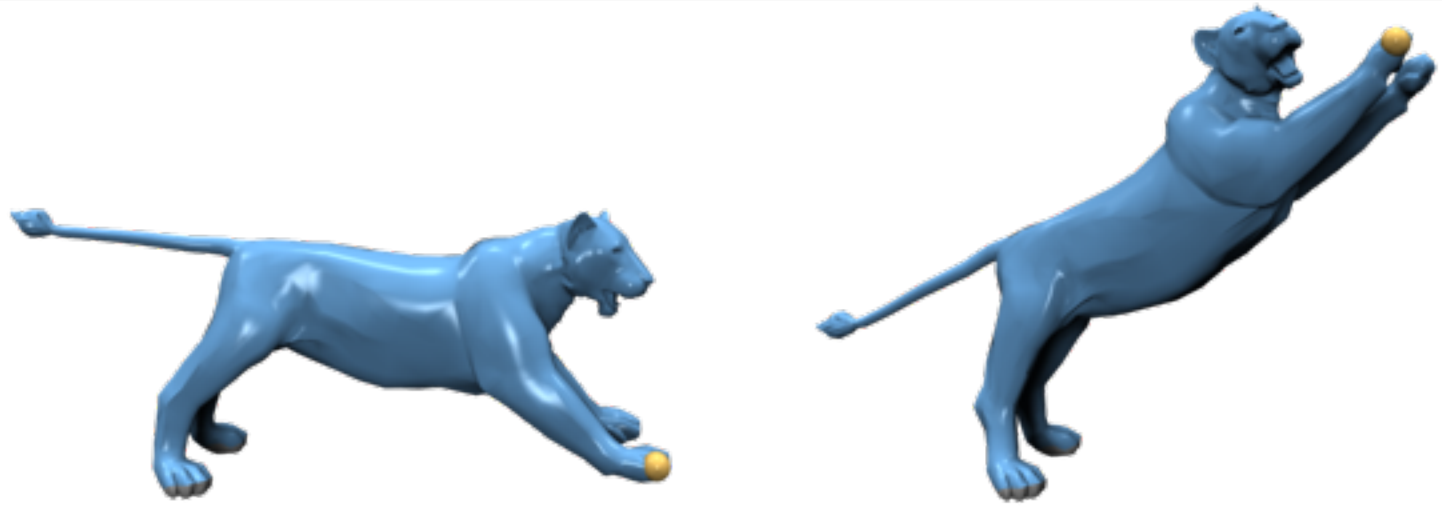
Example-Driven Deformation Based on Discrete Shells



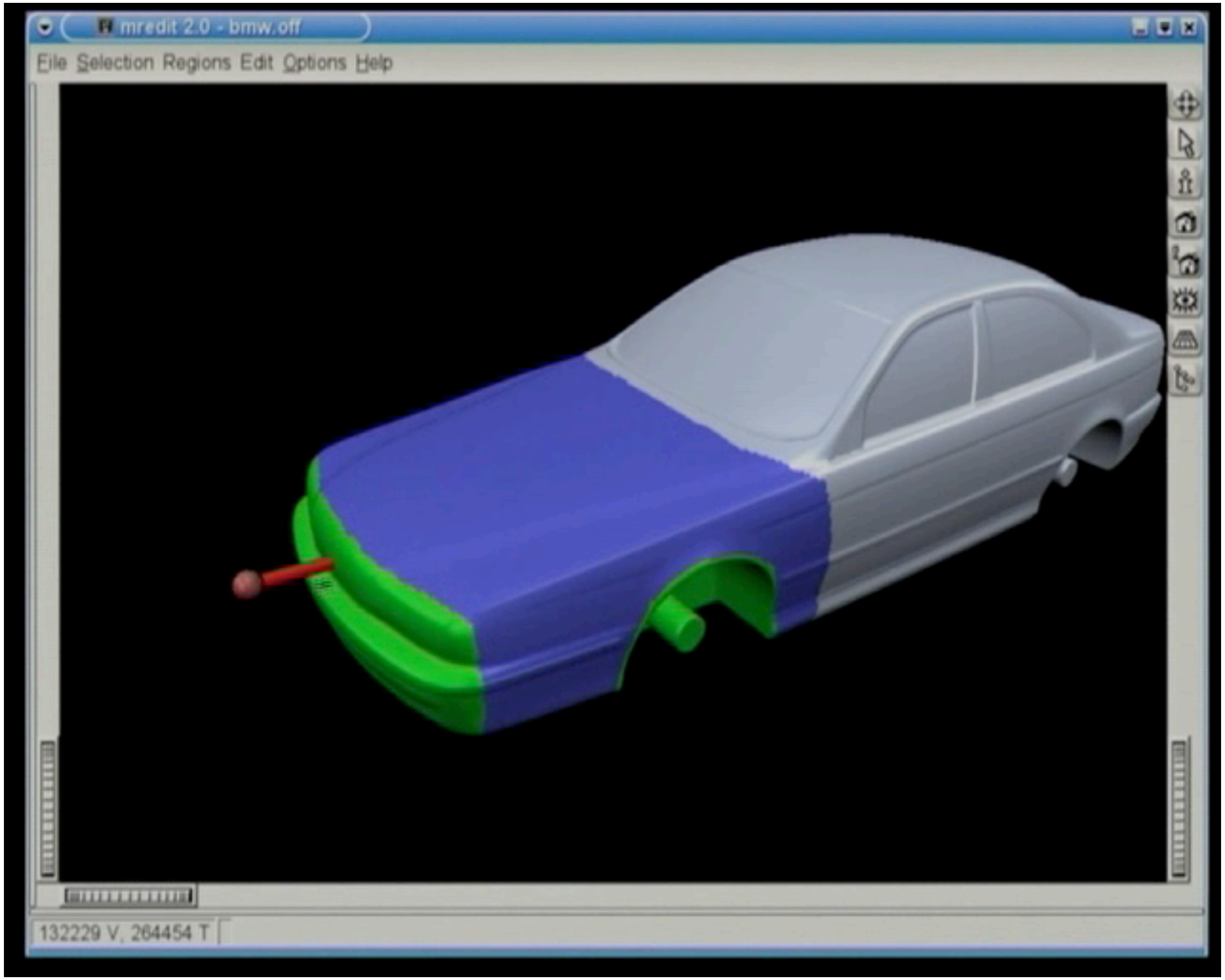
Stefan Fröhlich & Mario Botsch

Bielefeld University

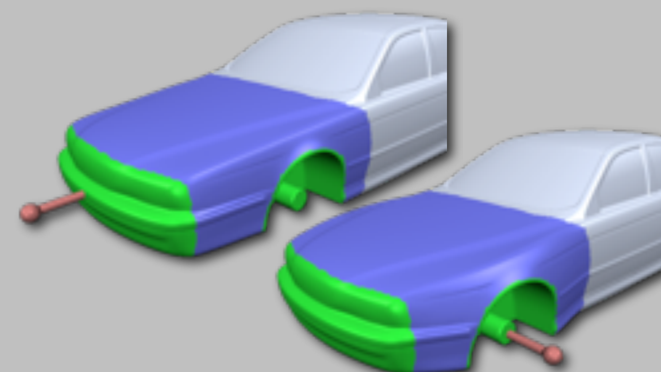
Universität Bielefeld

CITEC

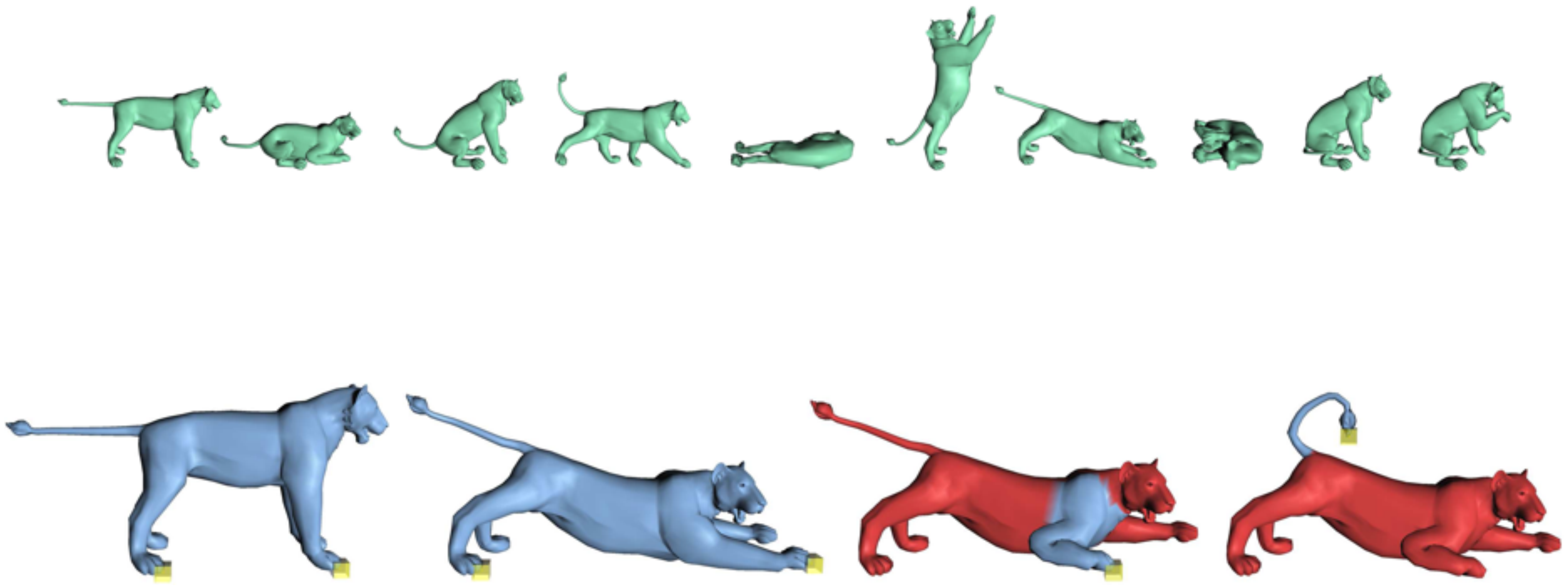
Physics-Based Deformations



Mesh Deformations

		
Physics-based deformation	✓	✗
Skeleton-based deformation	✗	✓
Example-based deformation	✓	✓

