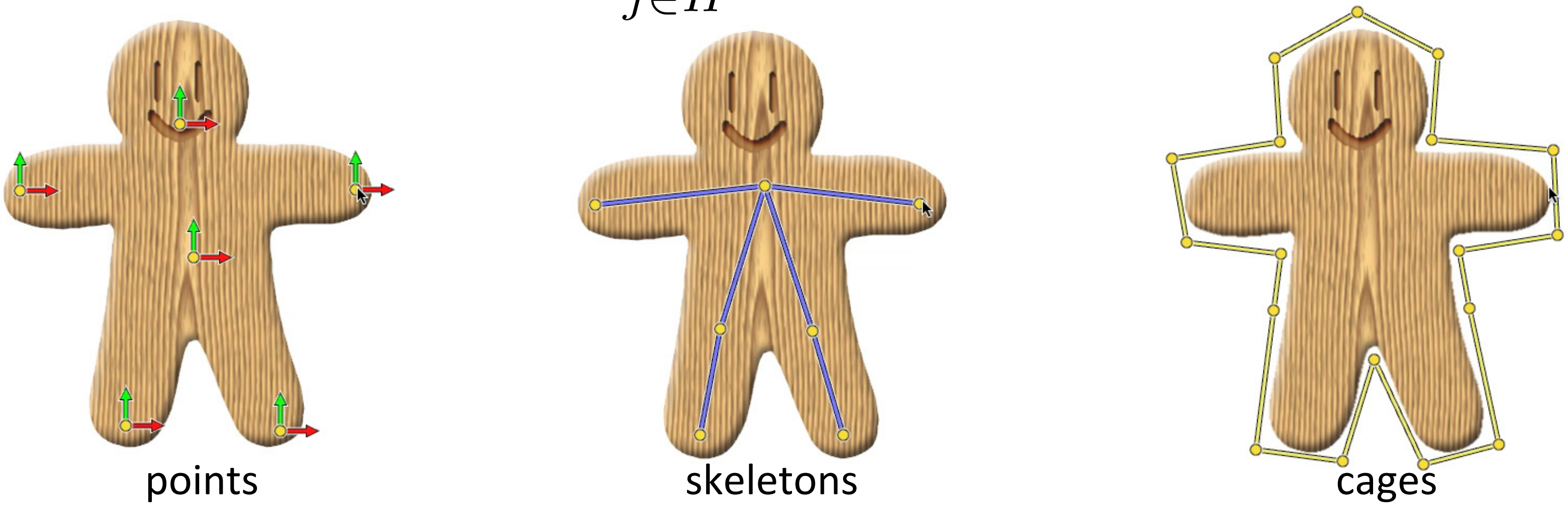
Linear Blend Skinning can't match quality but makes up in real-time performance

$$\mathbf{x}' = \sum_{j \in H} w_j(\mathbf{x}) T_j \mathbf{x}$$

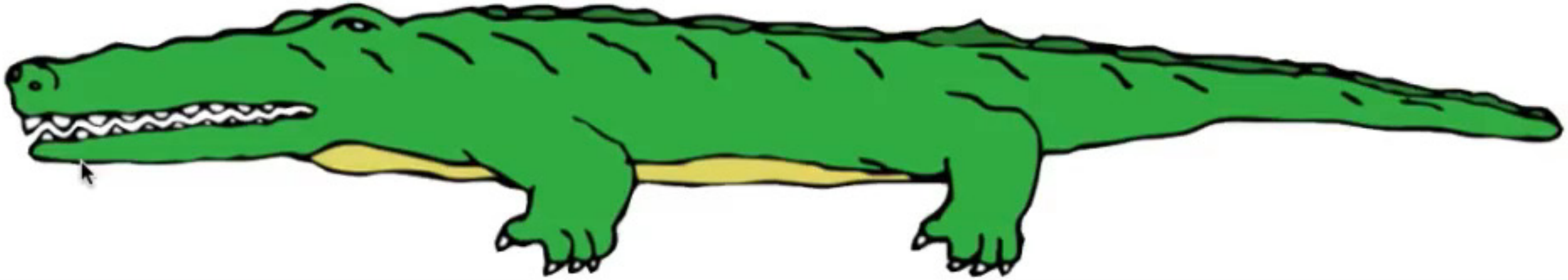


Warping and deformation tasks require different user interactions

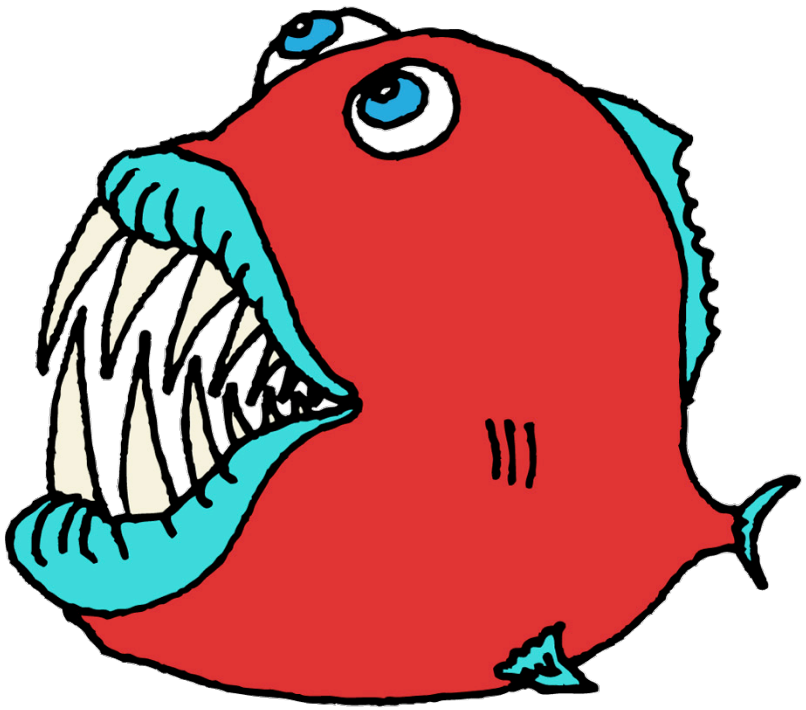
$$\mathbf{x}' = \sum_{j \in H} w_j(\mathbf{x}) T_j \mathbf{x}$$



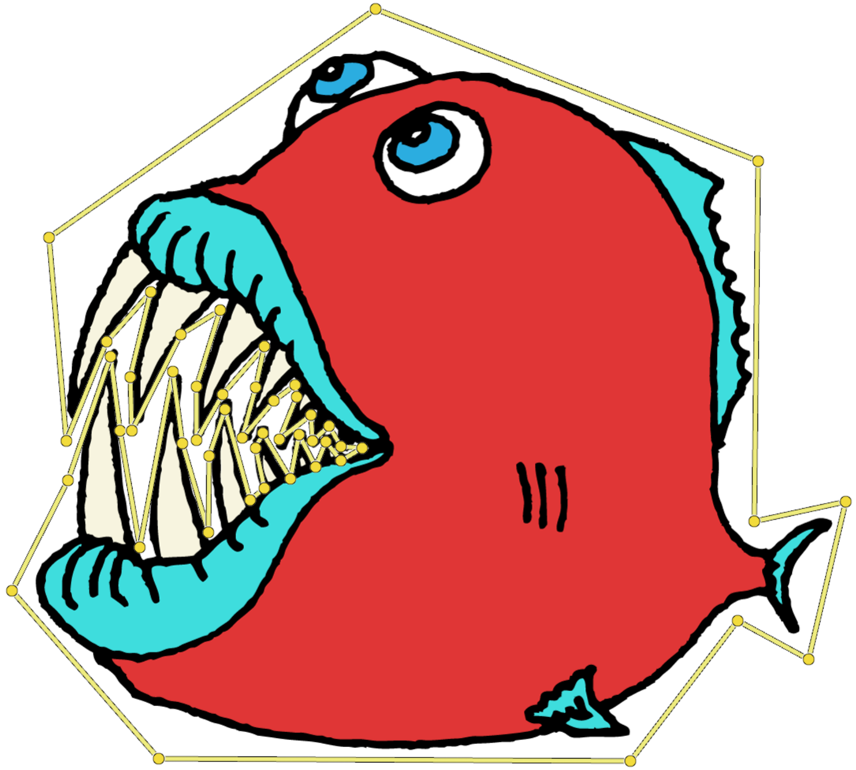
Each handle type has a specific task, more than just *different modeling metaphor*



