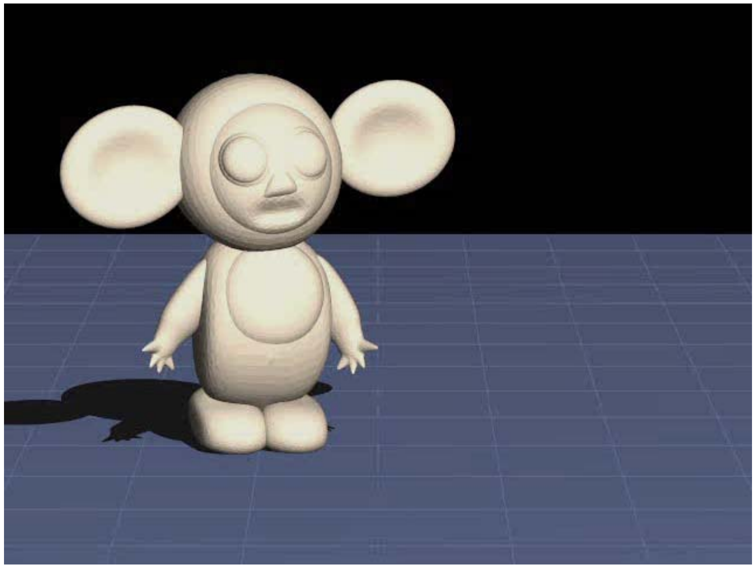
G22.3033-004, Spring 2009

Interactive Shape Modeling

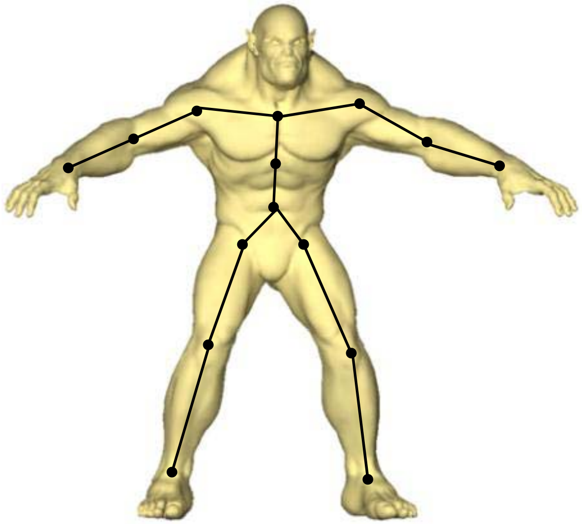
Skeletal deformation

Believable character animation

- Computers games and movies
- Skeleton: intuitive, low-dimensional subspace

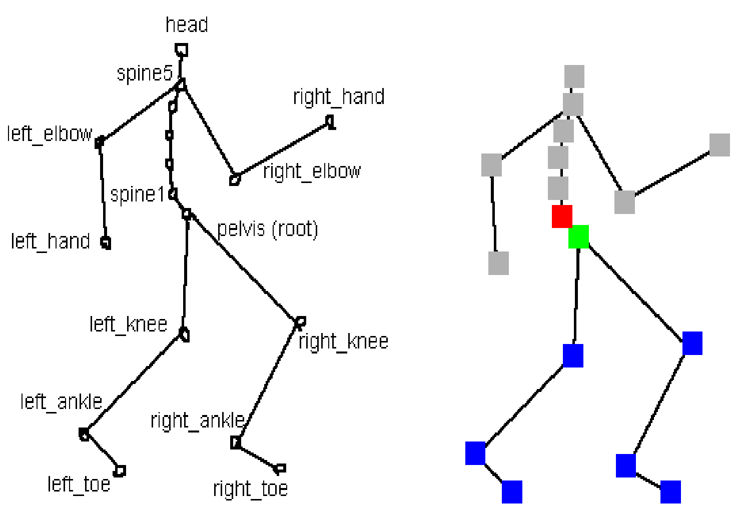


Clip courtesy of Ilya Baran

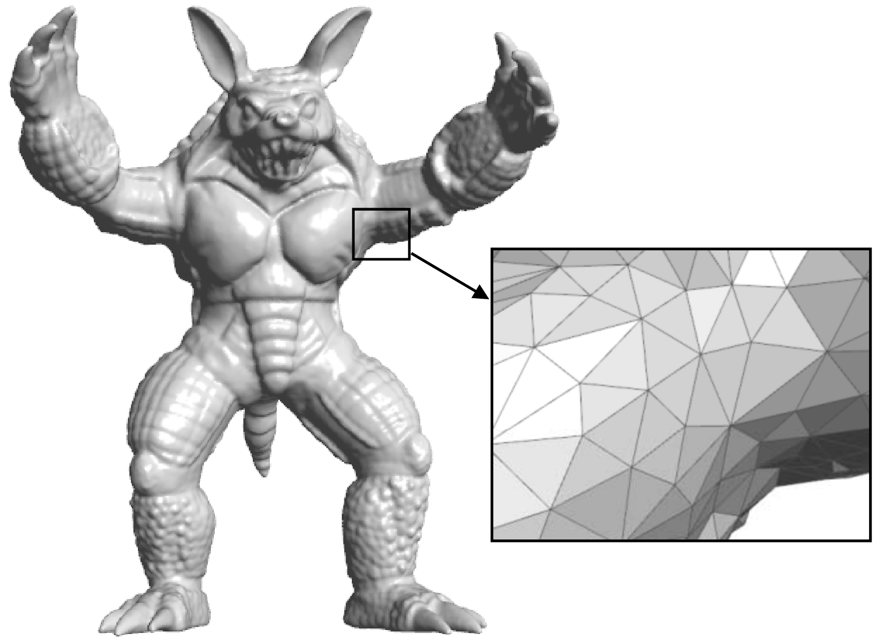


Discrete representation

- Skeleton:
  - collection of line segments
  - connected by joints

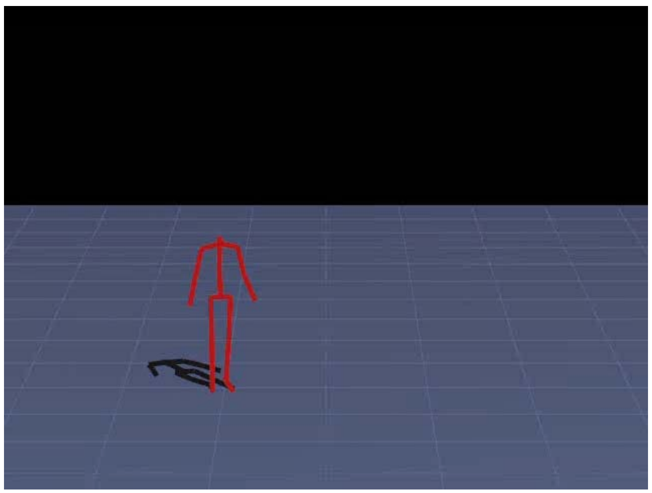


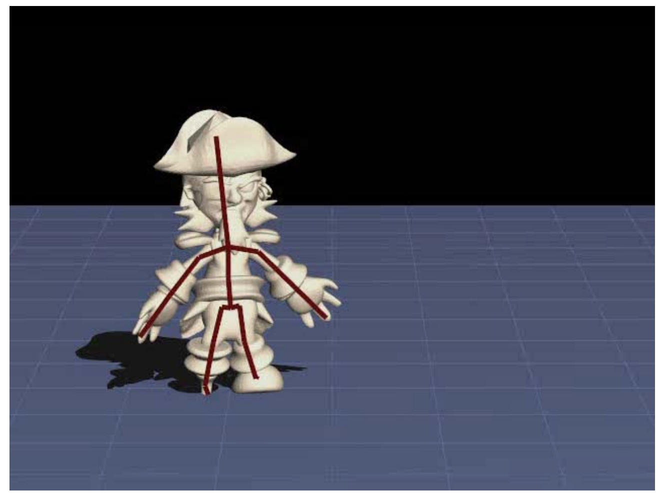
- Skin:
  - discrete samples of the surface
  - polygonal mesh



