COL781: Computer Graphics

30. Skinning

Ziva Dynamics

Announcements

Assignment 3 due at midnight tonight!

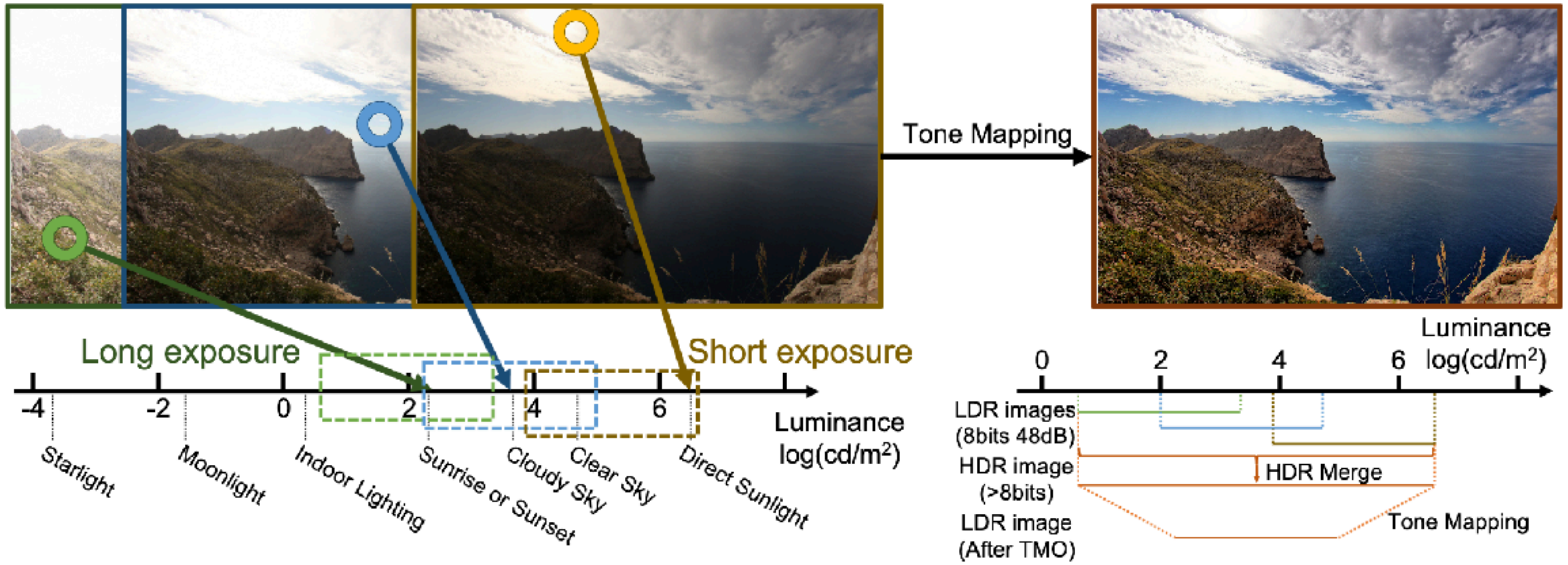
Assignment 4 is out: keyframing and simulation

Next semester: COL829 *Advanced Computer Graphics*
Potential topics:

- Modeling: mesh processing, surface reconstruction, level of detail, …
- Rendering: volume rendering, radiance fields, real-time global illumination, …
- Animation: character control, continuum mechanics, model reduction, …

Output of physics-based rendering is radiance, which can span many orders of magnitude!

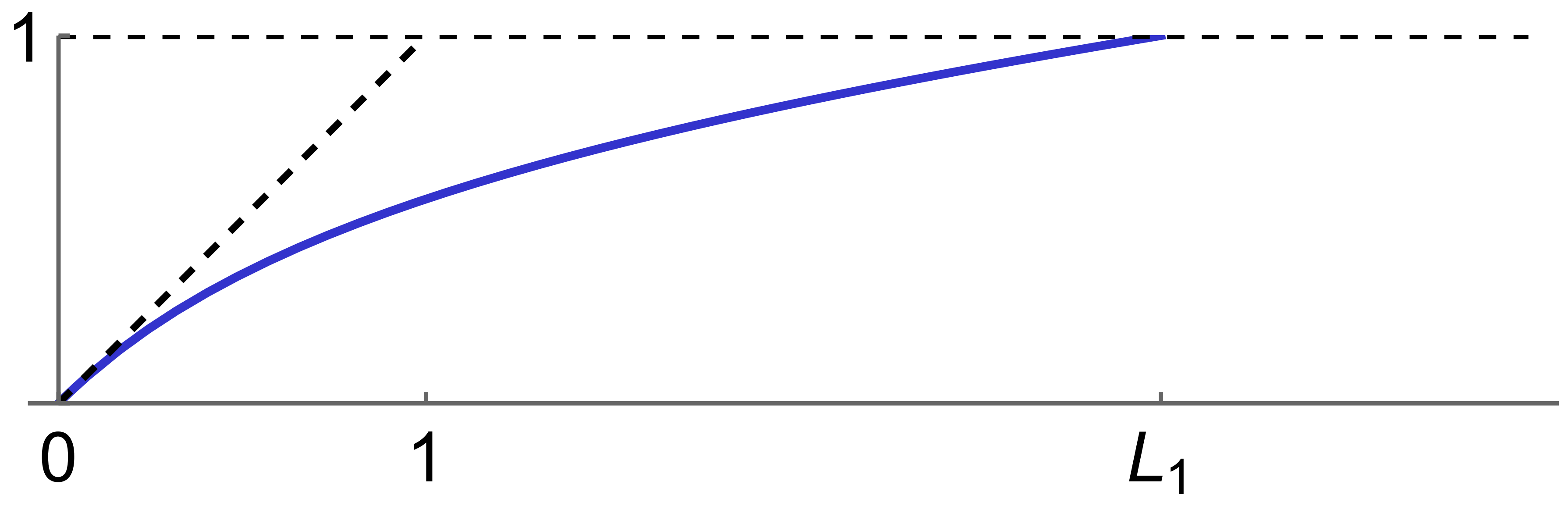
Displayable RGB values are in [0, 1]



Ou et al. 2020

Scale up/down by some (manually specified) exposure value. Then? Tone mapping

- Clamping: $L(x,y) \to \min(L(x,y), 1)$
- Reinhard operator: $L(x,y) \to \dfrac{L(x,y)(1 + L(x,y)/L_1^2)}{1 + L(x,y)}$

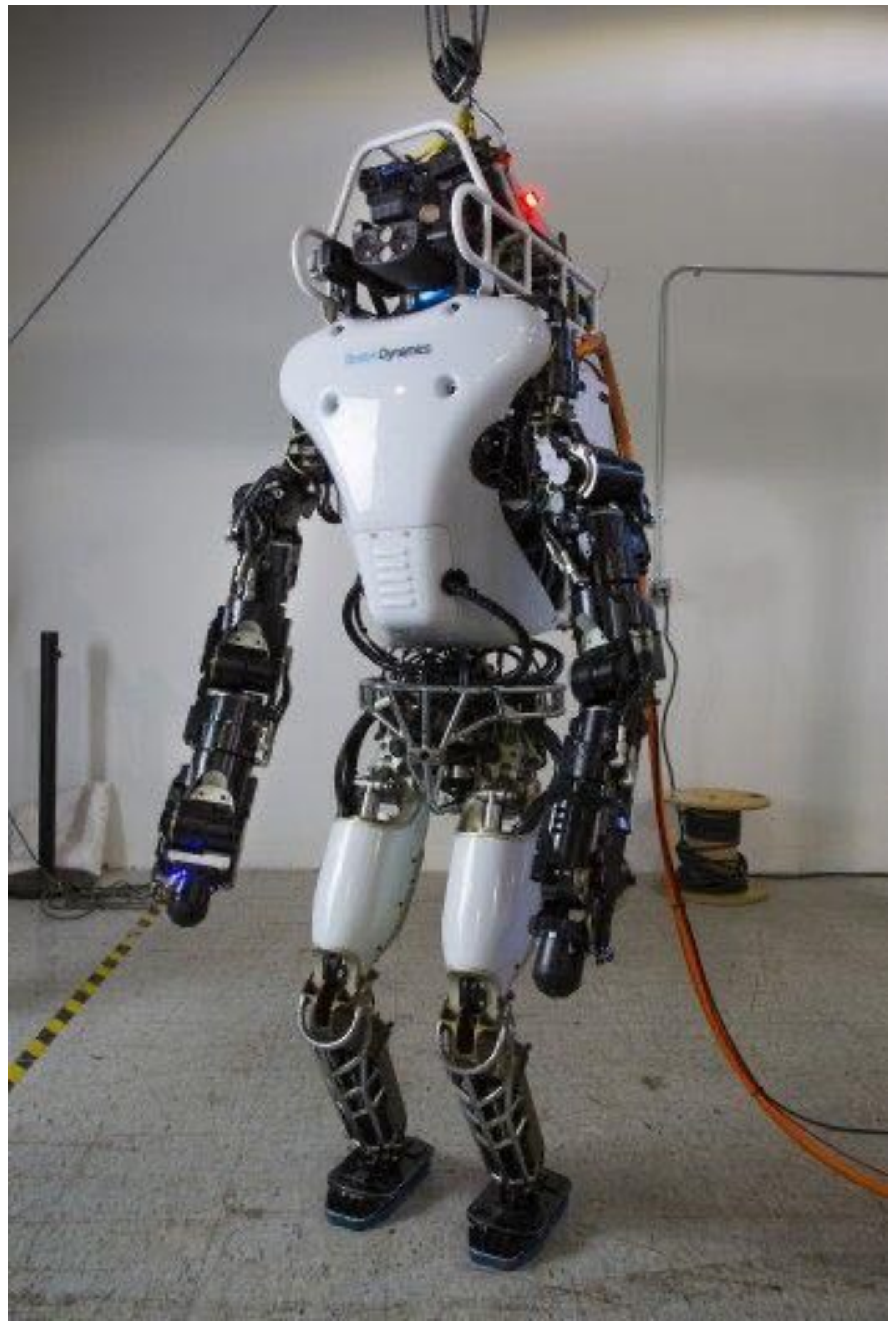


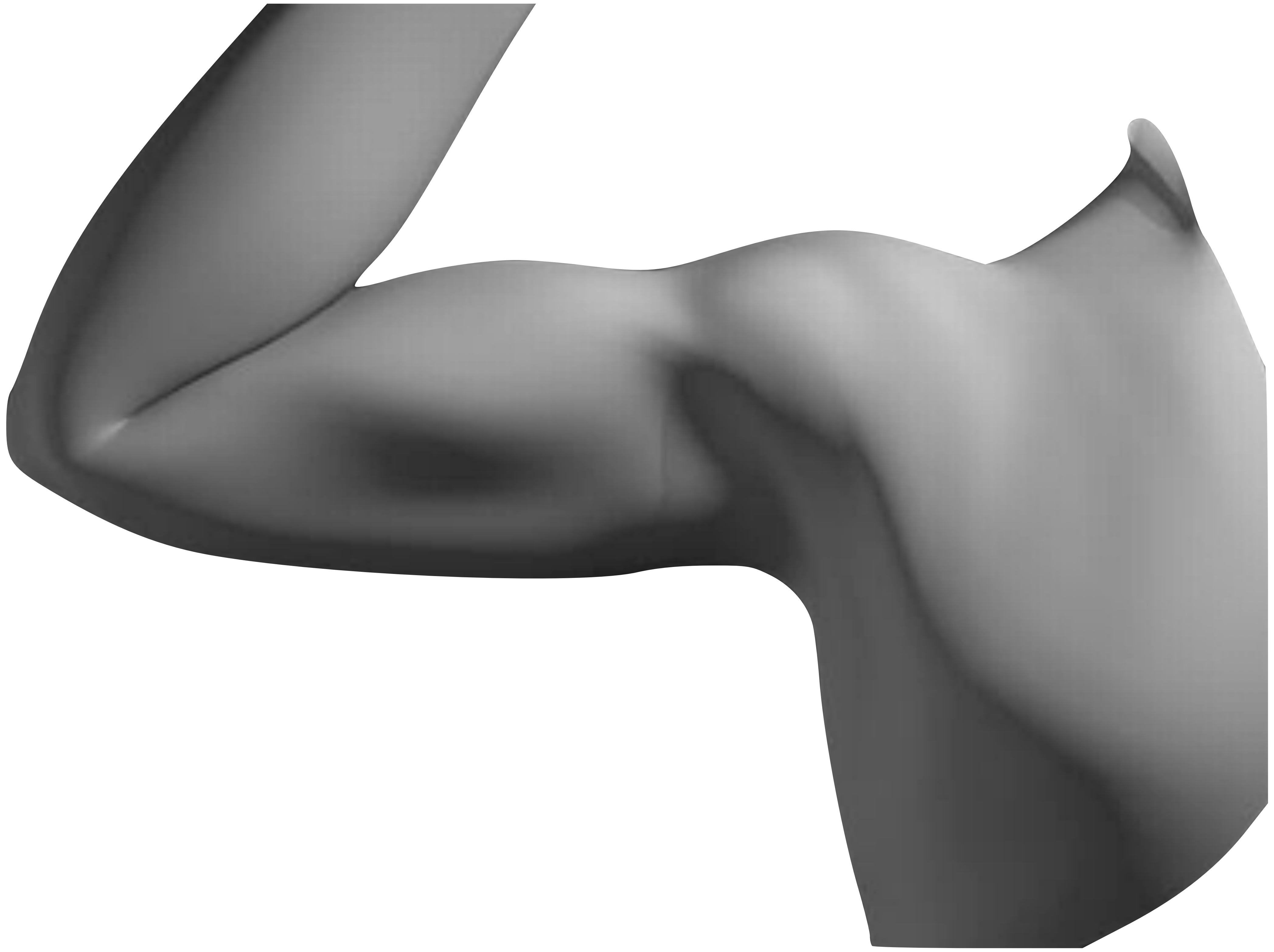
- Various other choices to control the "look"…

Highlights from Assignment 1

Rigid bone transformations may be sufficient for robots and toys with rigid parts.

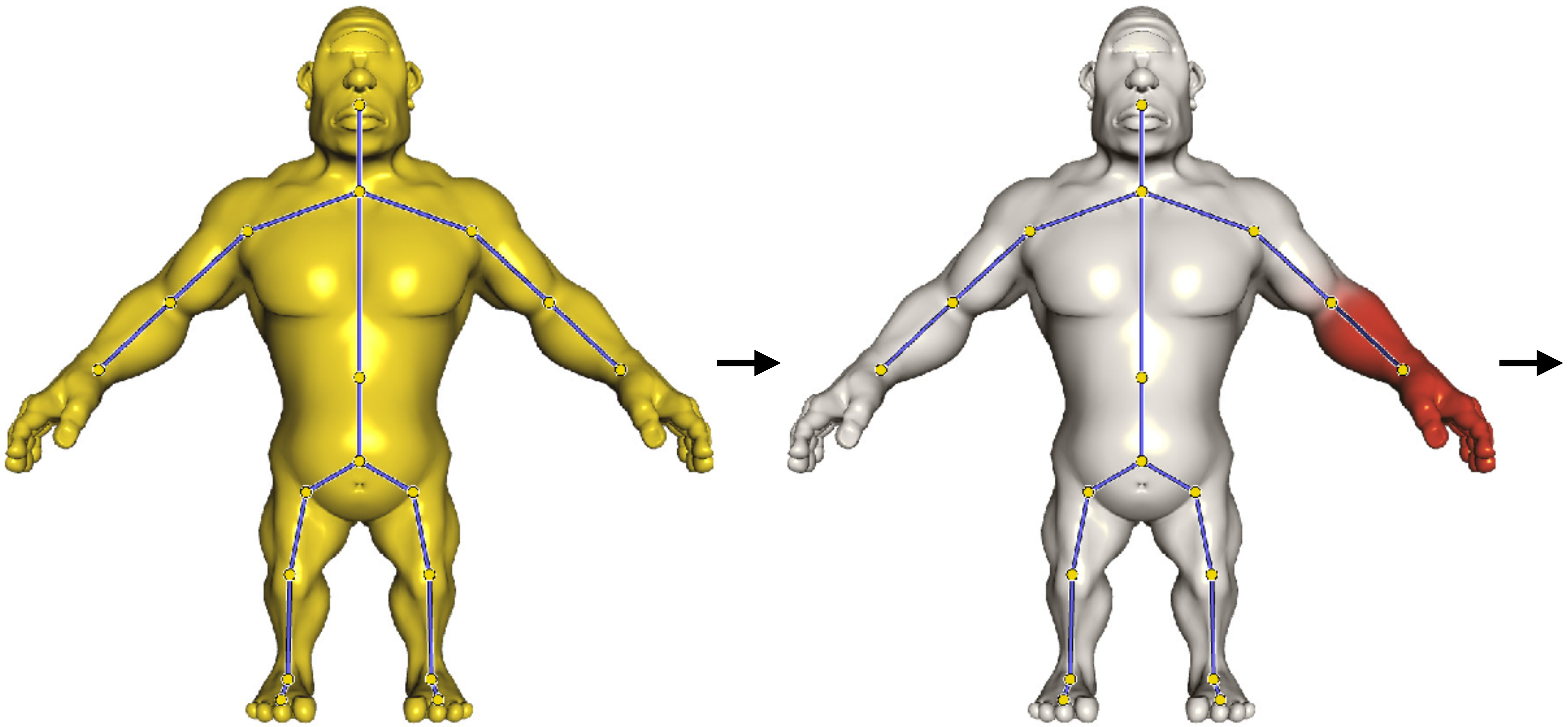
What about organic characters?