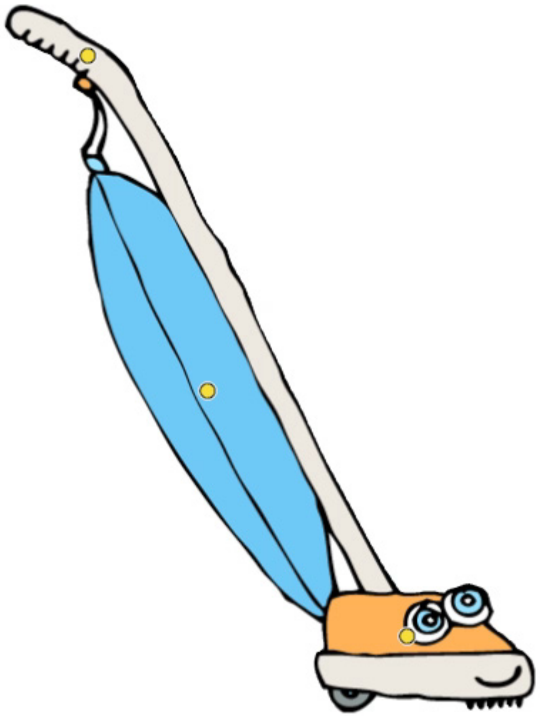
Cages can often be tedious to build and control

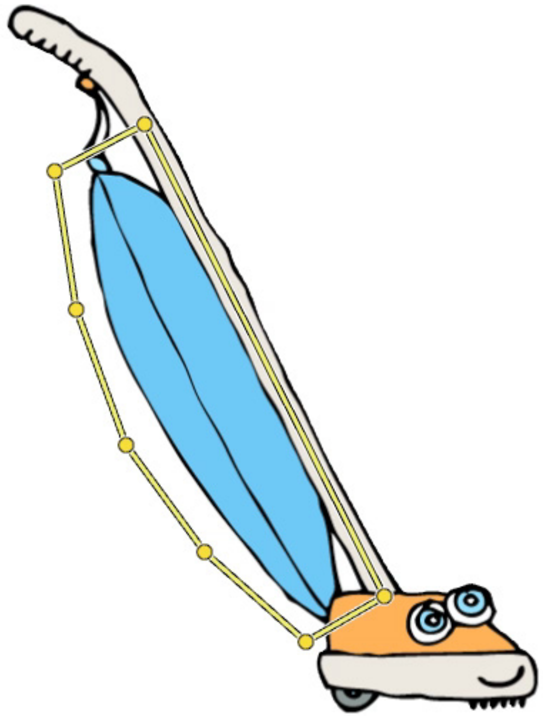


Cages can often be tedious to build and control

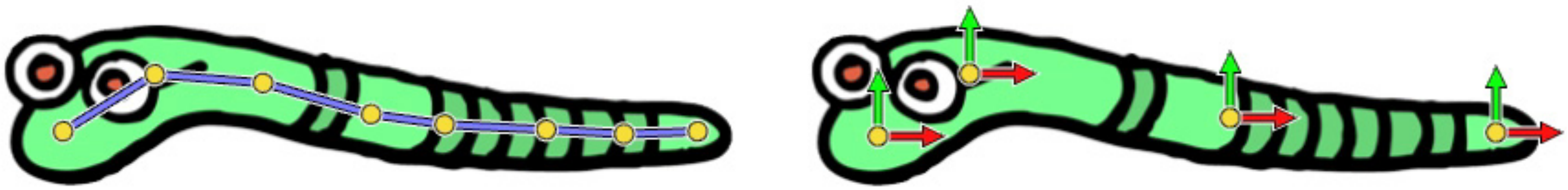


Points can only provide crude scaling

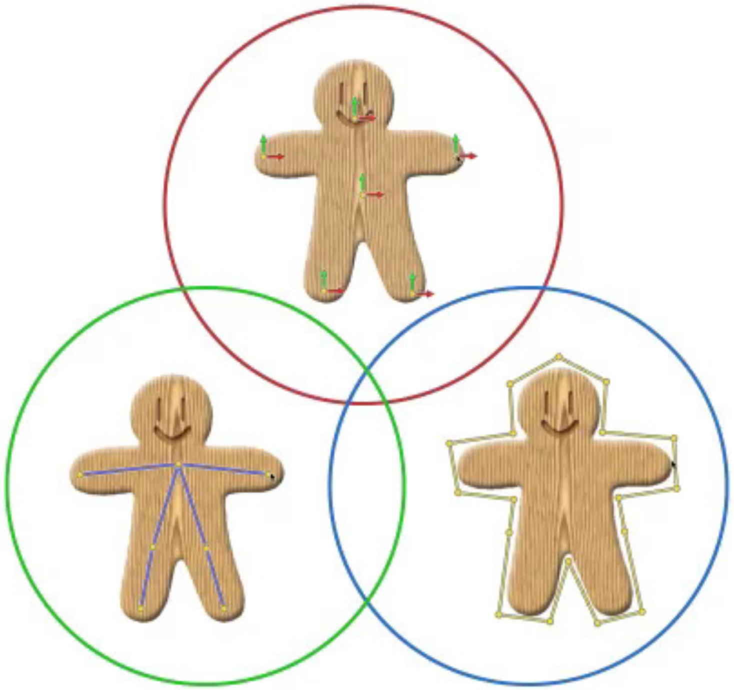




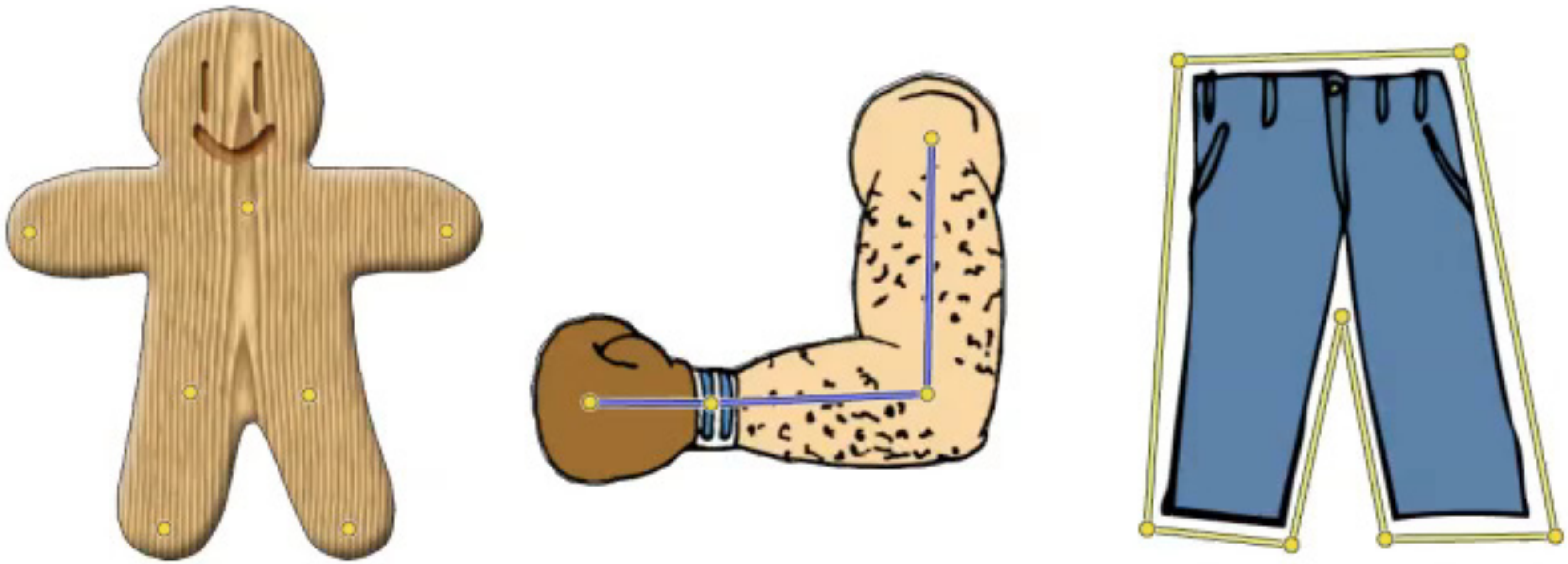
Skeletons may be too rigid or too cumbersome



We want to compute weights that unify points, skeletons and cages



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



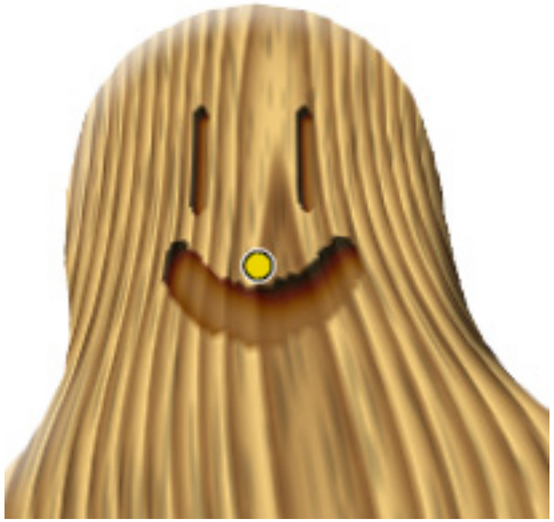
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Weights must be smooth everywhere, *especially* at handles