Skin + skeleton

- Skeleton defines the overall motion
- Skin moves with the skeleton

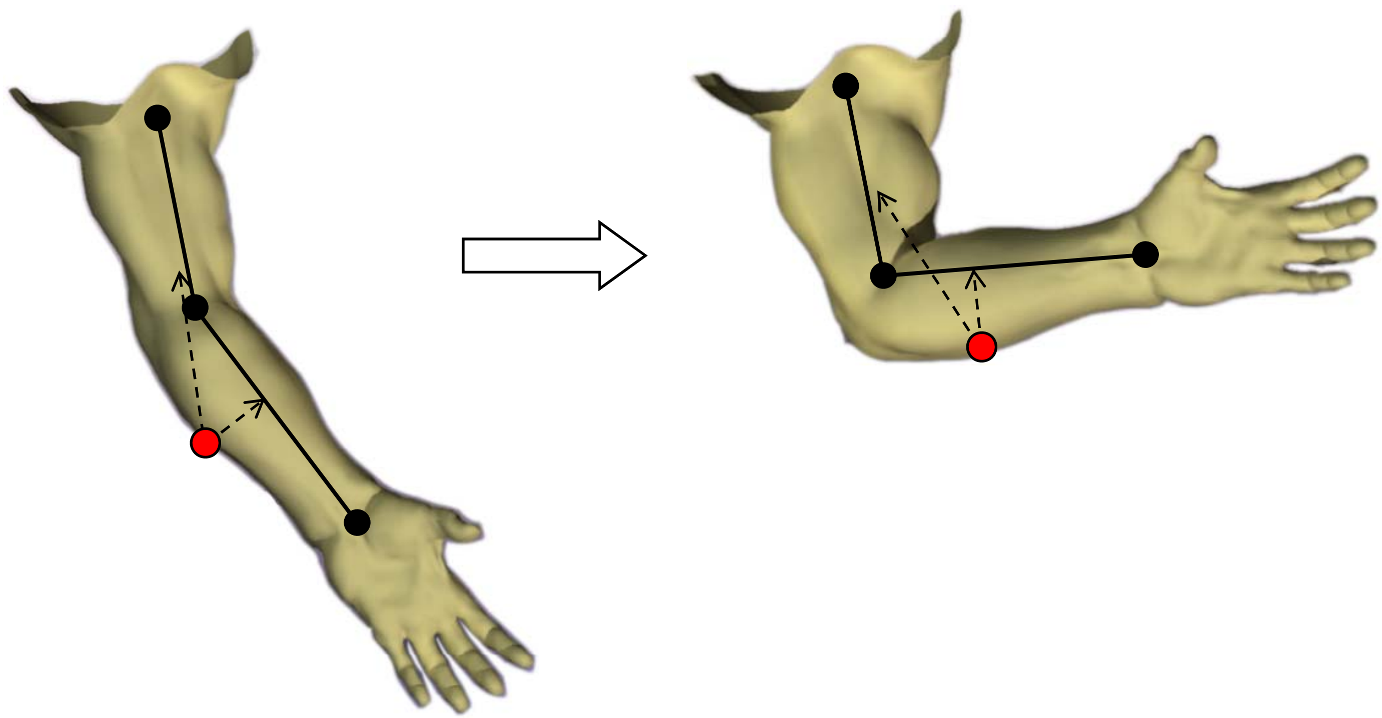




The process of building the skeleton and binding it to the skin mesh is called **rigging**.

Clips courtesy of Ilya Baran

Skeletal subspace deformation (SSD)

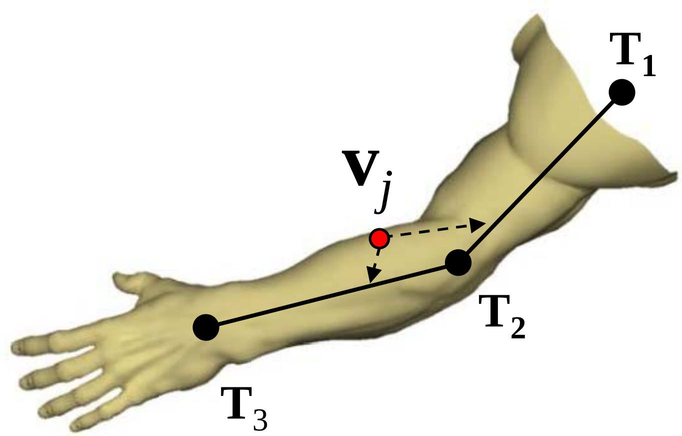


The artist needs to specify, for each point on the skin, how much it is influenced by the skeleton bones.

Skeletal subspace deformation (SSD)

- Affine combination of transformations

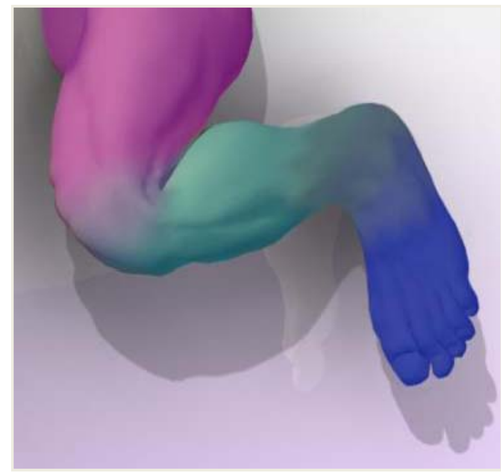
$$\mathbf{v}'_j = \sum_{k=1}^{K} w_{kj} \cdot \mathrm{T}_k \cdot \mathbf{v}_j$$

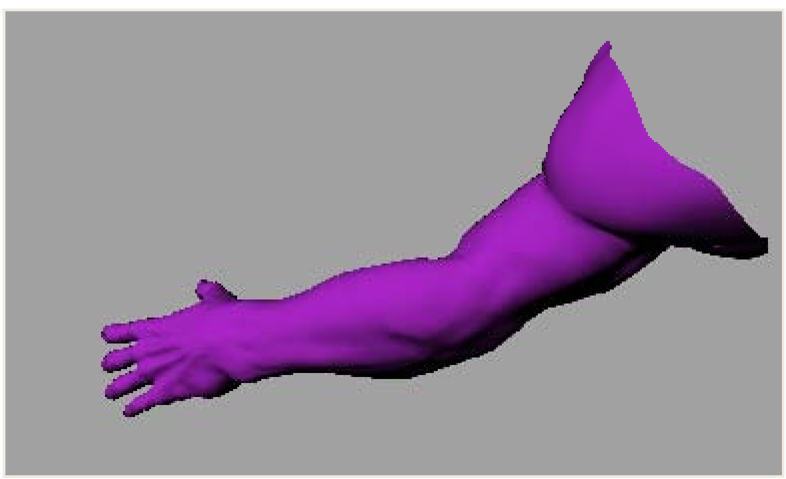


- De facto standard for interactive applications – simple + fast + works on the GPU

Skeletal subspace deformation (SSD)

- Hard to set up
- Visual artifacts
- No context



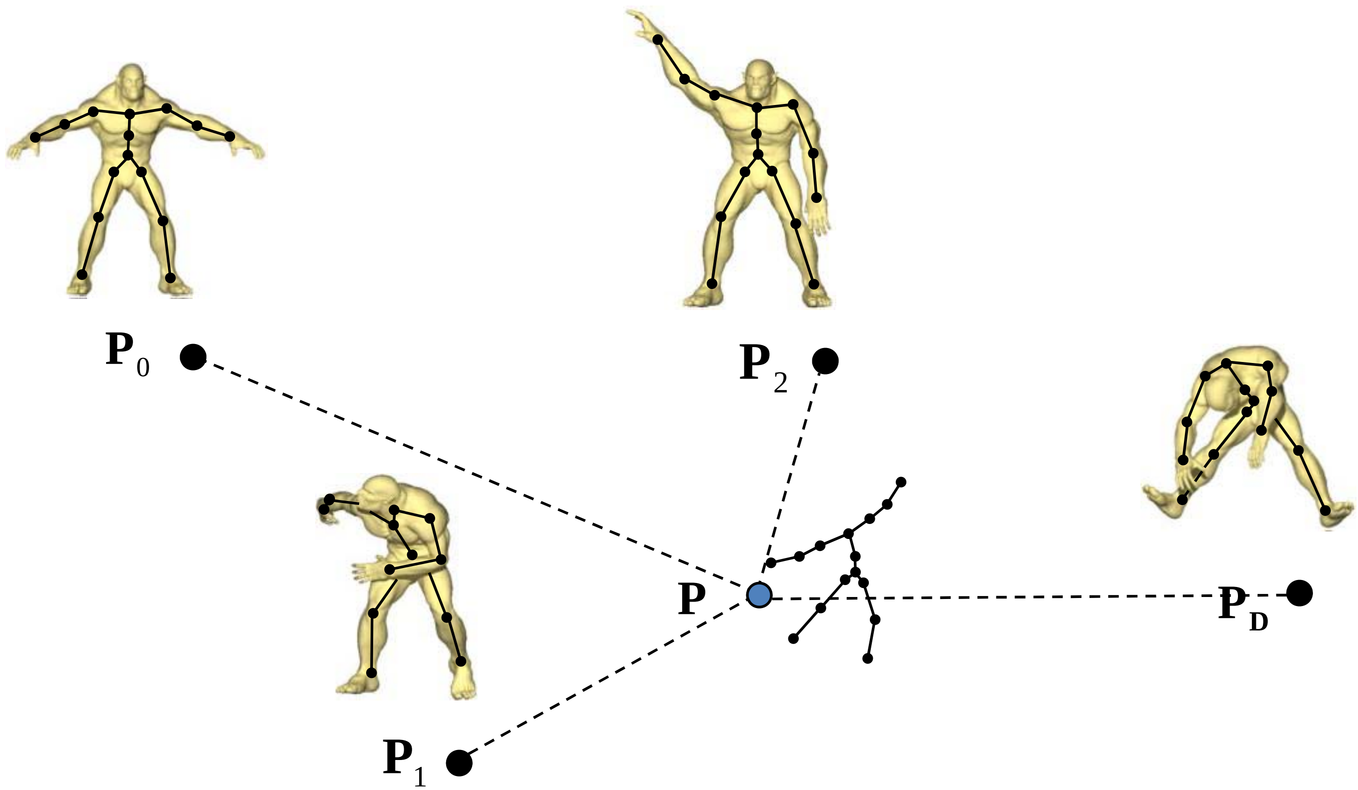


Pose space deformation (PSD)