J.P. Lewis

Skinning



Skeleton

Skinning weights

Deformed shape

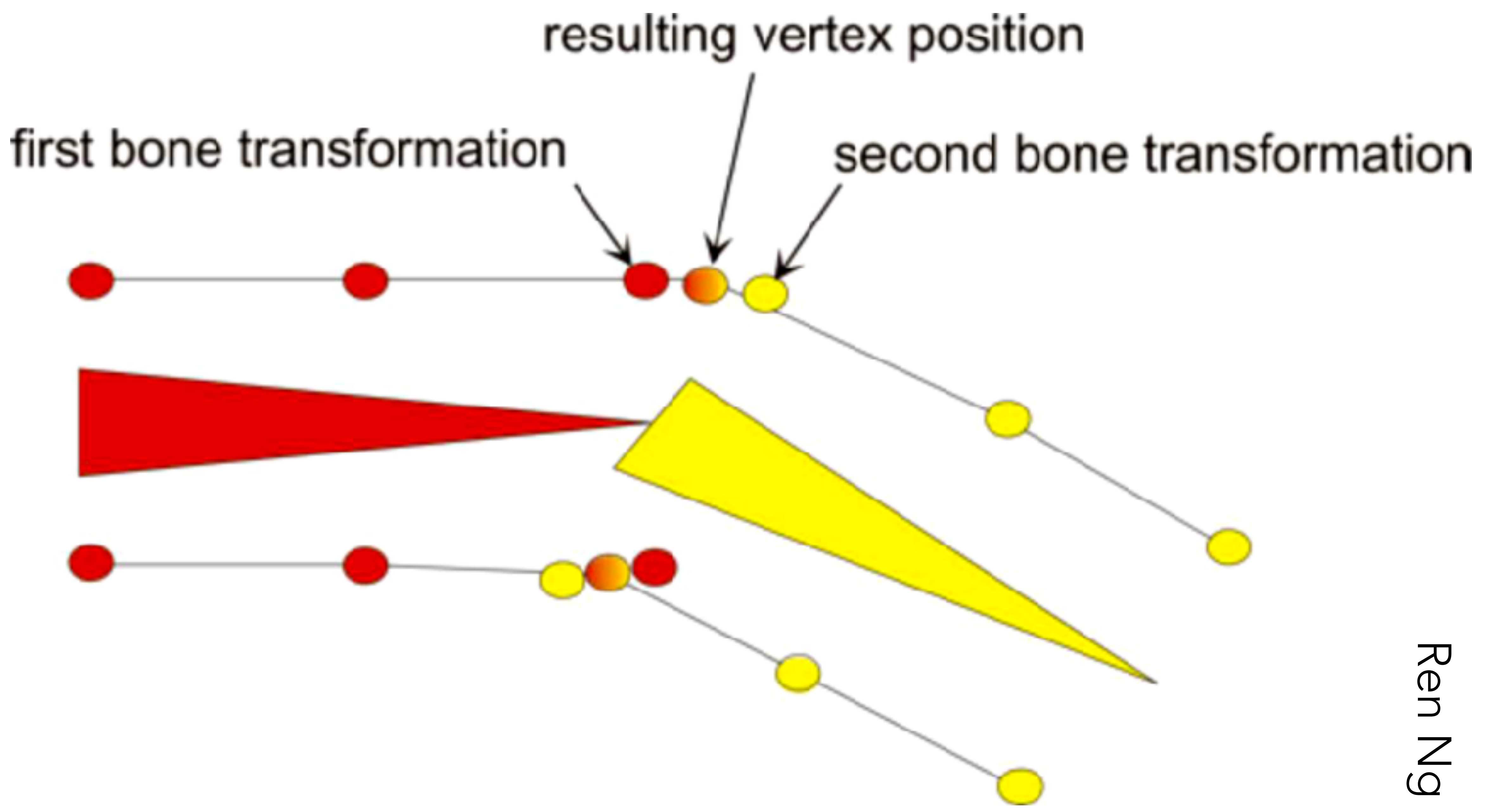
Alec Jacobson

Each vertex **v** may be affected by transformations from multiple bones.

Linear blend skinning: Final position is weighted average

$$\mathbf{v}' = \sum_{\text{bone } i} w_i \mathbf{T}_i \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$$

Of course, for a weighted average, we should have $\sum_i w_i = 1$ for each vertex



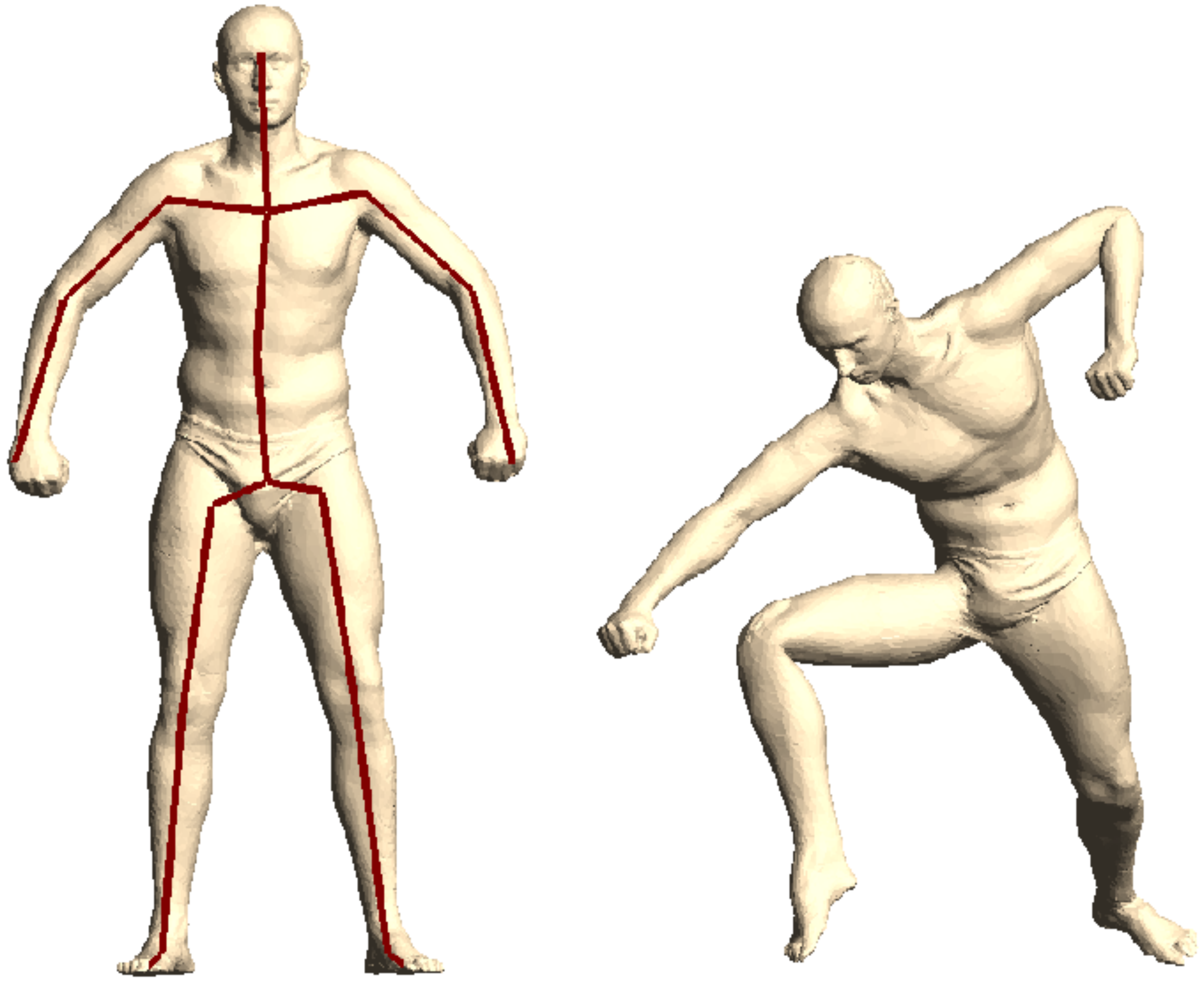
Ren Ng

Wait, to apply bone transformation $\mathbf{T}_i$, we need to have vertex $\mathbf{v}$ in the bone's coordinate frame…

$$\mathbf{v}' = \sum_{\text{bone } i} w_i \mathbf{T}_i \mathbf{B}_i^{-1} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$$

- $\mathbf{B}_i$: bone transformation in bind pose
- $\mathbf{T}_i$: bone transformation in deformed pose

From now on, we'll just call the product $\mathbf{T}_i$

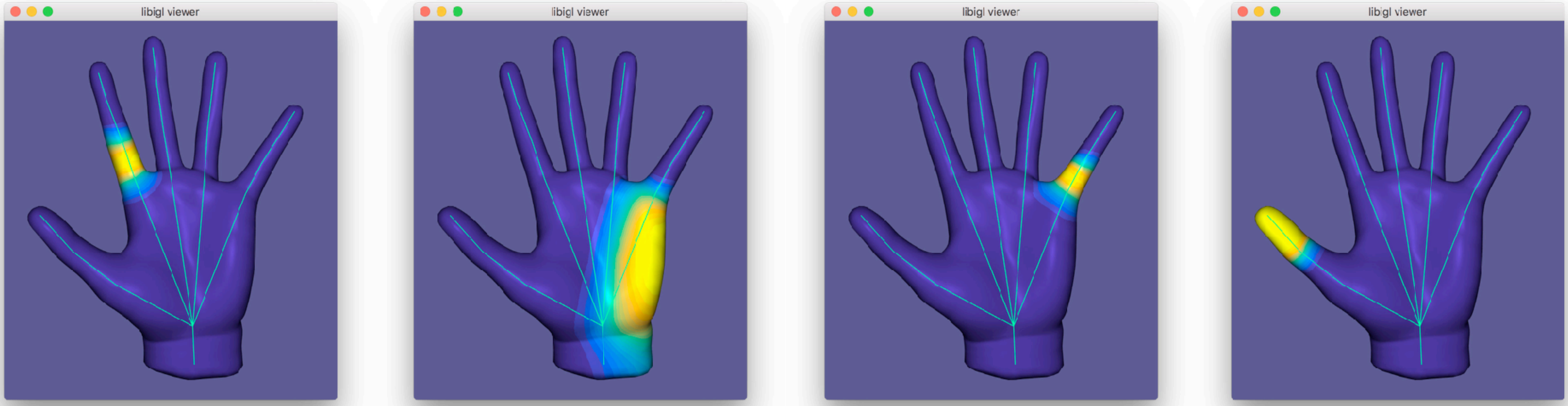


Bind pose

Deformed pose

Baran & Popovic 2007

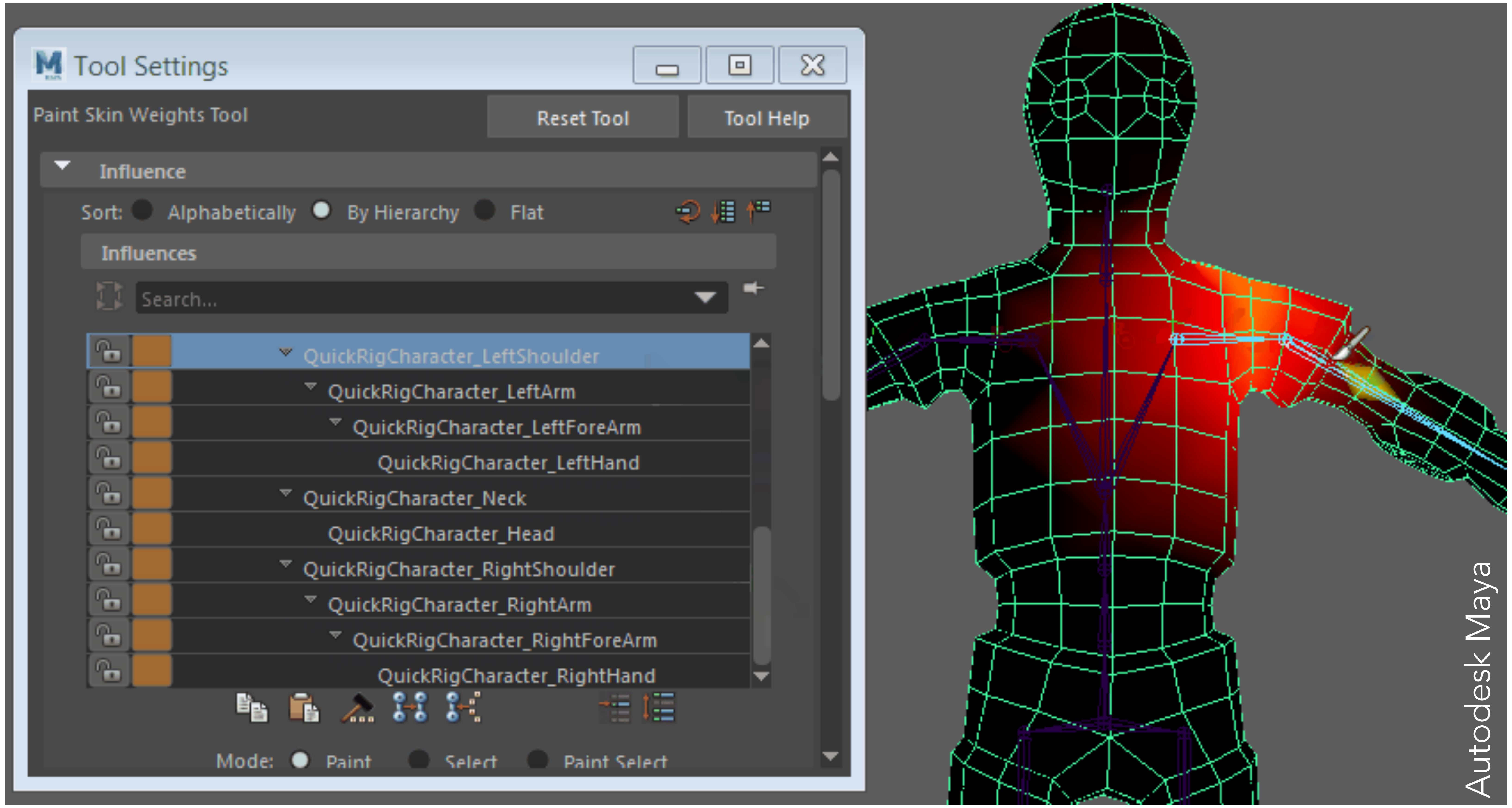
Example: Skinning weights



libigl tutorial

How to get the weights?

