Lecture 20:

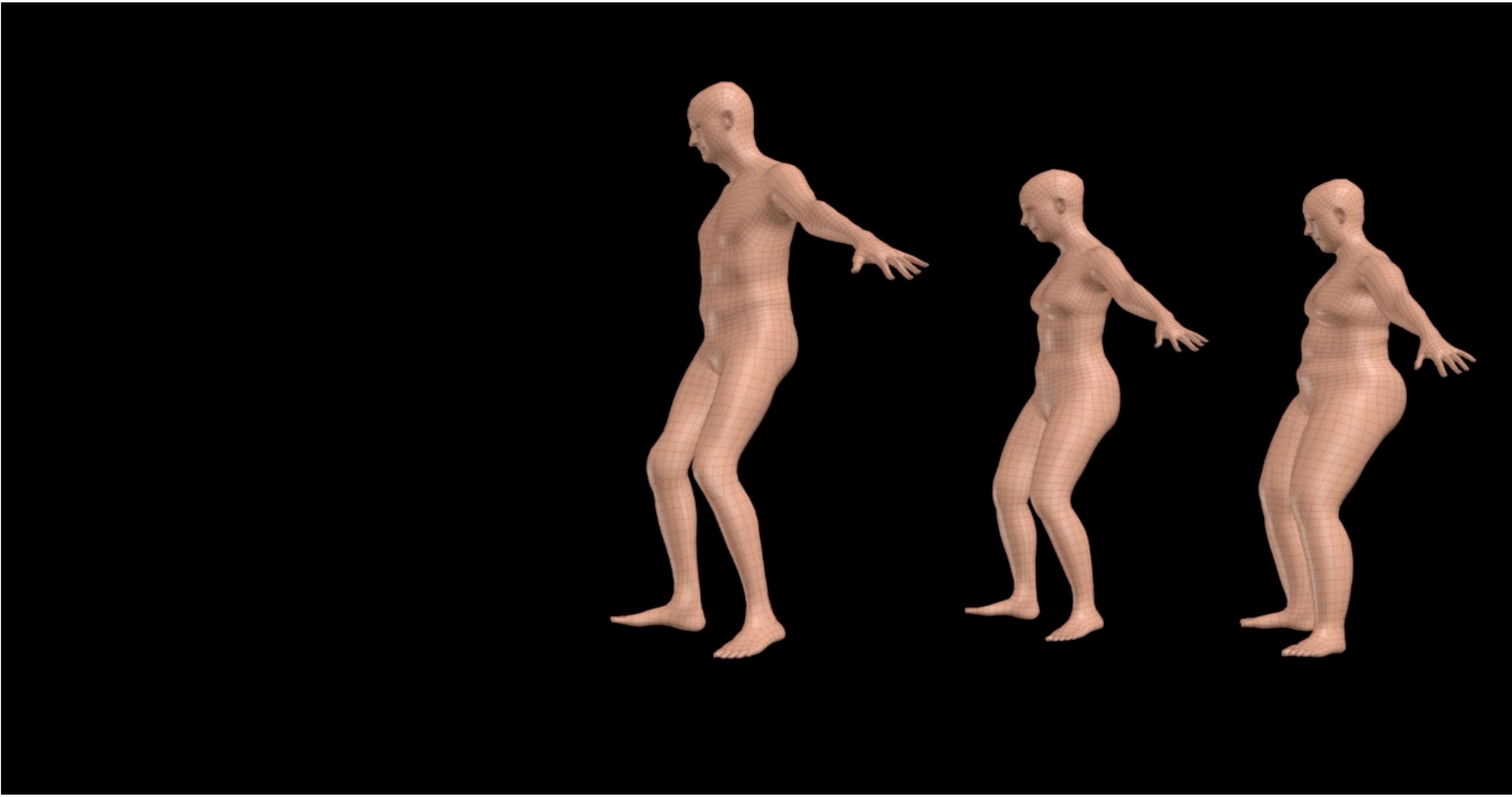
Introduction to Character Animations

Computer Graphics and Imaging
UC Berkeley CS184/284A

The majority of these slides courtesy of Michael Black, Gerard Pons-Moll, Alec Jacobson, Leon Sigal

Last Lecture: Physics Simulation (Cloth)

Today: Character animation



How to animate characters?