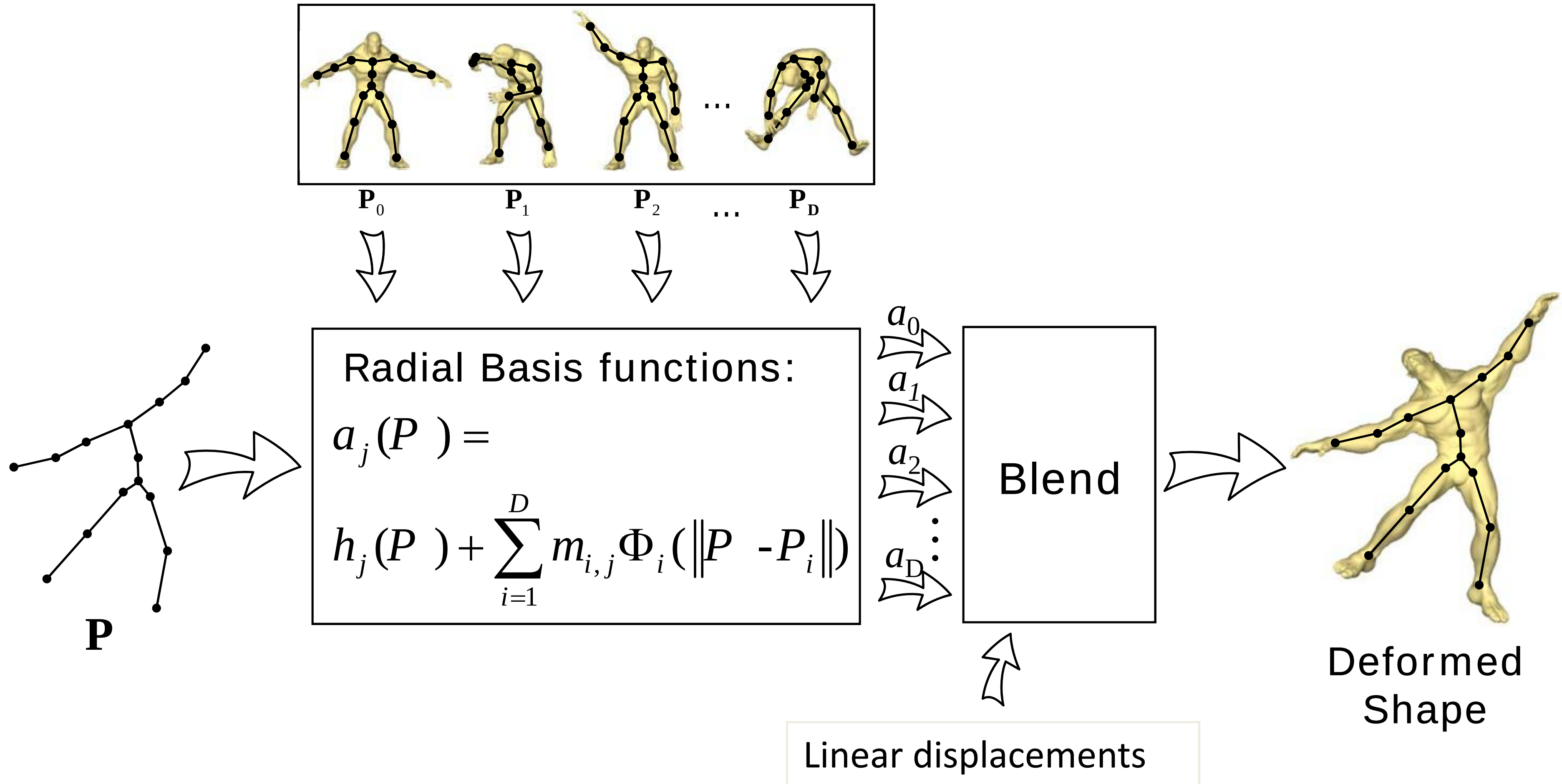
[Lewis et al. 2000, Sloan et al. 2001]

Each degree of freedom of the skeleton is a dimension:

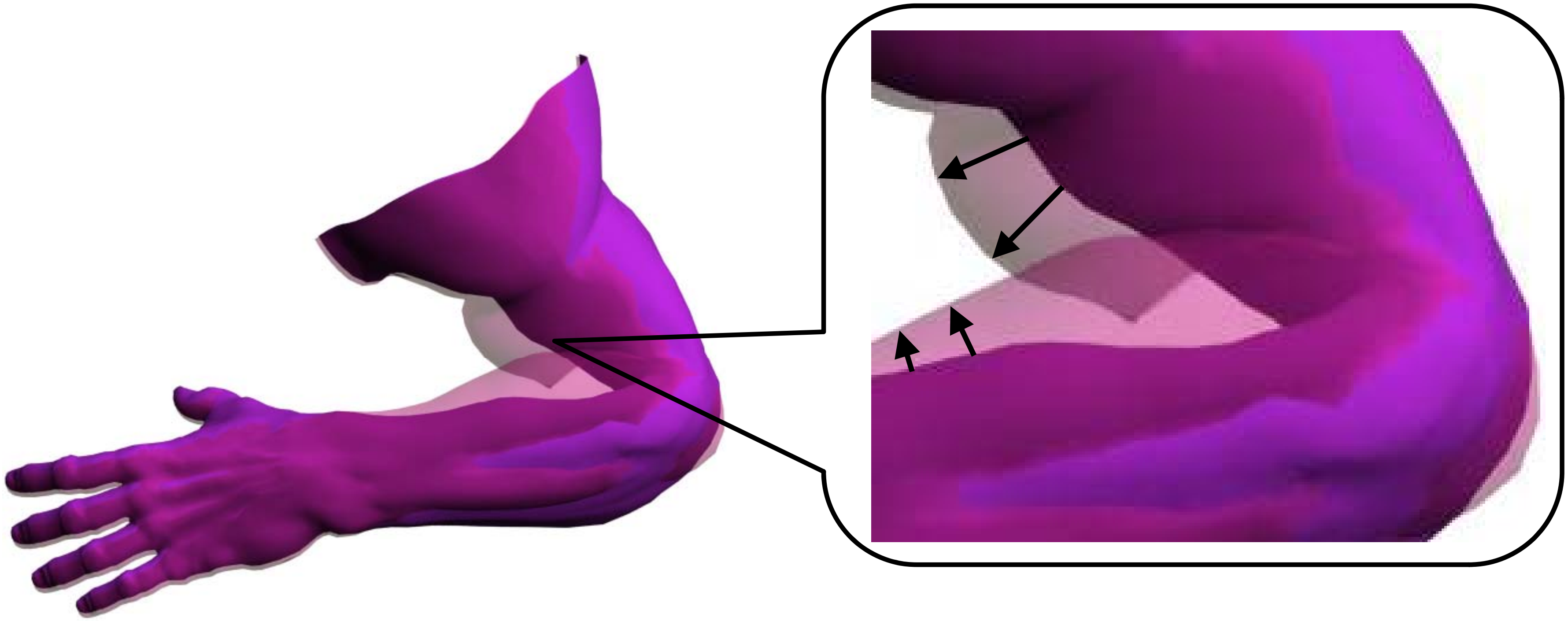
$$\mathbf{P} = (\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, \ldots, \alpha_K, \beta_K, \gamma_K)$$



Pose space deformation (PSD)

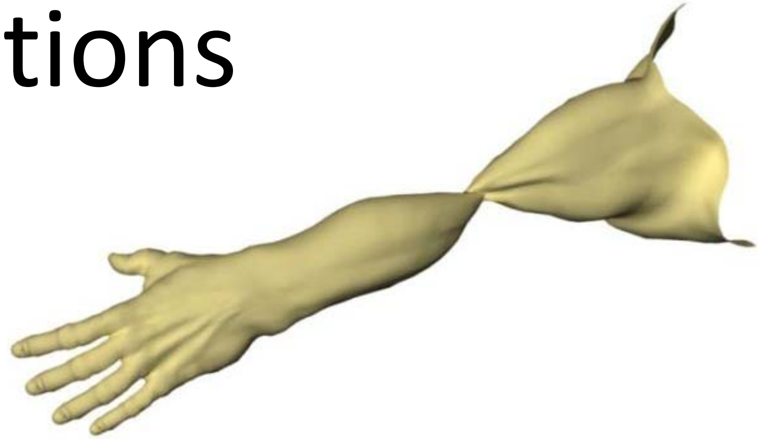


Pose space deformation (PSD)