One common way: User paints them manually on the mesh!

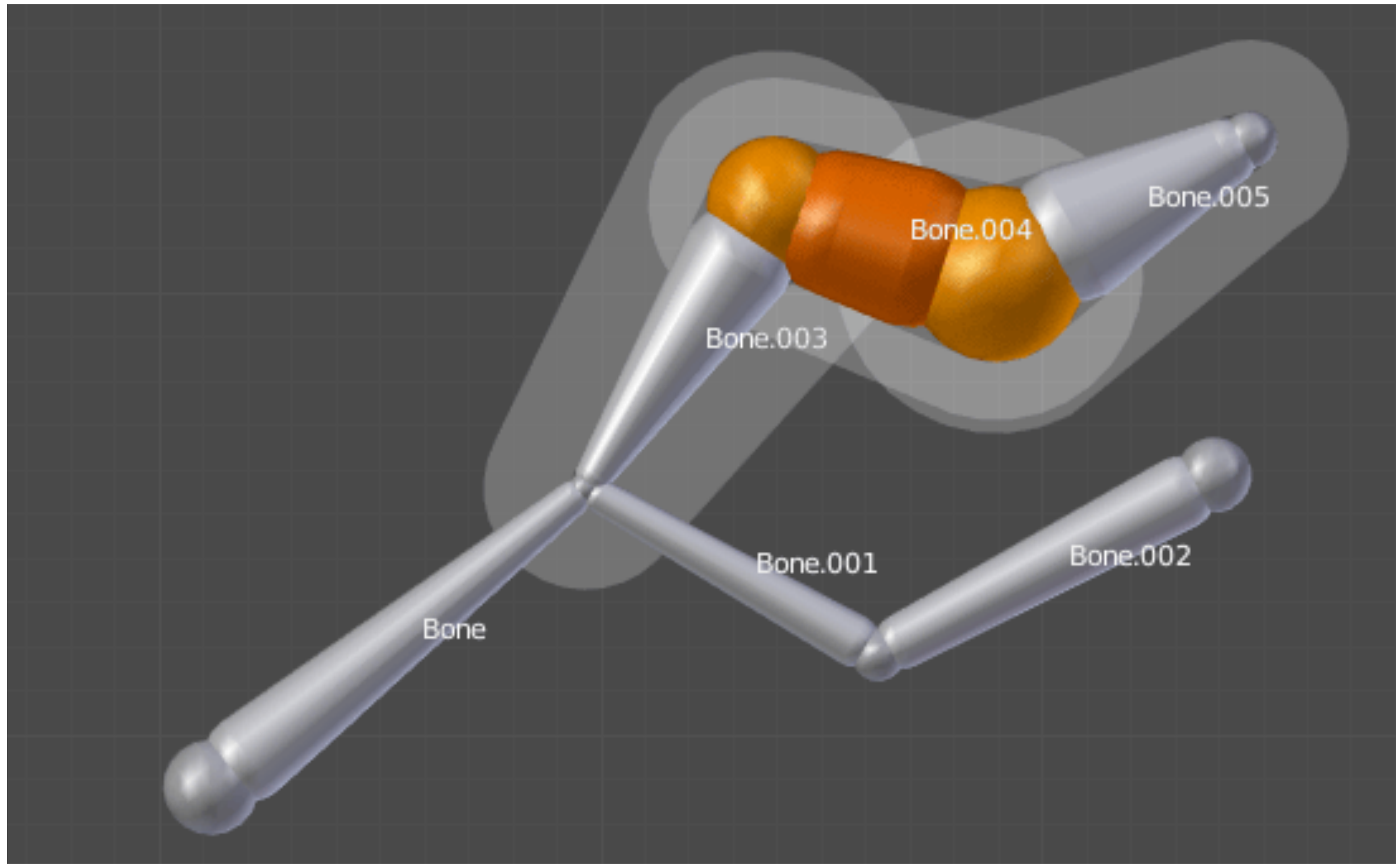


How to get the weights?

Various automatic methods, for example:

- **Envelopes**: manually specified region of influence
  - Influence function $\hat{w}_i(\mathbf{x}) = 1$ inside inner envelope, 0 outside outer envelope, smooth decay in between

$$w_i = \frac{\hat{w}_i(\mathbf{v})}{\sum_{\text{bone } j} \hat{w}_j(\mathbf{v})}$$

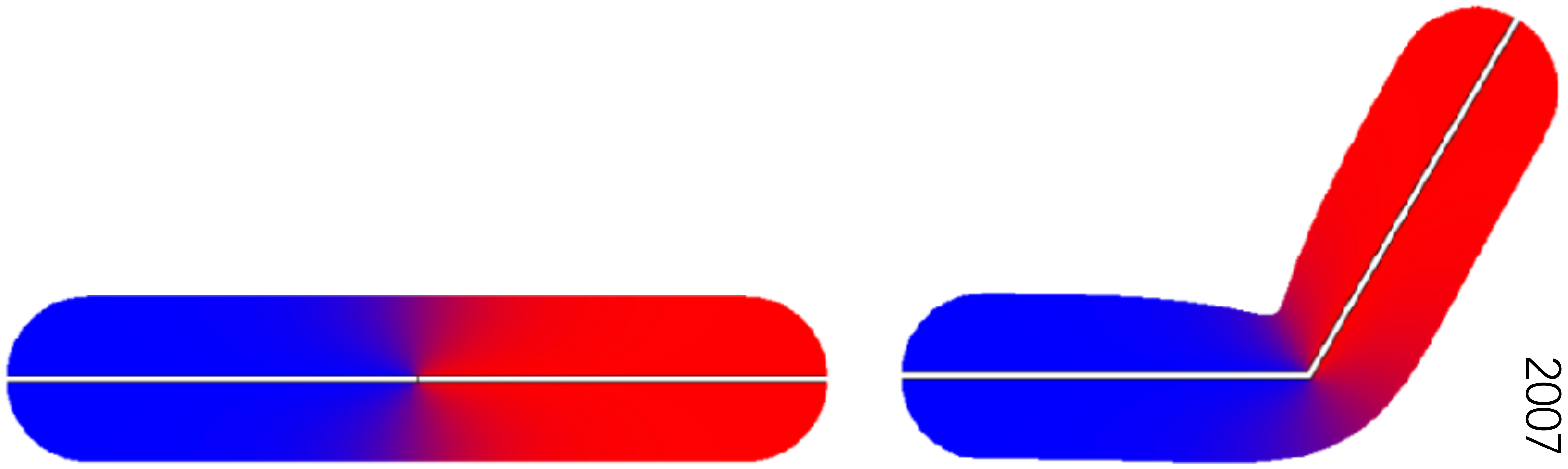


Blender manual

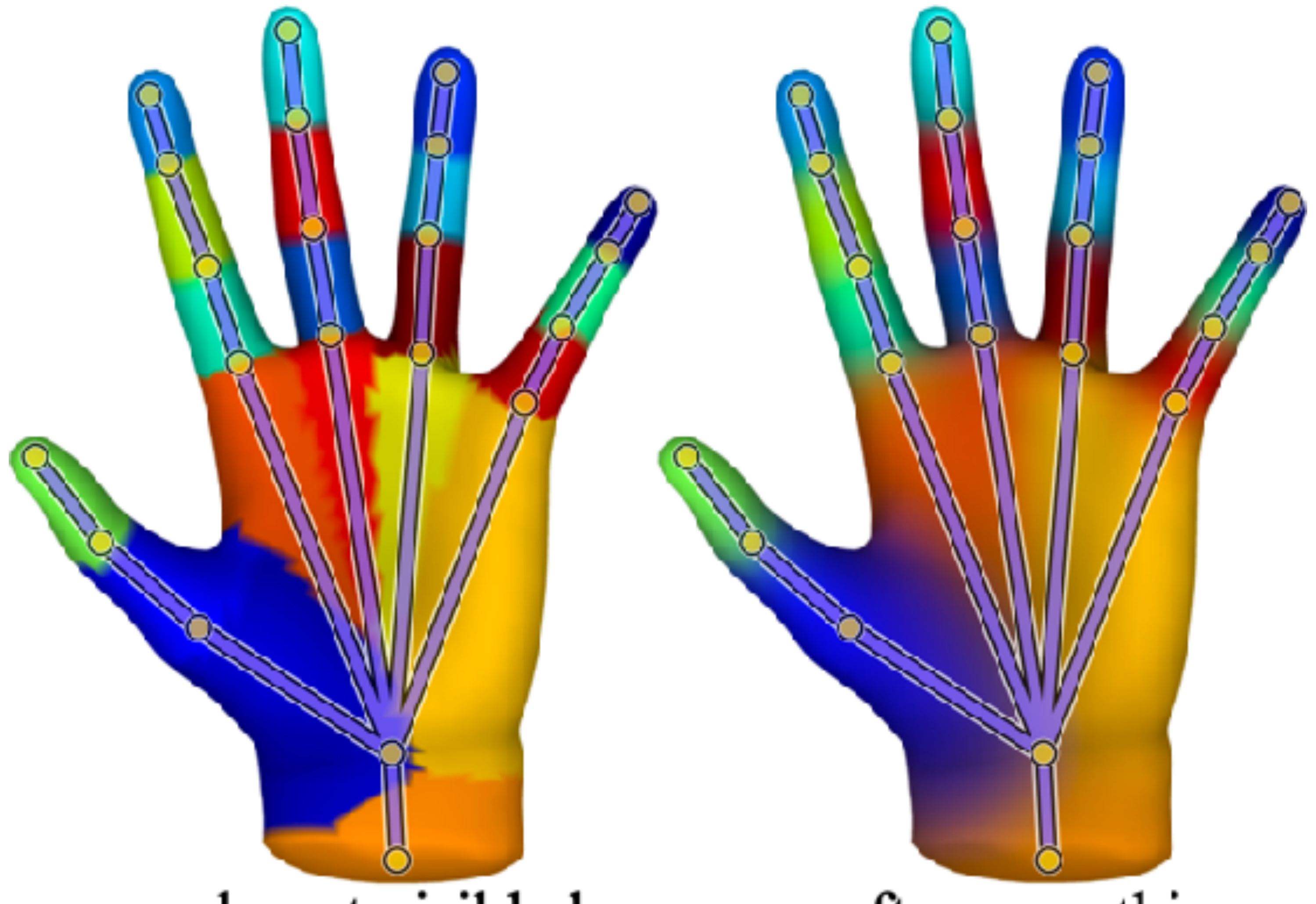
How to get the weights?

Various automatic methods, for example:

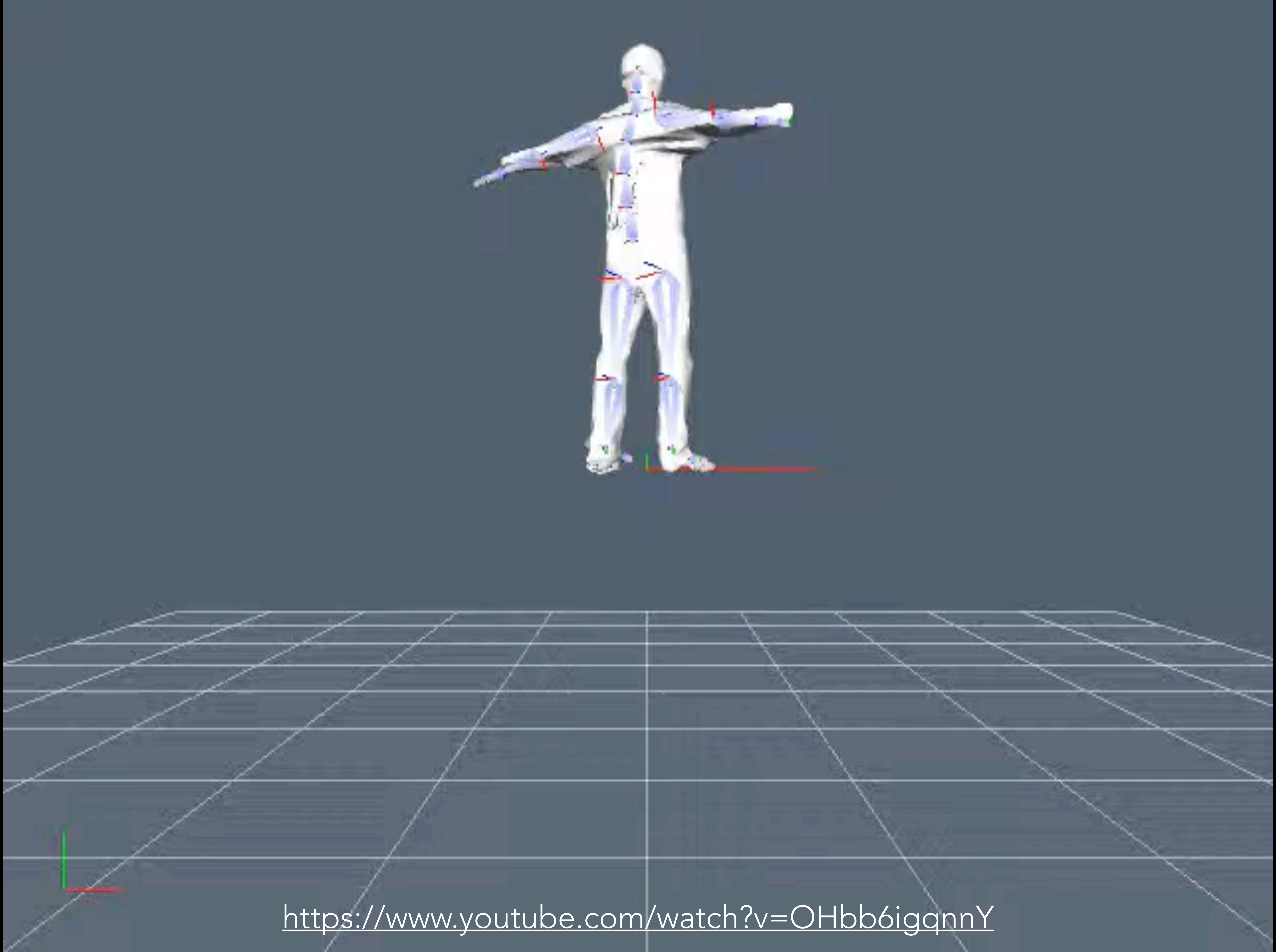
- **Envelopes**: manually specified region of influence
- **Bone heat** (Baran & Popovic 2007): solve PDE to get weights as smooth as possible (weight of bone i = equilibrium heat distribution when bone j is held at temperature δ_{ij})
  - Optimize weights for smoothness, locality, monotonicity, etc., e.g. "bounded biharmonic weights" (Jacobson et al. 2011)