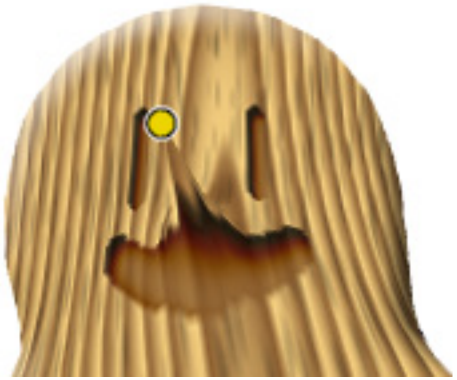
Our method

Extension of Harmonic Coordinates
[Joshi et al. 2005]

Weights must be smooth everywhere, *especially* at handles



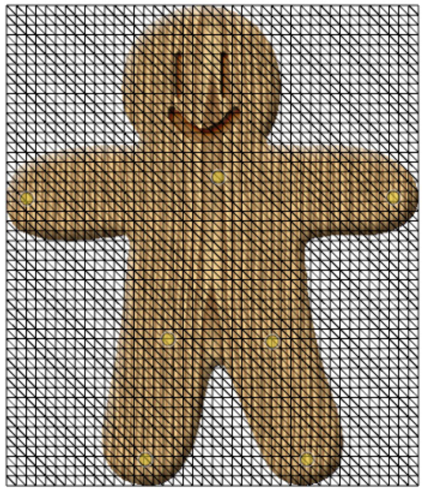
Our method



Extension of Harmonic Coordinates
[Joshi et al. 2005]

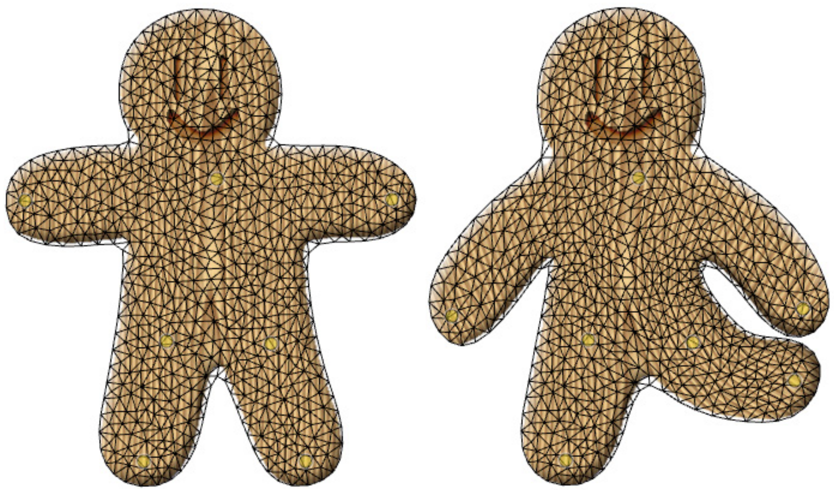
Shape-awareness ensures respect of domain's features

Our method

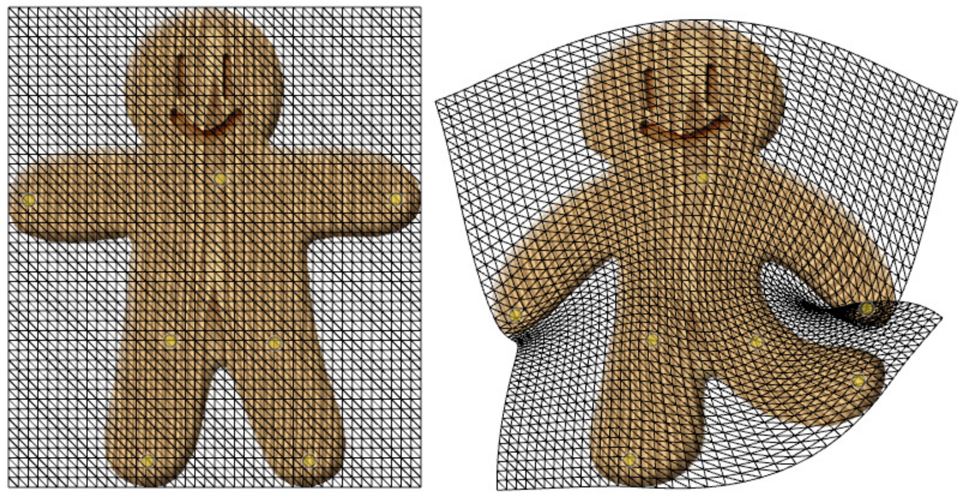


Non-shape-aware methods
e.g. [Schaefer et al. 2006]

Shape-awareness ensures respect of domain's features



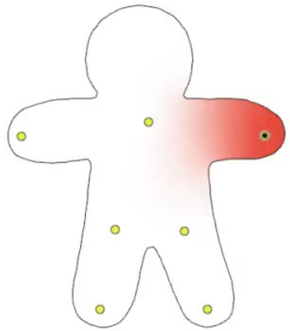
Our method



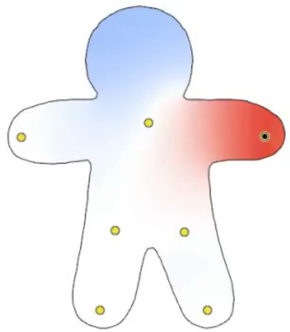
Non-shape-aware methods
e.g. [Schaefer et al. 2006]

Non-negative weights are necessary for intuitive response

Our method



Unconstrained biharmonic
[Botsch and Kobbelt 2004]



Weights must maintain other simple, but important properties

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

Partition of unity

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

w_j is linear along cage faces

Interpolation of handles

Weights must maintain other simple, but important properties

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

Partition of unity

Set of handles, aka mesh vertices under handles

Kronecker's delta

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

w_j is linear along cage faces

Interpolation of handles

Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

$$\Delta w_j = 0$$

Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

$$\Delta^2 w_j = 0$$

Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

Inverse distance,
weighted least-squares

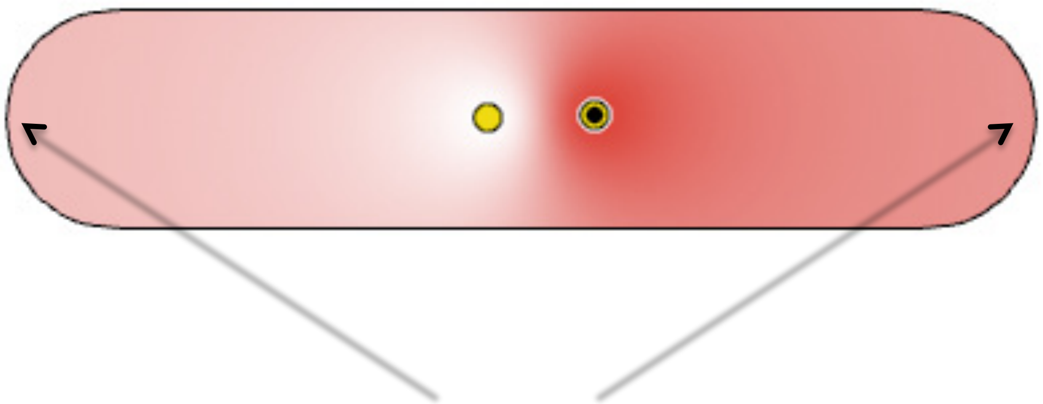
Inverse distance methods inherently suffer from *fall-off effect*

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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*



Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

How to support bones and cages?
How to make shape-aware?

Previous techniques only partially satisfy properties

	Harmonic Coordinates [Joshi et al. 2005]	Unconstrained biharmonic [Botsch and Kobbelt 2004]	Shepard interpolation [Shepard 1968]	Natural neighbors [Sibson 1981]
Smoothness	-	✓	✓	-
Non-negativity	✓	-	✓	✓
Shape-aware	✓	✓	-	-
Locality, sparsity	-	-	-	✓