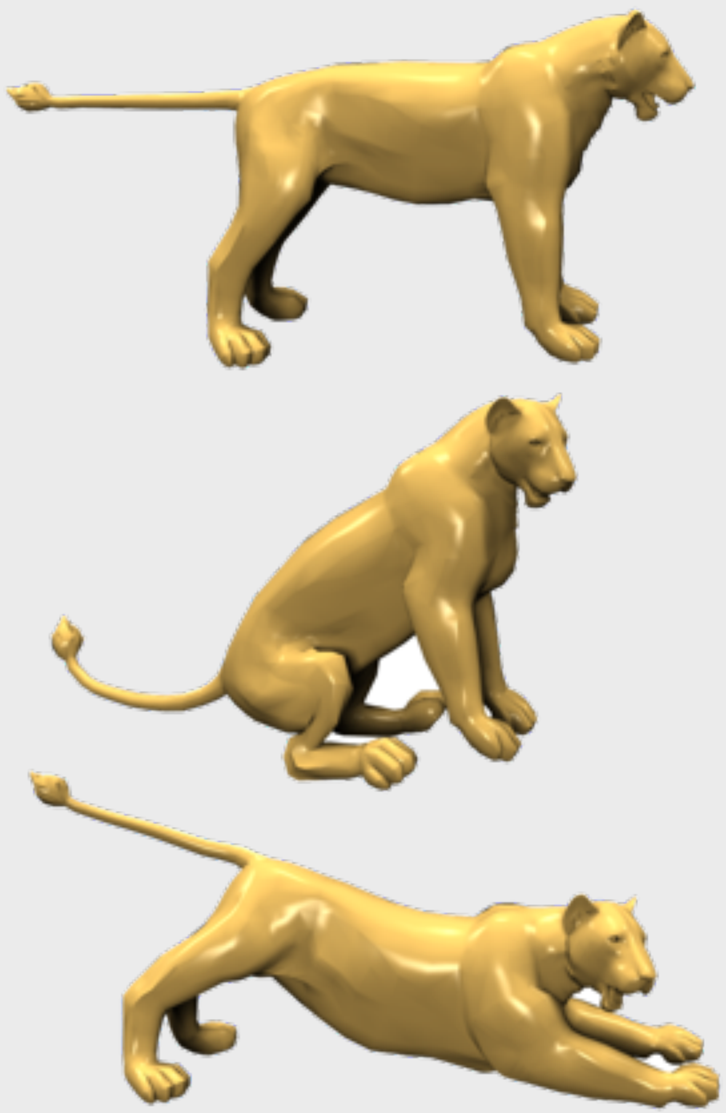
Mesh-Based Inverse Kinematics

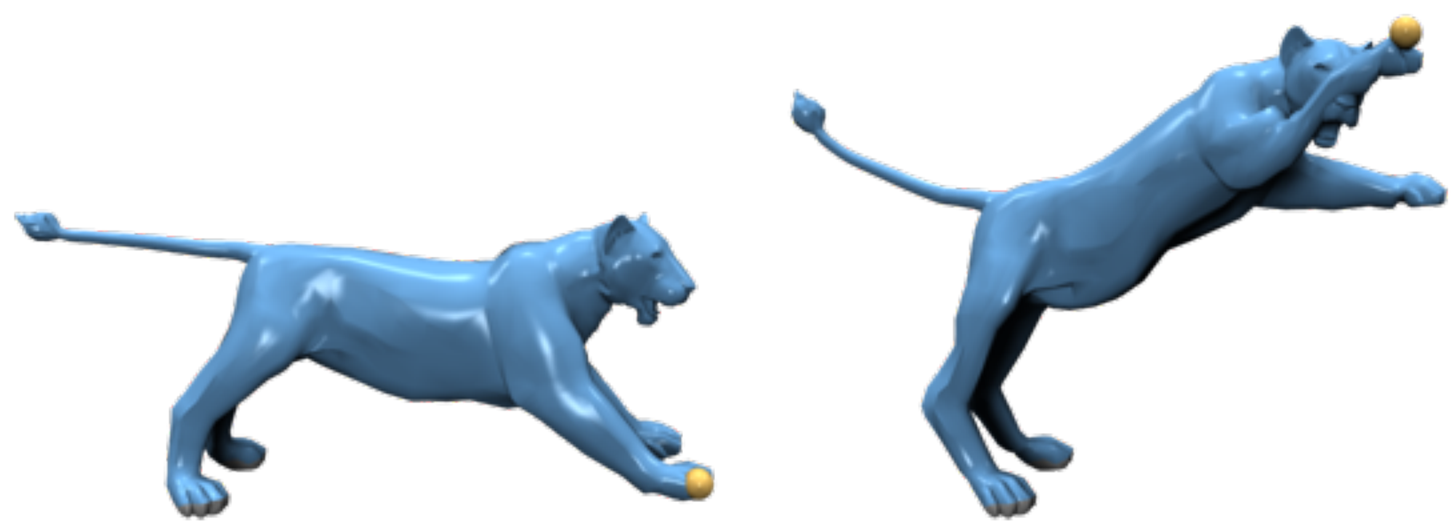


[Sumner et al, *Mesh-Based Inverse Kinematics*, SIGGRAPH 05]

Mesh-Based Inverse Kinematics

Example Poses

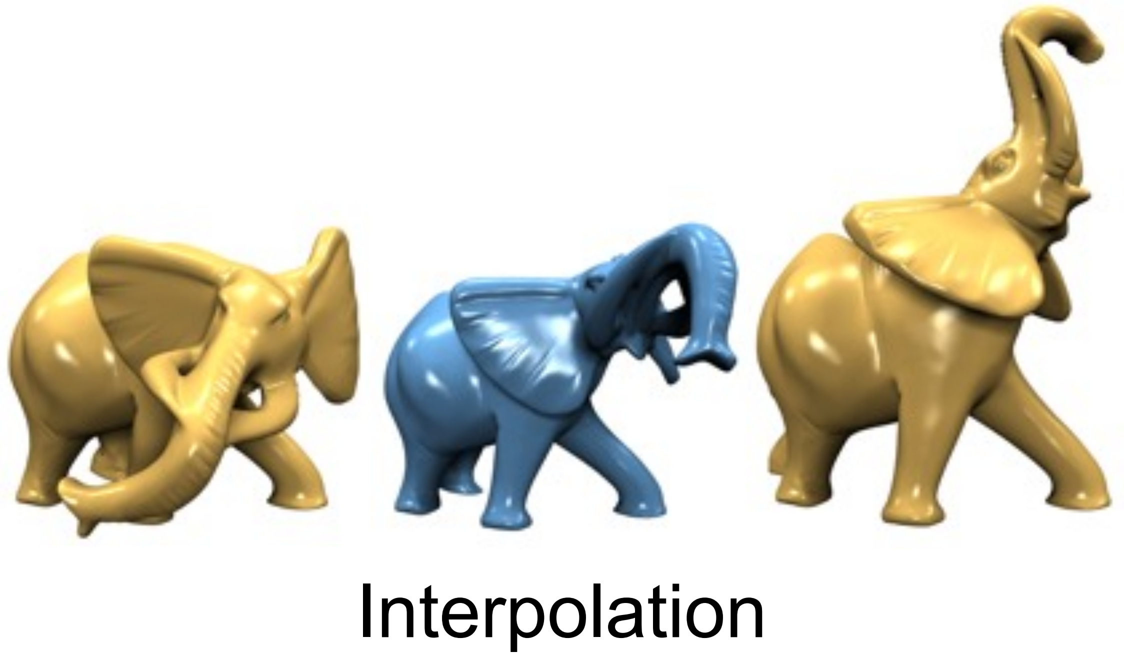


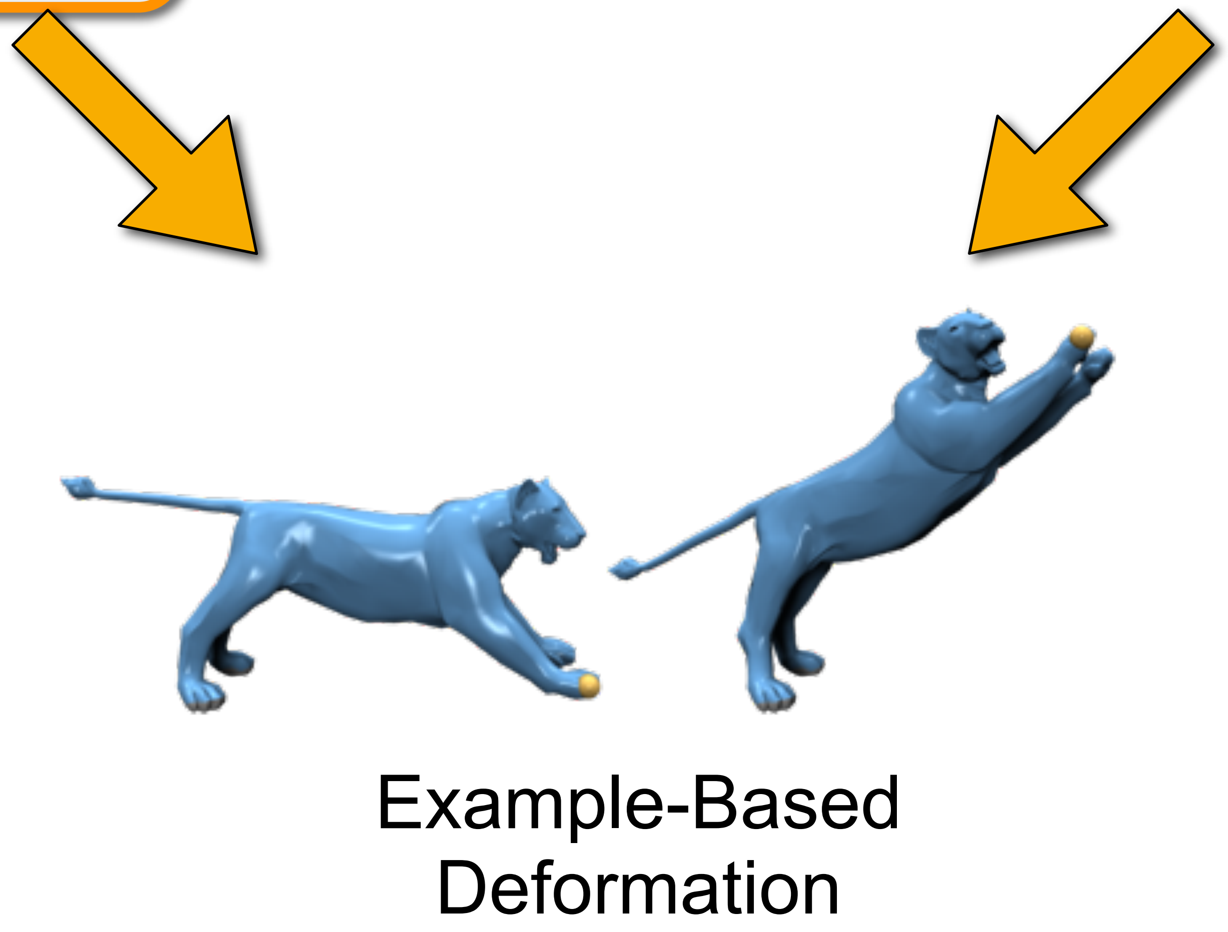


Example-Based Deformation

- **Interpolation**: Find interpolated mesh that best matches the user's constraints
- **Deformation**: Deform this mesh to exactly match the user's constraints