Baran & Popovic 2007



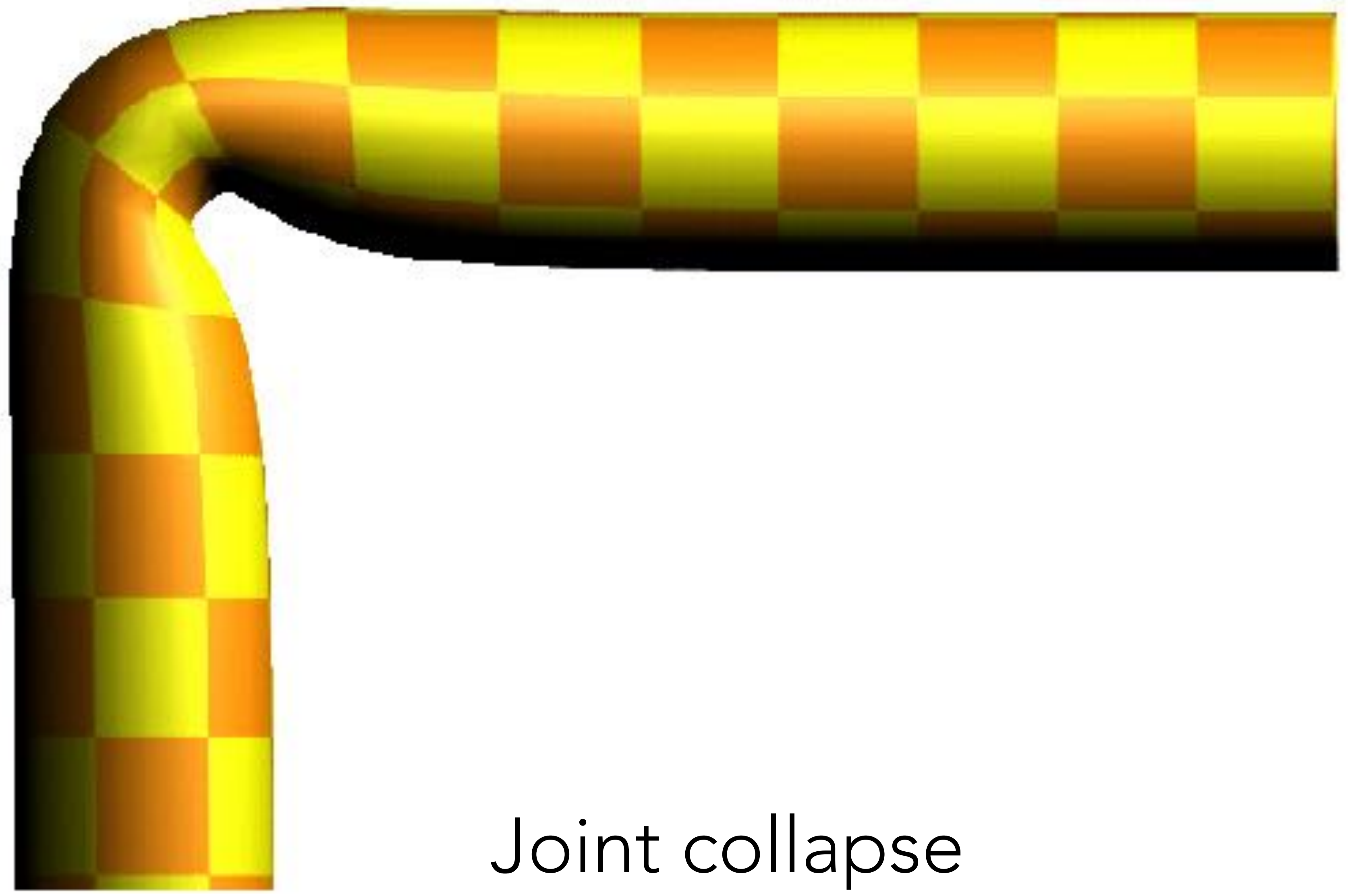
Jacobson 2014



https://www.youtube.com/watch?v=OHbb6igqnnY

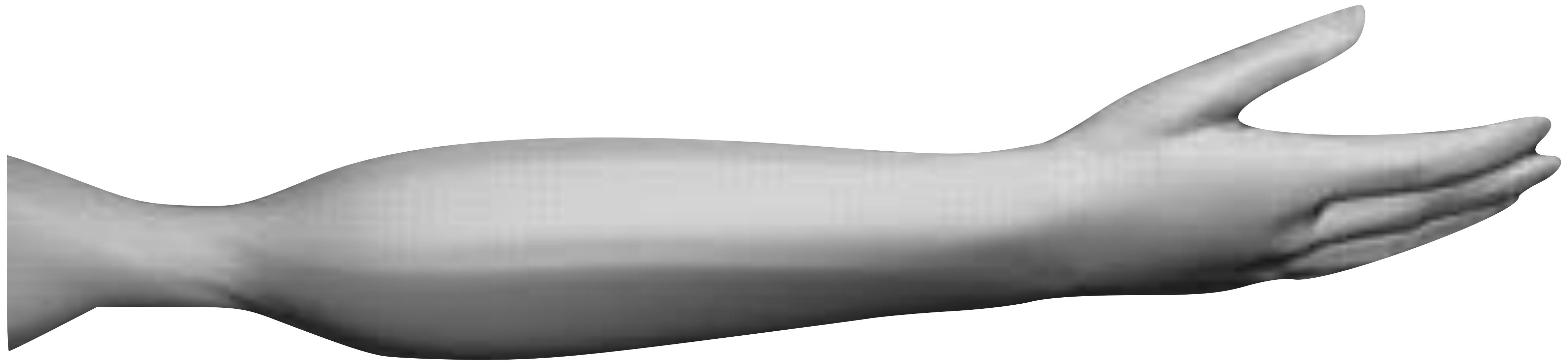
harrumphoid

Problems with linear blending



Ladislav Kavan

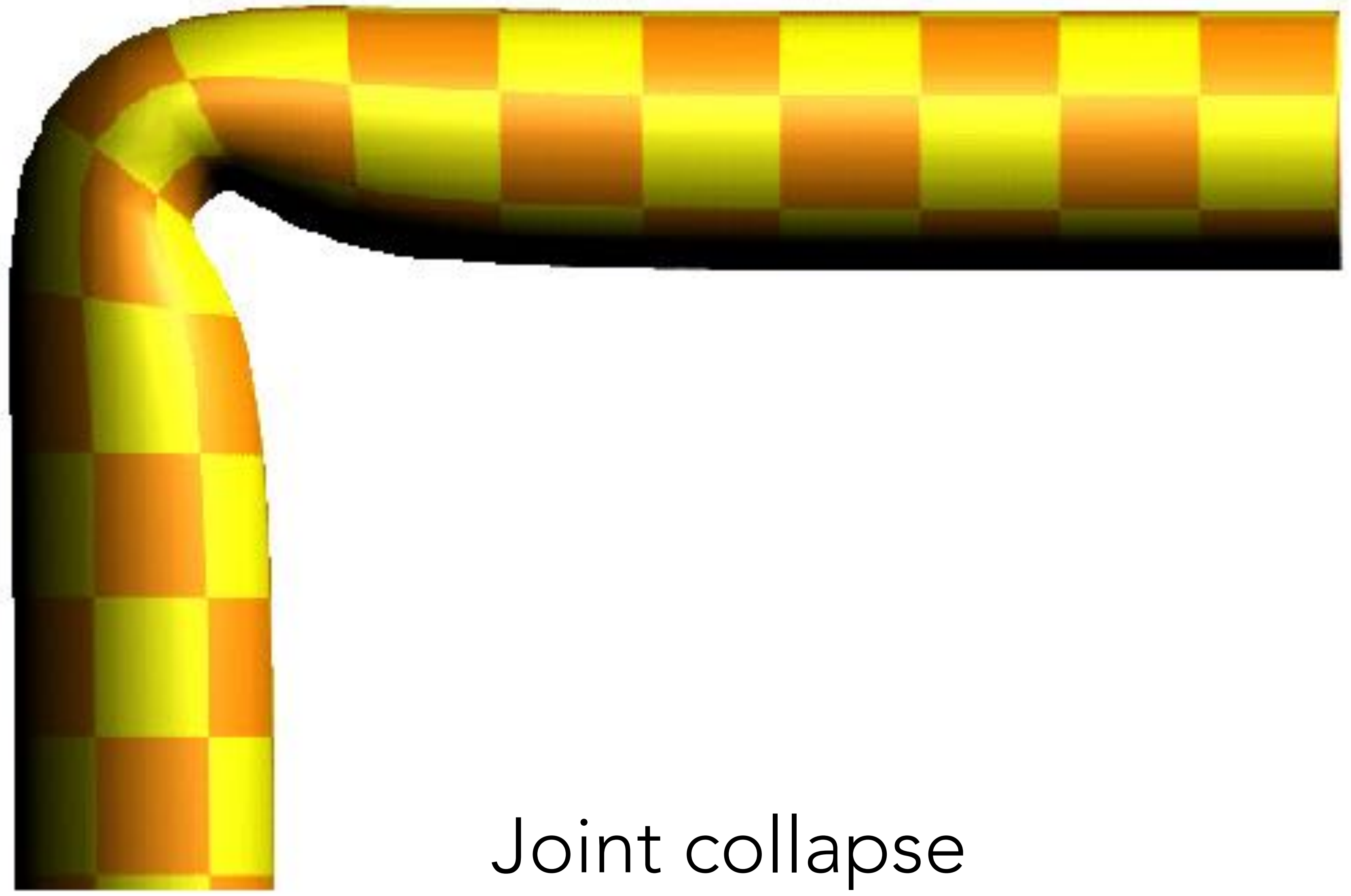
Joint collapse



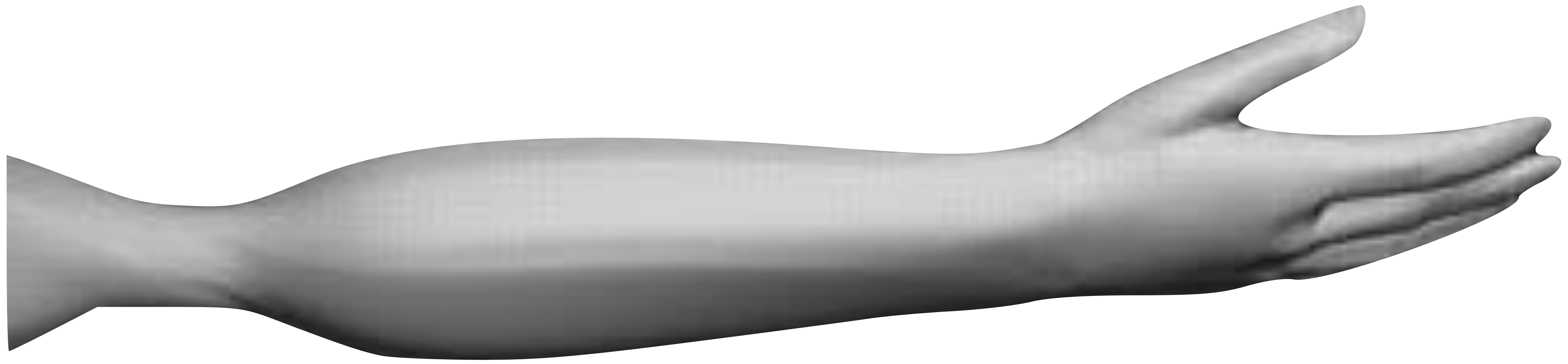
Lewis et al. 2000

Candy wrapper effect

Puzzle: Why does this happen?

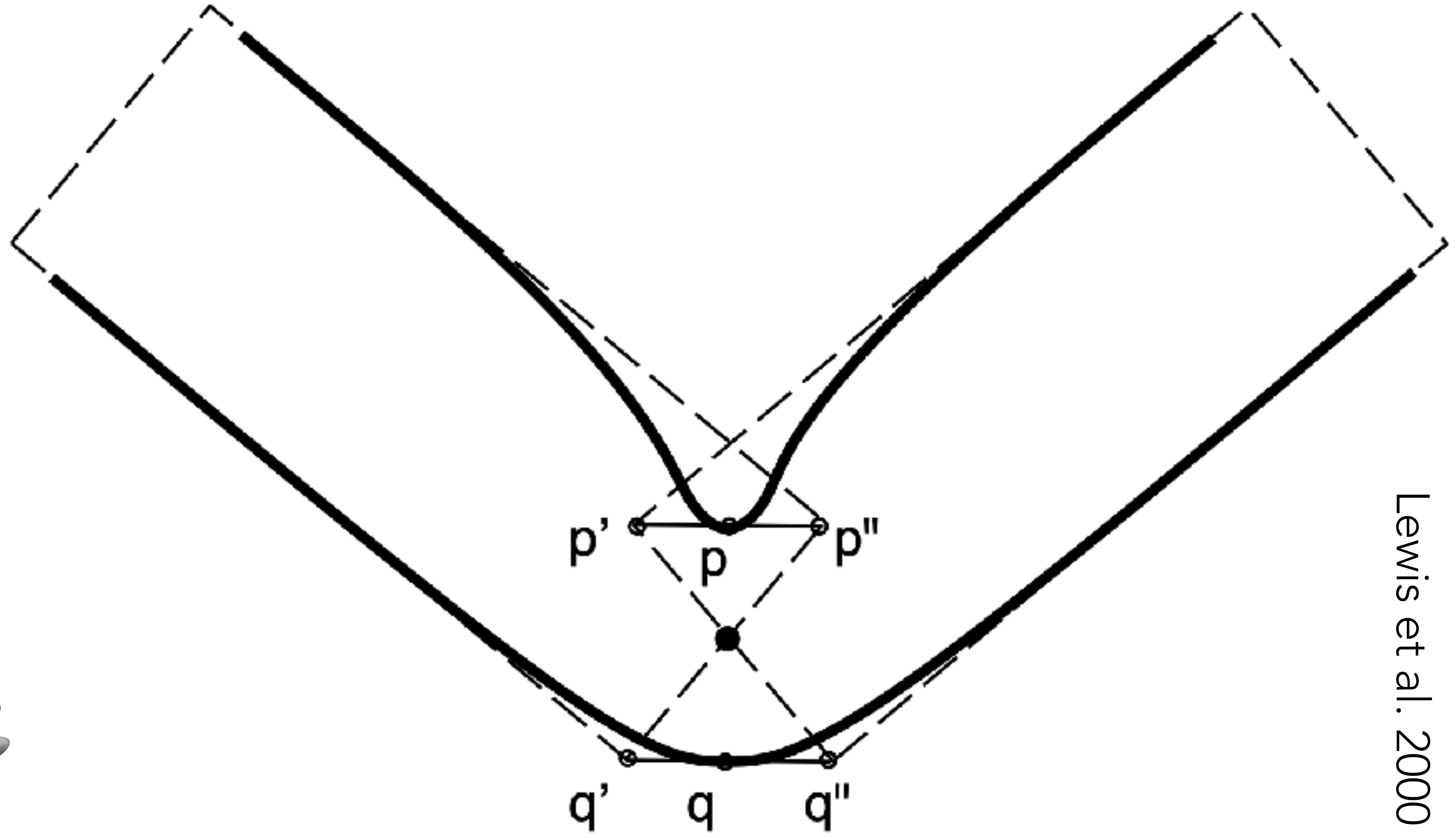


Joint collapse



Candy wrapper effect

Similar to problems with interpolating rotation matrices:



Lewis et al. 2000

Linearly averaging two rigid transformations does not give a rigid transformation!