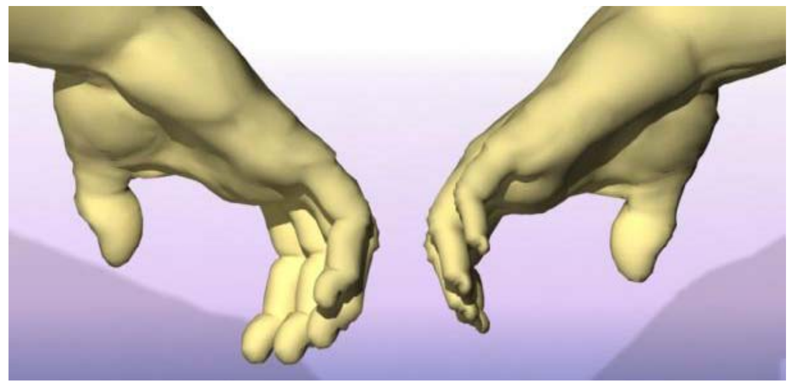


PSD limitations

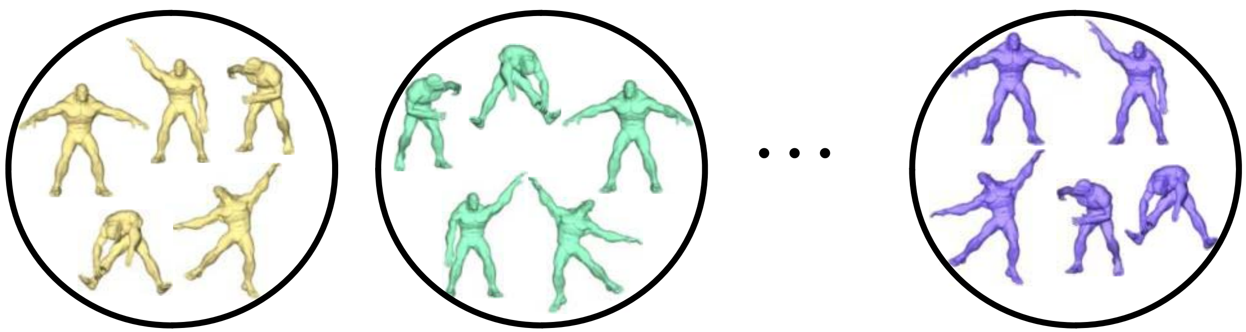
- SSD – artifacts, requires many examples + setup



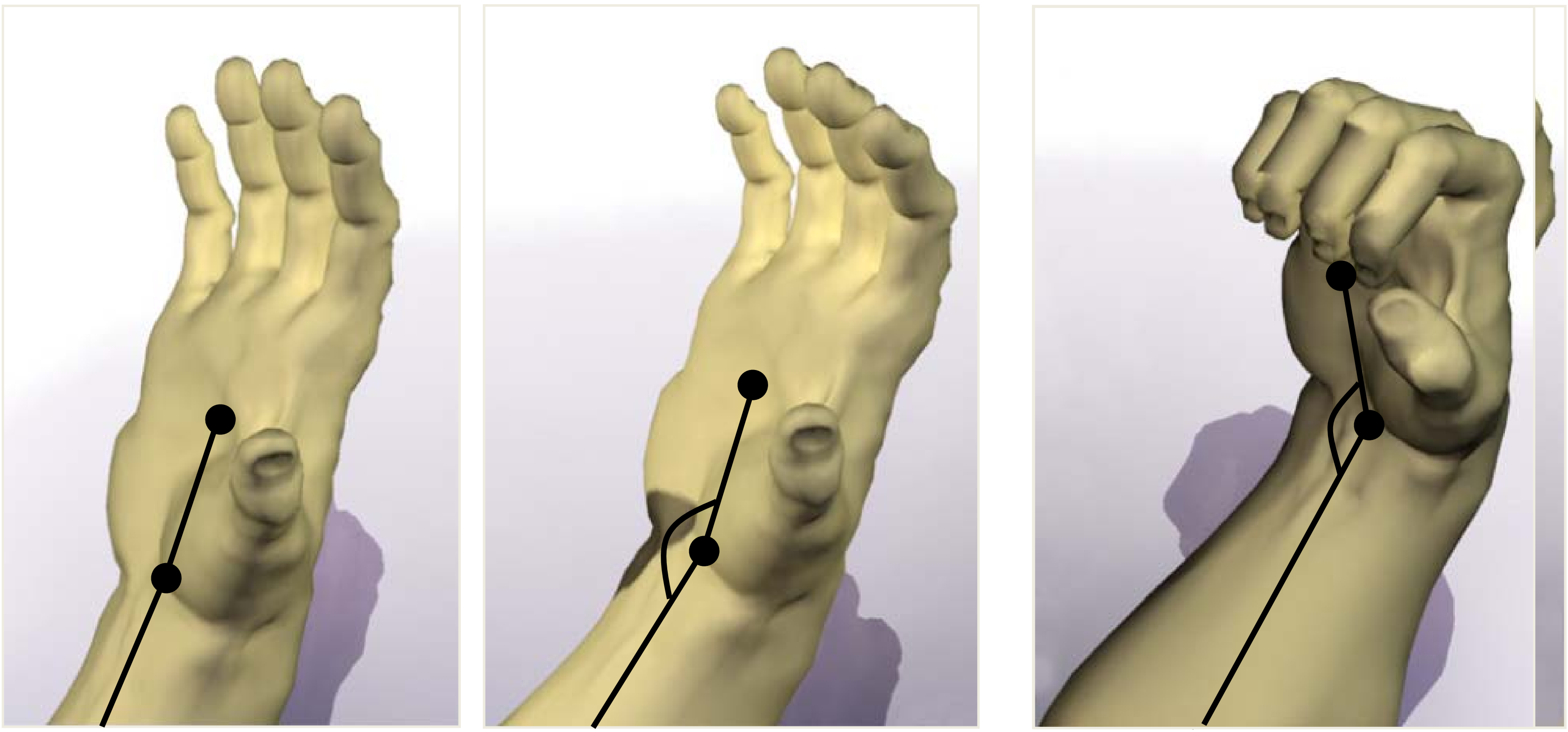
- Linear displacements – no rotation



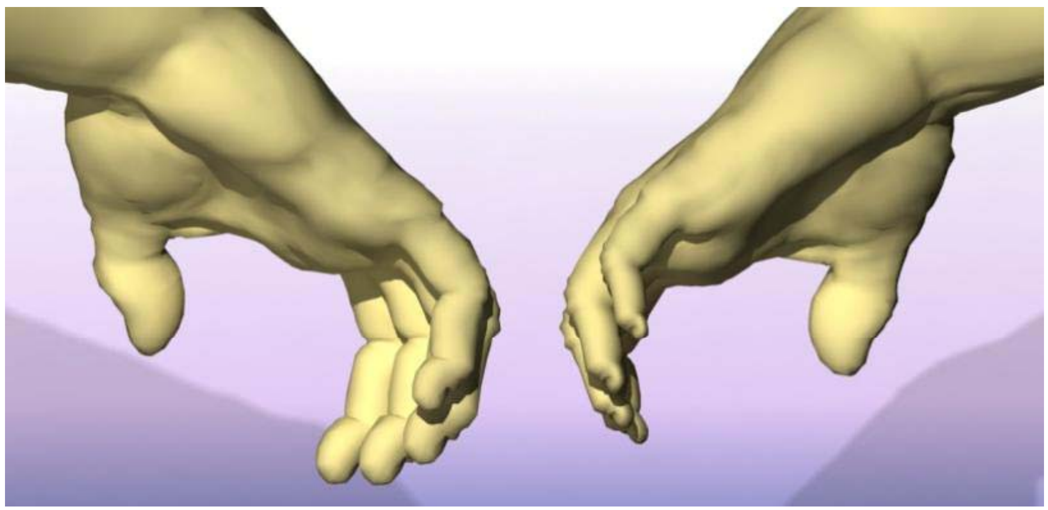
- High memory consumption, performance



Rotation interpolation and extrapolation



Linear displacements (PSD)