Outline

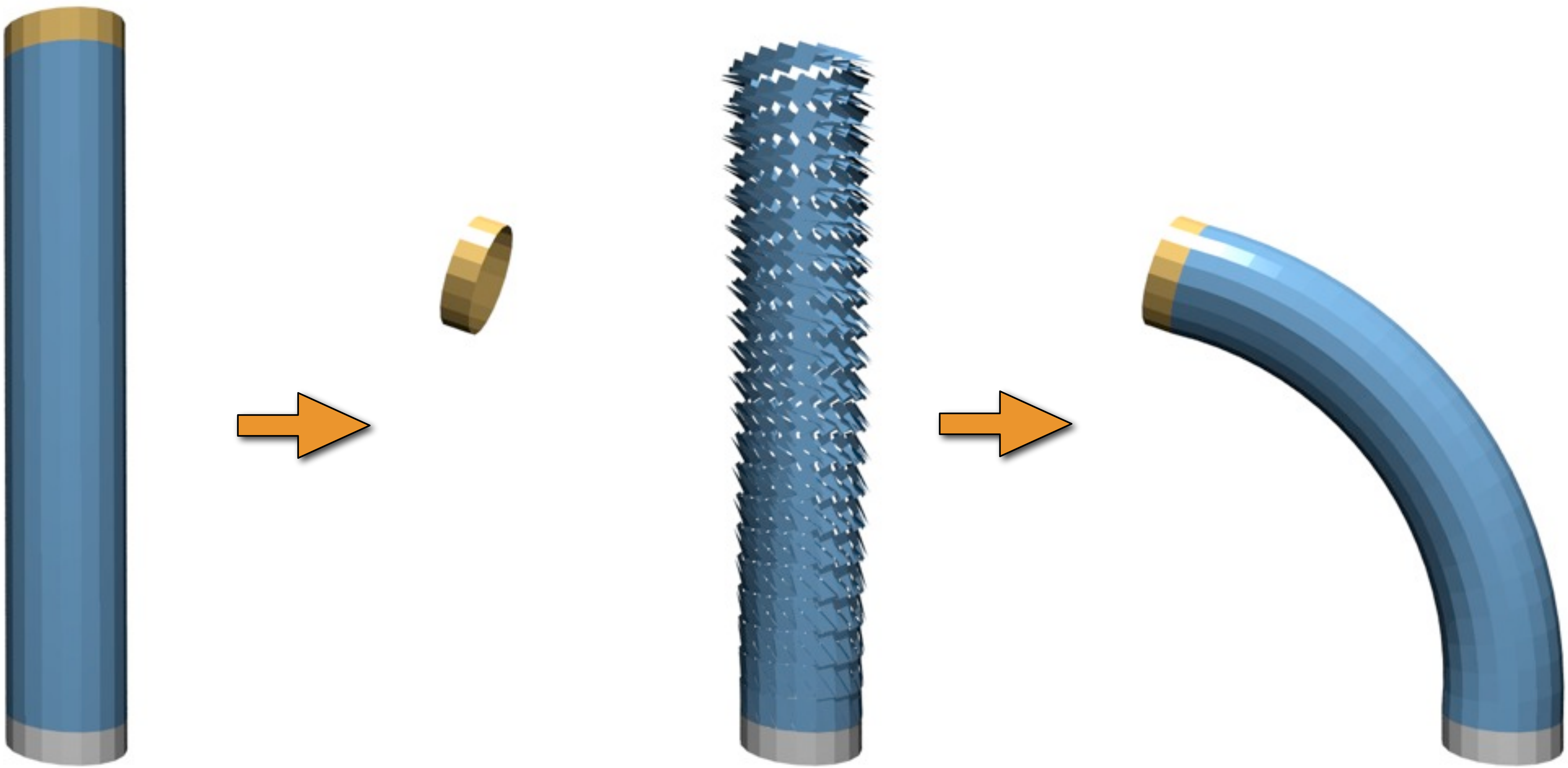




Shape Space Representation

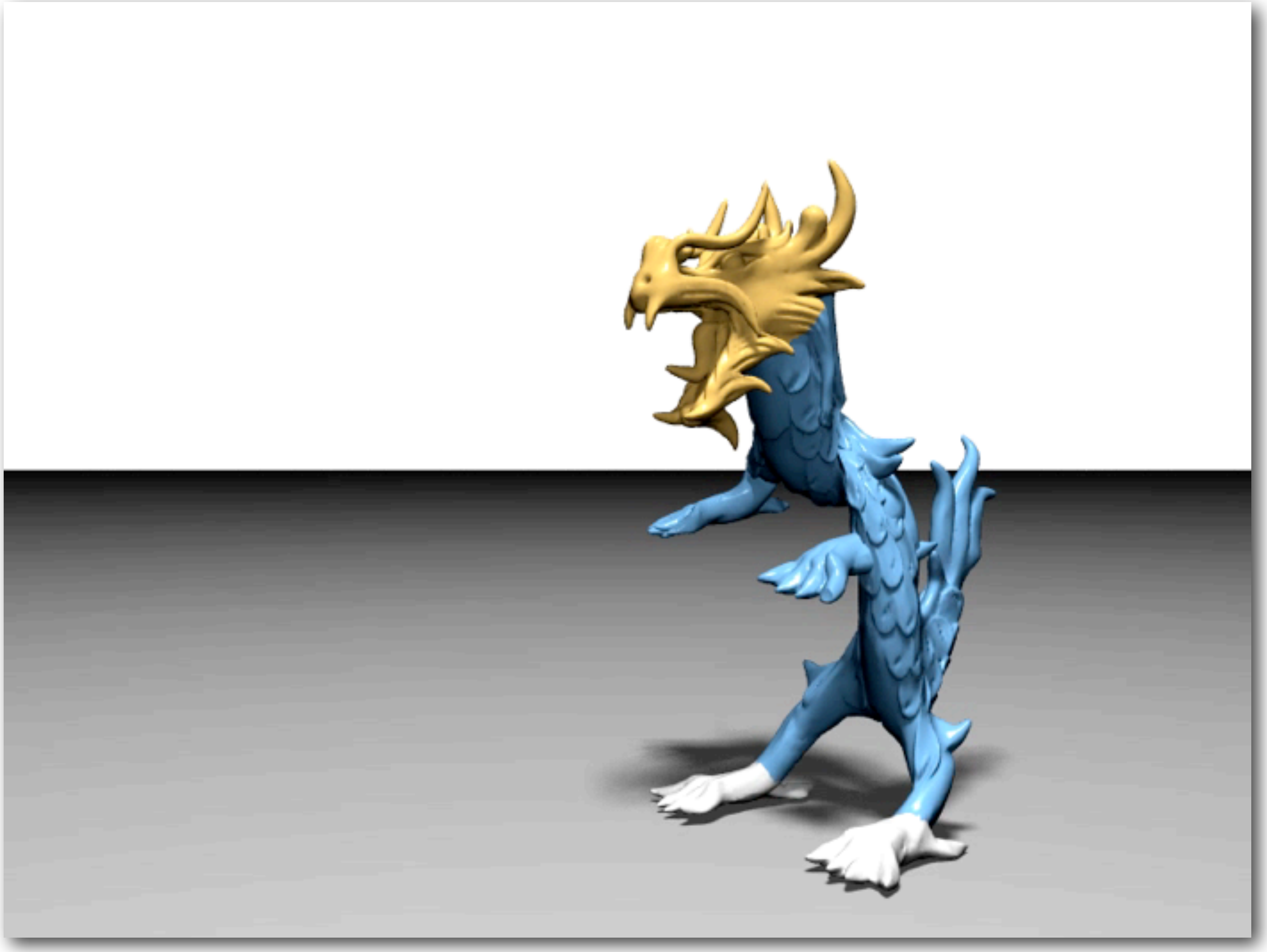
	Deformation	Interpolation
[Sumner 2005]	Deformation gradients	
ours		

Deformation Gradients



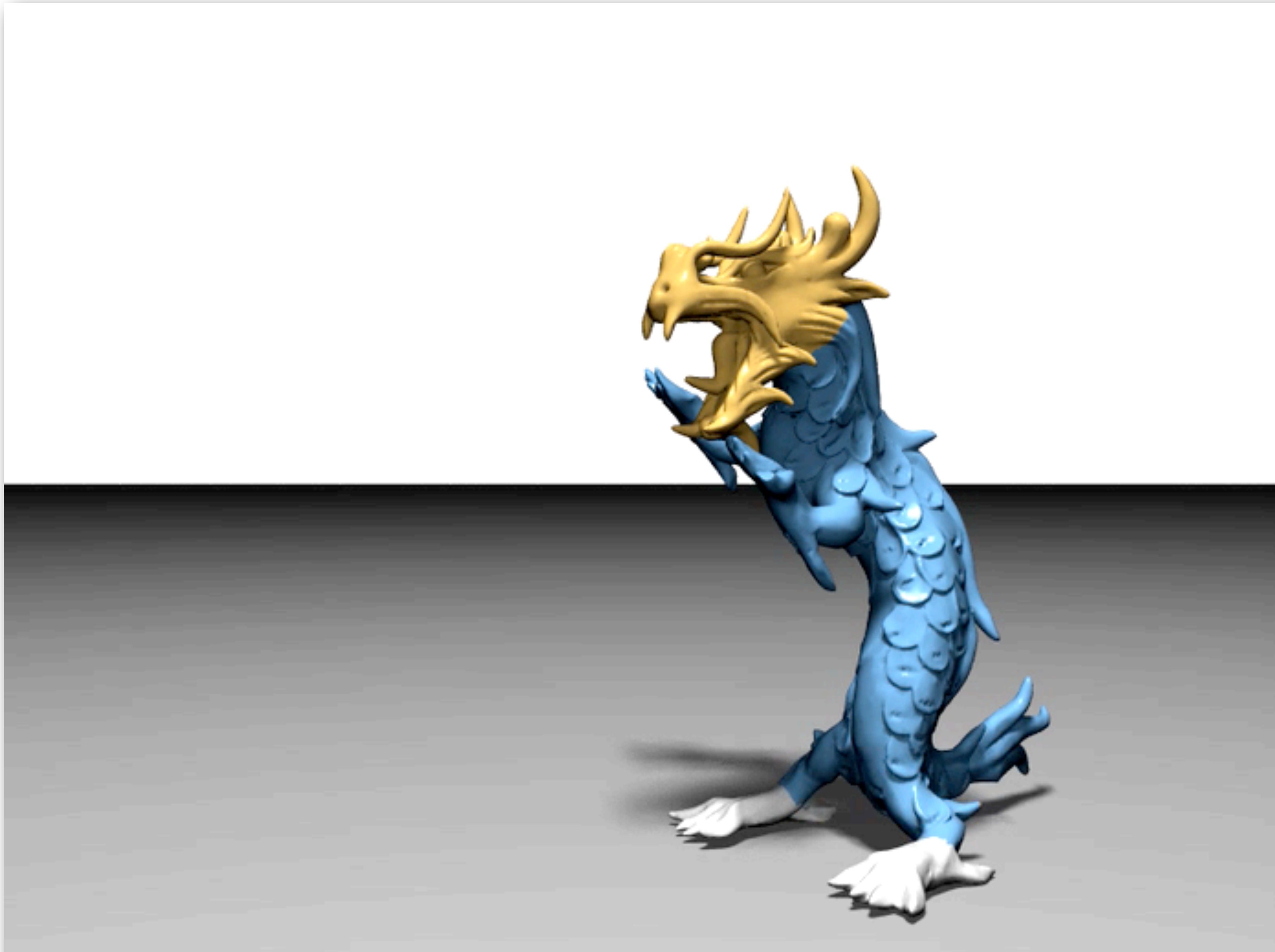
Original Rotated Gradients Reconstructed Mesh

Linear vs. Nonlinear



Linear deformation gradients

Linear vs. Nonlinear



Nonlinear deformation

Shape Space Representation

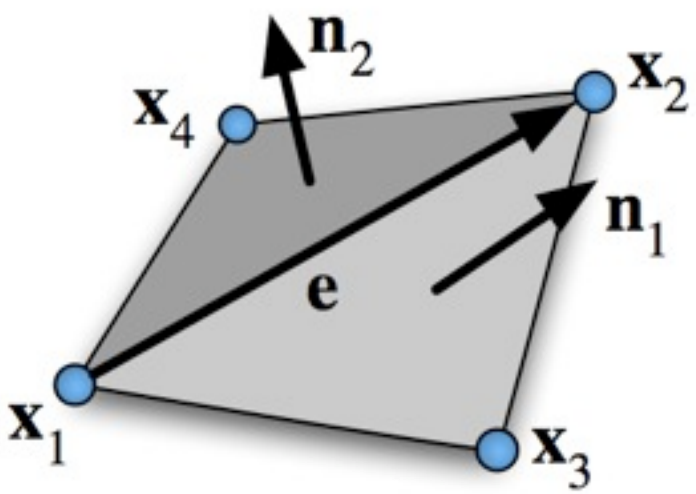
	Deformation	Interpolation
[Sumner 2005]	Deformation gradients	
ours	Edge lengths & dihedral angles	

Nonlinear Discrete Shells

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m) \;=\; \lambda \sum_e w_{s,e} \left(l_e - L_e\right)^2 \;+\; \mu \sum_e w_{b,e} \left(\theta_e - \Theta_e\right)^2$$

Stretching: change of edge length

Bending: change of dihedral angle



[Grinspun et al, SCA 2003]

Gauss-Newton Minimization

- Residual function

$$\mathbf{f}: \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \end{bmatrix} \mapsto \begin{bmatrix} \sqrt{\lambda\, w_{s,1}}\ (l_1 - L_1) \\ \vdots \\ \sqrt{\lambda\, w_{s,m}}\ (l_m - L_m) \\ \sqrt{\mu\, w_{b,1}}\ (\theta_1 - \Theta_1) \\ \vdots \\ \sqrt{\mu\, w_{b,m}}\ (\theta_m - \Theta_m) \end{bmatrix}, \qquad E(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T\, \mathbf{f}(\mathbf{x}) \to \min$$

