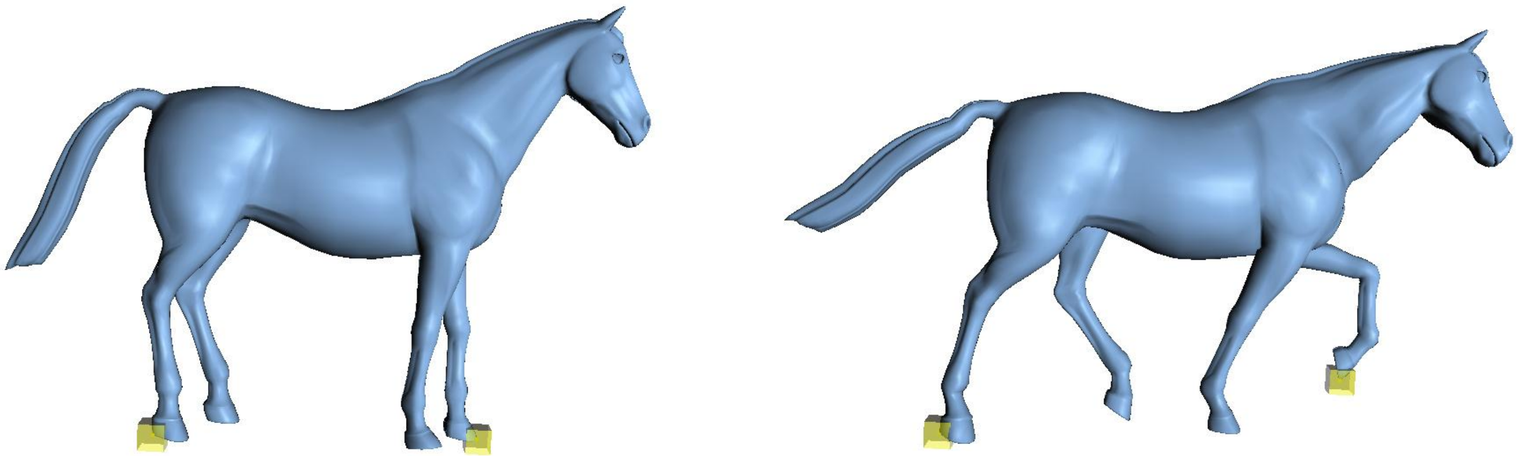
Inverse Kinematics for Reduced Deformable Models

Kevin G. Der   Robert W. Sumner[1]   Jovan Popović

Computer Science and Artificial Intelligence Laboratory   [1]ETH Zürich
Massachusetts Institute of Technology

Our Goal: Interaction

- mesh manipulation
- direct, intuitive control
- speed



Der, Sumner, and Popović

Related Work

- mesh editing

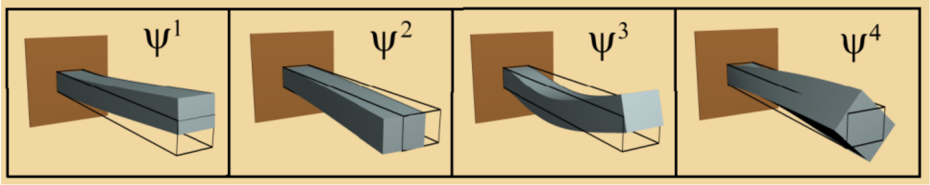
  [Zorin et al. 1997, Kobbelt et al. 1998, Sorkine 2005]

- inverse kinematics

  [Sumner et al. 2005, Grochow et al. 2003]

- reduced deformable models

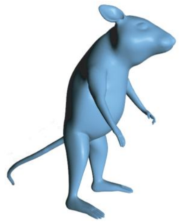
  [Pentland and Williams 1989, Alexa and Muller 2000, James and Twigg 2005]



[James and Barbic 2005]

Our Method

Mesh

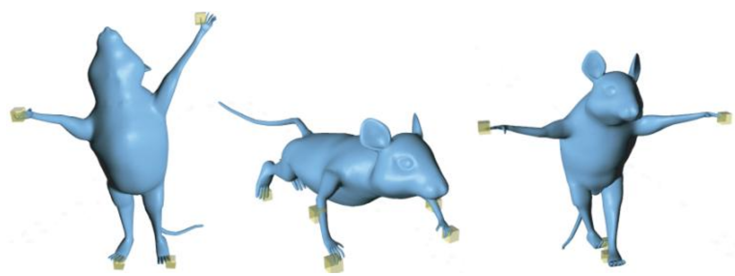


Examples

Reduced Deformable Model

decouple deformation complexity
from geometric complexity!