To tell a story you need characters that move around!

https://www.youtube.com/watch?v=QbhsMLD9Hb0

Content

Forward Kinematics

Skinning

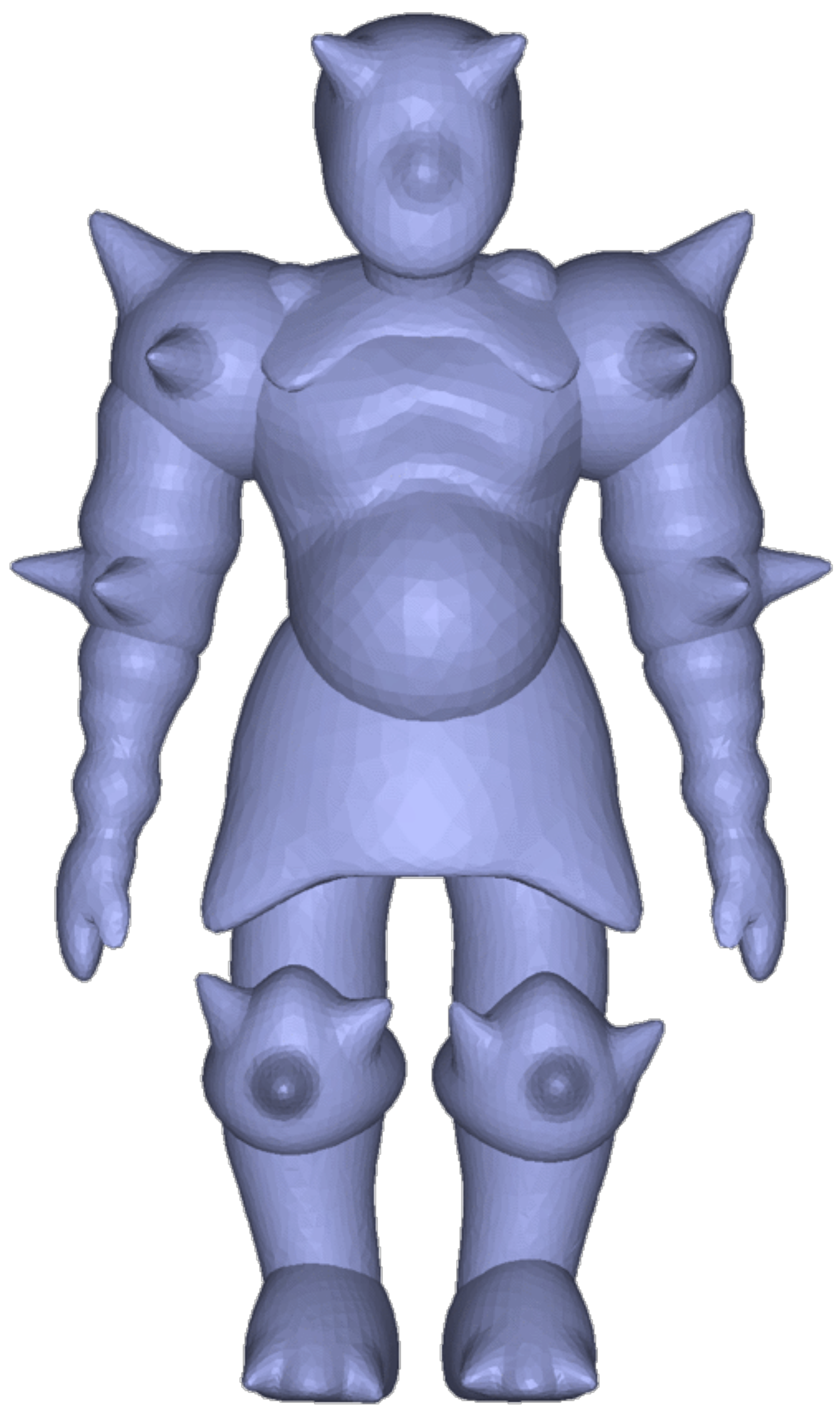
Blend Shapes

Data-driven body models (SMPL)