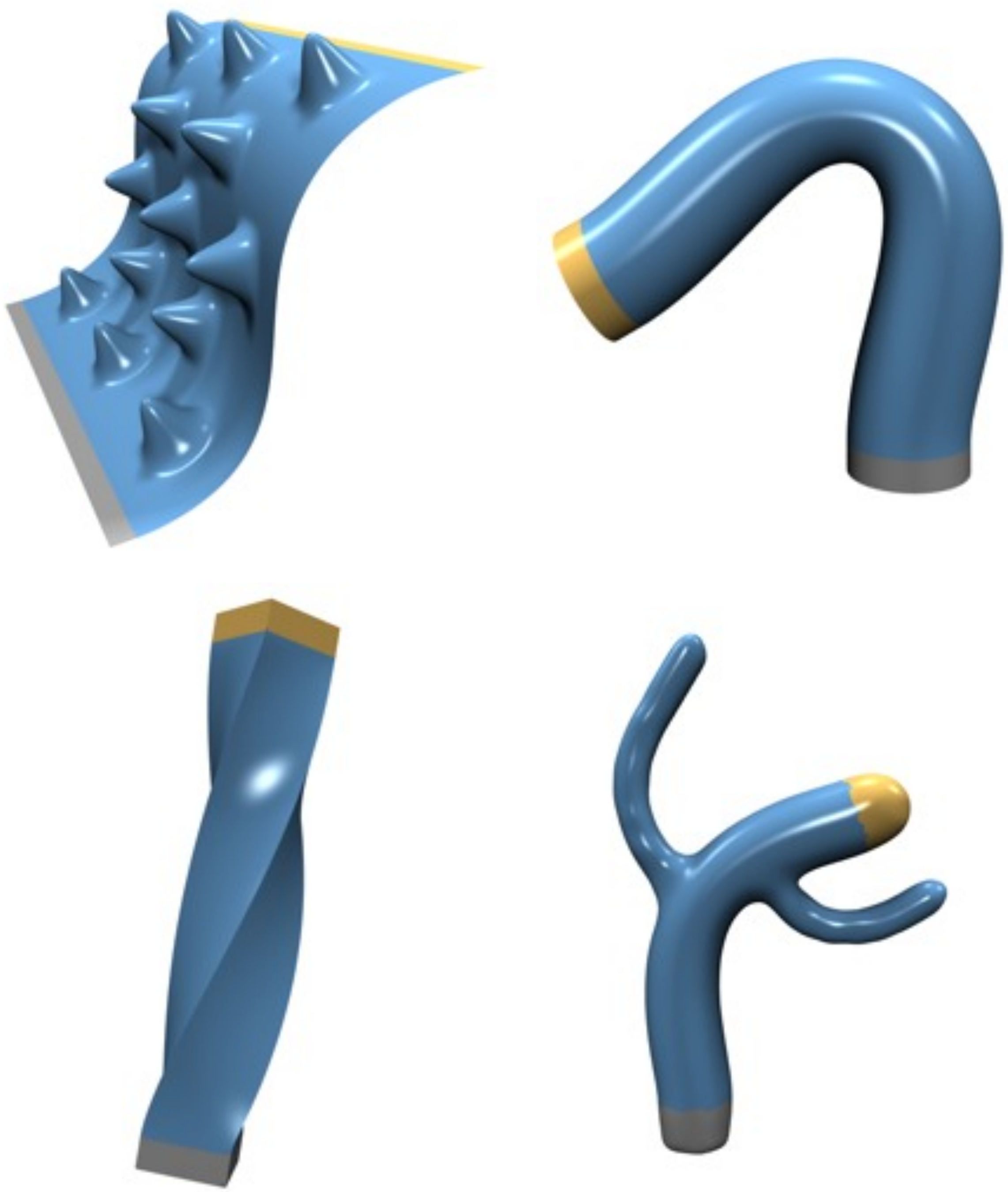
- Iterate until convergence

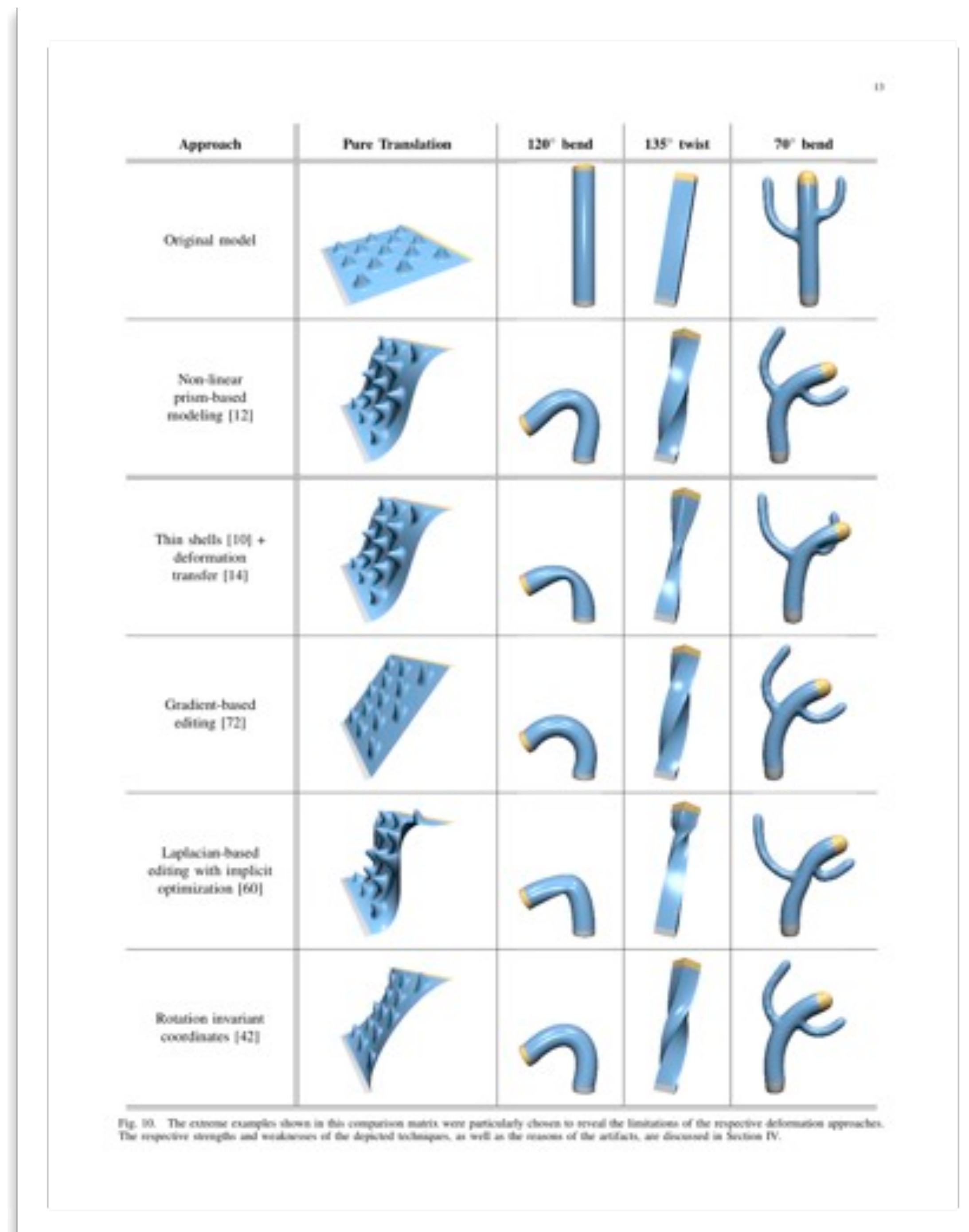
analytical derivatives

$$\mathbf{J}(\mathbf{x})^T\, \mathbf{J}(\mathbf{x})\ \boldsymbol{\delta}\ =\ -\mathbf{J}(\mathbf{x})^T\, \mathbf{f}(\mathbf{x})$$

$$\mathbf{x}\ \leftarrow\ \mathbf{x} + h\, \boldsymbol{\delta}$$

Deformation Results





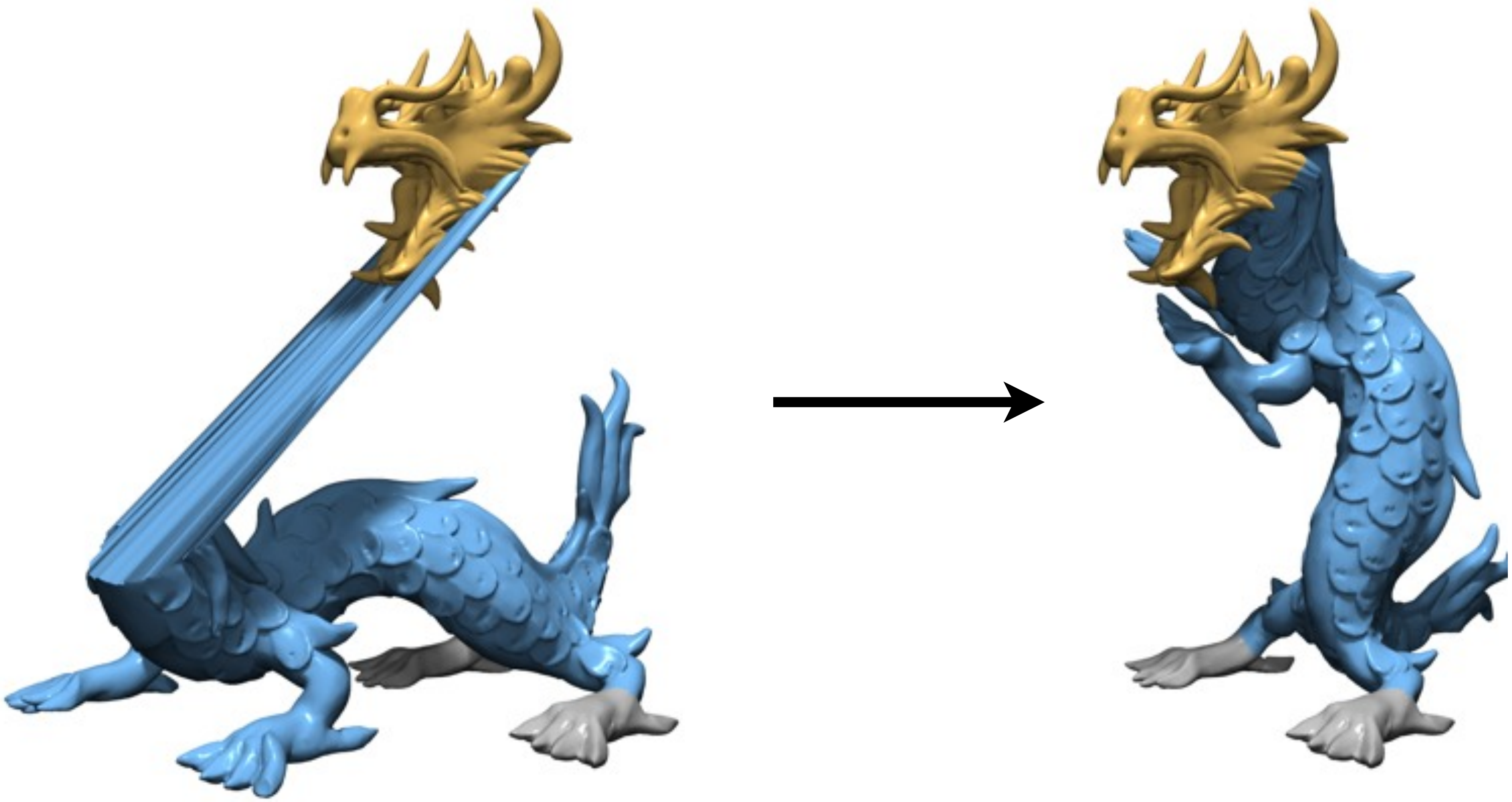
13

Approach	Pure Translation	120° bend	135° twist	70° bend
Original model				
Non-linear prism-based modeling [12]				
Thin shells [10] + deformation transfer [14]				
Gradient-based editing [72]				
Laplacian-based editing with implicit optimization [60]				
Rotation invariant coordinates [42]				

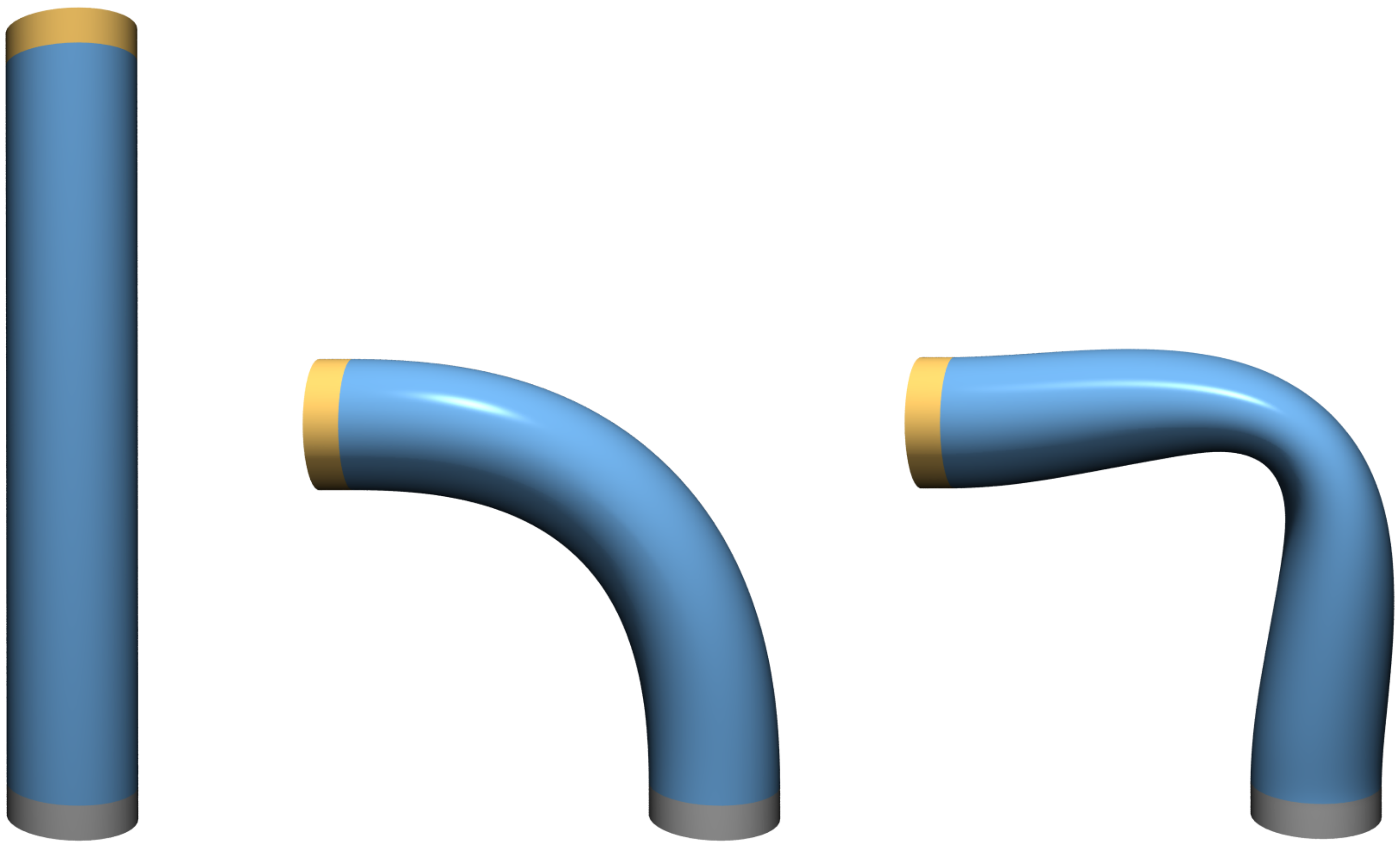
Fig. 10. The extreme examples shown in this comparison matrix were particularly chosen to reveal the limitations of the respective deformation approaches. The respective strengths and weaknesses of the depicted techniques, as well as the reasons of the artifacts, are discussed in Section IV.