Inverse Kinematics

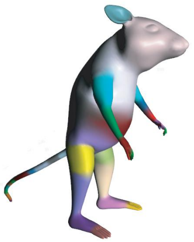


- completely automated animation pipeline

Reduced Deformable Model

Examples

Reduced Deformable Model



[James and Twigg 2005]

- expresses deformations compactly
- automatically constructed

Control parameters: $\{ \mathbf{F}_i , \mathbf{d}_i \}$

- non-rigid
- no hierarchy

RDM: Mesh Reconstruction

Control parameters: $\{ F_i, d_i \}$

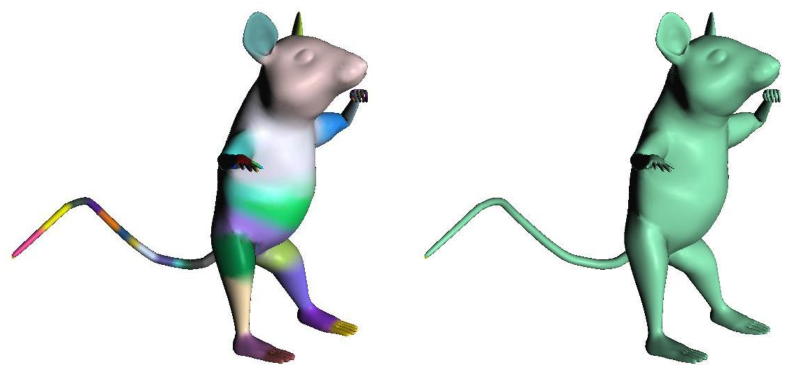
Skinning weights: $\{ \alpha_i(x) \}$

different values estimate each example

Skinning equation.

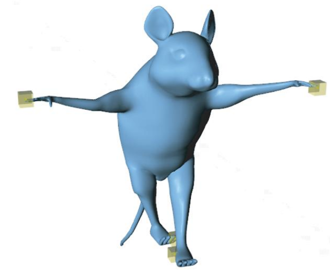
$$\mathbf{v} = \sum_i \alpha_i(\mathbf{x}) \cdot (F_i \mathbf{x} + d_i)$$

huge number *20 to 50*



Approx Exact

Inverse Kinematics

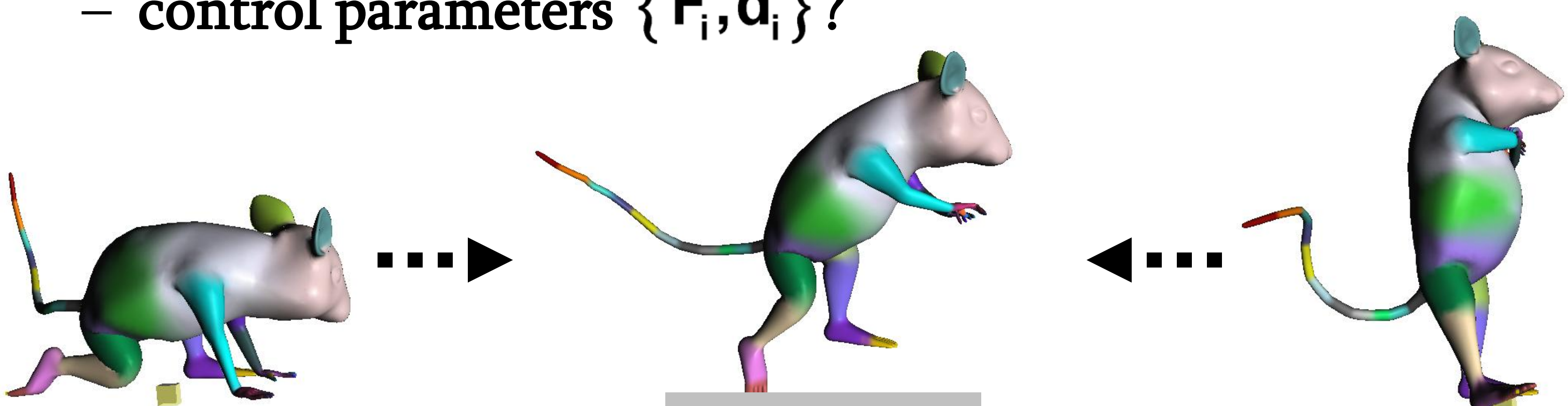


- find a semantically proper shape
- obtain natural poses by blending examples

- blend in which domain?
  - vertex positions $\{\mathbf{v}_i\}$?
  - control parameters $\{\mathbf{F}_i, \mathbf{d}_i\}$?