Context-Aware Skeletal Shape Deformation

Eurographics 2007

Ofir Weber
Olga Sorkine
Yaron Lipman
Craig Gotsman

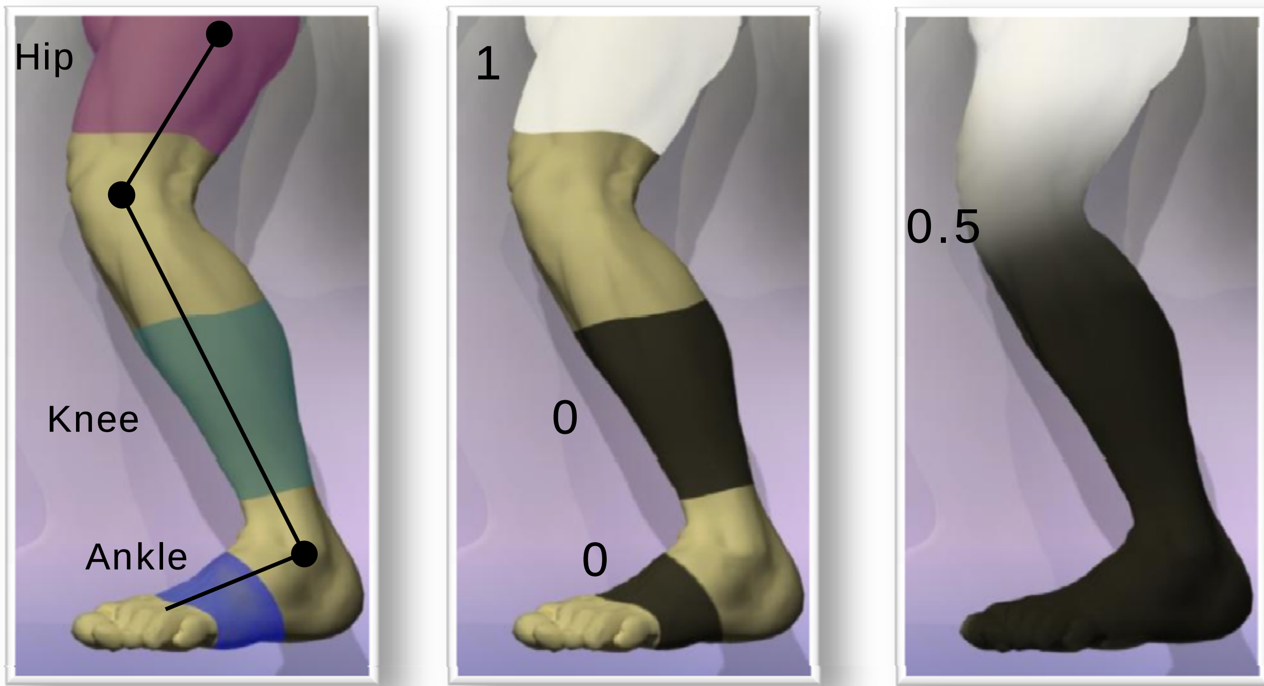
The contributions

- Replace SSD by detail-preserving mesh deformation
  [Sorkine et al. 2004, Sumner et al. 2004, Yu et al. 2004, Lipman et al. 2005, Zayer et al. 2005]
- Easy setup
- Differential morphing
- Sparse representation of example shapes

Other previous work

- Pose Space Deformation [Lewis et al. 2000, Sloan et al. 2001, Kry et al. 2002, Kurihara et al. 2004, Rhee et al. 2006]
- Detail-preserving mesh deformation [Sorkine et al. 2004, Sumner et al. 2004, Yu et al. 2004, Lipman et al. 2005, Zayer et al. 2005…]
  Survey: [Botsch and Sorkine 2008]
- MeshIK [Sumner et al. 2005, Der et al. 2006]
- SCAPE [Anguelov et al. 2005]

Detail-preserving deformation



$$\Delta \mathbf{w}_k = 0$$

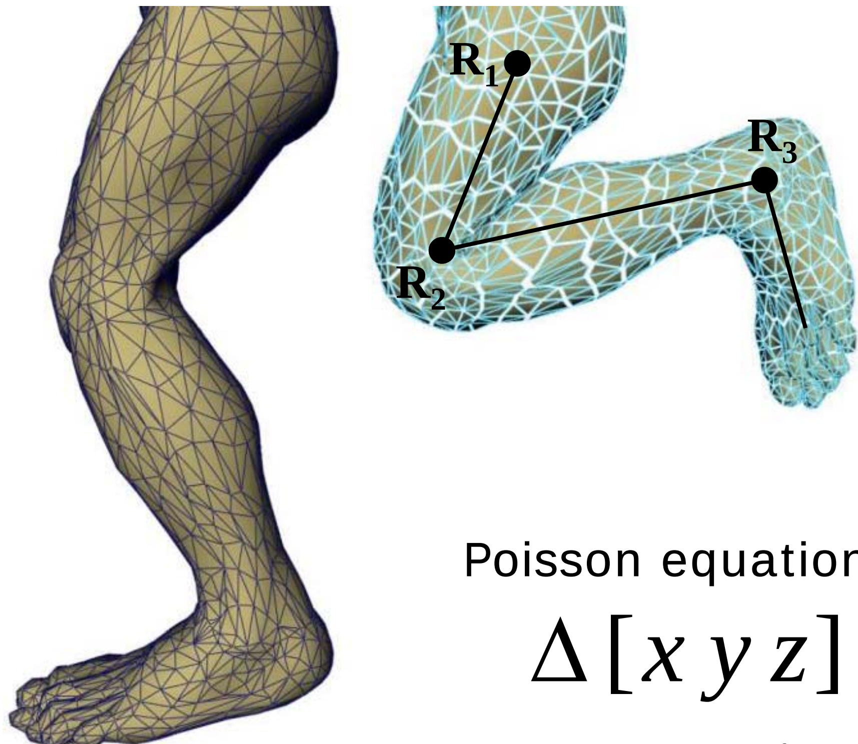
Dirichlet boundary conditions:
$\mathbf{w}_k(t_n) = 1$ for $t_n \in H_k$
$\mathbf{w}_k(t_n) = 0$ for $t_n \in H_l$ where $l \neq k$.

Blending rotations

For each face t:

$$\mathbf{R}(t) = \mathbf{w}_1(t)\mathbf{R}_1 \oplus \mathbf{w}_2(t)\mathbf{R}_2 \oplus \ldots \oplus \mathbf{w}_K(t)\mathbf{R}_K$$

$\oplus$: [Buss 93]
log-quaternion

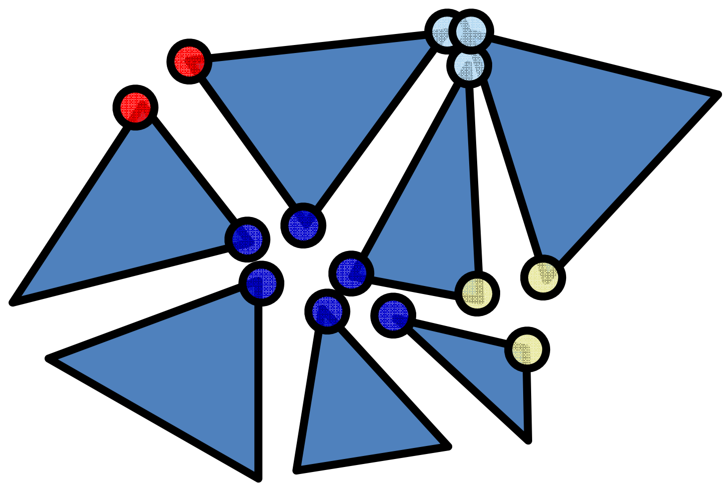


Poisson equation [Yu et al. 2004]

$$\Delta[x\,y\,z] = \mathrm{div}[\mathbf{R}]$$

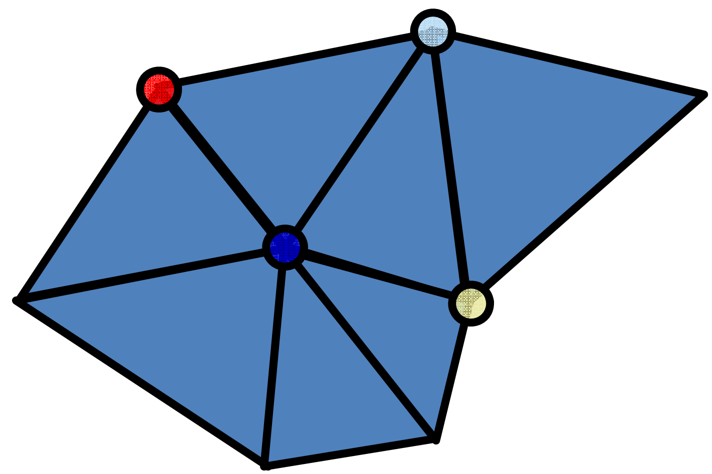
Sparse linear system

Poisson stitching



- The Poisson equation averages the different vertex positions
- Tries to preserve the shape and orientation of the triangles as much as possible