[Botsch & Sorkine, TVCG 08]

Robustness

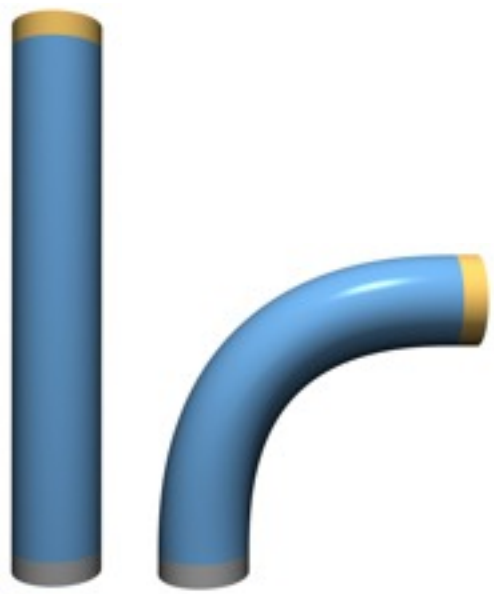


Global Stiffness Control



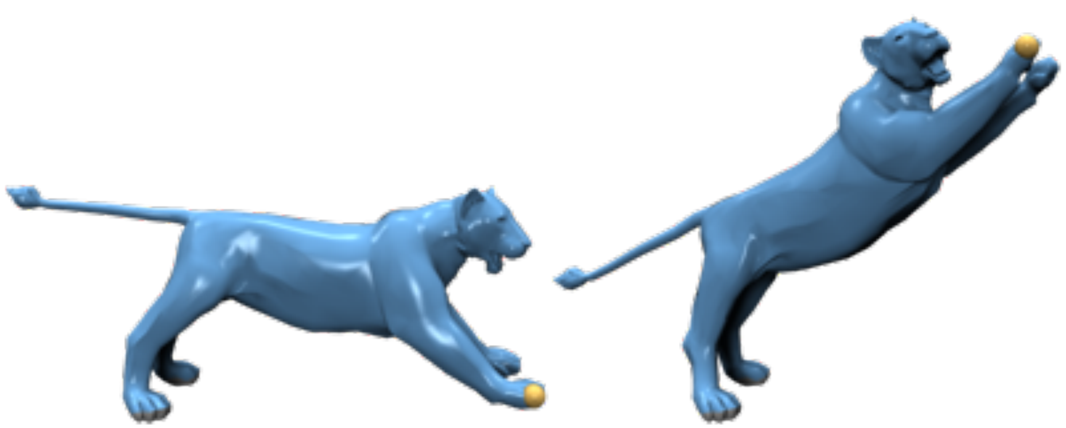
Local Stiffness Control

Outline



Deformation

Interpolation



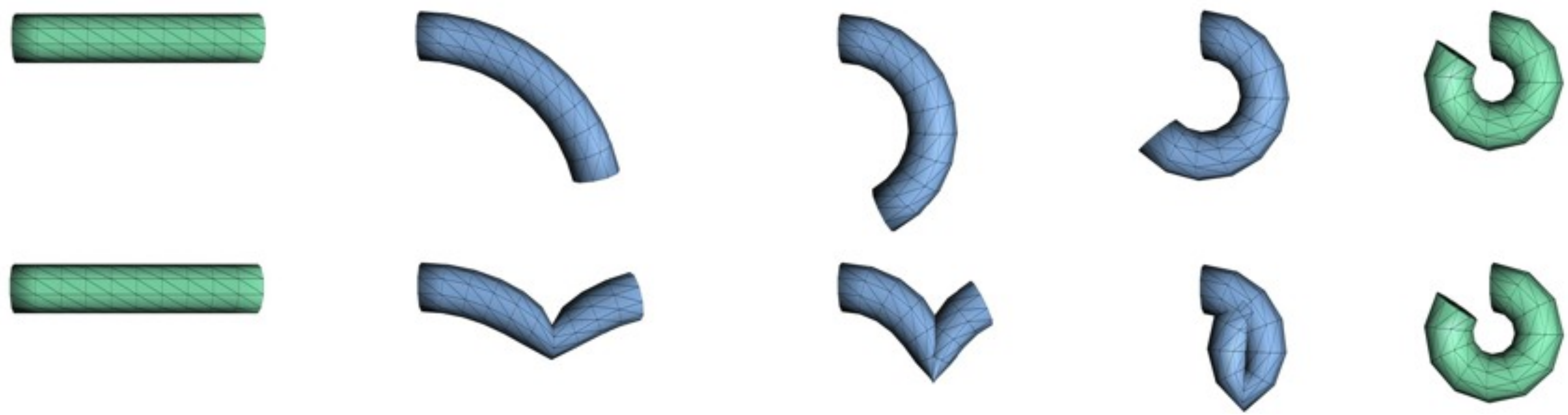
Example-Based
Deformation

Shape Space Representation

	Deformation	Interpolation
[Sumner 2005]	Deformation gradients	Deformation gradients
ours	Edge lengths & dihedral angles	

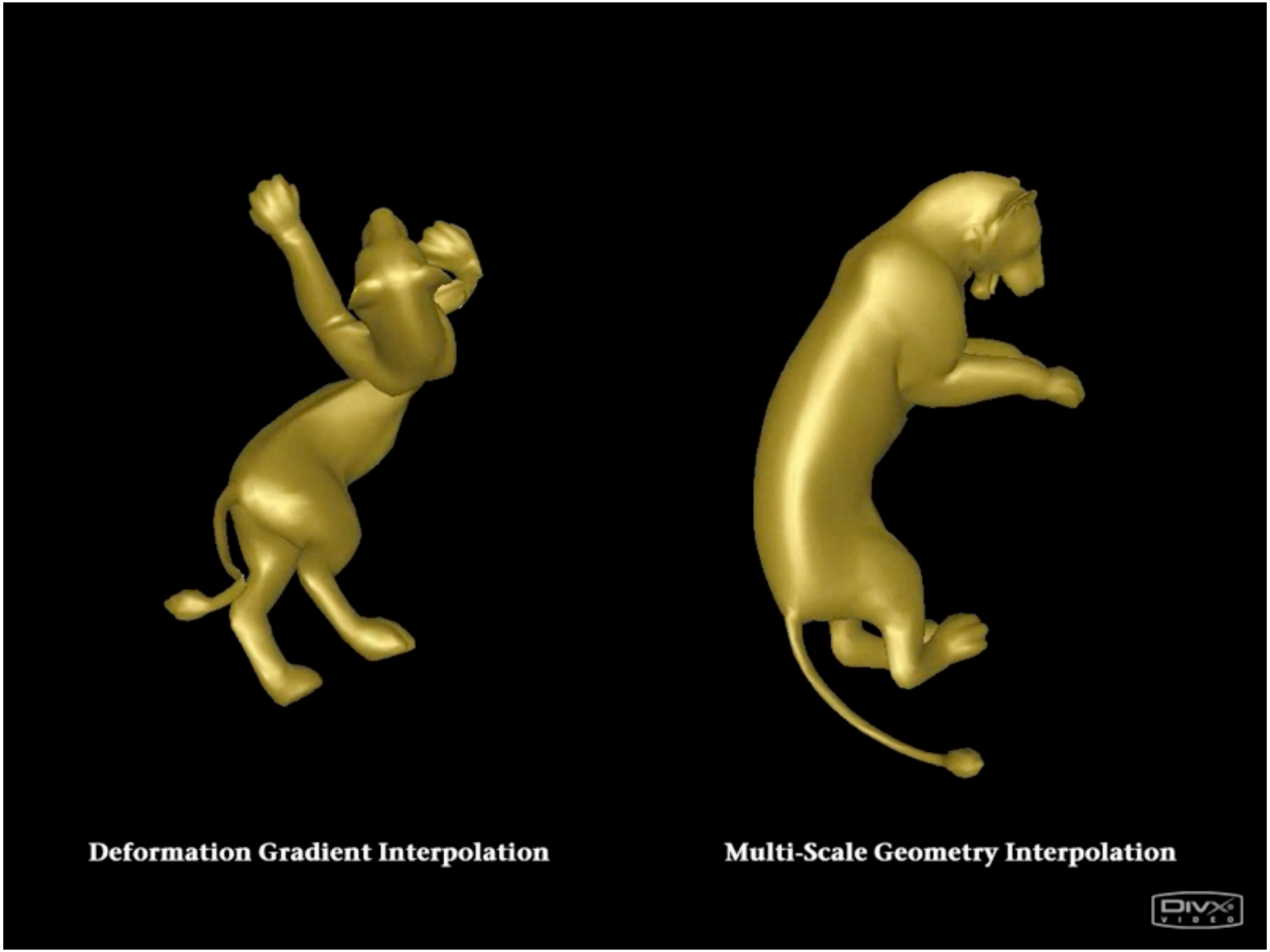
Gradient-Based Interpolation

- Limitation: rotation > 180 degrees



- Solution:
  - per-triangle rotation → per-edge dihedral angle [Lipman et al, SIG 2005; Winkler et al, EG 2010]

Mesh Interpolation



[Winkler et al, Eurographics 2010]

Shape Space Representation

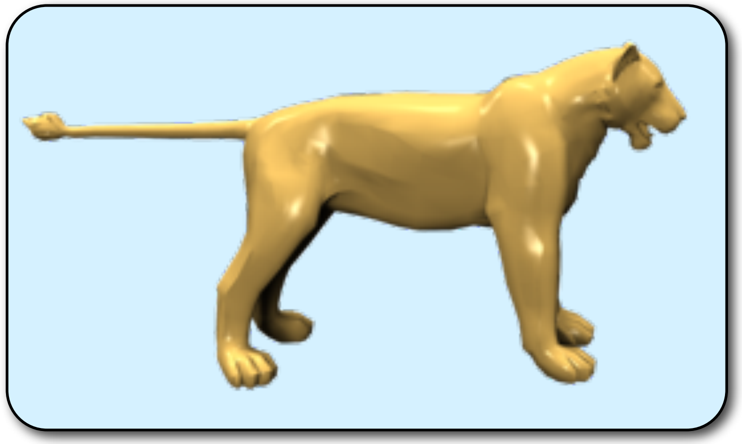
	Deformation	Interpolation
[Sumner 2005]	Deformation gradients	Deformation gradients
ours	Edge lengths & dihedral angles	Edge lengths & dihedral angles

Nonlinear Interpolation

- Linearly interpolate edge lengths and angles

$$l_e^* = L_e + \sum_{i=1}^{k} \alpha_i \left(L_e^{(i)} - L_e \right)$$

$$\theta_e^* = \Theta_e + \sum_{i=1}^{k} \alpha_i \left(\Theta_e^{(i)} - \Theta_e \right)$$



Nonlinear Interpolation

- Linearly interpolate edge lengths and angles

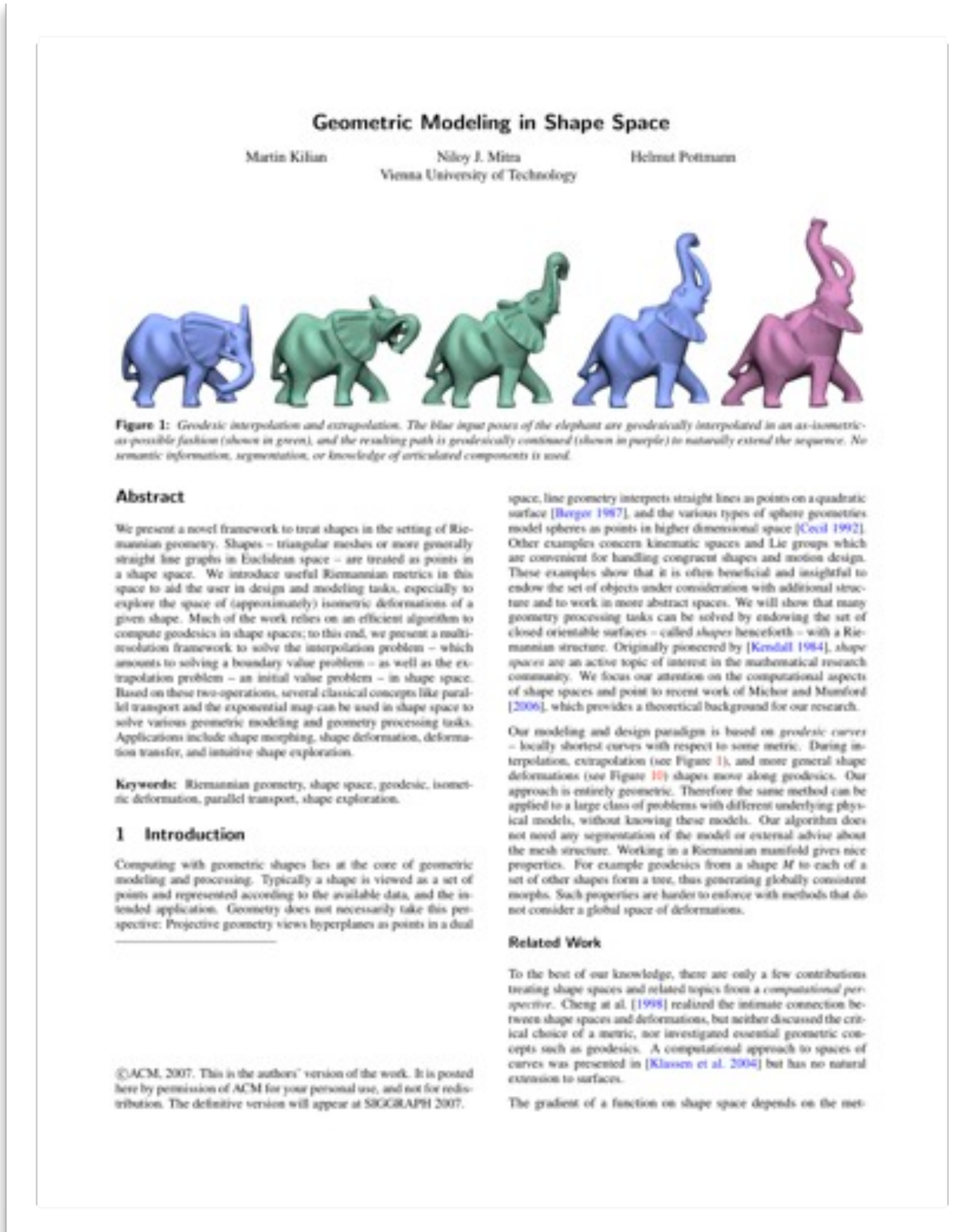
$$l_e^* = L_e + \sum_{i=1}^{k} \alpha_i \left(L_e^{(i)} - L_e \right)$$

$$\theta_e^* = \Theta_e + \sum_{i=1}^{k} \alpha_i \left(\Theta_e^{(i)} - \Theta_e \right)$$