Dual quaternions in 1 slide (not on the exam)

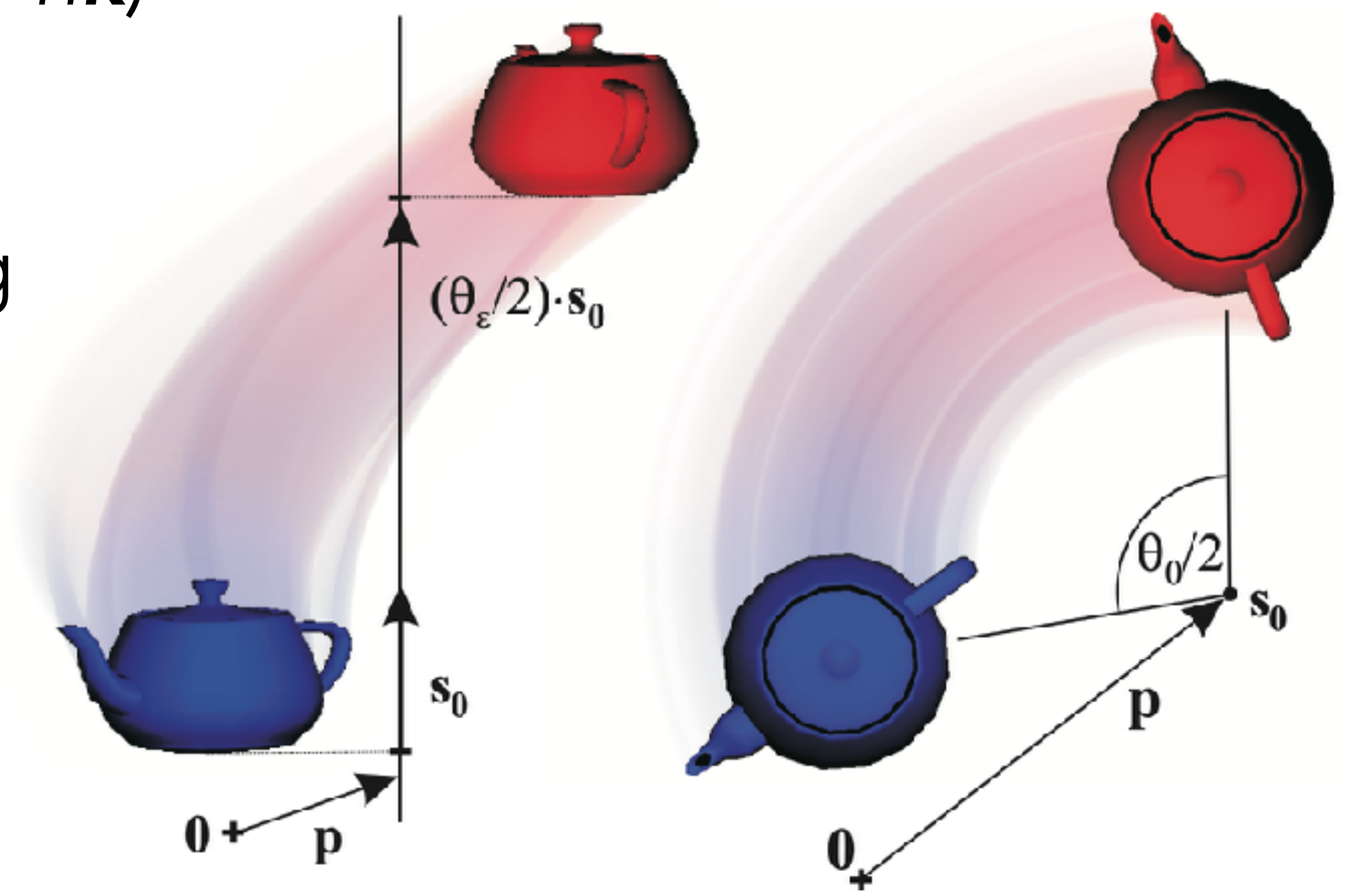
Quaternions: $\boldsymbol{q} = a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$

- Unit quaternions ($\boldsymbol{q}^*\boldsymbol{q} = 1$) represent rotations

Dual quaternions: $\hat{\boldsymbol{q}} = (a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}) + \varepsilon\,(e + f\boldsymbol{i} + g\boldsymbol{j} + h\boldsymbol{k})$

- Unit dual quaternions ($\hat{\boldsymbol{q}}^*\hat{\boldsymbol{q}} = 1 + 0\varepsilon$) represent rigid transformations: translation & rotation, no scaling

In both cases: easy to normalize after interpolation



Kavan et al. 2007

Linear blend skinning: $\mathbf{v}' = \sum_{\text{bone } i} w_i \mathbf{T}_i \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = \left(\sum_{\text{bone } i} w_i \mathbf{T}_i \right) \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$

- Take weighted average of bones' transformation matrices
- Apply to vertex

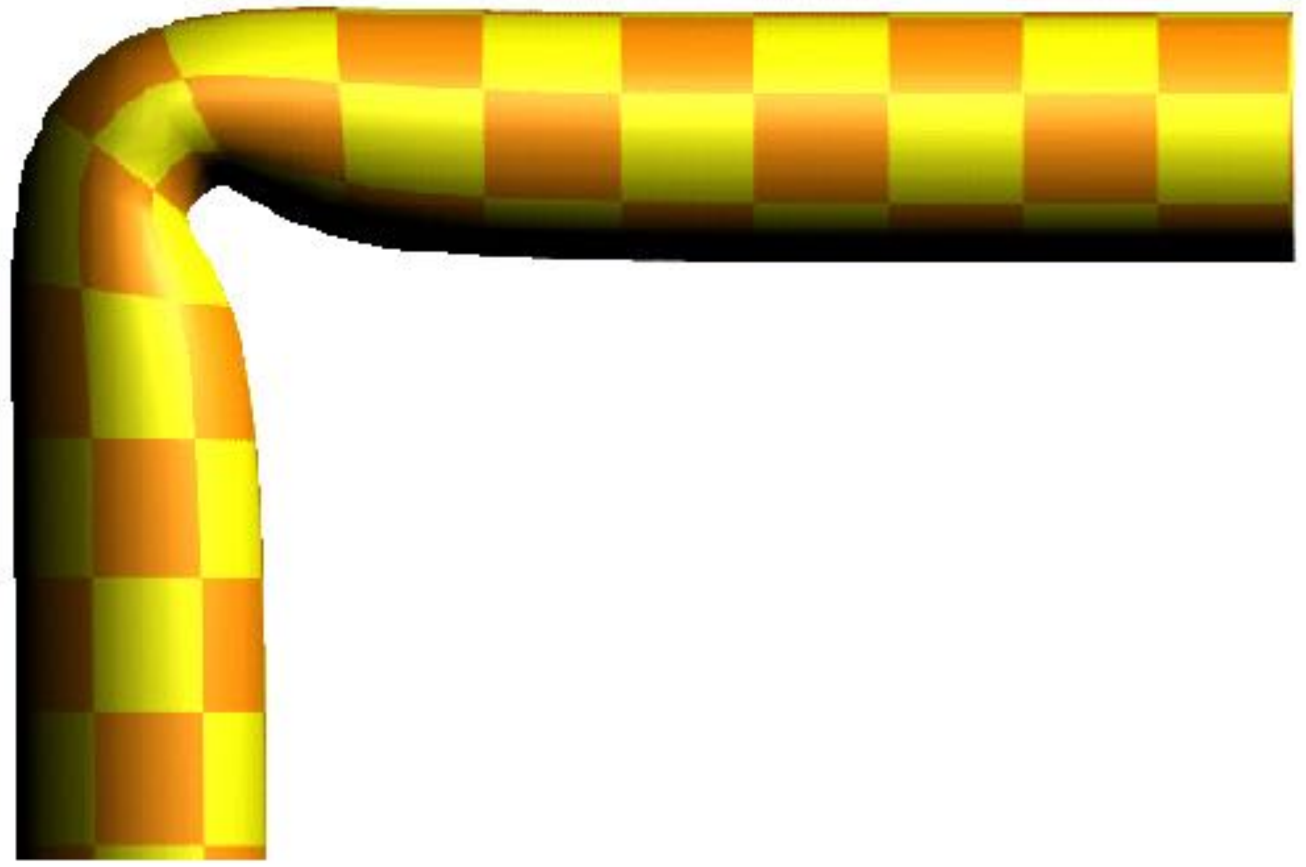
Dual quaternion skinning: represent bone transformations as DQs instead of matrices

- Take weighted average of bones' DQs
- Normalize to unit DQ $\Rightarrow$ rigid transformation!
- Apply to vertex

Linear blend skinning

Dual quaternion skinning

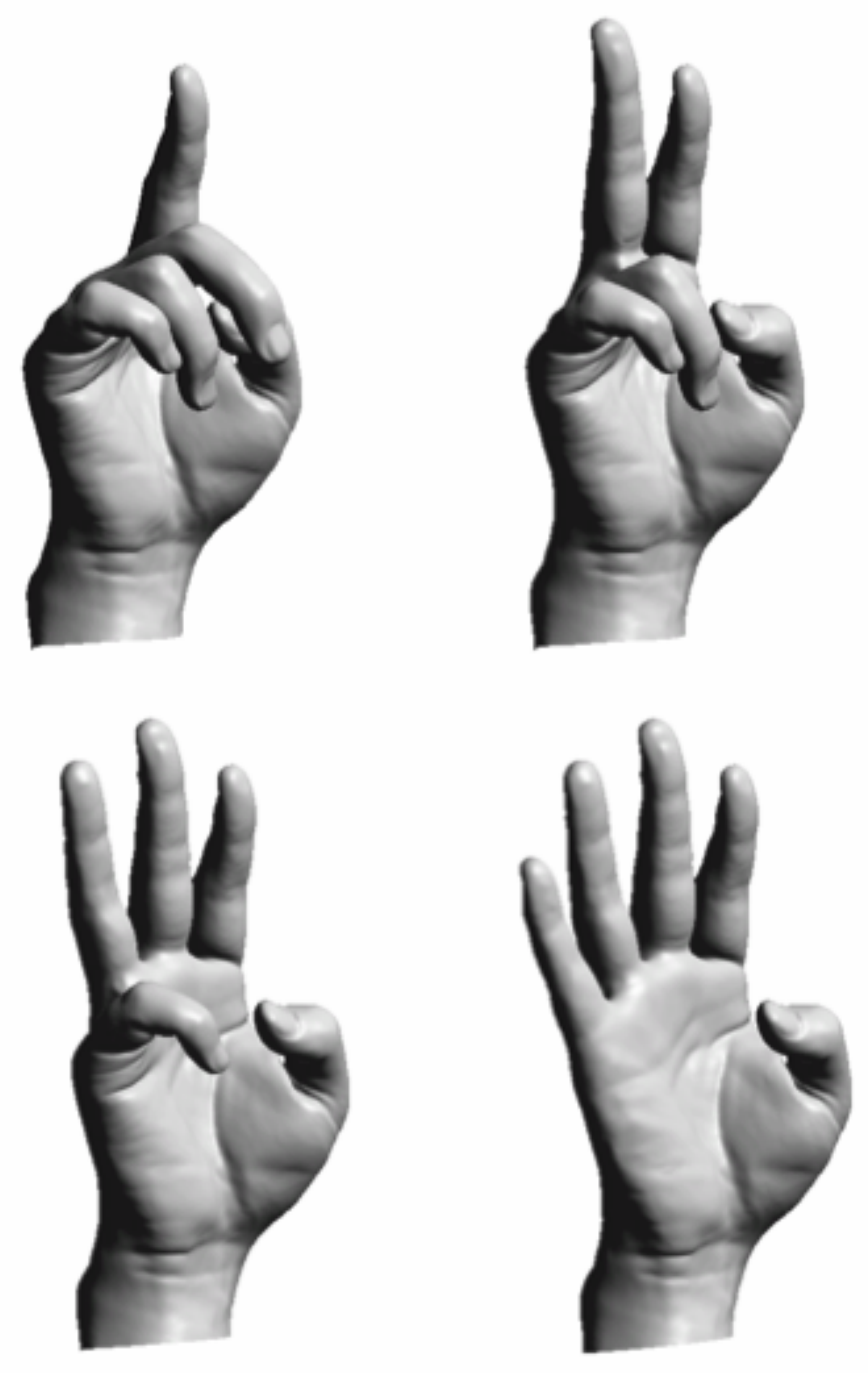
Kavan et al. 2007



Ladislav Kavan

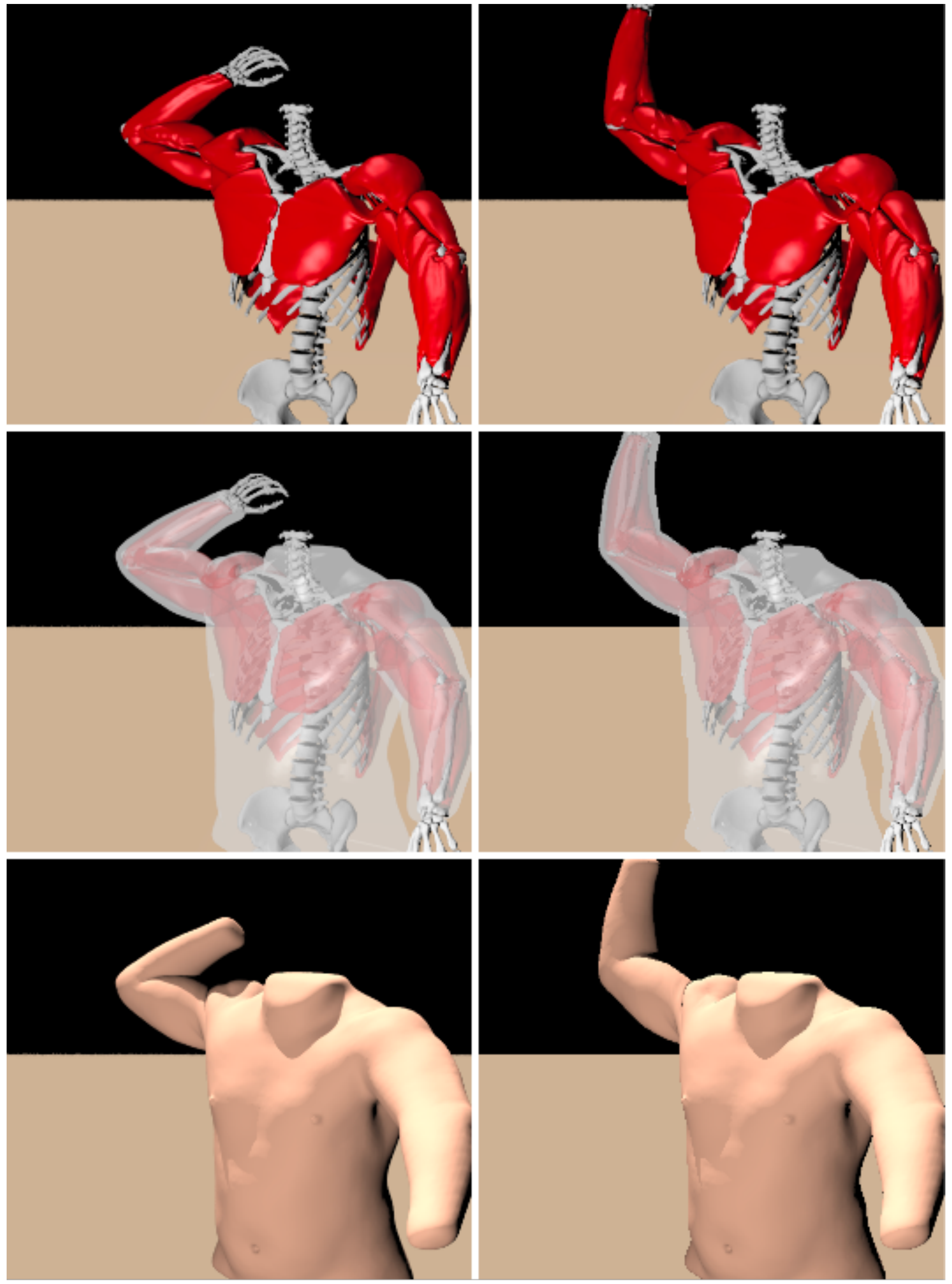
Lots of other techniques:

- Pose space deformation
- Elasticity-inspired deformers
- Implicit skinning
- Optimized centers of rotation
- Direct delta mush
- …