$$\Delta^2 w_j = 0$$

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

$$0 \leq w_j(\mathbf{x}) \leq 1$$

Partition of unity

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

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

$$0 \leq w_j(\mathbf{x}) \leq 1$$

Solve independently and normalize

$$w_j(\mathbf{x}) = \frac{w_j(\mathbf{x})}{\sum_{i \in H_k} w_i(\mathbf{x})}$$

Weights optimized as precomputation at bind-time

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

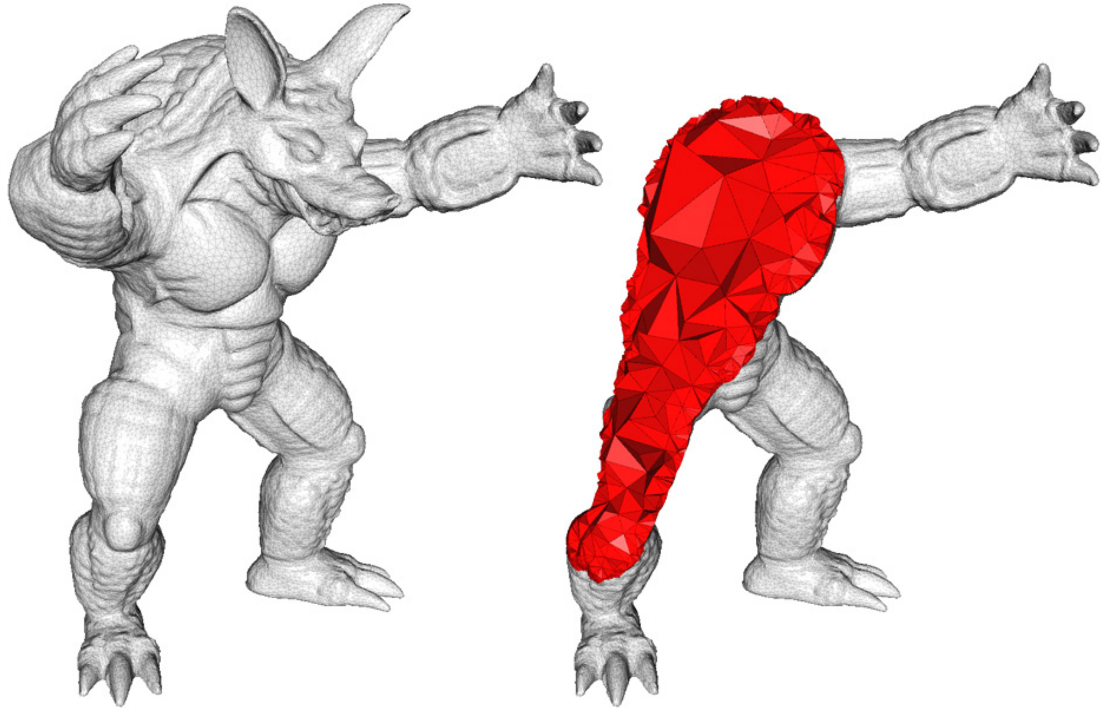
$$w_j\big|_{H_k} = \delta_{jk}$$

w_j is linear along cage faces

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

FEM discretization

- 2D → Triangle mesh
- 3D → Tet mesh

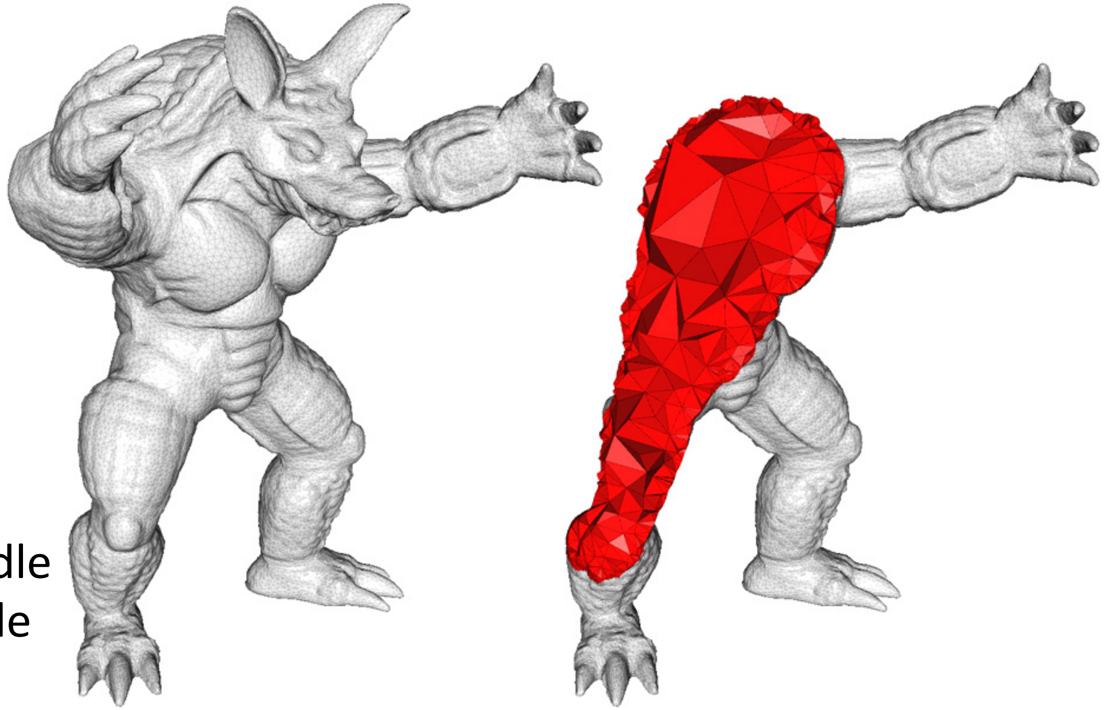


Weights optimized as precomputation at bind-time

$$\arg\min_{w_j} \frac{1}{2} \int_\Omega \|\Delta w_j\|^2 dV$$
$$w_j\big|_{H_k} = \delta_{jk}$$
w_j is linear along cage faces
$$0 \leq w_j(\mathbf{x}) \leq 1$$

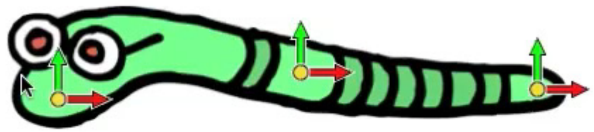
Sparse quadratic programming with constant inequality constraints