- Use interpolated values as target values

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m) = \lambda \sum_e w_{s,e} \left(l_e - l_e^* \right)^2 + \mu \sum_e w_{b,e} \left(\theta_e - \theta_e^* \right)^2$$

Interpolation Results



Geometric Modeling in Shape Space

Martin Kilian Niloy J. Mitra Helmut Pottmann
Vienna University of Technology

Figure 1: *Geodesic interpolation and extrapolation. The blue input poses of the elephant are geodesically interpolated in an as-isometric-as-possible fashion (shown in green), and the resulting path is geodesically continued (shown in purple) to naturally extend the sequence. No semantic information, segmentation, or knowledge of articulated components is used.*

Abstract

We present a novel framework to treat shapes in the setting of Riemannian geometry. Shapes – triangular meshes or more generally straight line graphs in Euclidean space – are treated as points in a shape space. We introduce useful Riemannian metrics in this space to aid the user in design and modeling tasks, especially to explore the space of (approximately) isometric deformations of a given shape. Much of the work relies on an efficient algorithm to compute geodesics in shape spaces; to this end, we present a multi-resolution framework to solve the interpolation problem – which amounts to solving a boundary value problem – as well as the extrapolation problem – an initial value problem – in shape space. Based on these two operations, several classical concepts like parallel transport and the exponential map can be used in shape space to solve various geometric modeling and geometry processing tasks. Applications include shape morphing, shape deformation, deformation transfer, and intuitive shape exploration.

Keywords: Riemannian geometry, shape space, geodesic, isometric deformation, parallel transport, shape exploration.

1 Introduction

Computing with geometric shapes lies at the core of geometric modeling and processing. Typically a shape is viewed as a set of points and represented according to the available data, and the intended application. Geometry does not necessarily take this perspective: Projective geometry views hyperplanes as points in a dual space, line geometry interprets straight lines as points on a quadratic surface [Berger 1987], and the various types of sphere geometries model spheres as points in higher dimensional space [Cecil 1992]. Other examples concern kinematic spaces and Lie groups which are convenient for handling congruent shapes and motion design. These examples show that it is often beneficial and insightful to endow the set of objects under consideration with additional structure and to work in more abstract spaces. We will show that many geometry processing tasks can be solved by endowing the set of closed orientable surfaces – called *shapes* henceforth – with a Riemannian structure. Originally pioneered by [Kendall 1984], *shape spaces* are an active topic of interest in the mathematical research community. We focus our attention on the computational aspects of shape spaces and point to recent work of Michor and Mumford [2006], which provides a theoretical background for our research.

Our modeling and design paradigm is based on *geodesic curves* – locally shortest curves with respect to some metric. During interpolation, extrapolation (see Figure 1), and more general shape deformations (see Figure 10) shapes move along geodesics. Our approach is entirely geometric. Therefore the same method can be applied to a large class of problems with different underlying physical models, without knowing these models. Our algorithm does not need any segmentation of the model or external advise about the mesh structure. Working in a Riemannian manifold gives nice properties. For example geodesics from a shape M to each of a set of other shapes form a tree, thus generating globally consistent morphs. Such properties are harder to enforce with methods that do not consider a global space of deformations.

Related Work

To the best of our knowledge, there are only a few contributions treating shape spaces and related topics from a *computational perspective*. Cheng at al. [1998] realized the intimate connection between shape spaces and deformations, but neither discussed the critical choice of a metric, nor investigated essential geometric concepts such as geodesics. A computational approach to spaces of curves was presented in [Klassen et al. 2004] but has no natural extension to surfaces.

The gradient of a function on shape space depends on the met-

[Kilian et al, SIG 2007]

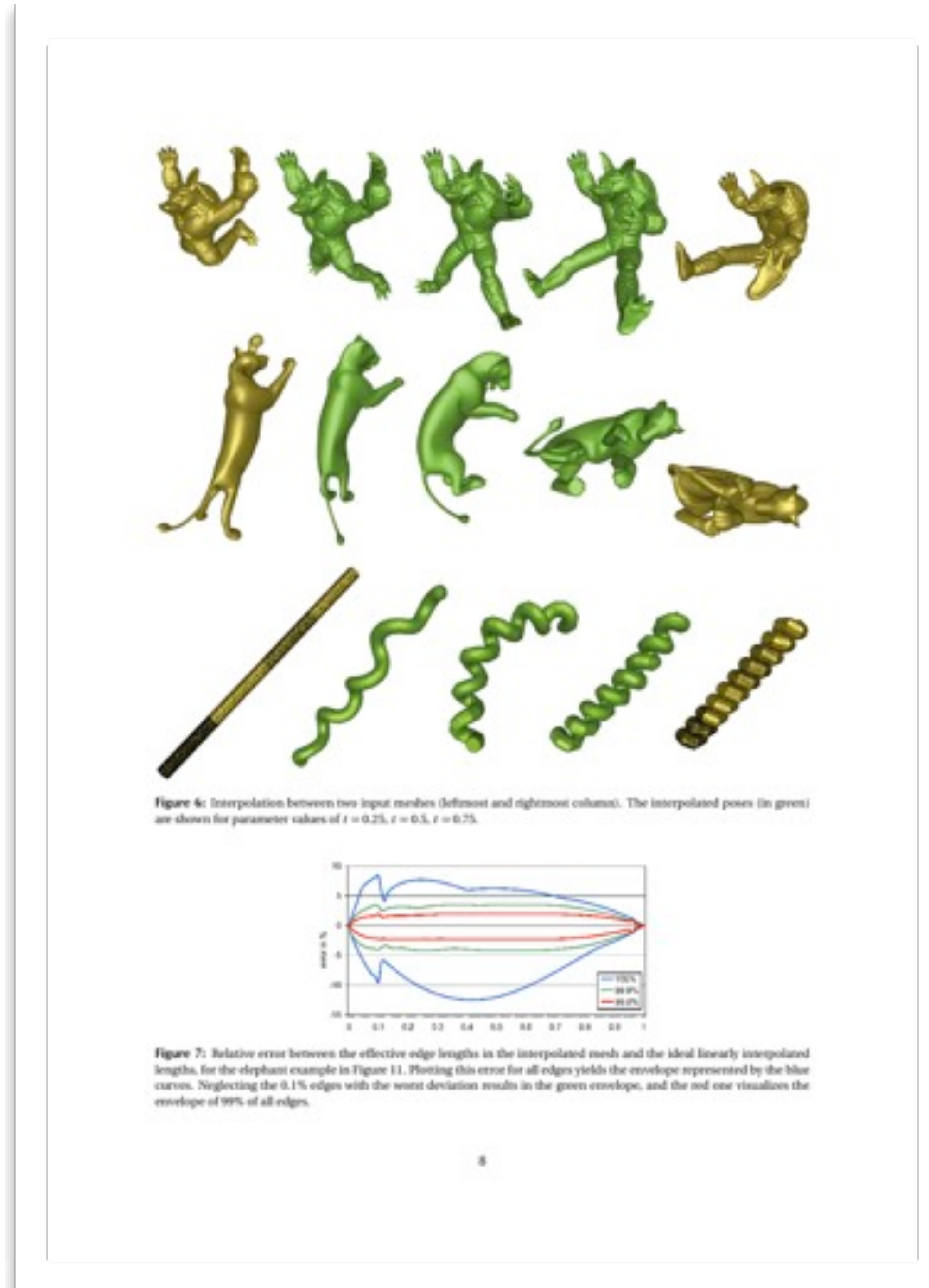
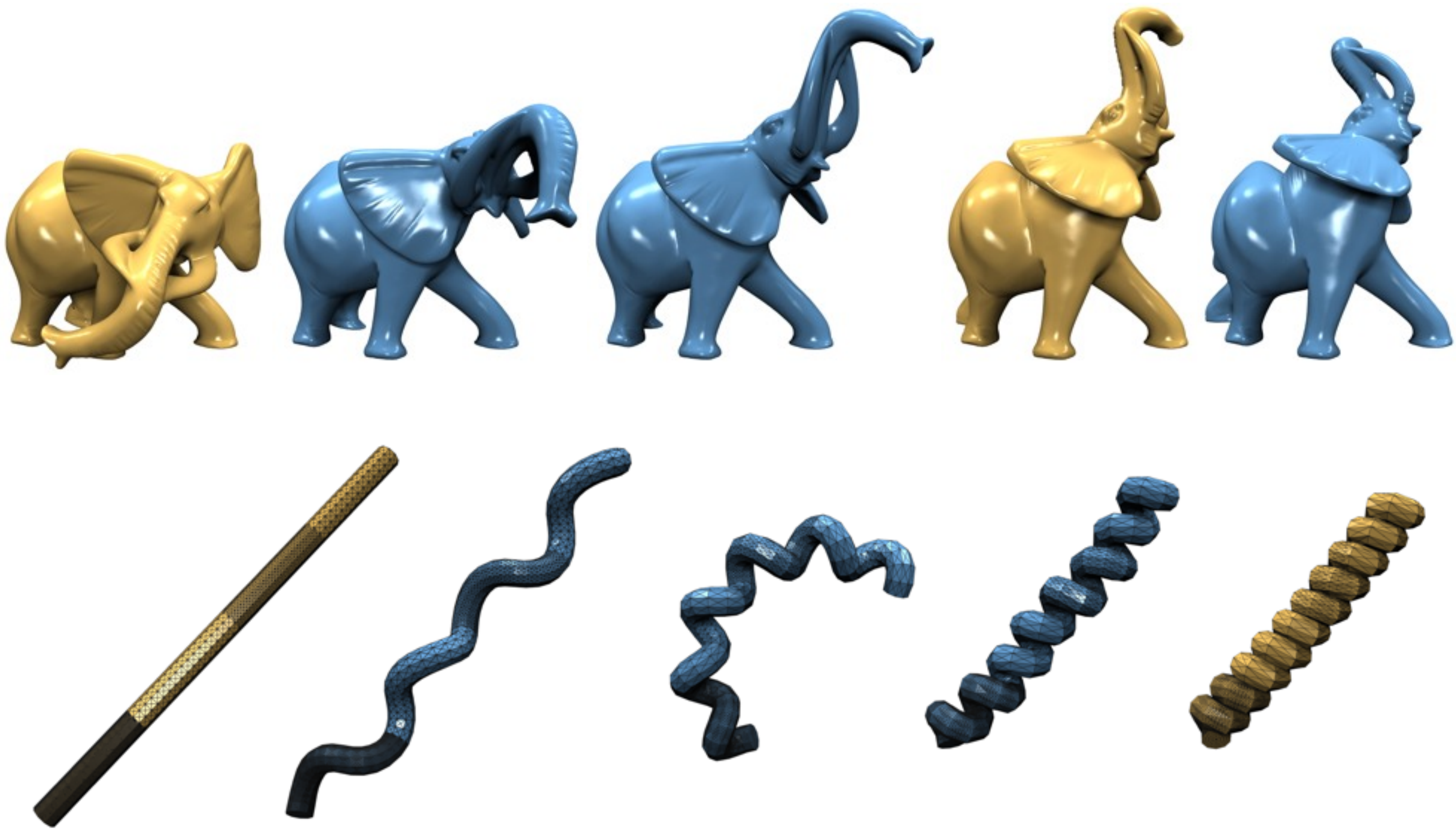


Figure 6: Interpolation between two input meshes (leftmost and rightmost column). The interpolated poses (in green) are shown for parameter values of $t = 0.25$, $t = 0.5$, $t = 0.75$.