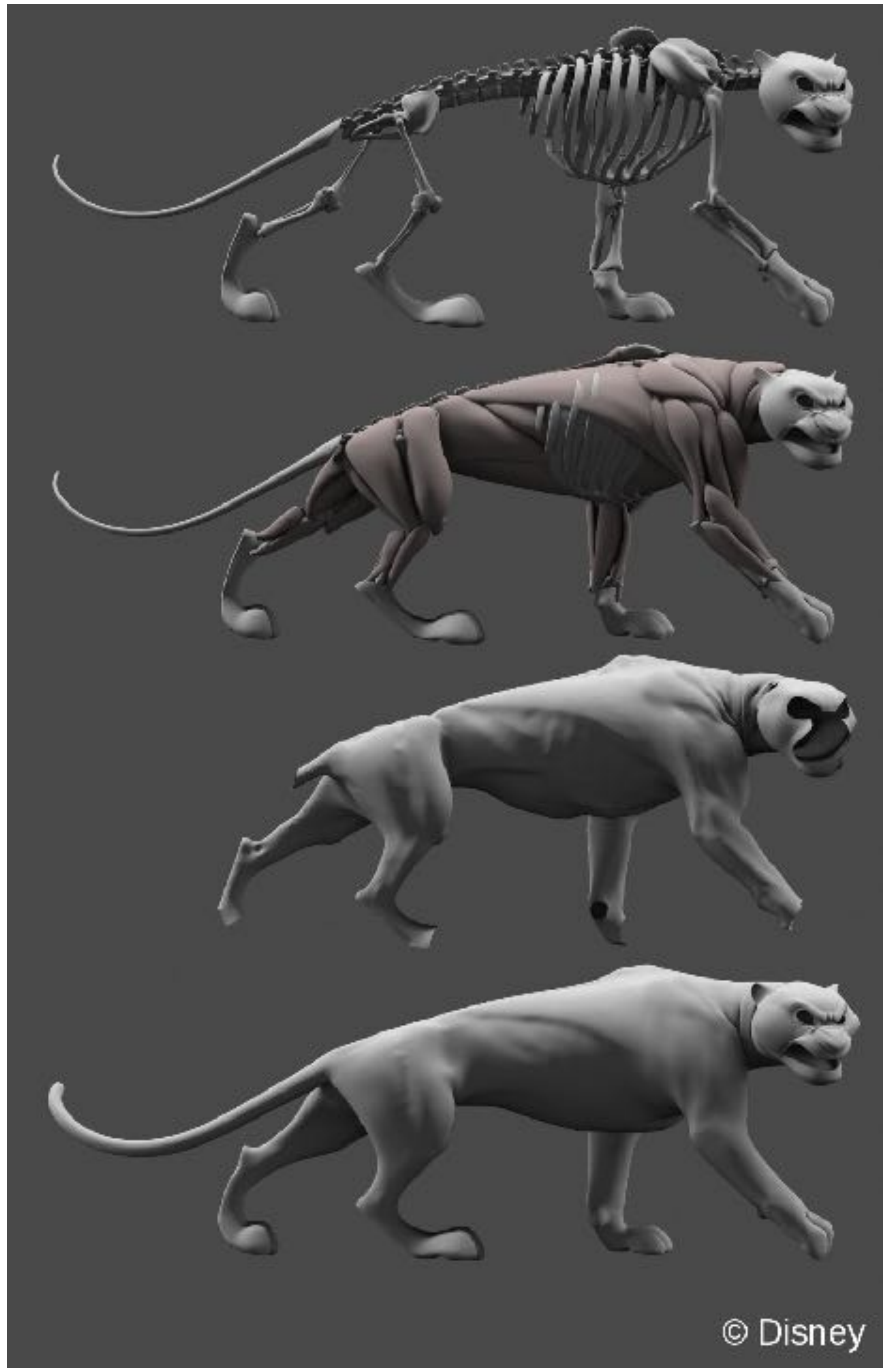


Kurihara & Miyata 2004

Beyond geometric skinning

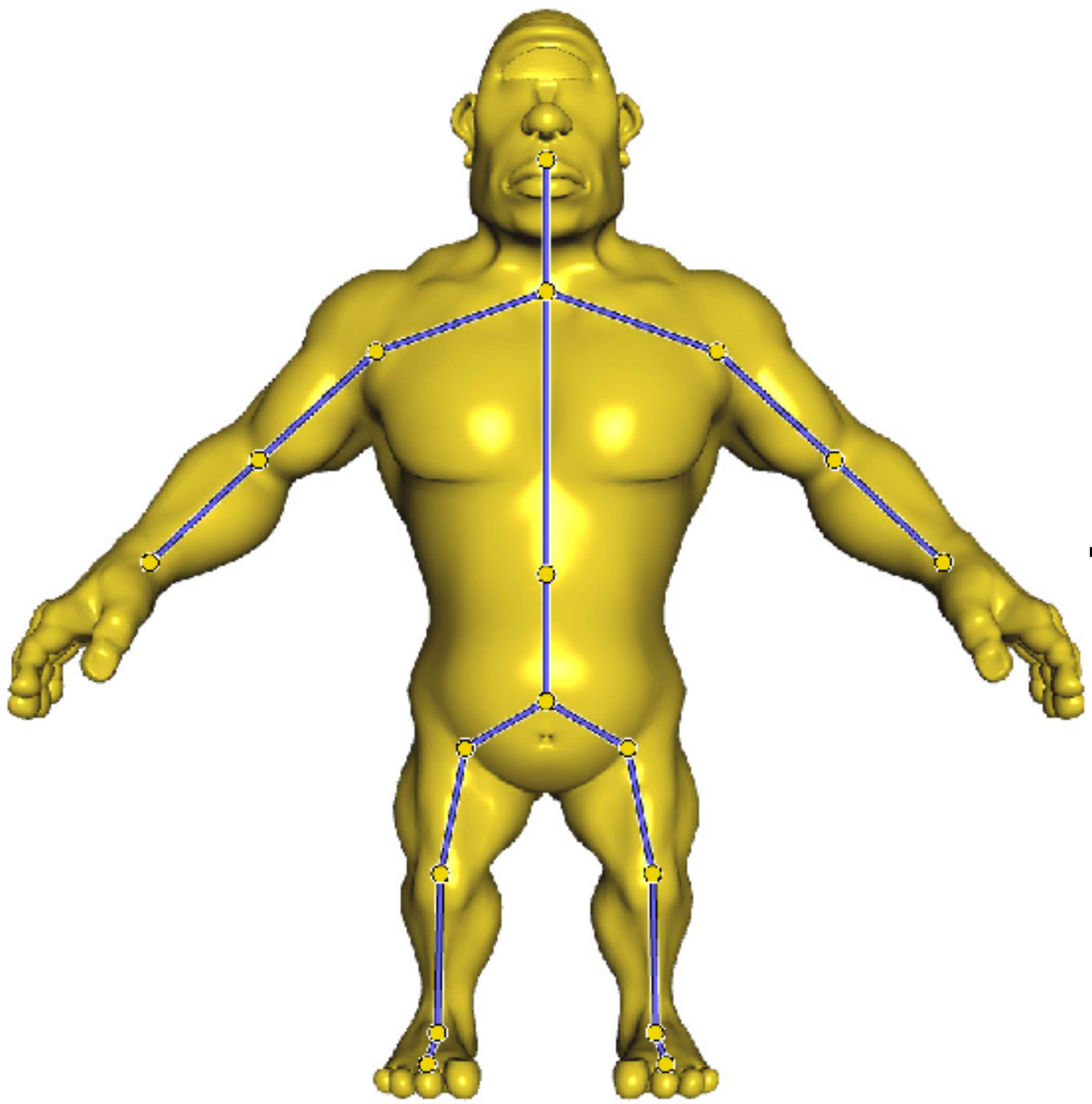


Teran et al. 2005

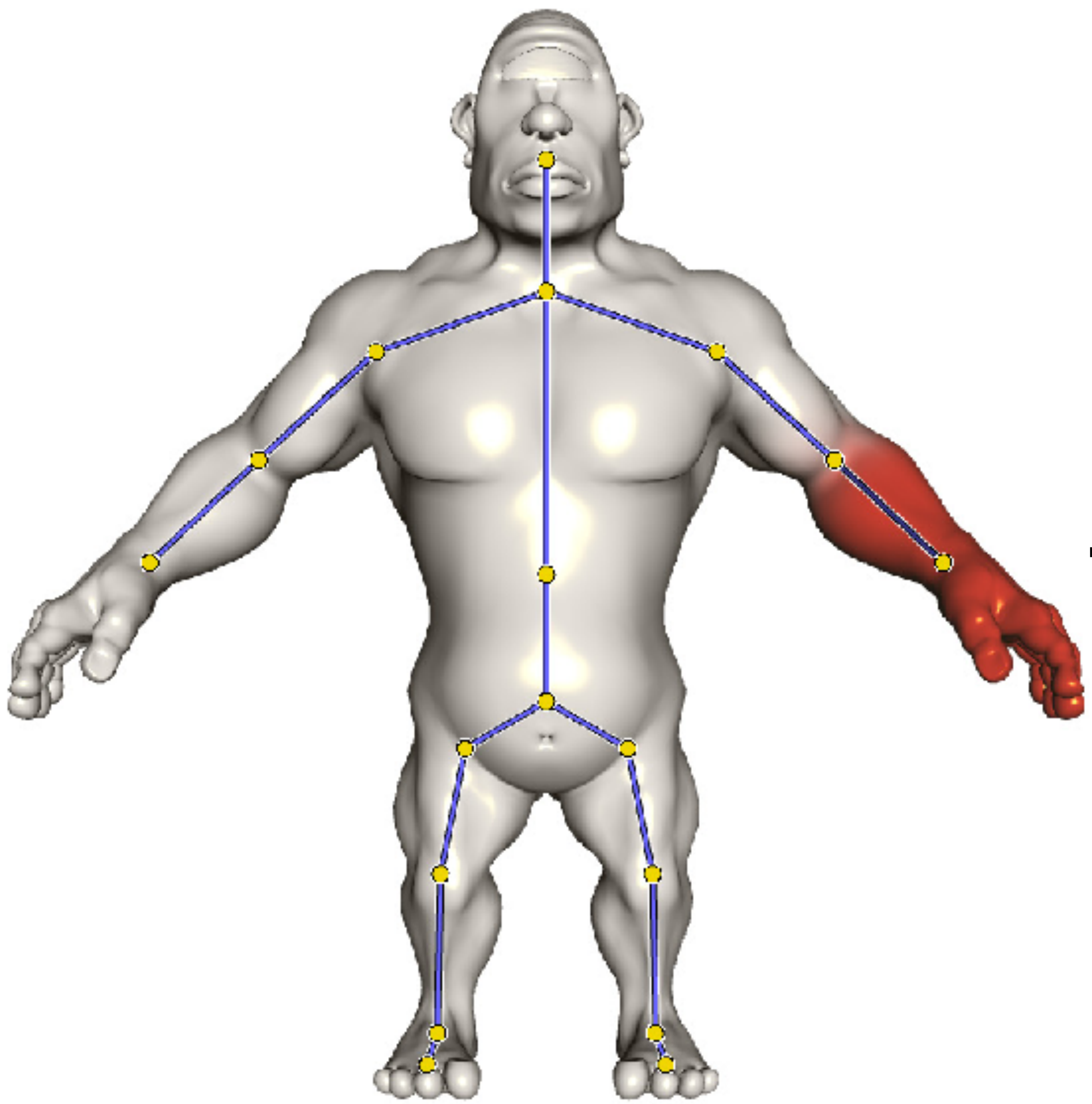


Milne et al. 2016

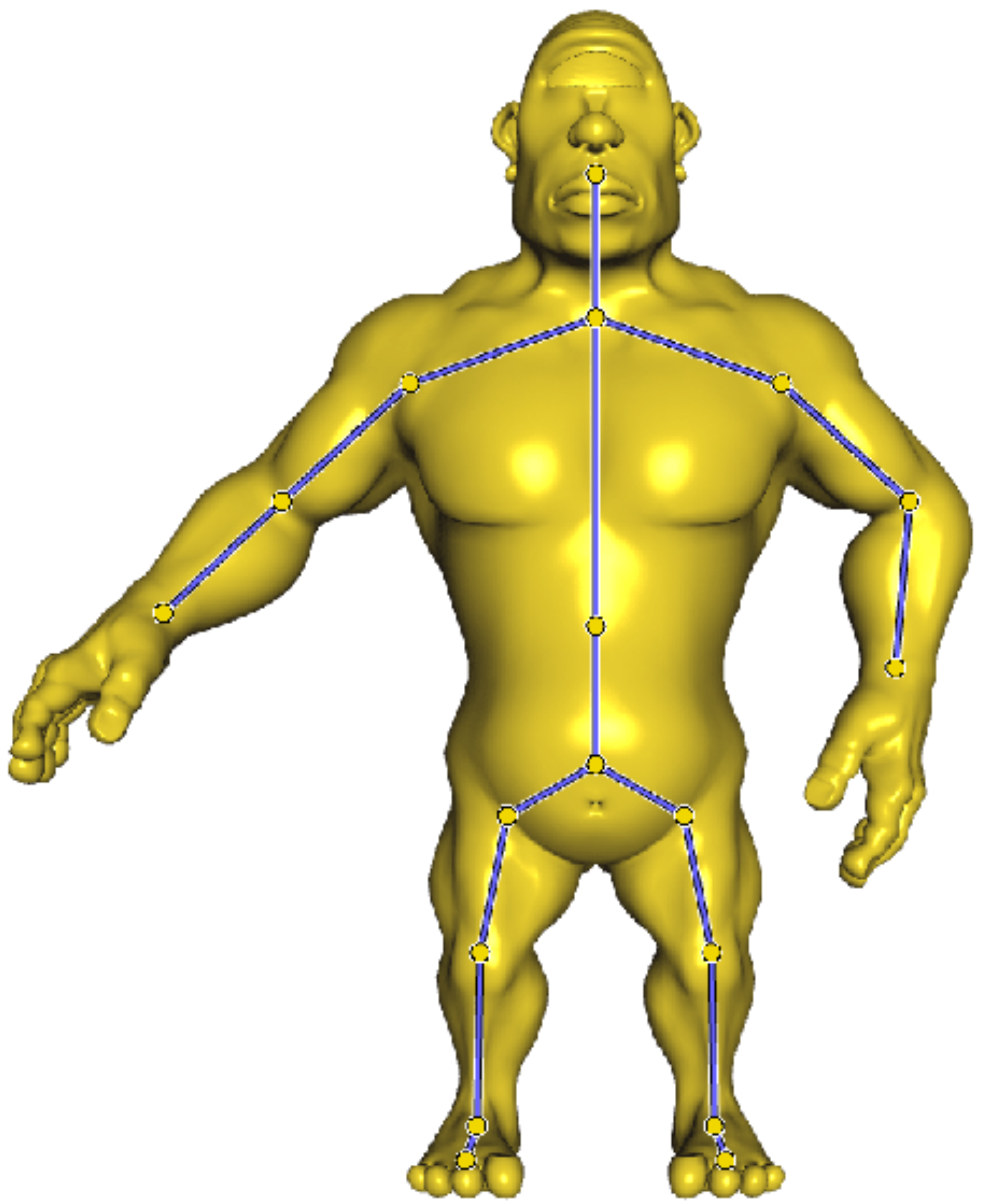
Character animation wrap-up



Define skeleton