Blending function.

$\mathbf{m}(\beta, \mathbf{f}_1, \mathbf{f}_2, \ldots)$

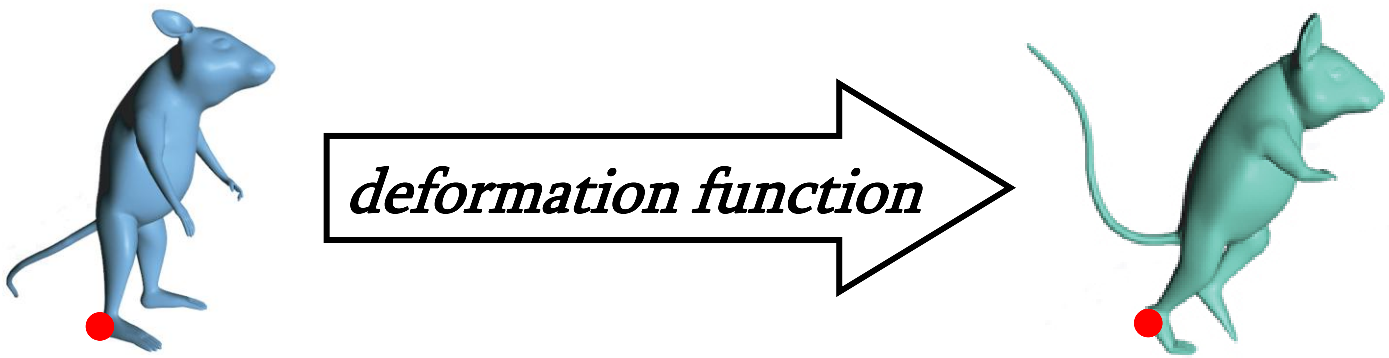


Desired

Der, Sumner, and Popović

Deformation Gradients

- simple 3x3 transformation matrices
- describe deformation of each example



Skinning equation.

$$\mathbf{v} = \sum_i \alpha_i(\mathbf{x}) \cdot (\mathbf{F}_i \mathbf{x} + \mathbf{d}_i)$$

Vertex deformation gradient.

$$\mathbf{D}_x \mathbf{v} = \sum_i (\mathbf{F}_i \cdot \mathbf{G}_1 + \mathbf{d}_1 \cdot \mathbf{G}_2)$$

Der, Sumner, and Popović

Deformation Gradients

$$D_x v = \sum_i \left(F_i \cdot G_1 + d_1 \cdot G_2 \right)$$

$$f = G \; t$$

flattened control matrices

constant matrix

flattened deformation gradients of vertices

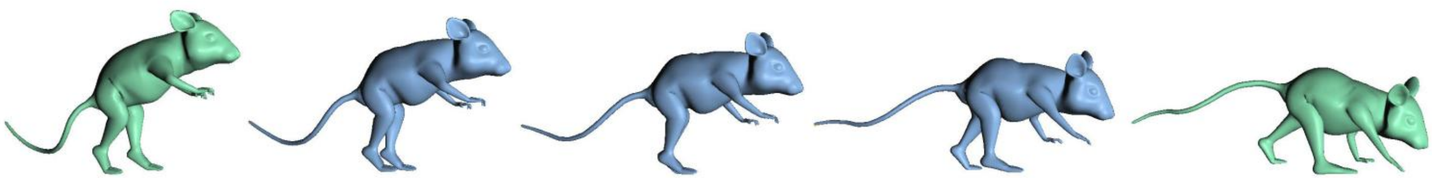
for each example: have t $\rightarrow$ get f

Shape Blending

- combine example deformation gradients

Blending function.