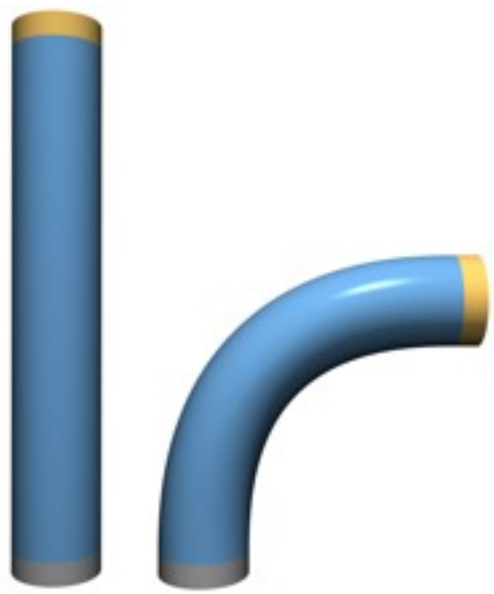
Figure 7: Relative error between the effective edge lengths in the interpolated mesh and the ideal linearly interpolated lengths, for the elephant example in Figure 11. Plotting this error for all edges yields the envelope represented by the blue curves. Neglecting the 0.1% edges with the worst deviation results in the green envelope, and the red one visualizes the envelope of 99% of all edges.

[Winkler et al, EG 2010]

Interpolation Results



Outline



Deformation

Interpolation

Example-Based
Deformation

Example-Based Deformation

	Deformation	Interpolation
[Sumner 2005]	Deformation gradients	Deformation gradients
ours	Edge lengths & dihedral angles	Edge lengths & dihedral angles

Example-Based Deformation

- Unknowns are now:
  - Vertex positions $\mathbf{x}_1, \ldots, \mathbf{x}_m$
  - Interpolation weights $\alpha_1, \ldots, \alpha_k$

$$E(\mathbf{x}_1, \ldots, \mathbf{x}_m, \alpha_1, \ldots, \alpha_k) = \lambda \sum_e w_{s,e} \left(l_e - l_e^*\right)^2 + \mu \sum_e w_{b,e} \left(\theta_e - \theta_e^*\right)^2$$

$$l_e^* = L_e + \sum_{i=1}^{k} \alpha_i \left(L_e^{(i)} - L_e \right)$$

$$\theta_e^* = \Theta_e + \sum_{i=1}^{k} \alpha_i \left(\Theta_e^{(i)} - \Theta_e \right)$$

Gauss-Newton Minimization

- Residual function

$$\mathbf{f}: \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ \vdots \\ x_n \\ y_n \\ z_n \\ \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} \mapsto \begin{bmatrix} \sqrt{\lambda\, w_{s,1}}\,(l_1 - l_1^*) \\ \vdots \\ \sqrt{\lambda\, w_{s,m}}\,(l_m - l_m^*) \\ \sqrt{\mu\, w_{b,1}}\,(\theta_1 - \theta_1^*) \\ \vdots \\ \sqrt{\mu\, w_{b,m}}\,(\theta_m - \theta_m^*) \end{bmatrix}$$

$$l_e^* = L_e + \sum_{i=1}^{k} \alpha_i \left(L_e^{(i)} - L_e \right)$$

$$\theta_e^* = \Theta_e + \sum_{i=1}^{k} \alpha_i \left(\Theta_e^{(i)} - \Theta_e \right)$$

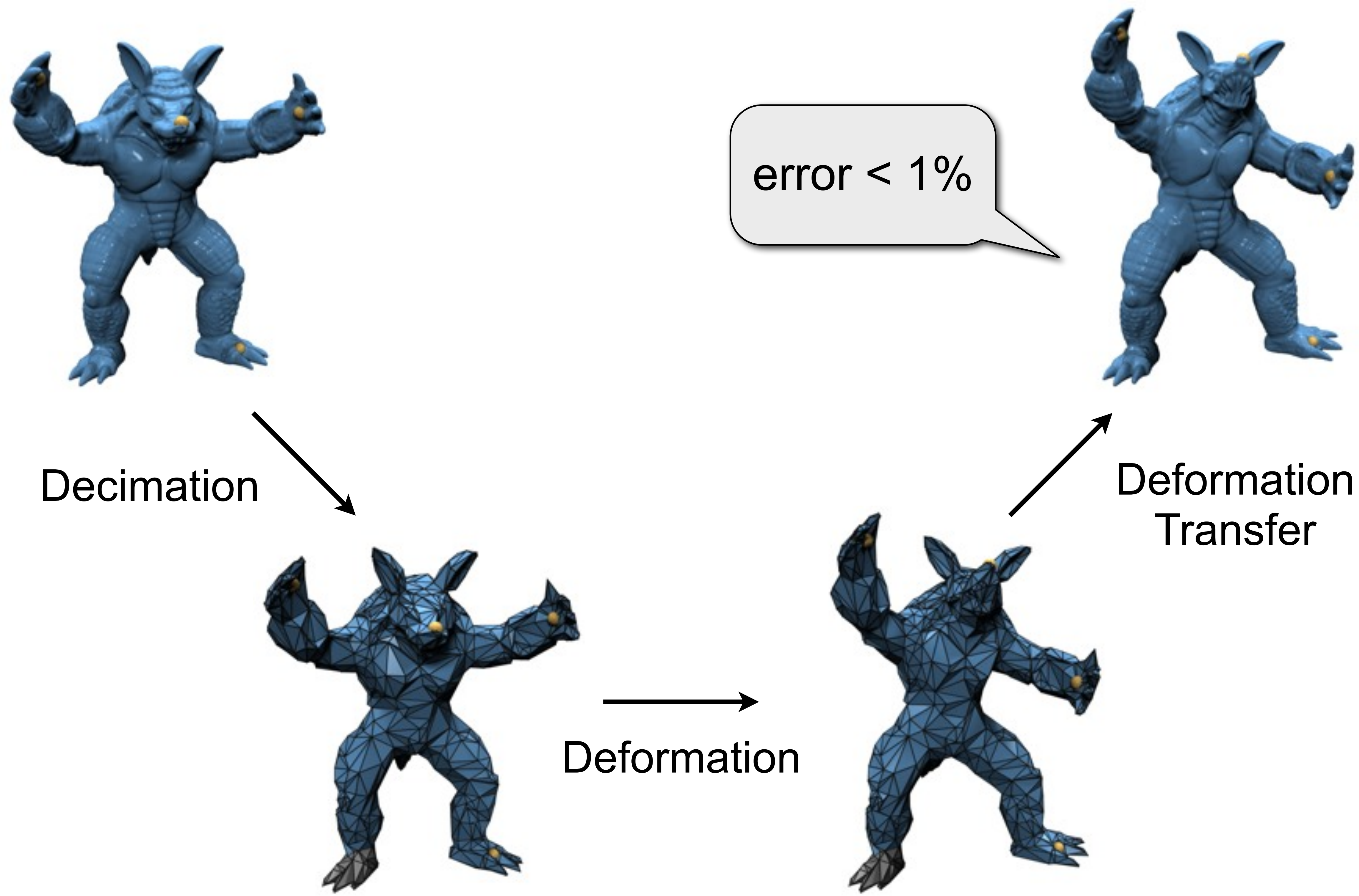
- Iterate until convergence

analytical derivatives

$$\mathbf{J}(\mathbf{x})^T\, \mathbf{J}(\mathbf{x})\; \boldsymbol{\delta} \;=\; -\mathbf{J}(\mathbf{x})^T\, \mathbf{f}(\mathbf{x})$$

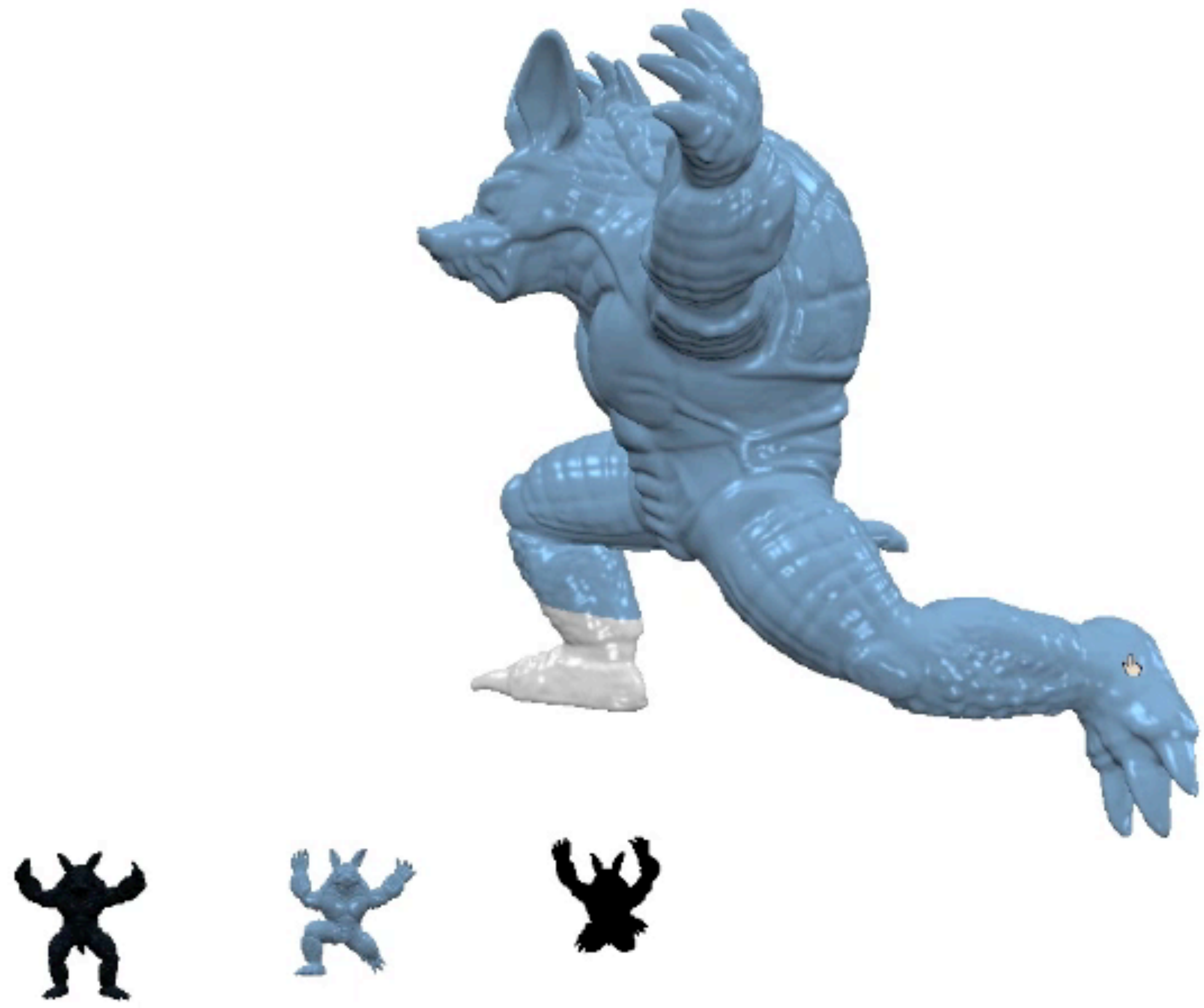
$$\mathbf{x} \;\leftarrow\; \mathbf{x} + h\, \boldsymbol{\delta}$$