- 2D → less than second per handle
- 3D → tens of seconds per handle

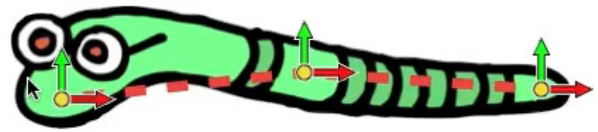


Variational formulation allows additional, problem-specific constraints

Rotations at point handles may be computed automatically based on translations

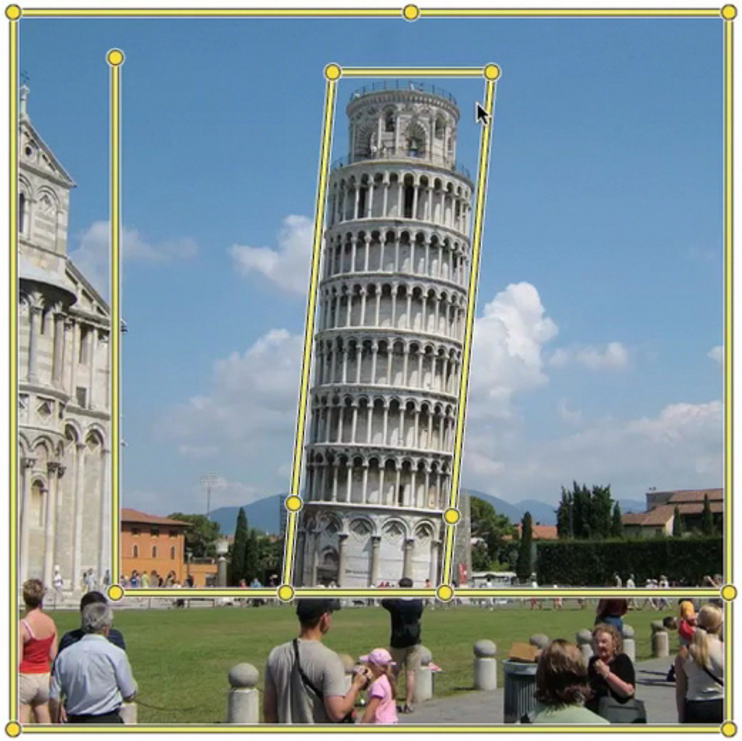


without pseudo-edges

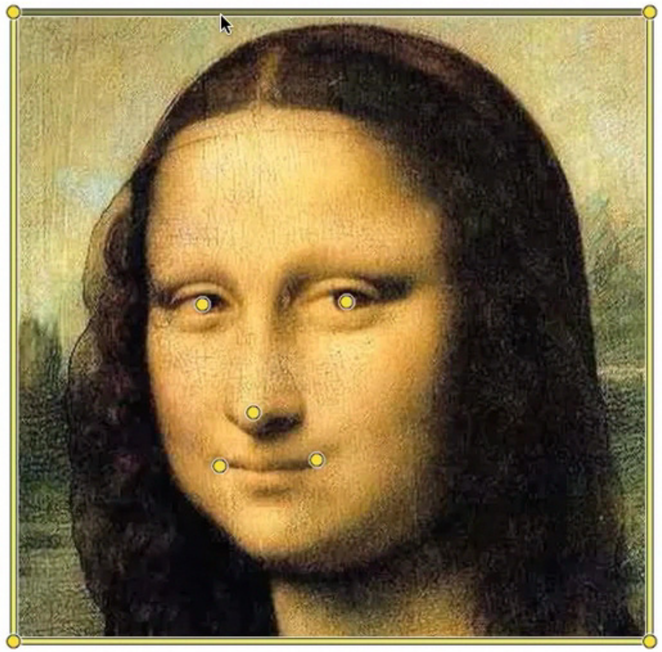


with pseudo-edges

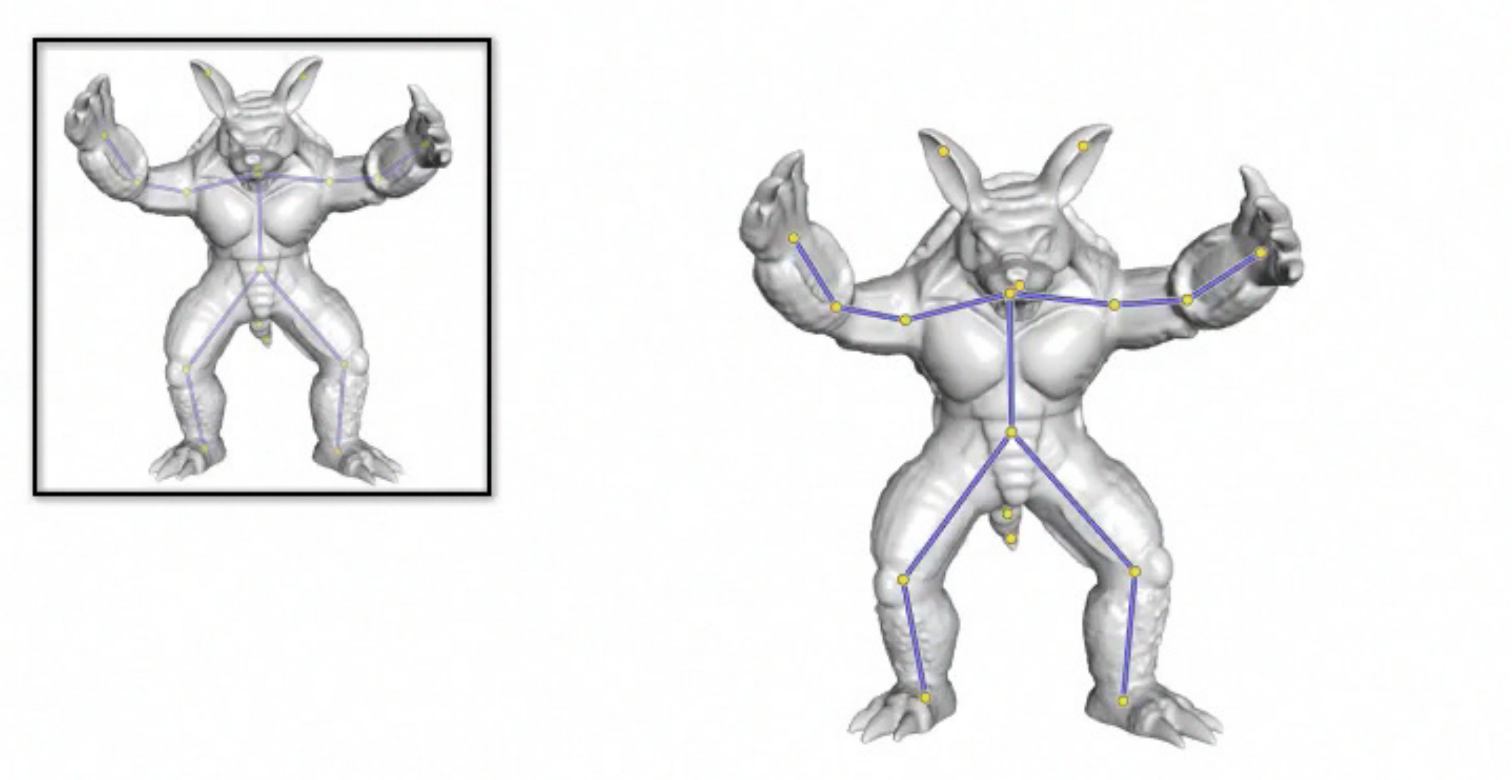
Open cages allow arbitrary line constraints



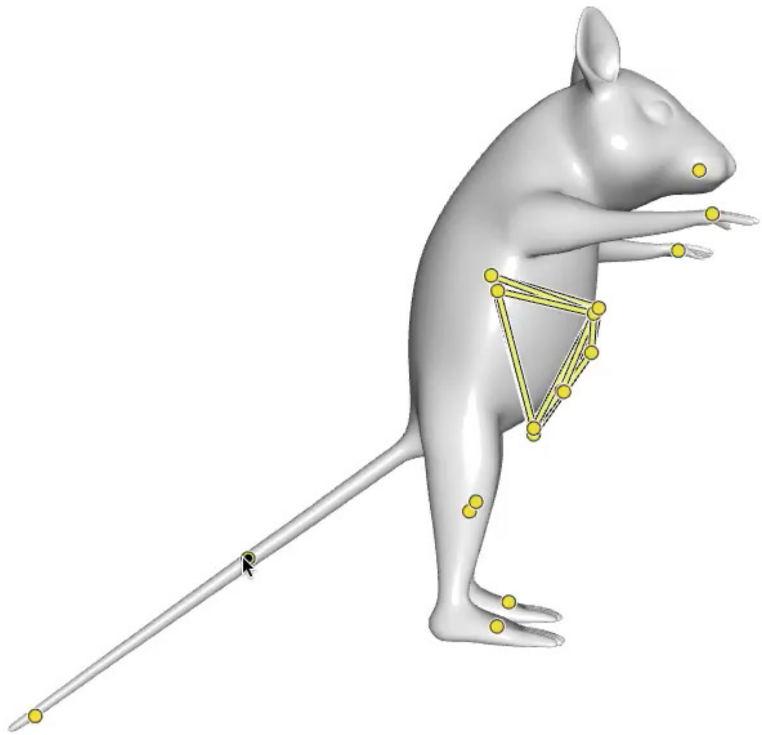
Some tasks are more easily accomplished by controlling both points and lines



Weights in 3D also retain nice properties, …

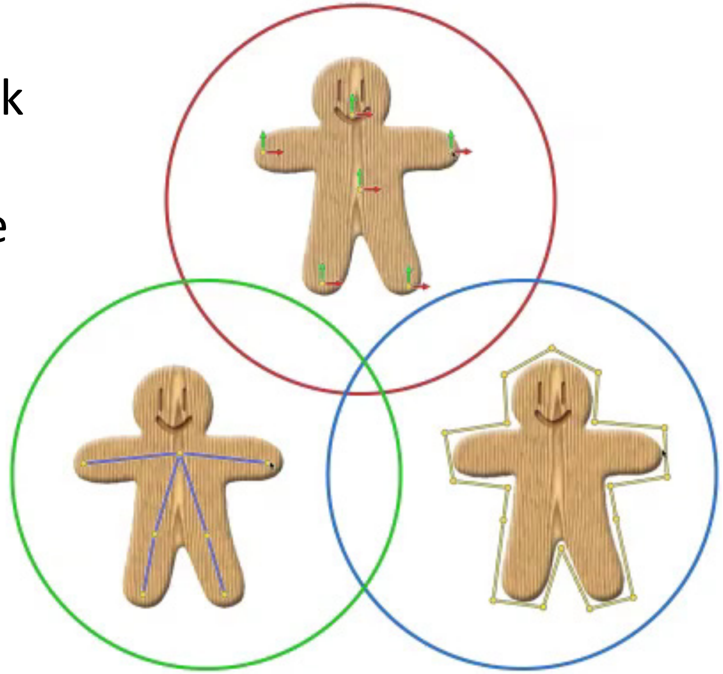


… and unify handle types



In summary, BBW unify popular handle types for intuitive, real-time deformations

- Point, skeleton and cage handles have applications dependent on setting and task
- Bounded biharmonic weights unify handle types
- BBW are:
  - smooth, even at handles
  - local, shape-aware and sparse
  - always between 0 and 1



Future work for bounded biharmonic weights

- Missing degrees of freedom point handles
- Sparsity heuristics, improve precomputation time and storage
- Other uses of bounded biharmonic functions

We gratefully acknowledge…

Jaakko Lehtinen, Bob Sumner and Denis Zorin for illuminating discussions

Annie Ytterberg for her narration of the accompanying video

Yang Song for help implementing the rigidity brush and 2D remeshing

This work was supported in part by an NSF award IIS-0905502 and by a gift from Adobe Systems.