$$m(\beta, f_1, f_2, \ldots)$$



- recover the control parameters given deformation gradients

$$t^* = \arg\min_t \left\| Gt - m(\beta) \right\|^2$$

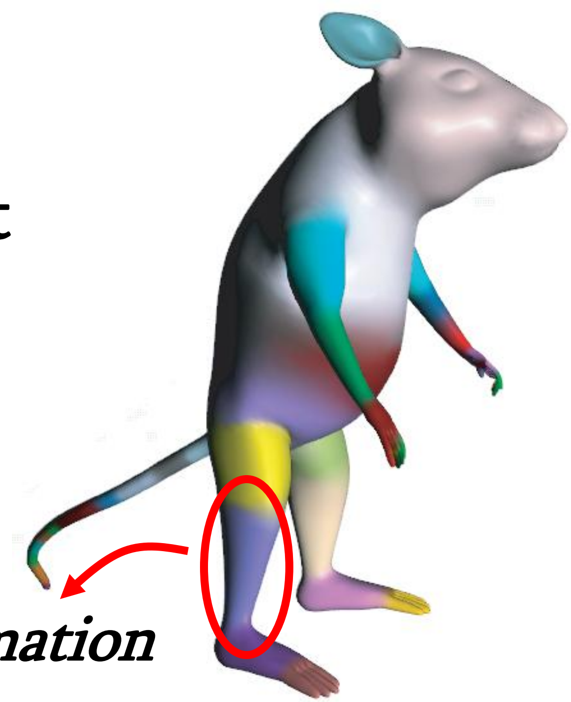
Has closed form solution.

solving for reduced basis!
but still slow

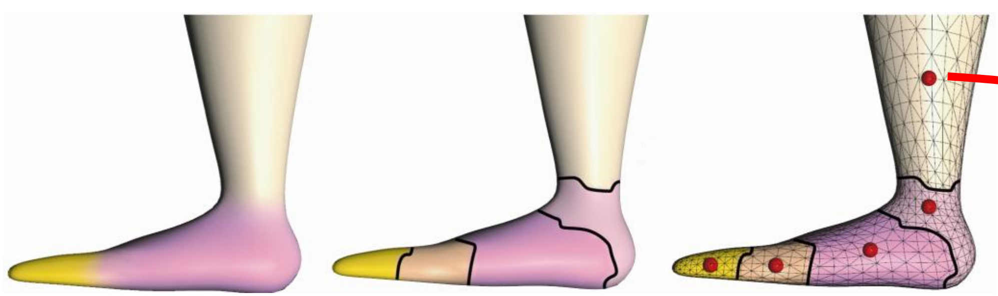
Der, Sumner, and Popović

Proxy Vertices

- evaluating deformation gradients at every vertex is undesirable
- summarize using a few vertices



same deformation gradient!