Motion Capture

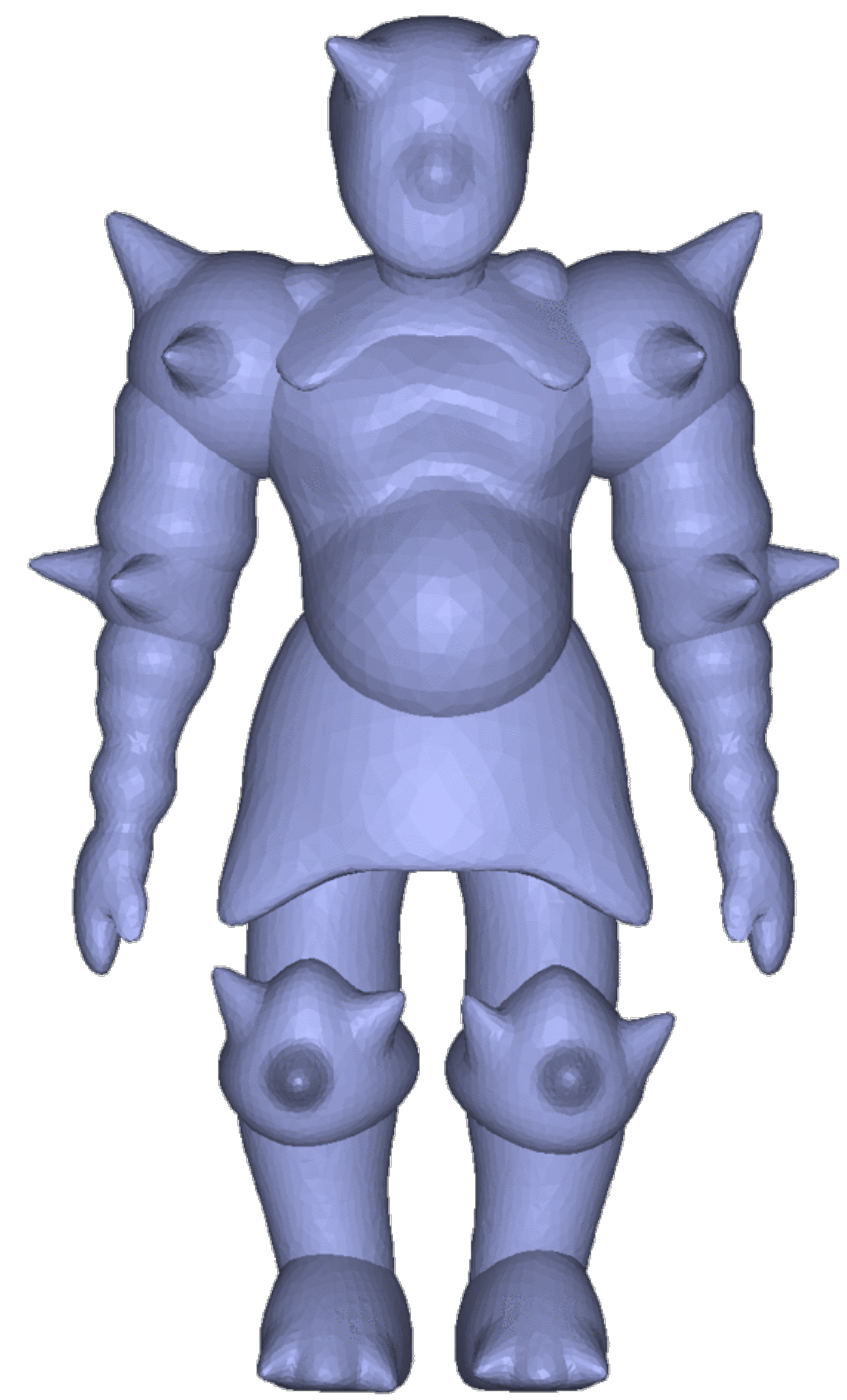
Super SMPL Demo

Modeling 3D Deformation



Full space : # Vertices x 3 DoF

physical simulation,
geometry based deformation

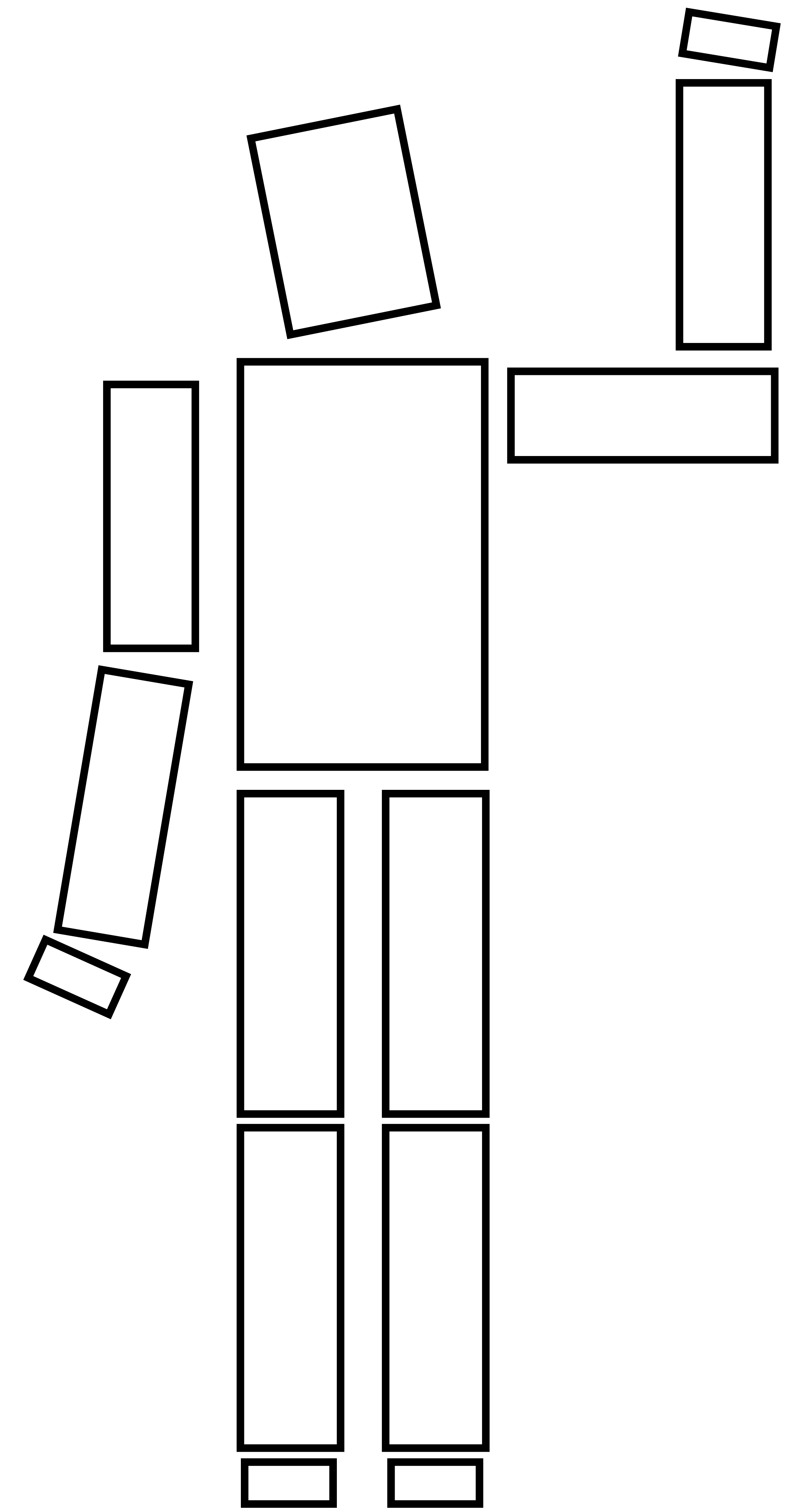


Skinning space: # Bones x [9,6] DoF

character animation

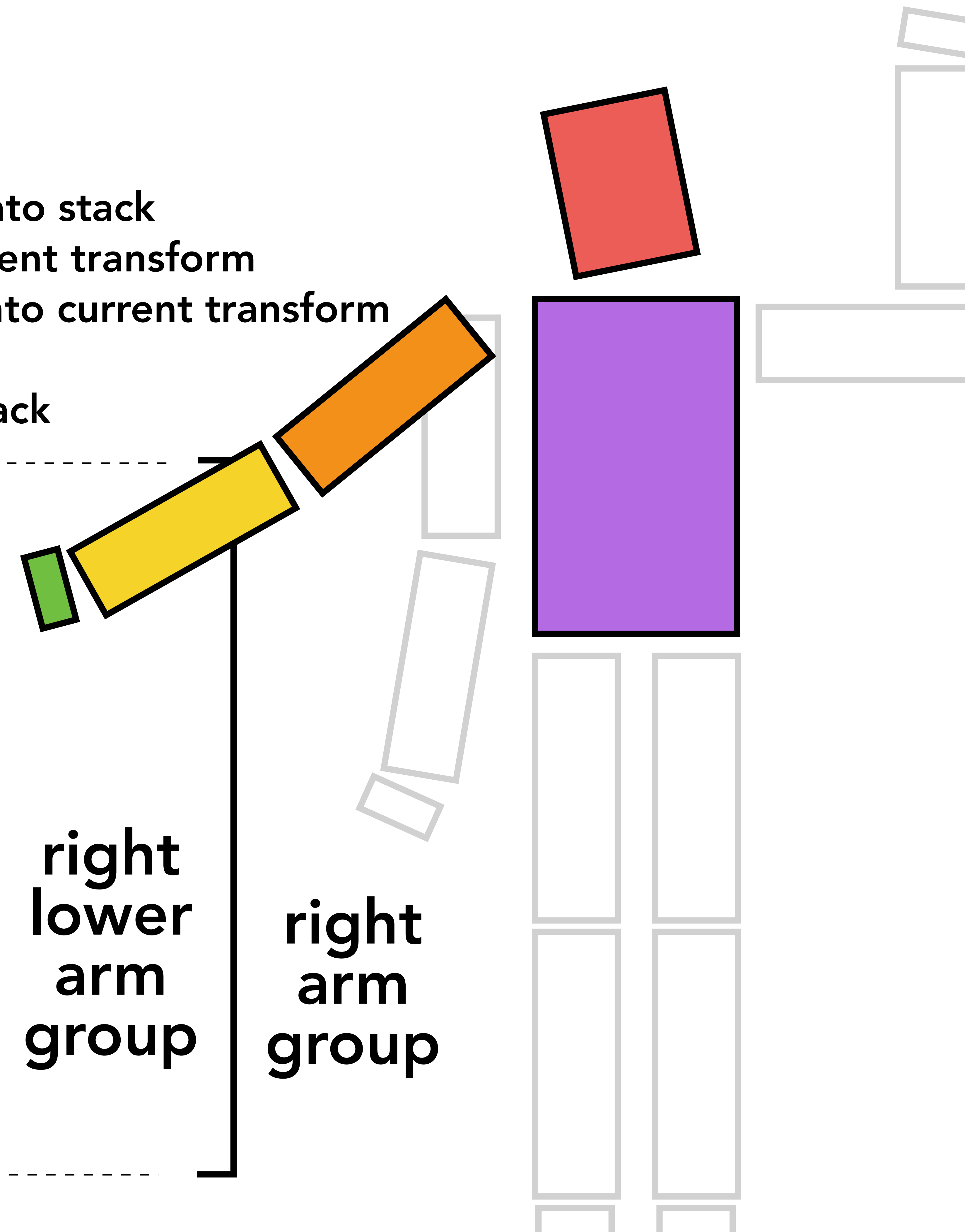
Recall this skeleton from Transforms lecture

torso
- head
- right arm
  - upper arm
    - lower arm
      - hand
- left arm
  - upper arm
    - lower arm
      - hand
- right leg
  - upper leg
    - lower leg
      - foot
- left leg
  - upper leg
    - lower leg
      - foot



Skeleton - Hierarchical Representation

```
translate(0, 10);
    drawTorso();
        pushmatrix();  // push a copy of transform onto stack
            translate(0, 5);  // right-multiply onto current transform
            rotate(headRotation);  // right-multiply onto current transform
            drawHead();
        popmatrix();  // pop current transform off stack
        pushmatrix();
            translate(-2, 3);
            rotate(rightShoulderRotation);
            drawUpperArm();
            pushmatrix();
                translate(0, -3);
                rotate(elbowRotation);
                drawLowerArm();
                pushmatrix();
                    translate(0, -3);
                    rotate(wristRotation);
                    drawHand();
                popmatrix();
            popmatrix();
        popmatrix();
        ....
```