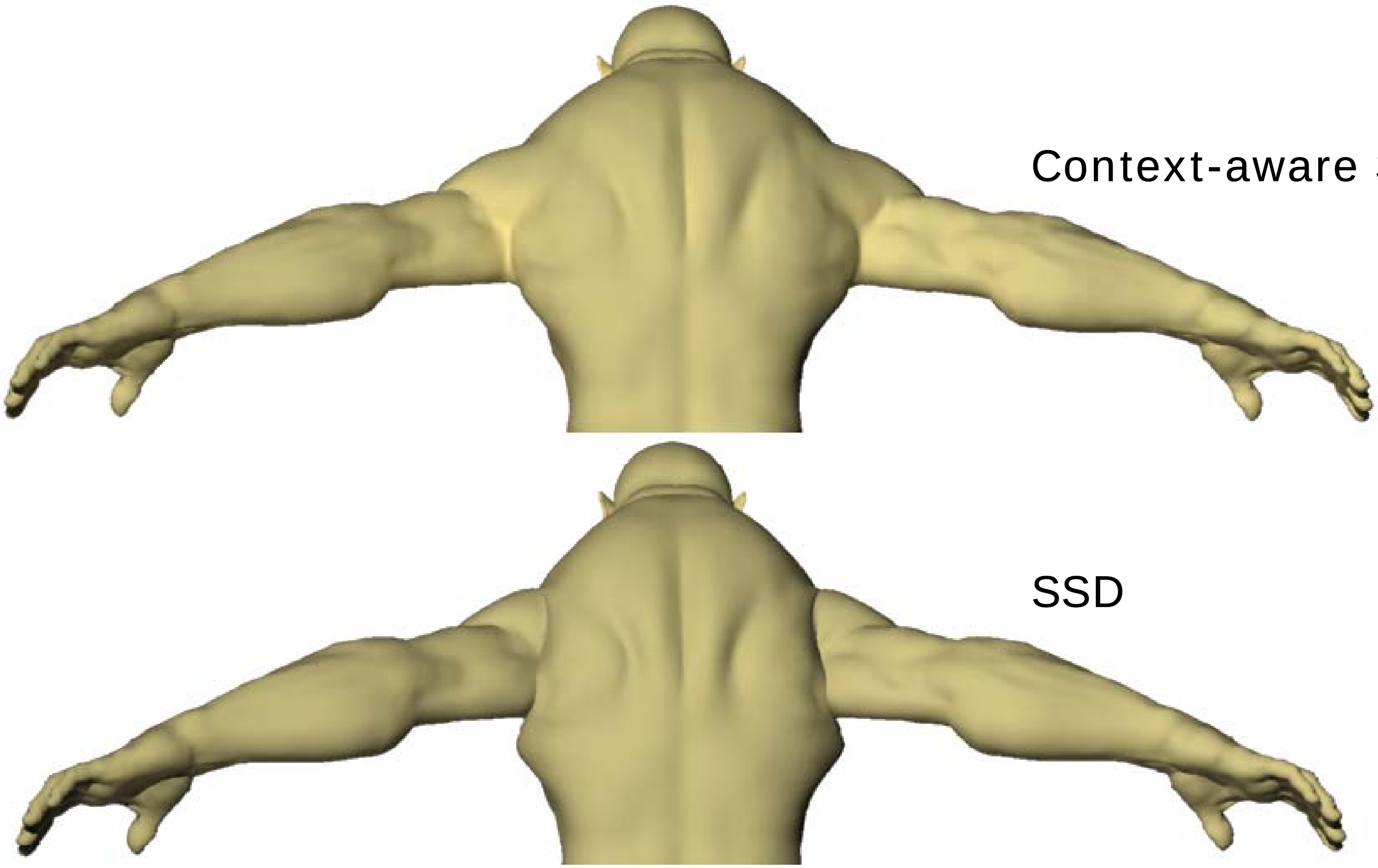
Poisson stitching



- The Poisson equation averages the different vertex positions
- Tries to preserve the shape and orientation of the triangles as much as possible