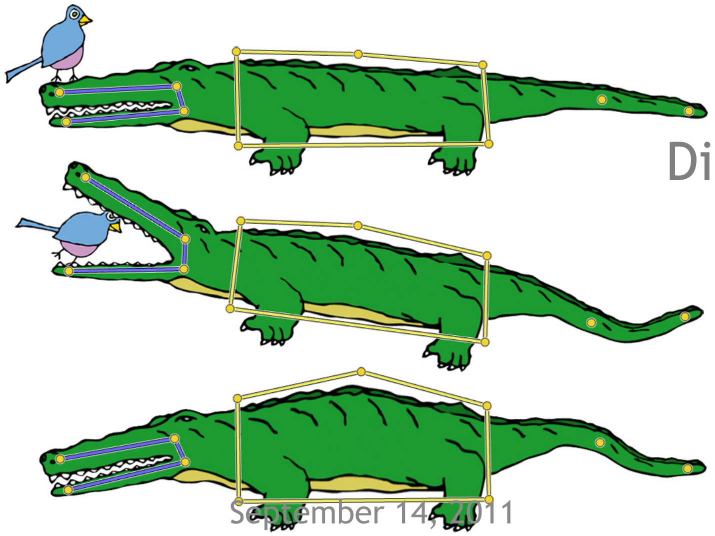
-

Bounded Biharmonic Weights for Real-Time Deformation

MATLAB code: http://igl.ethz.ch/projects/bbw/

Alec Jacobson
Ilya Baran
Jovan Popović
Olga Sorkine

NYU, ETH Zurich
Disney Research, Zurich
Adobe Systems, Inc.
NYU, ETH Zurich

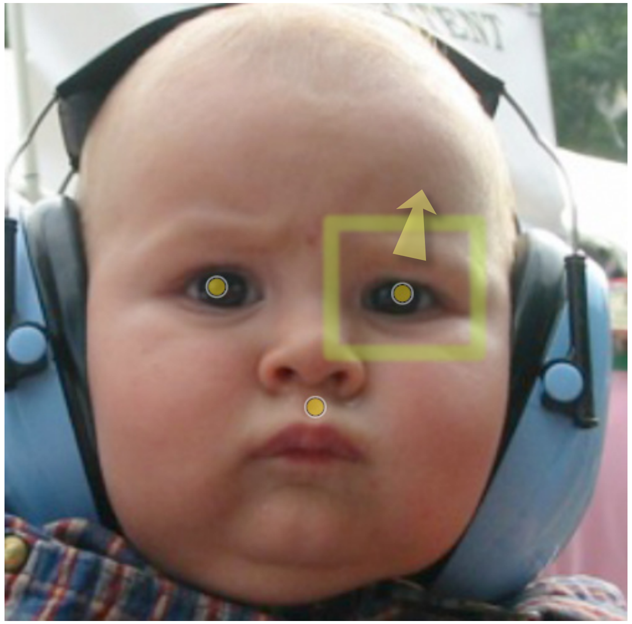


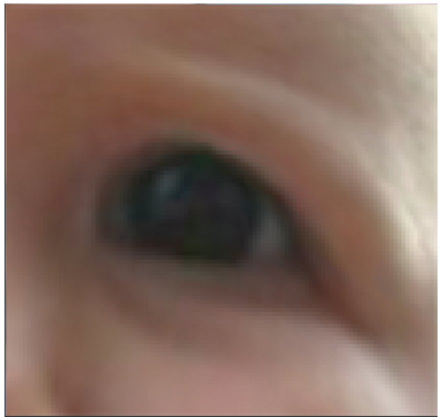
INTERACTIVE GEOMETRY LAB

September 14, 2011

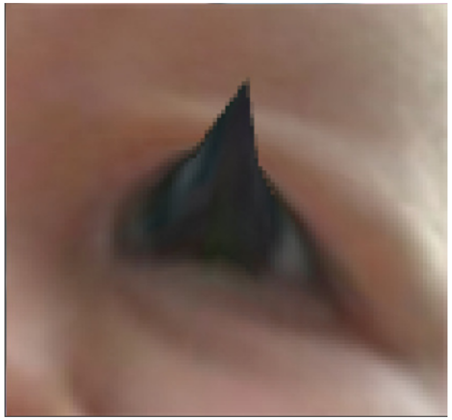
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Weights should be smooth everywhere, *especially* at handles





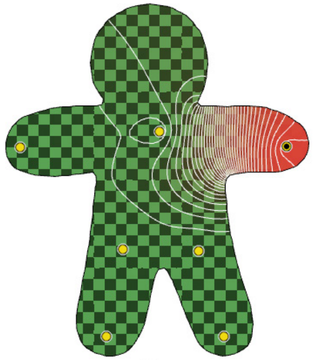
Our method



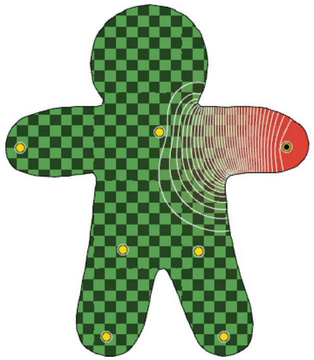
Extension of Harmonic Coordinates
[Joshi et al. 2005]

Sparsity helps maintain local influence

Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]

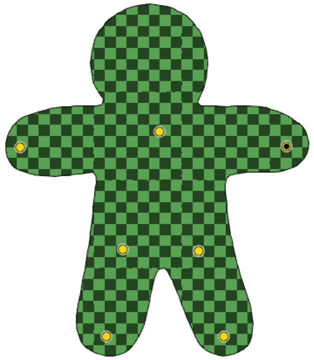


Our method

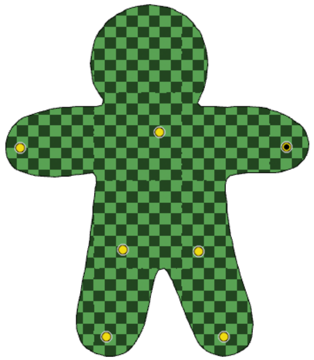


Sparsity helps maintain local influence

Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]

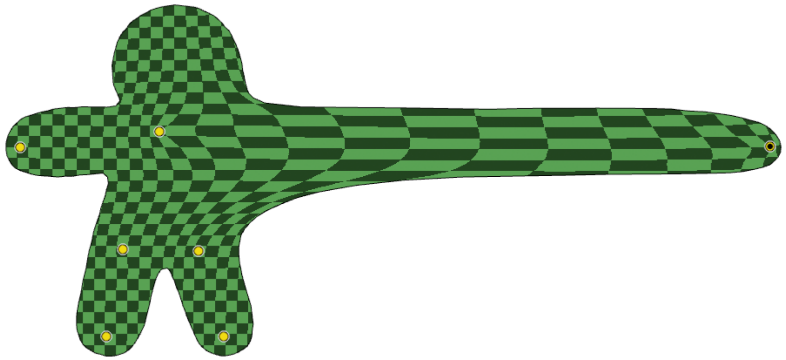


Our method

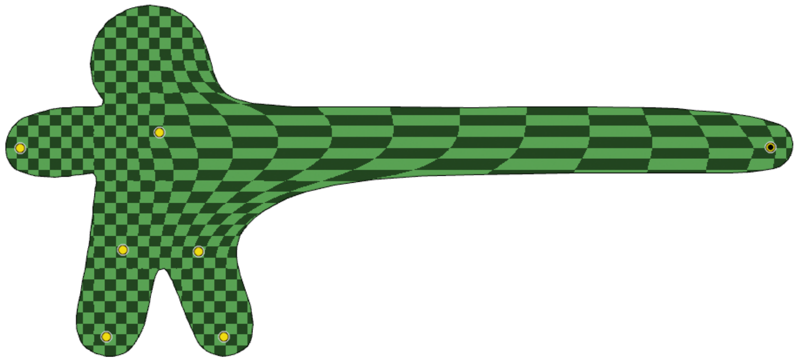


Sparsity helps maintain local influence

Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]

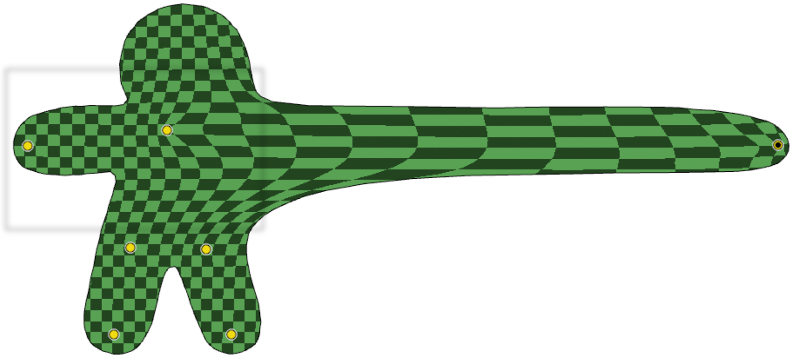


Our method

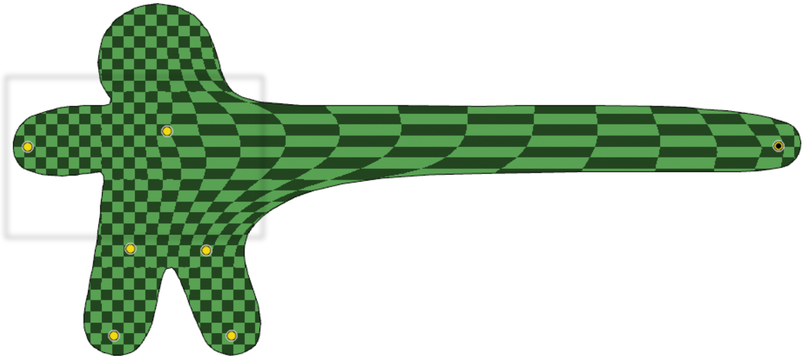


Sparsity helps maintain local influence

Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]