Forward Kinematics

Forward Kinematics

- Describes the motion of articulated character
- Given relative joint angles, compute the absolute bone transformation to apply to the bone
- Animation = Angles over time

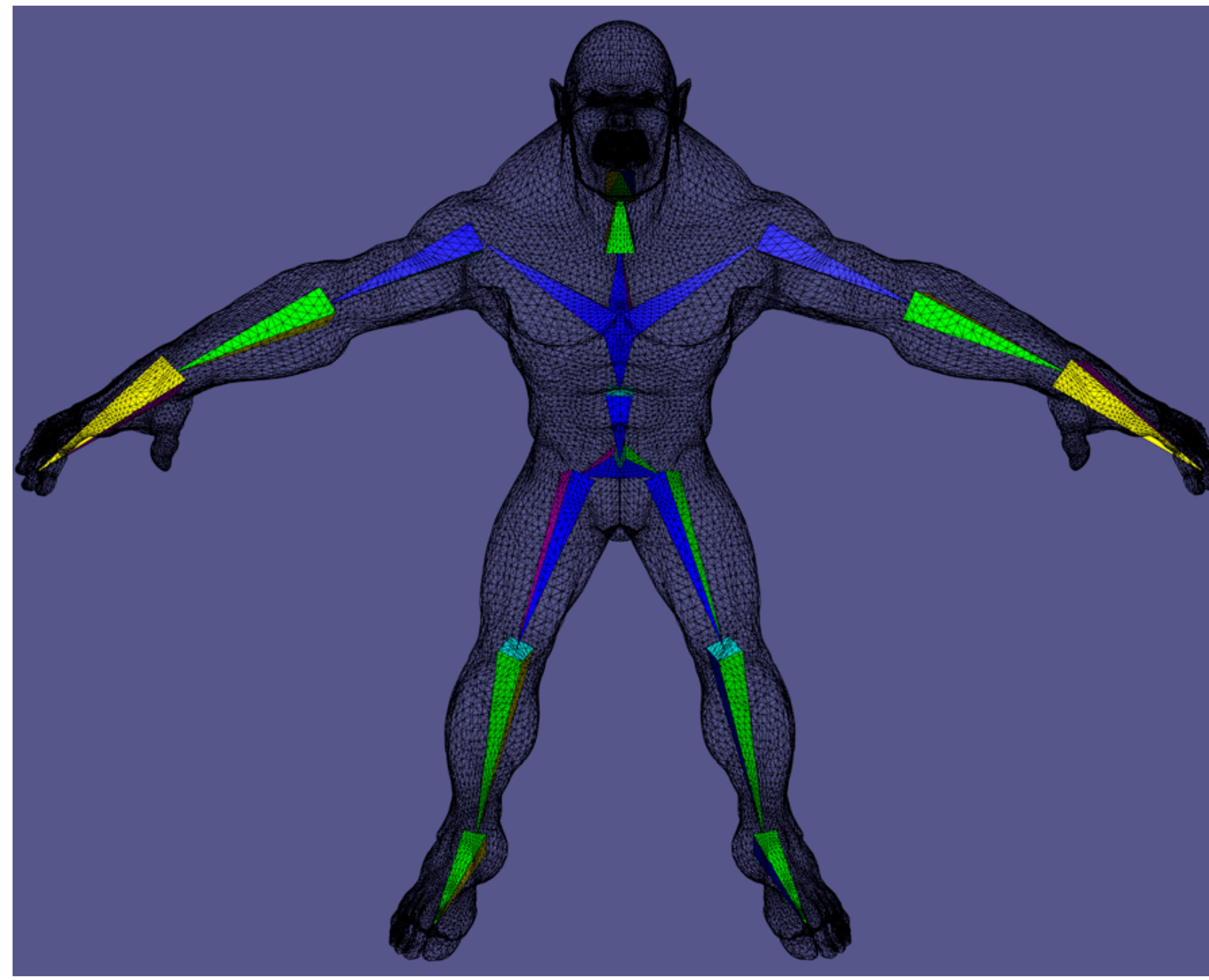
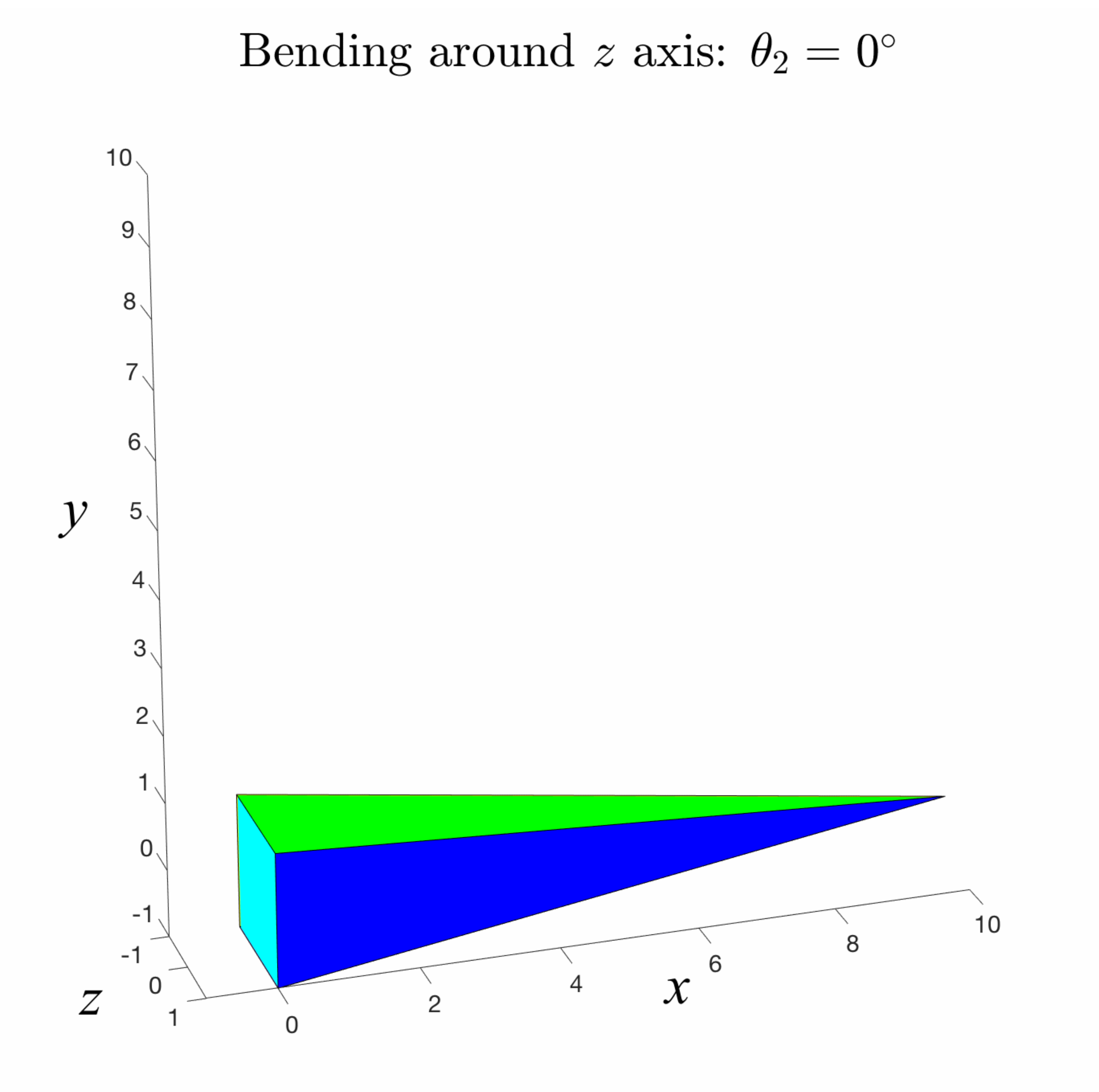