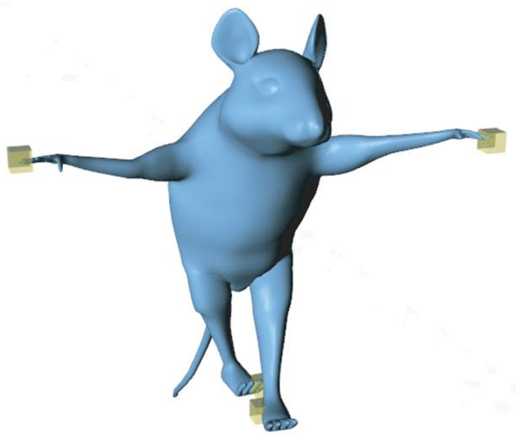


controls *vertex groups*

weighted centroid of the group's vertices

Der, Sumner, and Popović

Inverse Kinematics



$$t^*, \beta^* = \arg\min_{t,\beta} \left\| Gt - m(\beta) \right\|^2$$

$$+ \text{constraints}$$

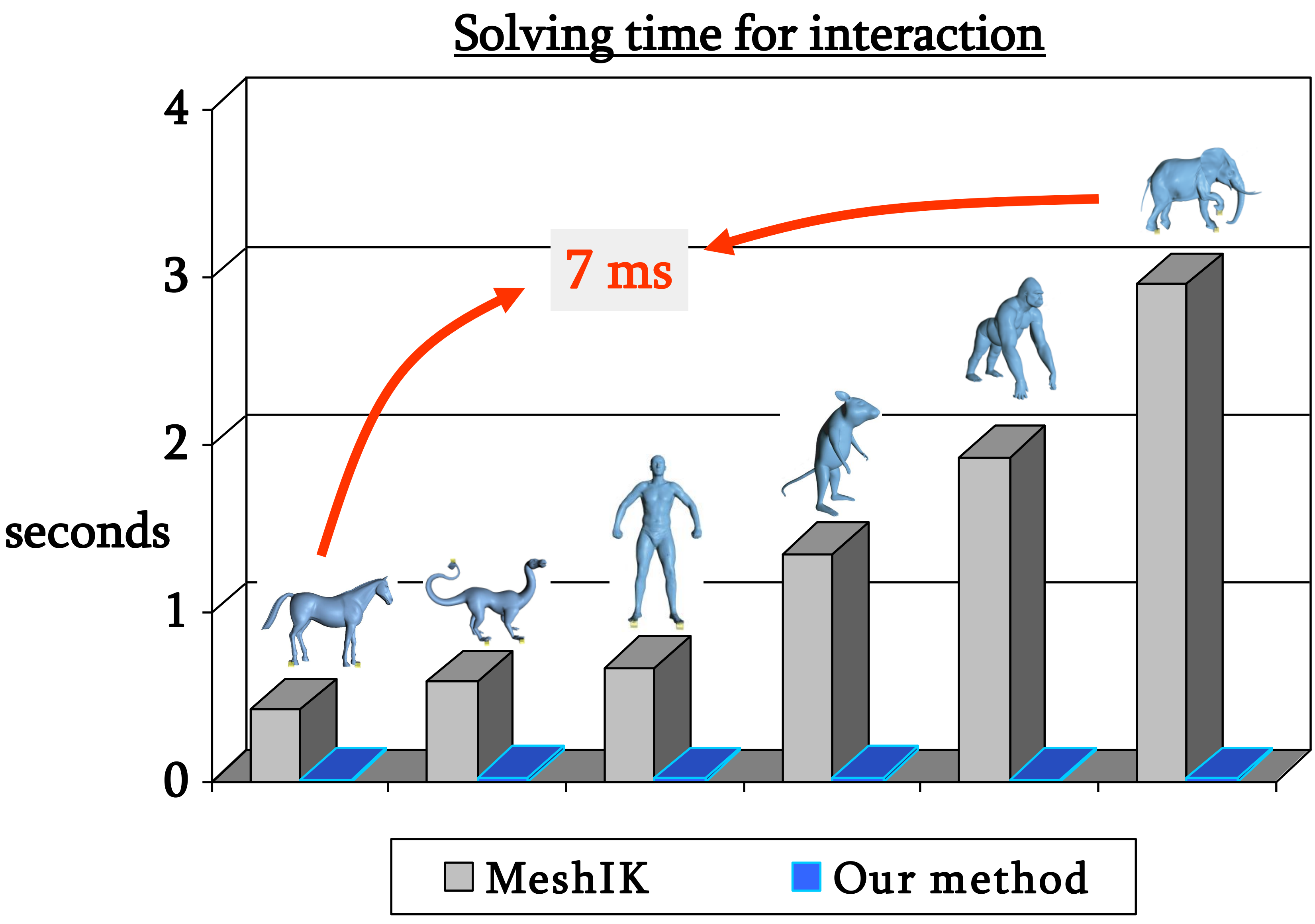
"Find best natural pose that meets user constraints"

Solving the constrained, nonlinear optimization:

- eliminate constraints
- linearize and iterate
- exploit constants

Der, Sumner, and Popović

Results

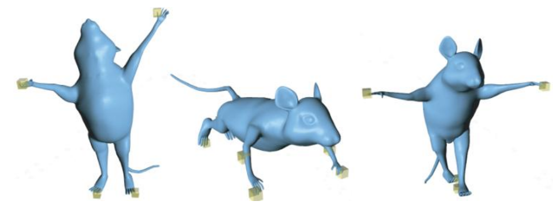


Der, Sumner, and Popović

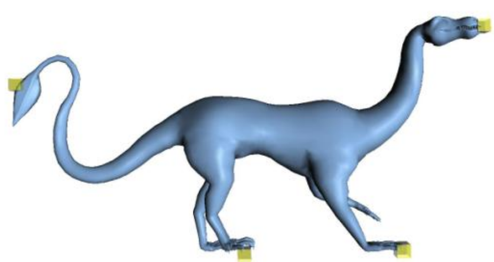
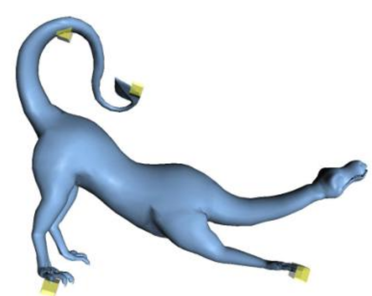
Conclusion

- interactive control of reduced deformable models





- future work
  - error correction for new poses
  - model transfer between meshes
  - other reduced deformable models

Acknowledgements

Dragomir Anguelov
Rutgers Image Understanding Laboratory

John Reed UROP Fund
Pixar Animation Studios

MIT Computer Graphics Group