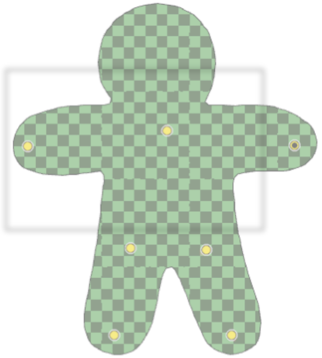


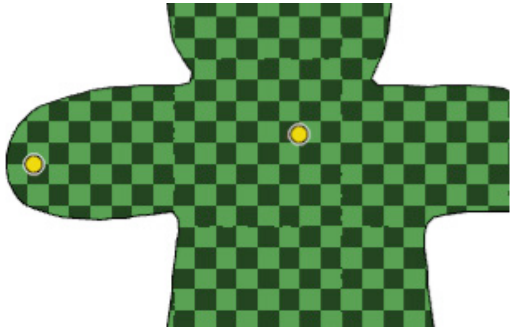
Our method



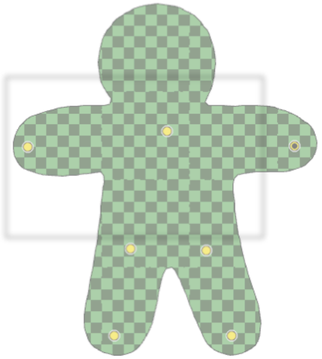
Sparsity helps maintain local influence

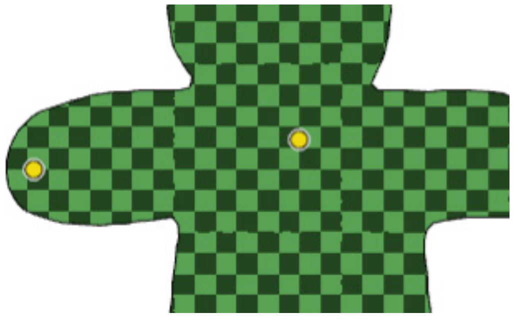
Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]





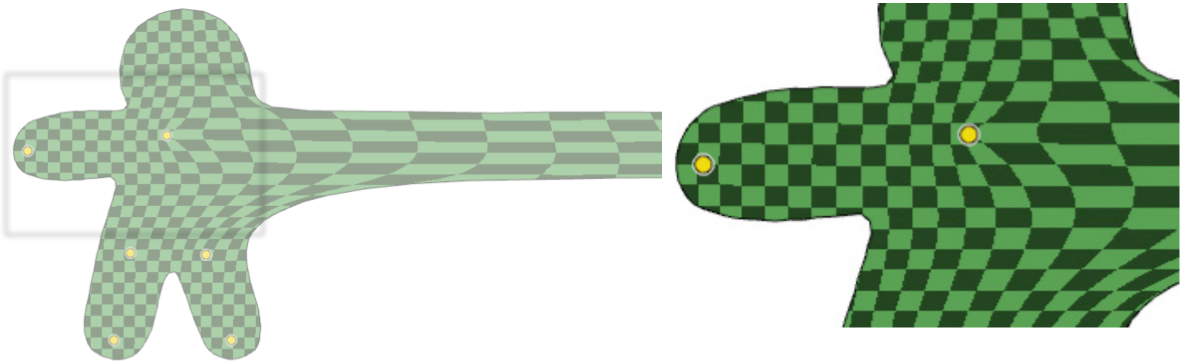
Our method



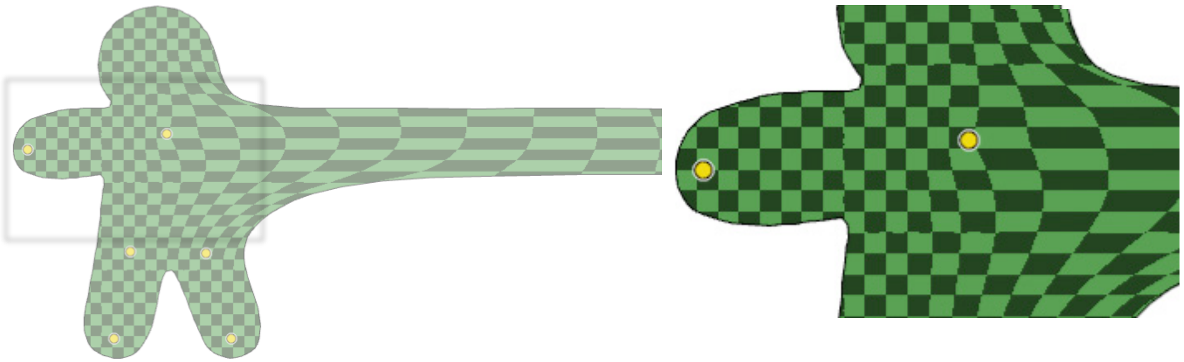


Sparsity helps maintain local influence

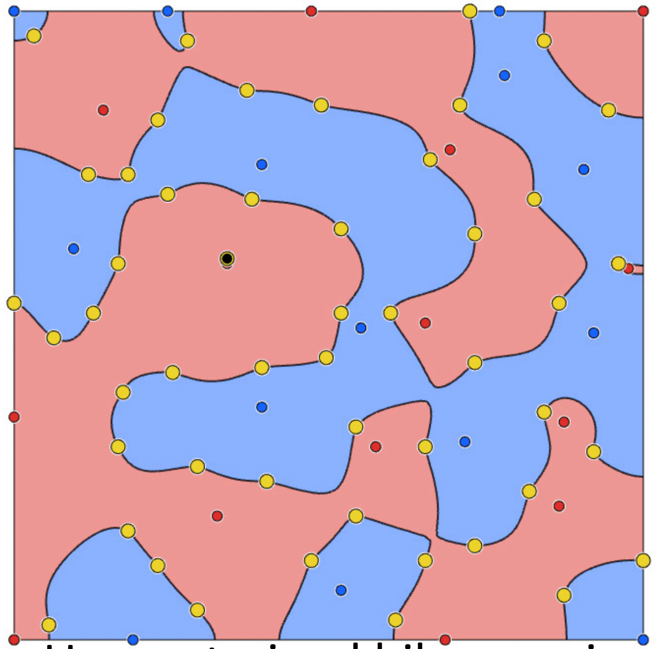
Smoothed extension of Harmonic Coordinates [Joshi et al. 2005]



Our method

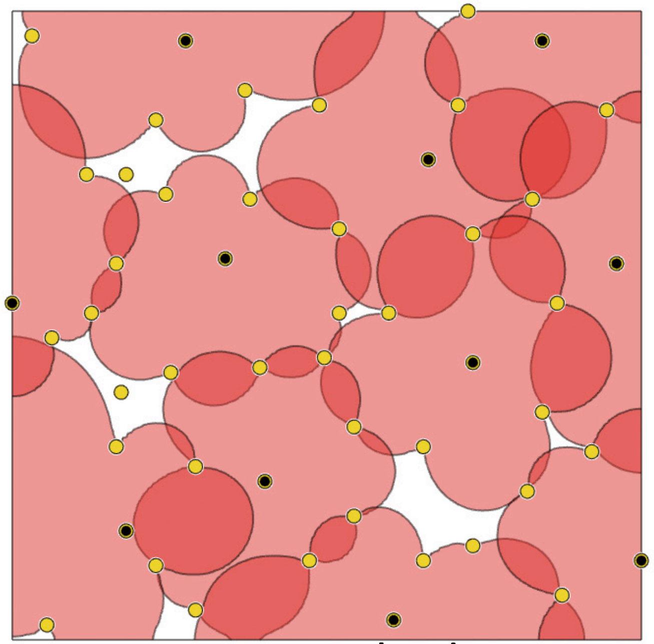


Boundedness also helps maintain local influence

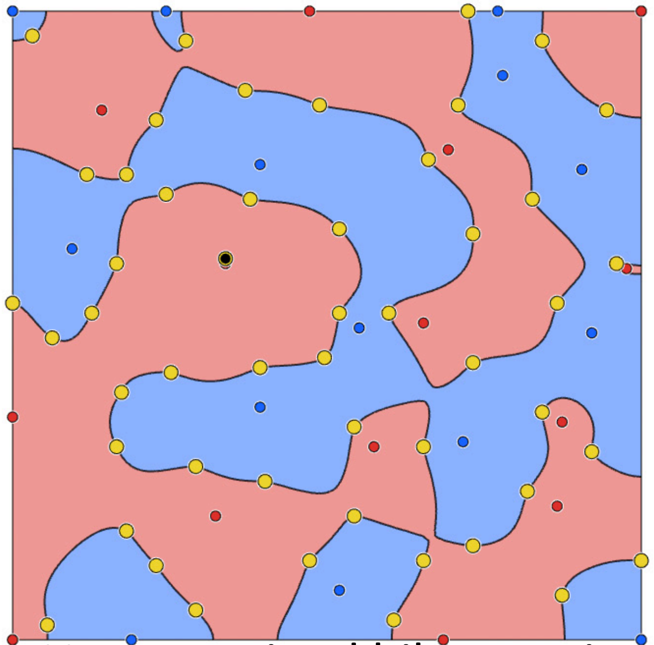


Unconstrained biharmonic
[Botsch and Kobbelt 2004]