Attach skin

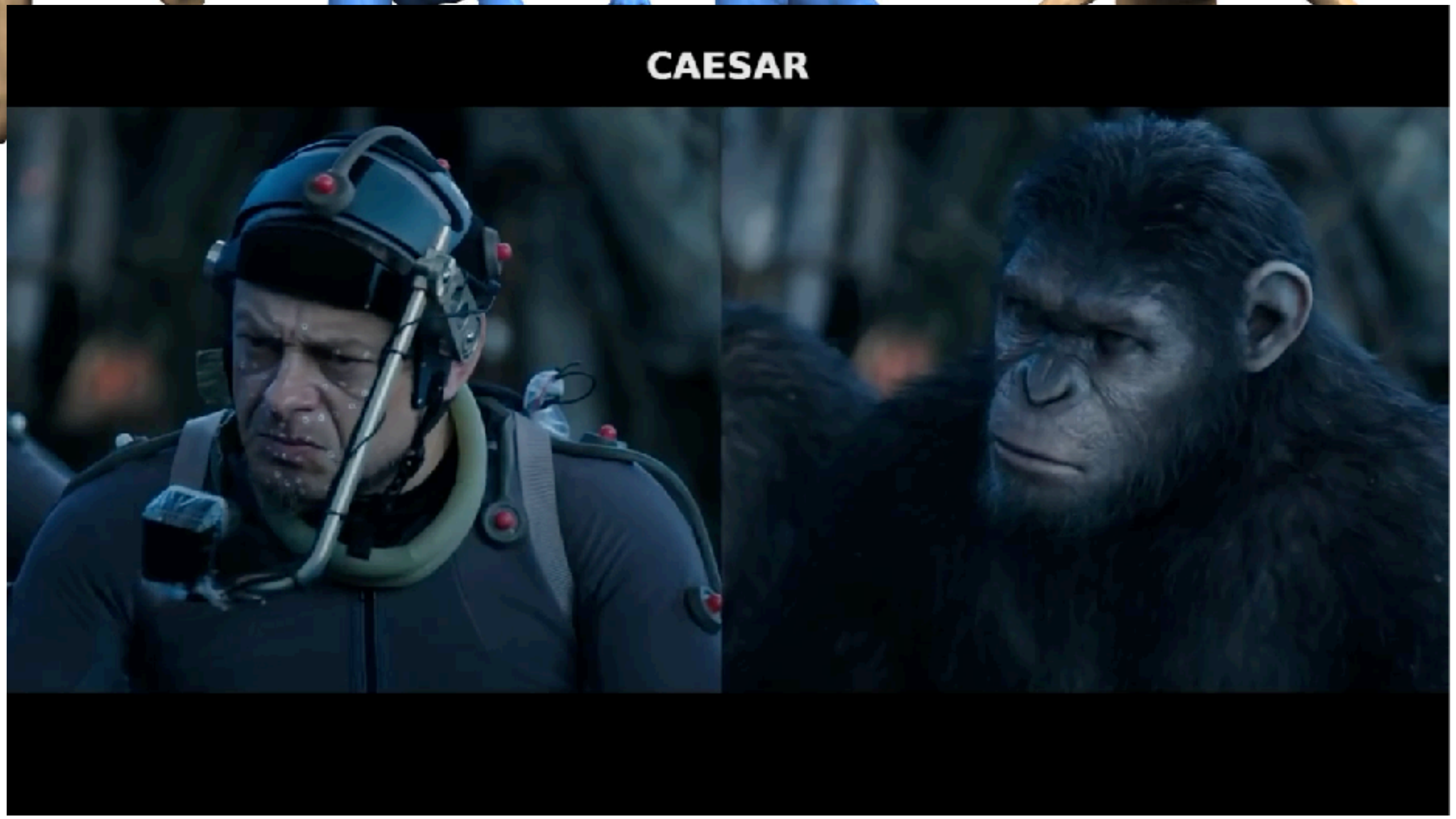


Move controls (joints and/or end effectors) to animate

Alec Jacobson

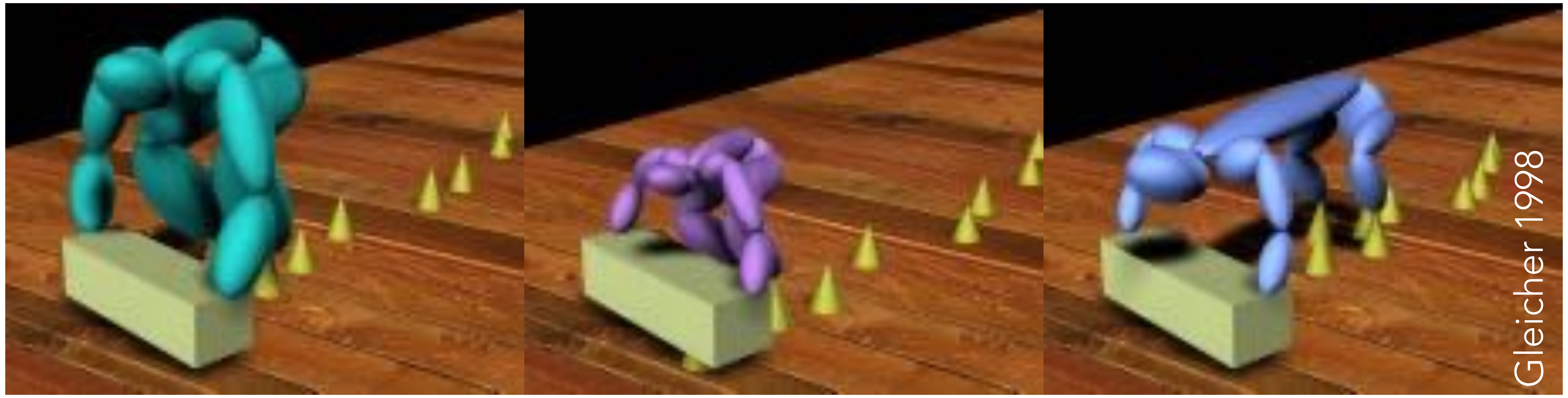
Where does the motion data come from?

Keyframe animation



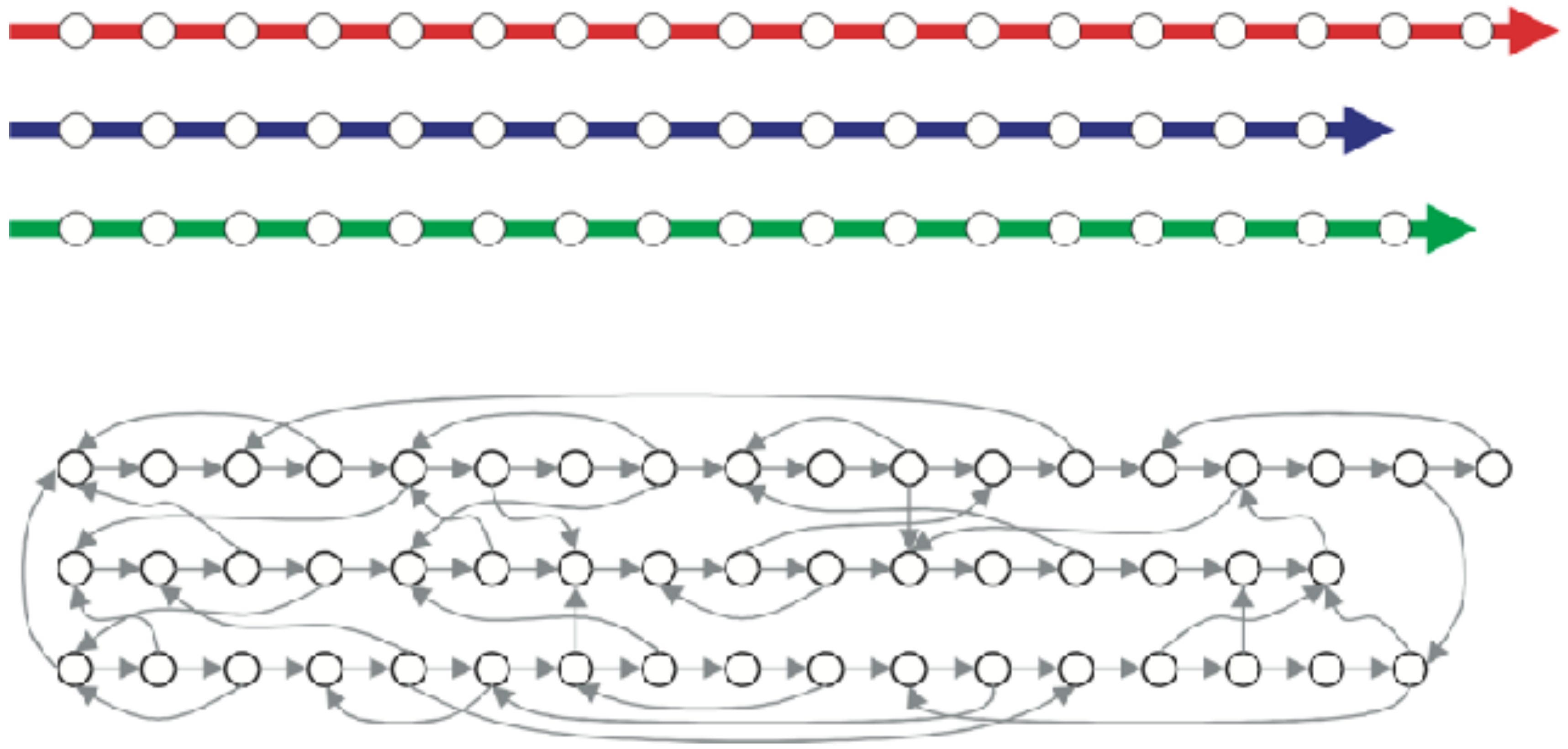
Motion capture ("mocap")