Hierarchical Optimization

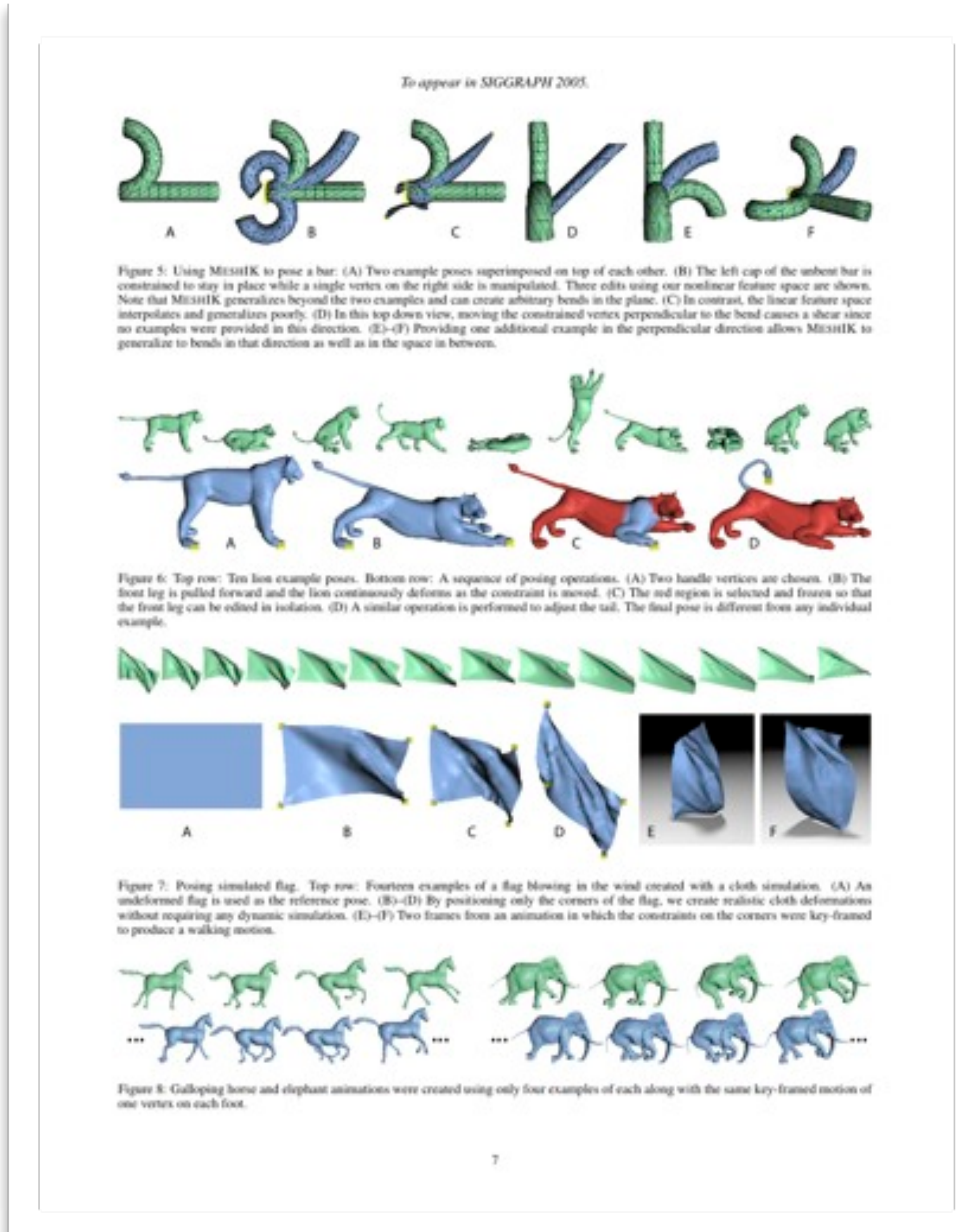


Results

Deformation with MeshIK



Example-Based Deformation



A B C D E F

Figure 5: Using MESHIK to pose a bar: (A) Two example poses superimposed on top of each other. (B) The left cap of the unbent bar is constrained to stay in place while a single vertex on the right side is manipulated. Three edits using our nonlinear feature space are shown. Note that MESHIK generalizes beyond the two examples and can create arbitrary bends in the plane. (C) In contrast, the linear feature space interpolates and generalizes poorly. (D) In this top down view, moving the constrained vertex perpendicular to the bend causes a shear since no examples were provided in this direction. (E)–(F) Providing one additional example in the perpendicular direction allows MESHIK to generalize to bends in that direction as well as in the space in between.

A B C D

Figure 6: Top row: Ten lion example poses. Bottom row: A sequence of posing operations. (A) Two handle vertices are chosen. (B) The front leg is pulled forward and the lion continuously deforms as the constraint is moved. (C) The red region is selected and frozen so that the front leg can be edited in isolation. (D) A similar operation is performed to adjust the tail. The final pose is different from any individual example.

A B C D E F

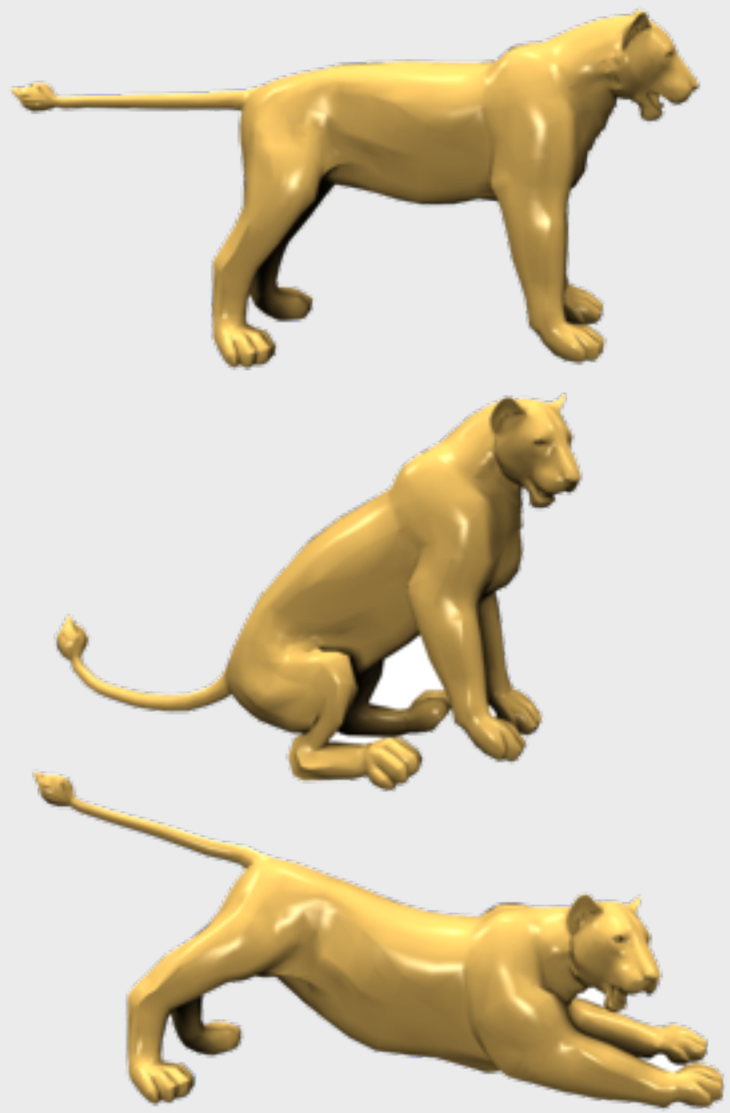
Figure 7: Posing simulated flag. Top row: Fourteen examples of a flag blowing in the wind created with a cloth simulation. (A) An undeformed flag is used as the reference pose. (B)–(D) By positioning only the corners of the flag, we create realistic cloth deformations without requiring any dynamic simulation. (E)–(F) Two frames from an animation in which the constraints on the corners were key-framed to produce a walking motion.

Figure 8: Galloping horse and elephant animations were created using only four examples of each along with the same key-framed motion of one vertex on each foot.

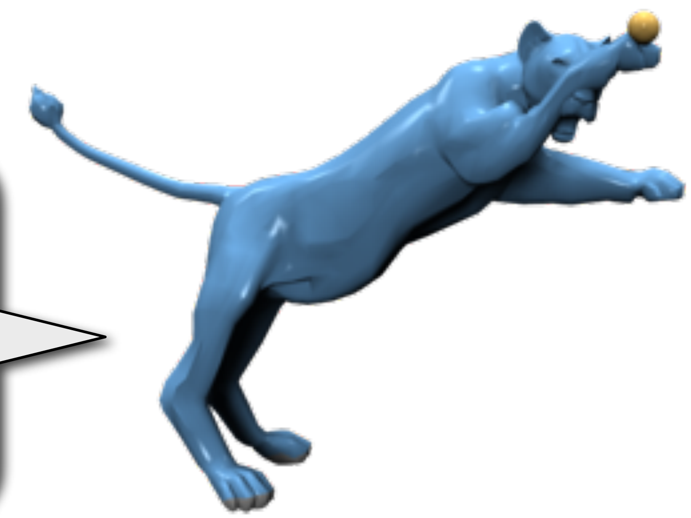
[Sumner et al, SIG 2005]

Results

Example Poses

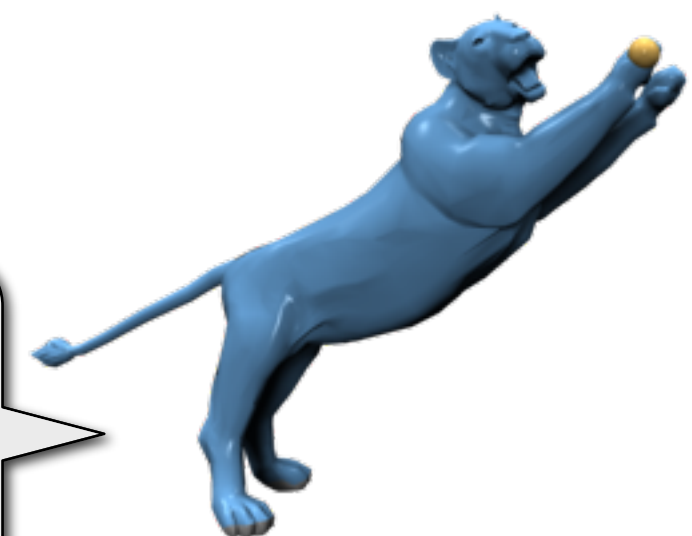


per-triangle global rotations



[Sumner 2005]

per-edge local rotations



ours

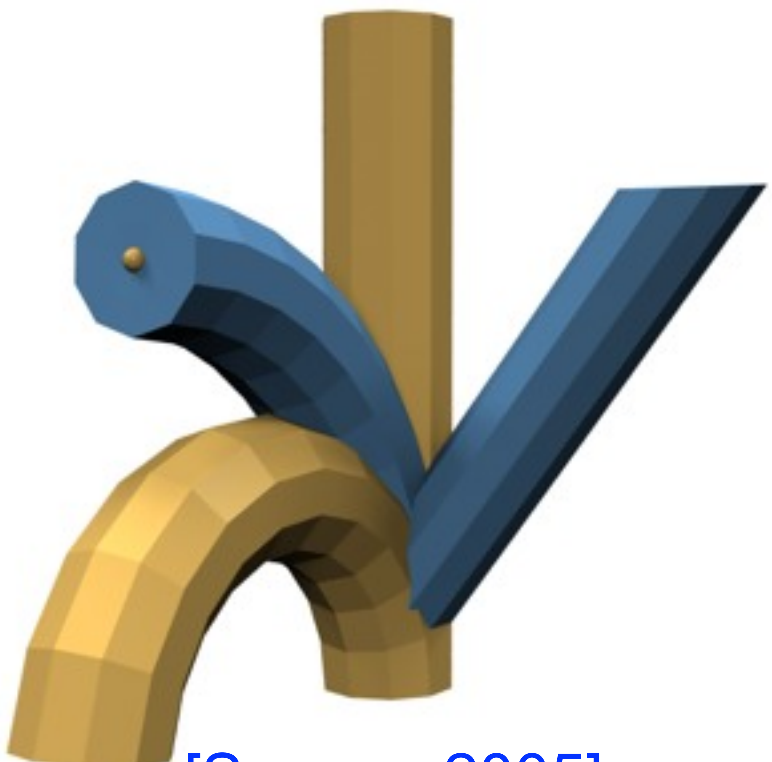
Results

Example Poses

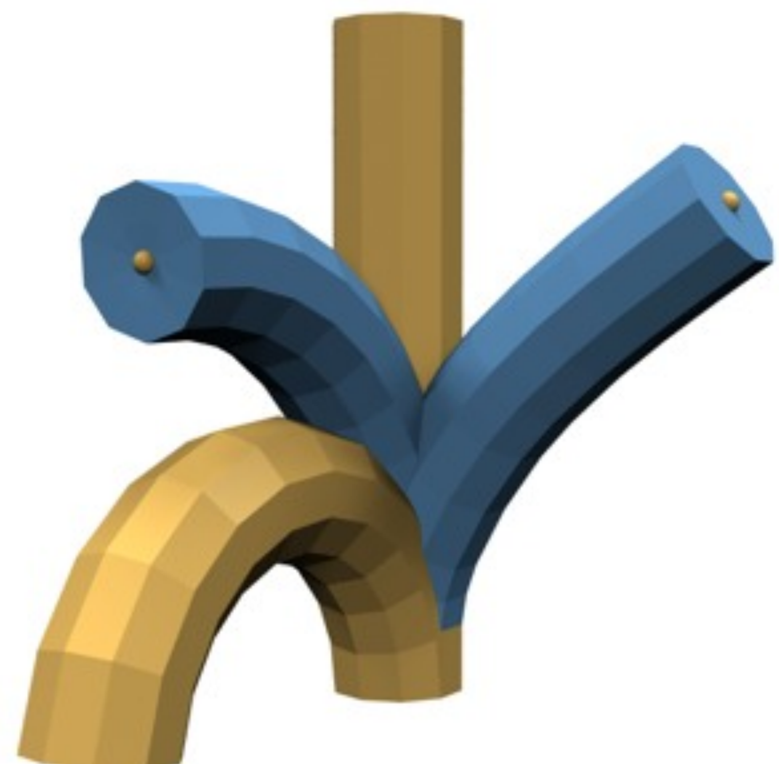
[Sumner 2005]

ours

Results



[Sumner 2005]



ours

linear deformation

nonlinear deformation

Timings: Highres Optimization

Model	#Vertices	Gauss-Newton
Helix	612	18ms
Lion	5k	377ms
Elephant	40k	3.1s
Dragon	50k	5.2s
Armadillo	166k	-

Timings: Hierarchical Optmization





Model	#Vertices	Gauss-Newton	Def. Trans.
Helix	612	18ms	
Lion	5k	~30ms	23ms
Elephant	40k	~30ms	271ms
Dragon	50k	~30ms	323ms
Armadillo	166k	~30ms	1278ms

Thanks for your attention!

References

– S. Fröhlich, M. Botsch, ***Example-Driven Deformations Based on Discrete Shells***, Computer Graphics Forum 30(8), 2011

Acknowledgments

– Bob Sumner: Sharing models and executables

– Funding: DFG, Center of Excellence CITEC