Setup

Comparison to SSD



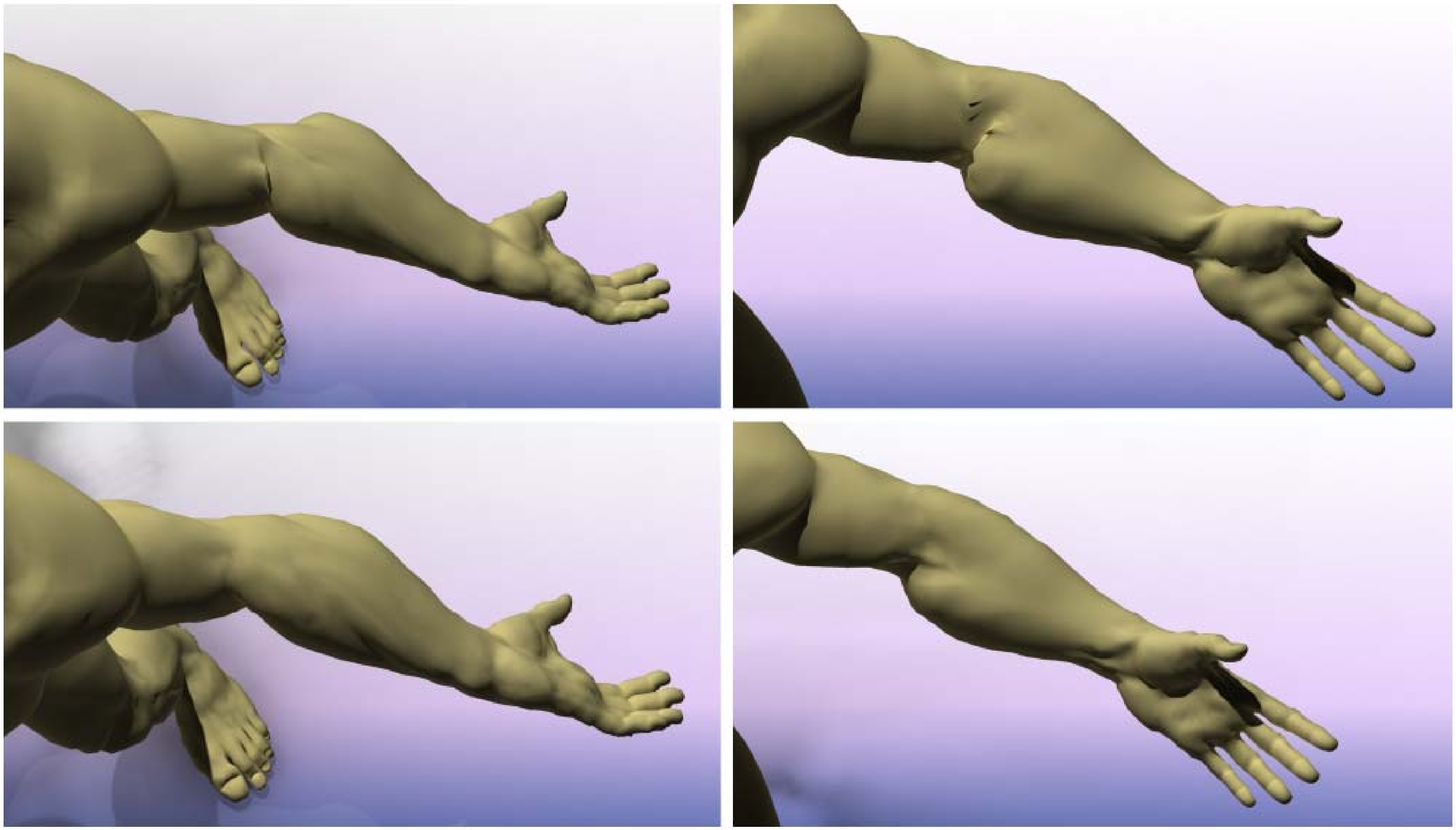
Context-aware SSD

SSD

Comparison to SSD

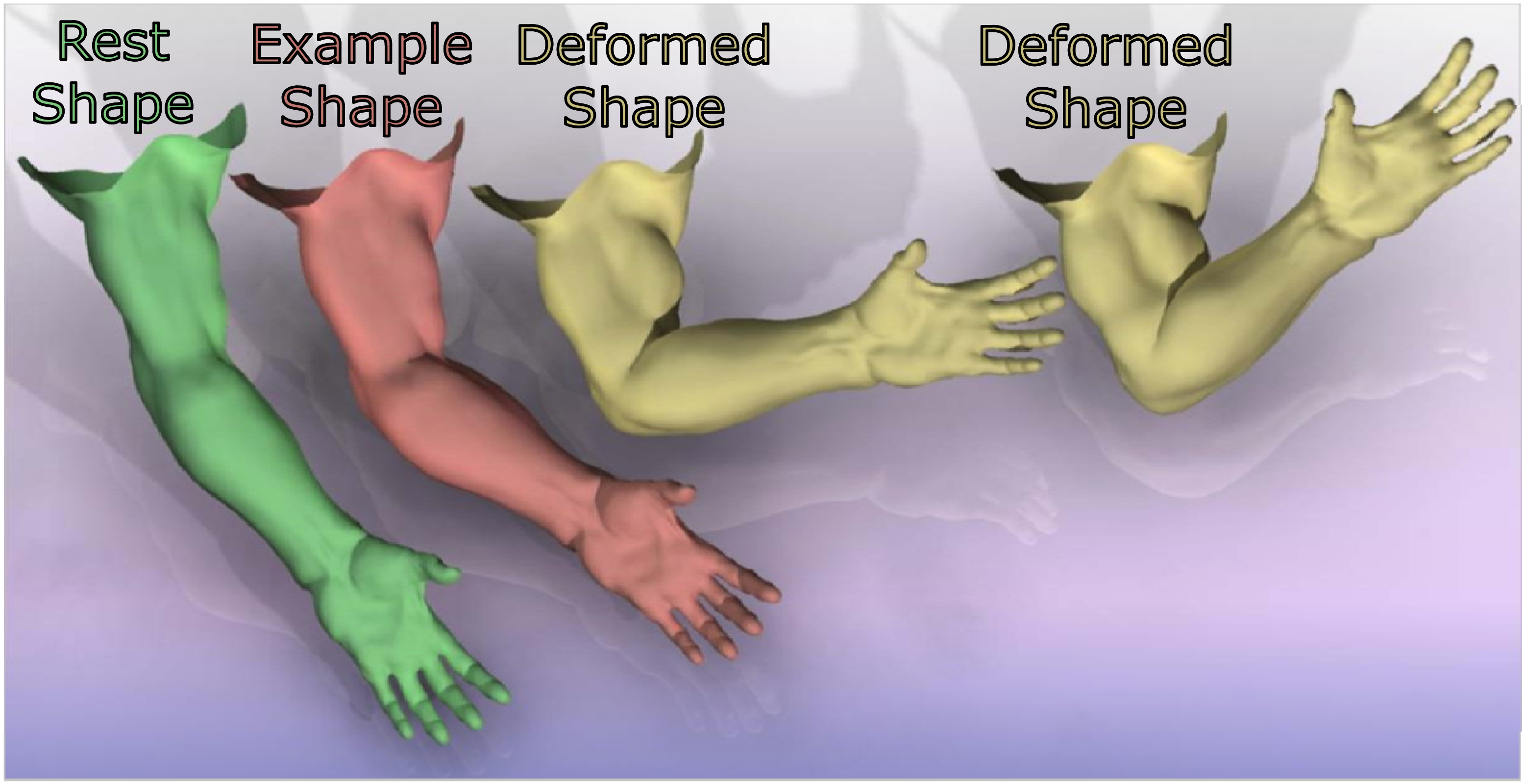
SSD

CASSD

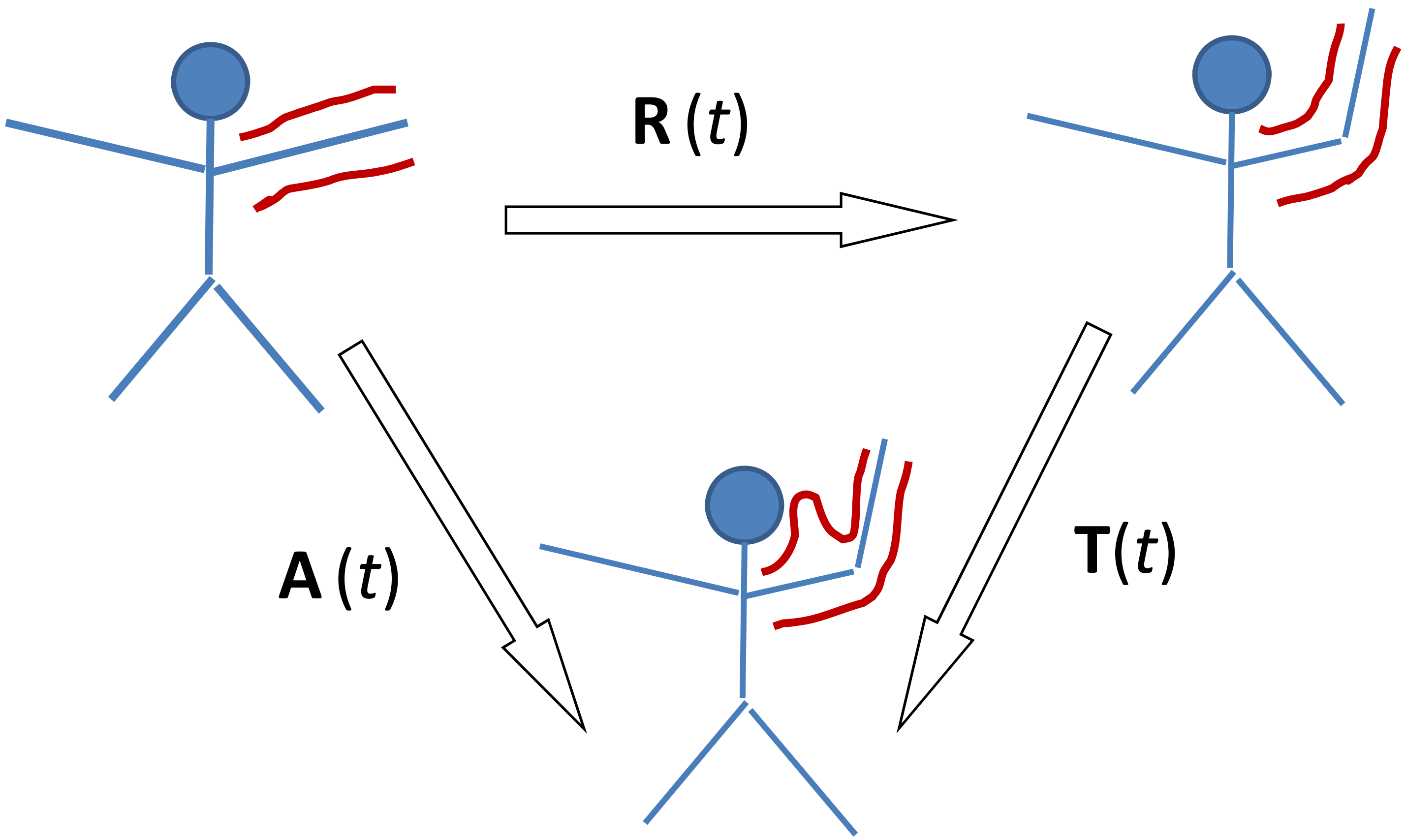


Video: 0:0:18

Using context – examples

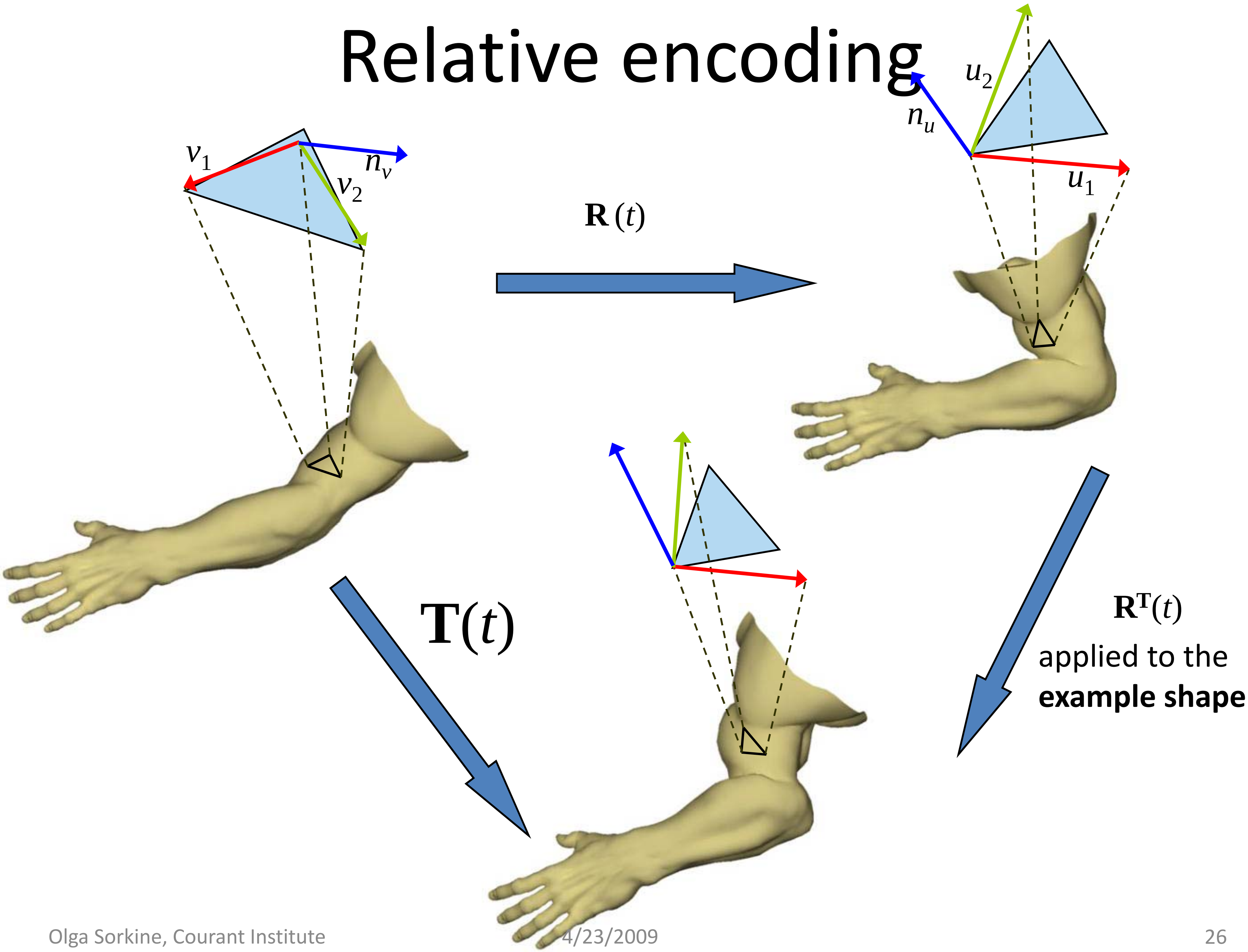


Relative encoding

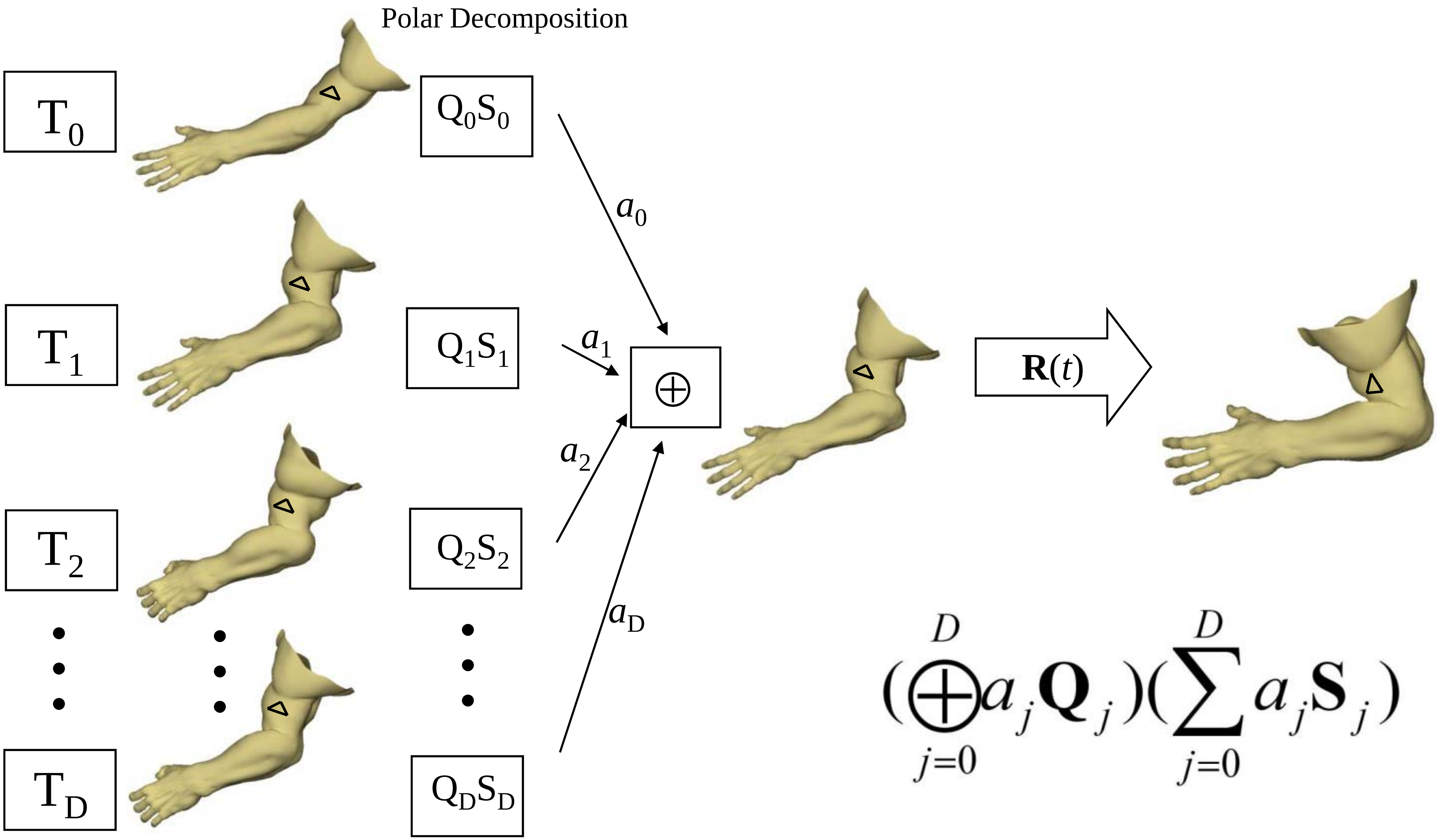


$$\mathbf{A}(t) = \mathbf{T}(t)\,\mathbf{R}(t)$$
$$\mathbf{T}(t) = \mathbf{A}(t)\,\mathbf{R}^{\mathrm{T}}(t)$$

Relative encoding

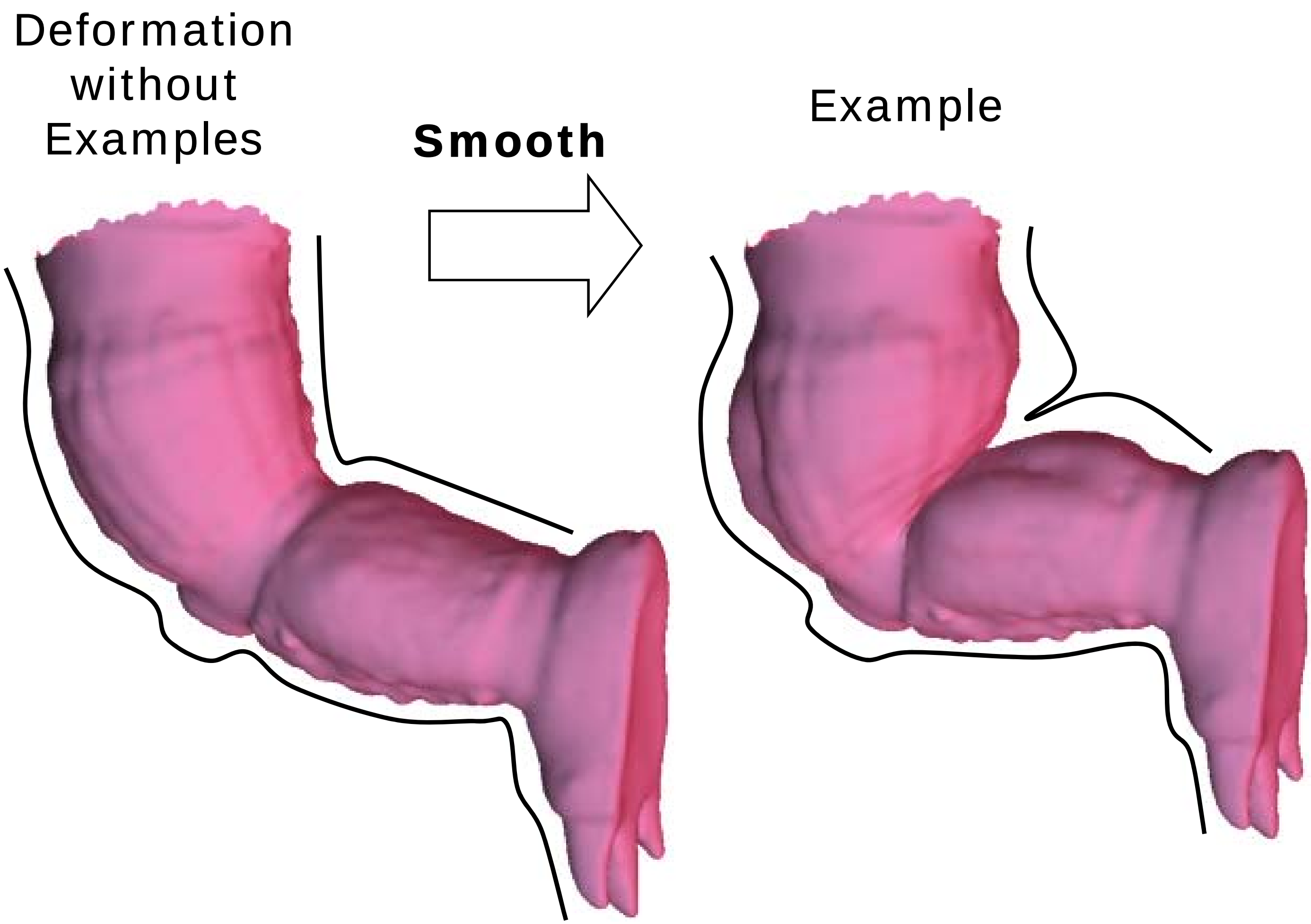


Blending transformations

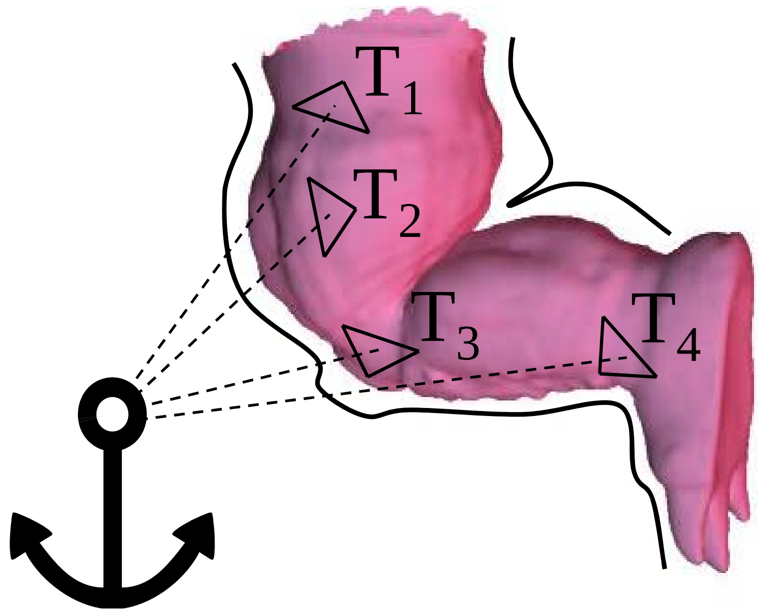


$$\left(\bigoplus_{j=0}^{D} a_j \mathbf{Q}_j\right)\left(\sum_{j=0}^{D} a_j \mathbf{S}_j\right)$$

Smooth Difference



Compact Representation



$$\Delta T = 0$$

Boundary conditions:
known T's at anchors

- Transformations varies smoothly
- Laplace equation
- Less than 5% memory
- Evaluation only at anchors – performance
- Greedy selection

See Least-squares Meshes
[Sorkine and Cohen-Or 2004]

One more result…

Video: 3:55

Conclusions

- Detail-preserving skeletal shape deformation
- Easy setup
- No or small number of examples
- Interpolation and meaningful extrapolation
- Sparse representation of examples

Limitations and extensions

- No dynamics
- The greedy algorithm is not optimal
- Map to GPU $\rightarrow$ Wang et al. SIGGRAPH 2007