Mocap editing

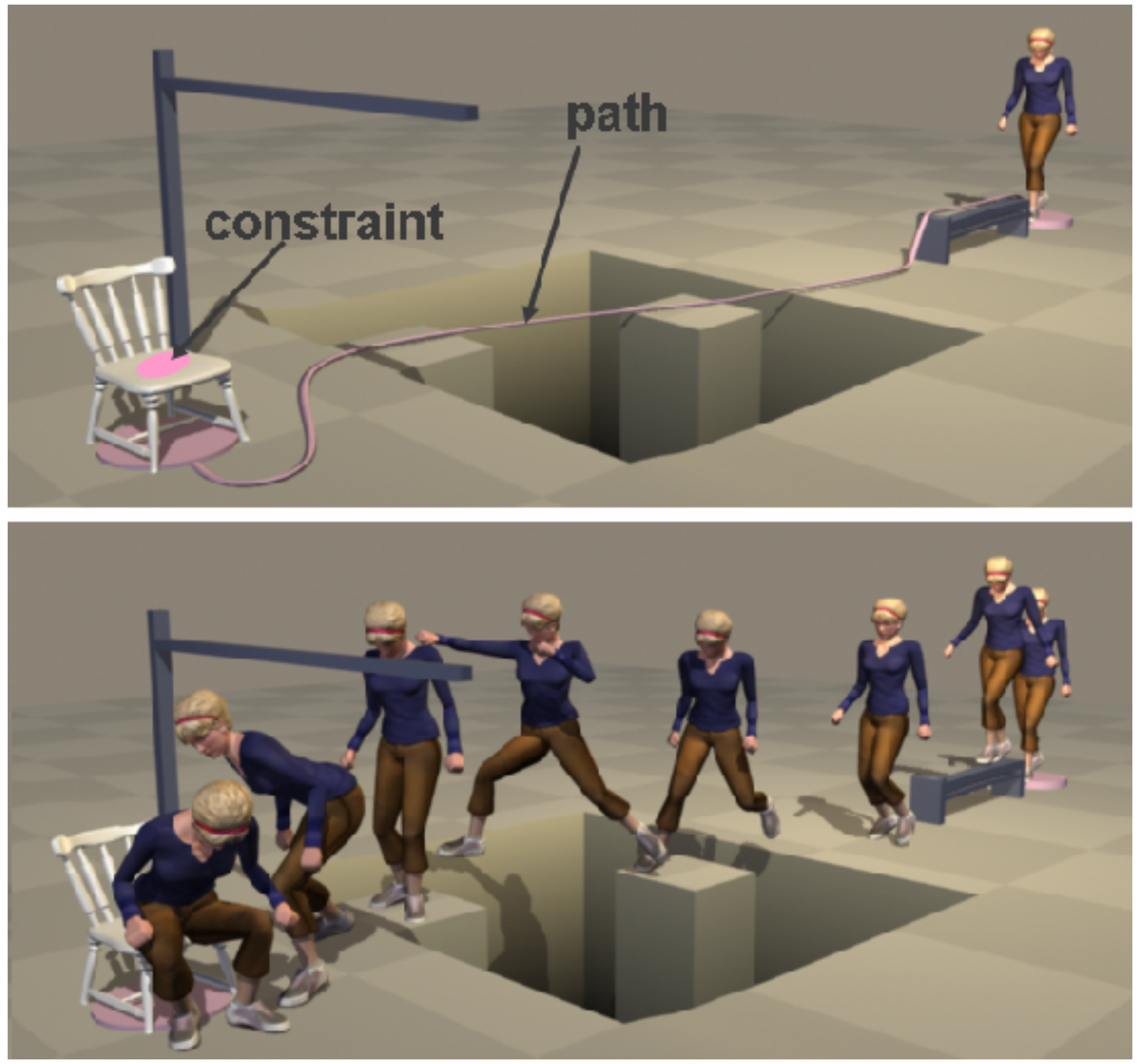


Retargeting

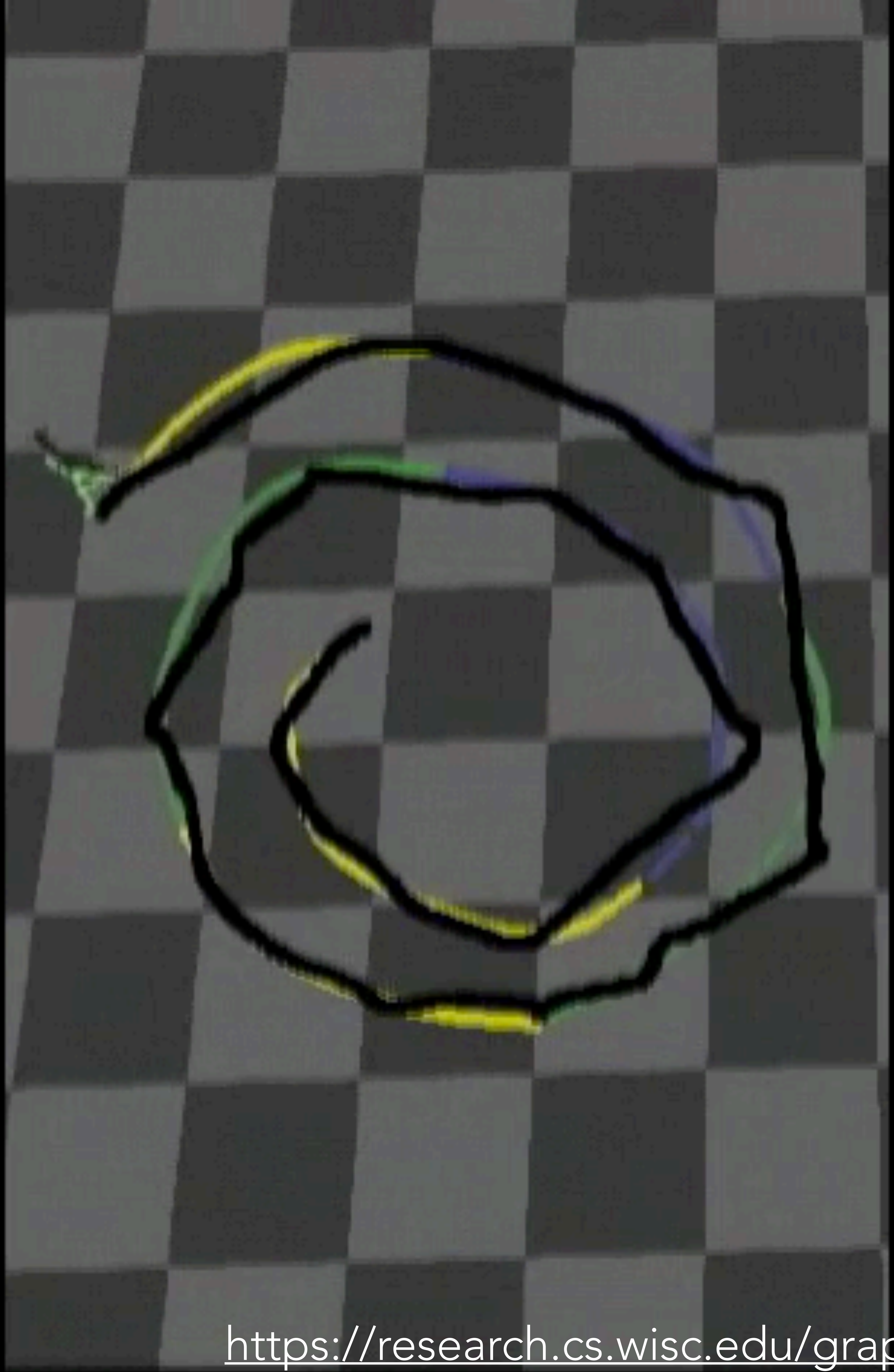
Lee et al. 2002

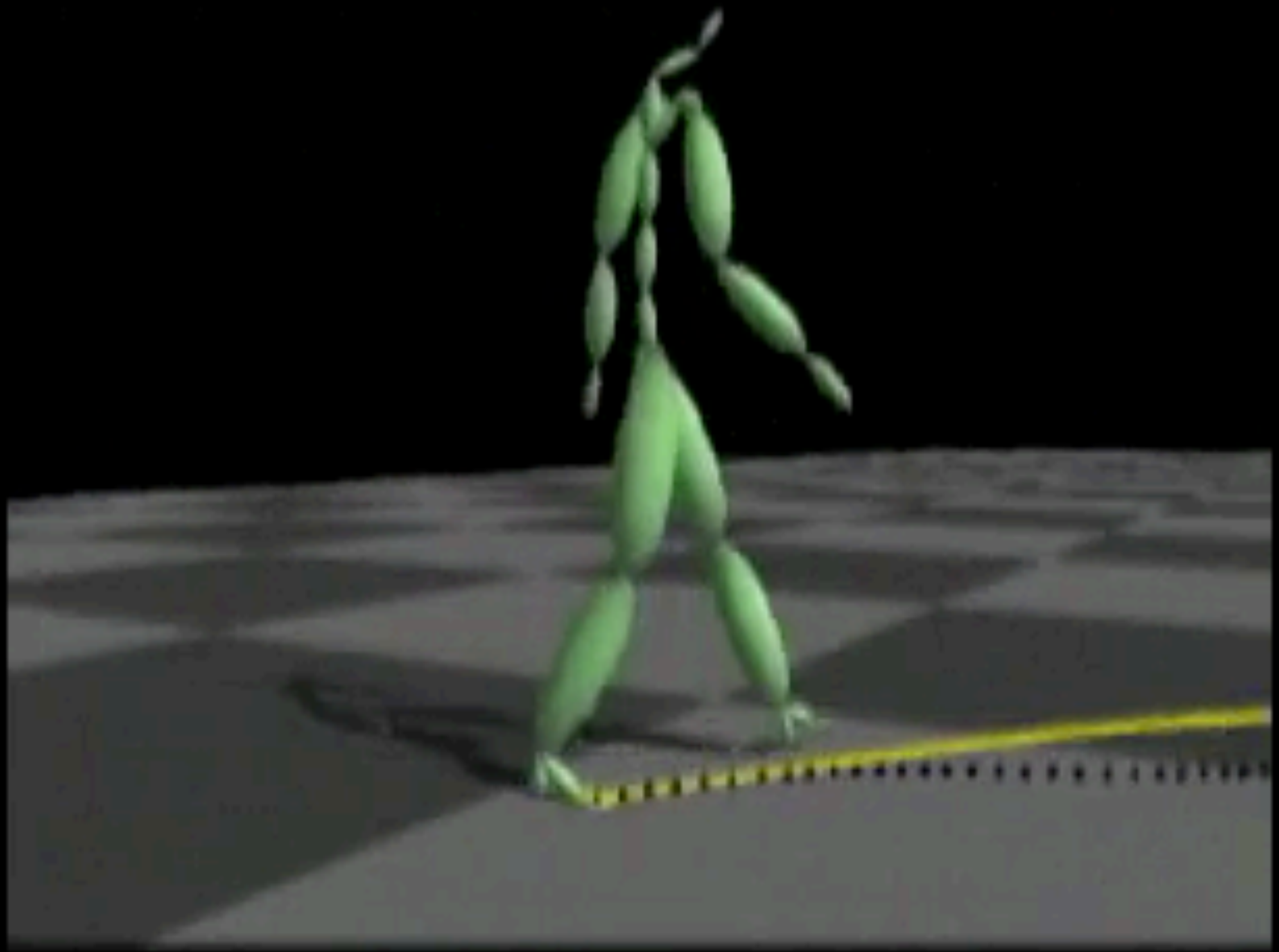


Motion graphs



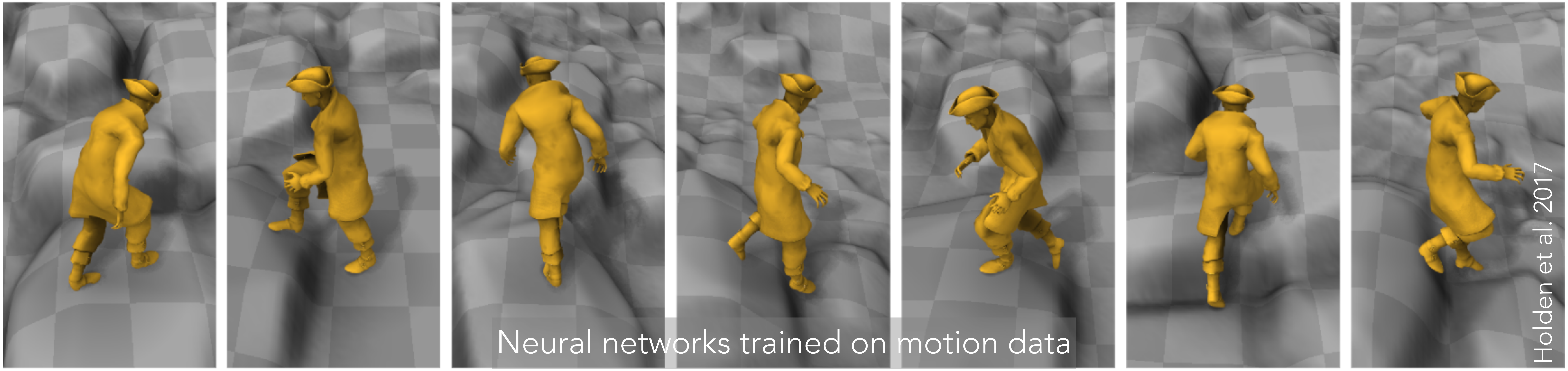
Safonova & Hodgins 2007

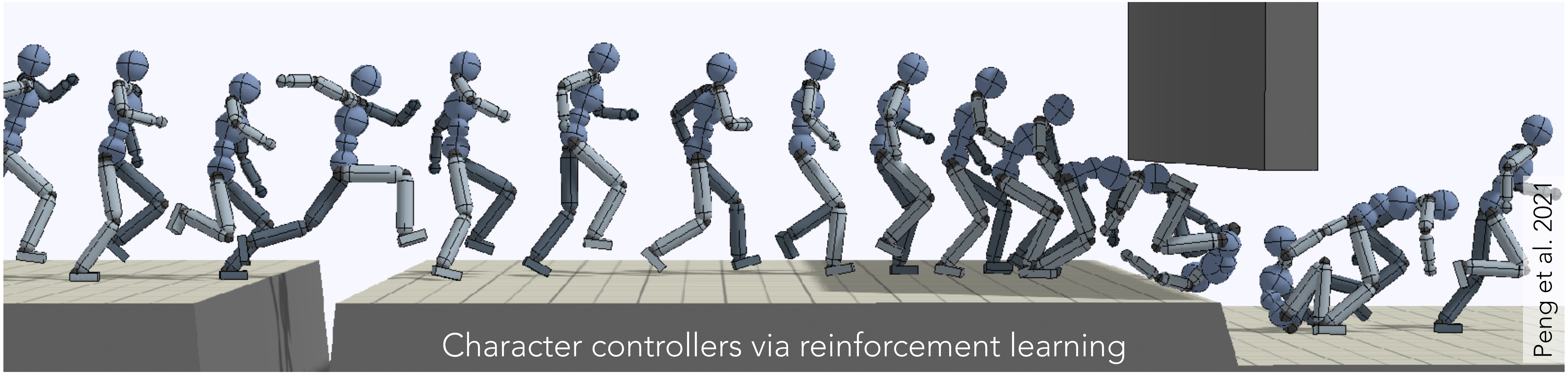


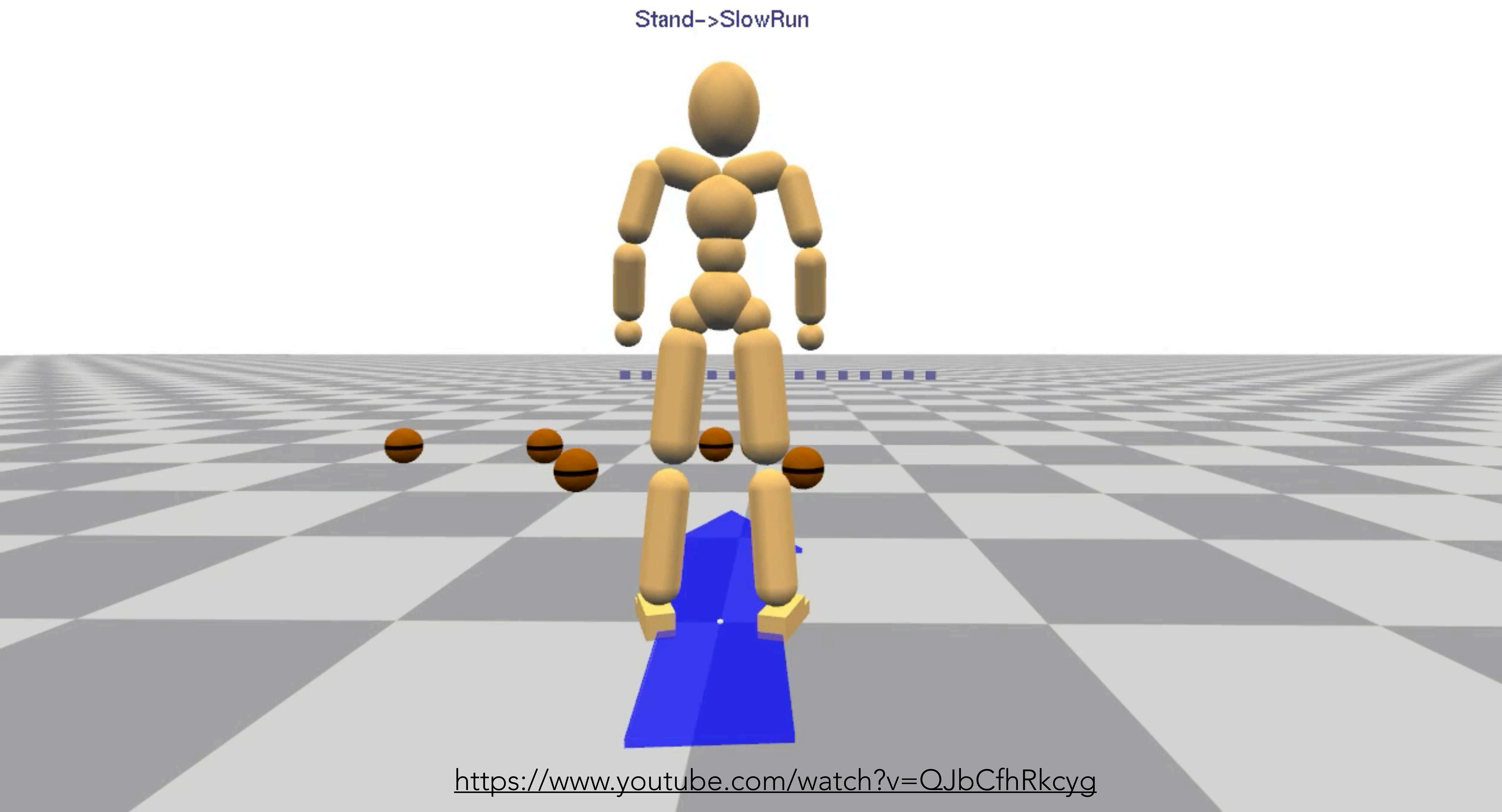


https://research.cs.wisc.edu/graphics/Gallery/kovar.vol/MoGraphs/

Kovar et al. 2002


Neural networks trained on motion data
Holden et al. 2017


Character controllers via reinforcement learning
Peng et al. 2021



https://www.youtube.com/watch?v=QJbCfhRkcyg

Liu et al. 2016

Physics-based animation

for things hard to animate using keyframing or motion capture…

Fluids

http://physbam.stanford.edu/~fedkiw/

Losasso et al. 2008

Cloth

https://www.cs.columbia.edu/cg/ESIC/esic.html

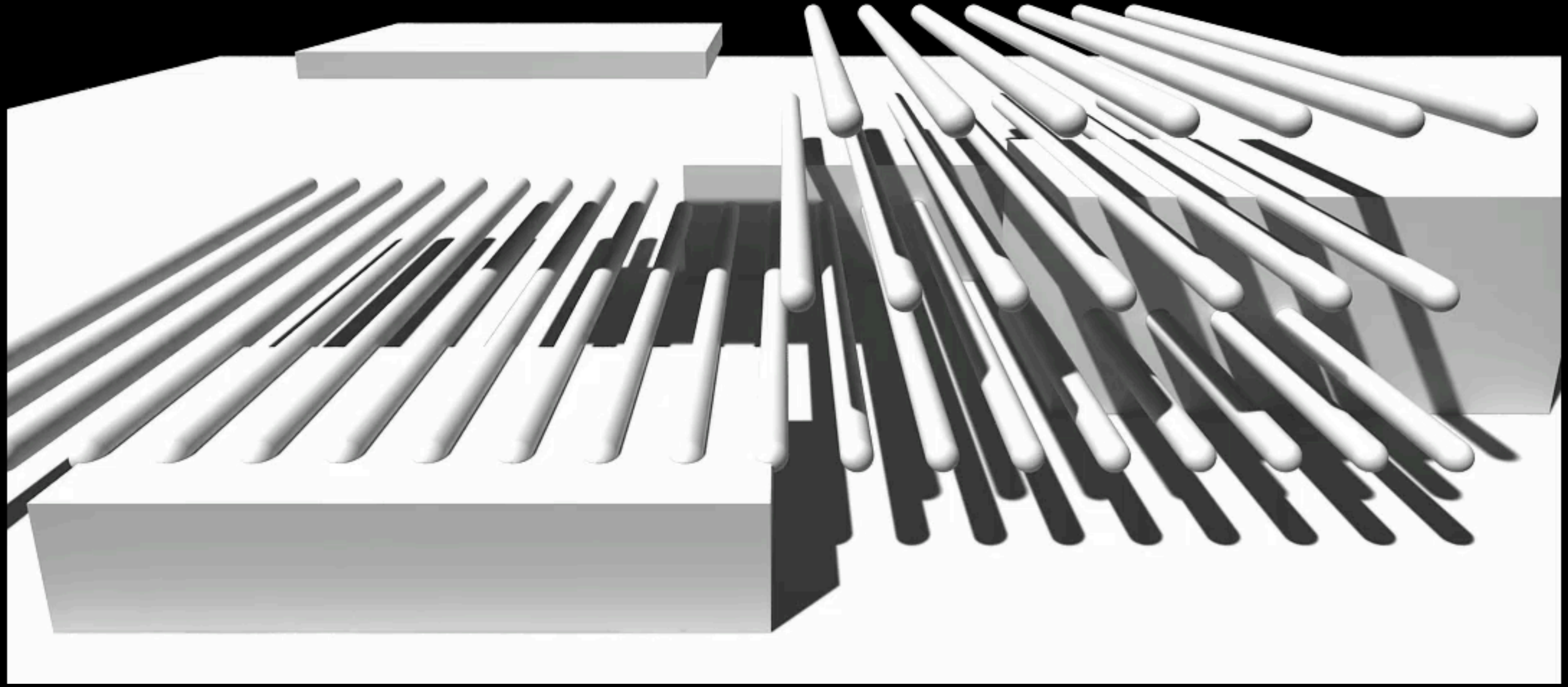
Goldenthal et al. 2007

Fracture

https://www.youtube.com/watch?v=eB2iBY-HjYU

Müller et al. 2013

The simplest physics-based characters :)



Lewin et al. 2013

https://researchportal.bath.ac.uk/en/publications/rod-constraints-for-simplified-ragdolls

Next class

We'll start simple: **Particle system** = collection of (usually non-interacting) particles in motion