Boundedness also helps maintain local influence



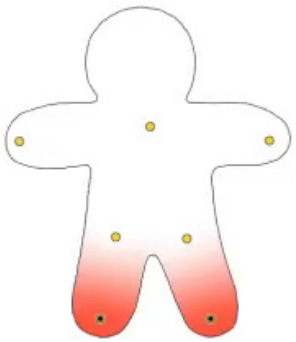
Our method



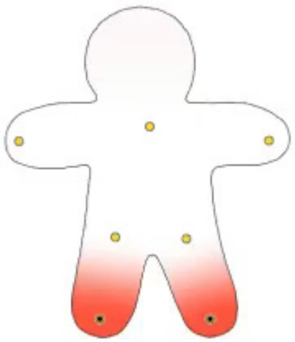
Unconstrained biharmonic

[Botsch and Kobbelt 2004]

Spurious local maxima also cause unintuitive response



Our method

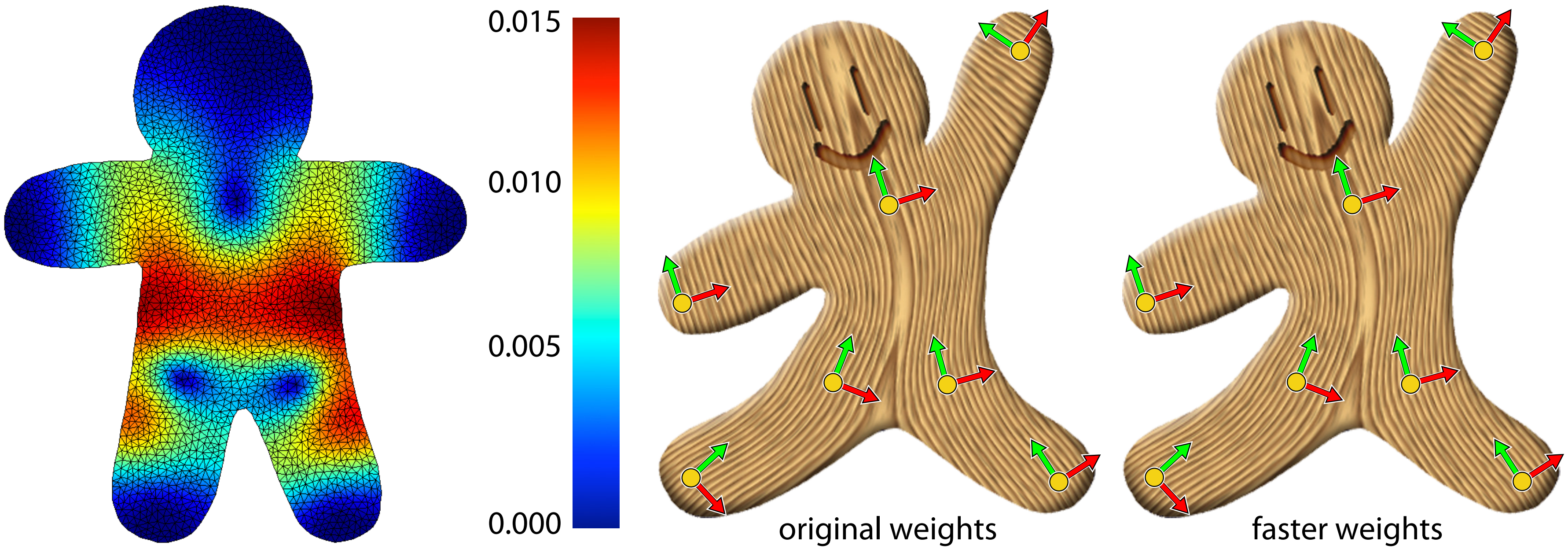


Extension of unconstrained biharmonic
[Botsch and Kobbelt 2004]

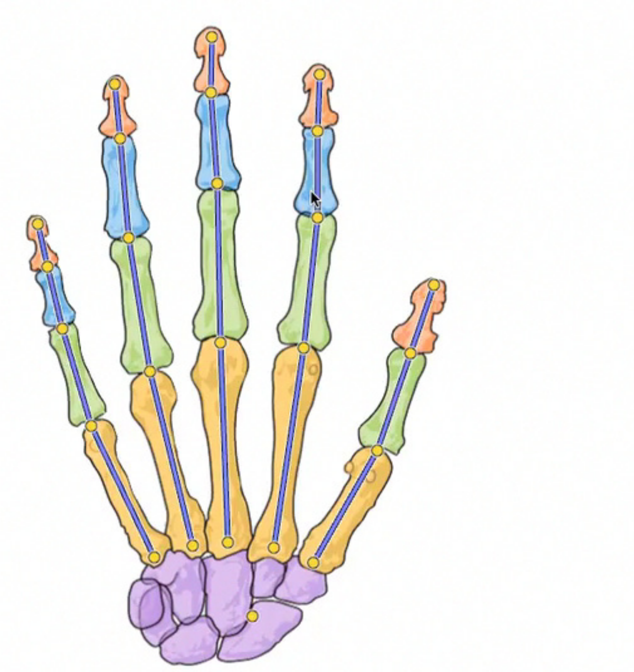
Weights propagate transformations at handles to shape in real-time

Translate Rotate Scale

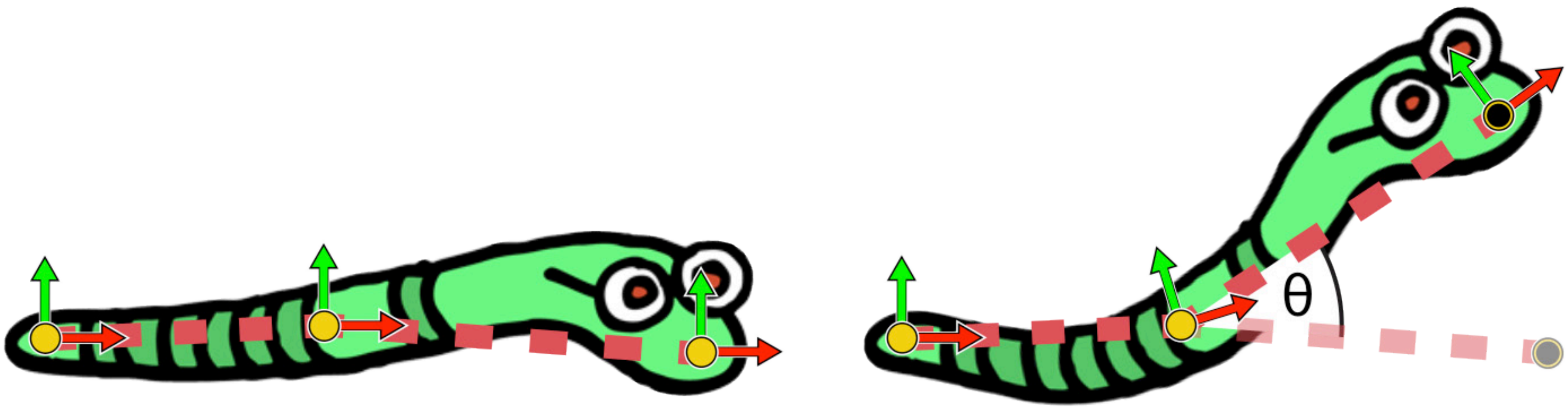
Dropping partition of unity as explicit constraint does not effect quality



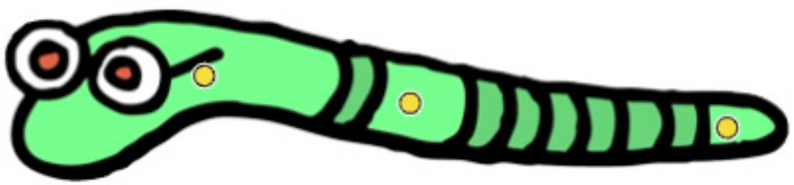
Weights may also define an intuitive, shape-aware depth ordering in 2D



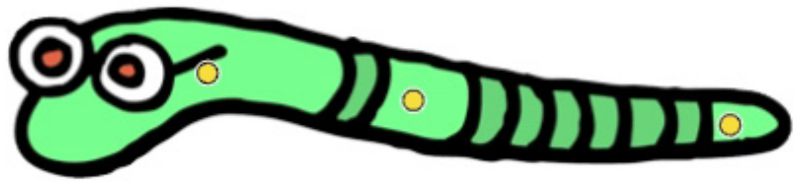
Rotations at point handles may be computed automatically based on translations



Alternative skinning methods may also take advantage of bounded biharmonic weights



Linear blend skinning



Dual quaternion skinning