Figure courtesy of Alec Jacobson

“Canonical” Bone

Each bone has its own coordinate space

At origin with length l and (relative) rotation R



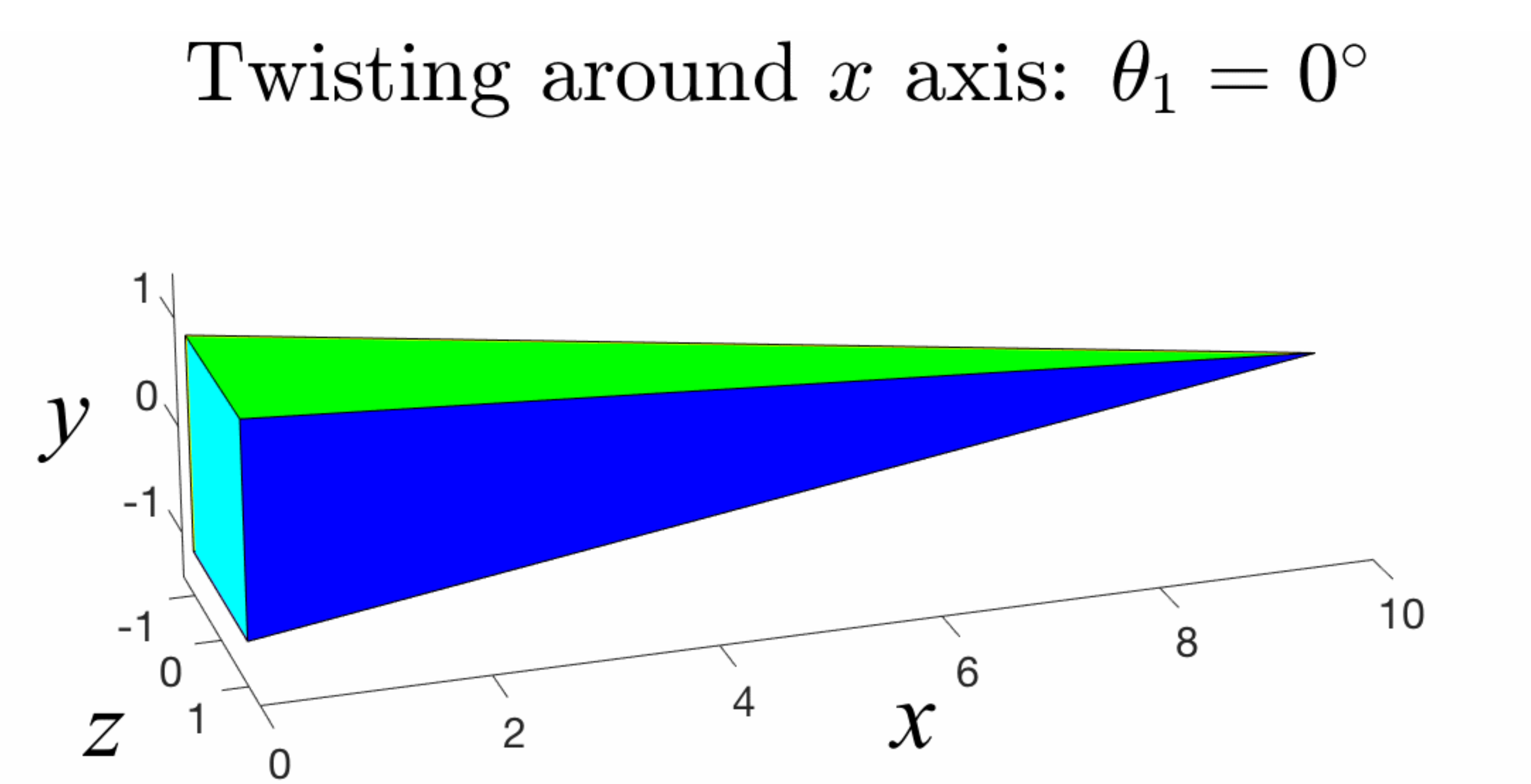


Figure courtesy of Alec Jacobson

Rest Bone / T-Pose

Each bone has a pre-set "undeformed" configuration

This is the configuration when all relative rotation is Identity, hence "rest" or T-pose.

Defined by end points of the bones

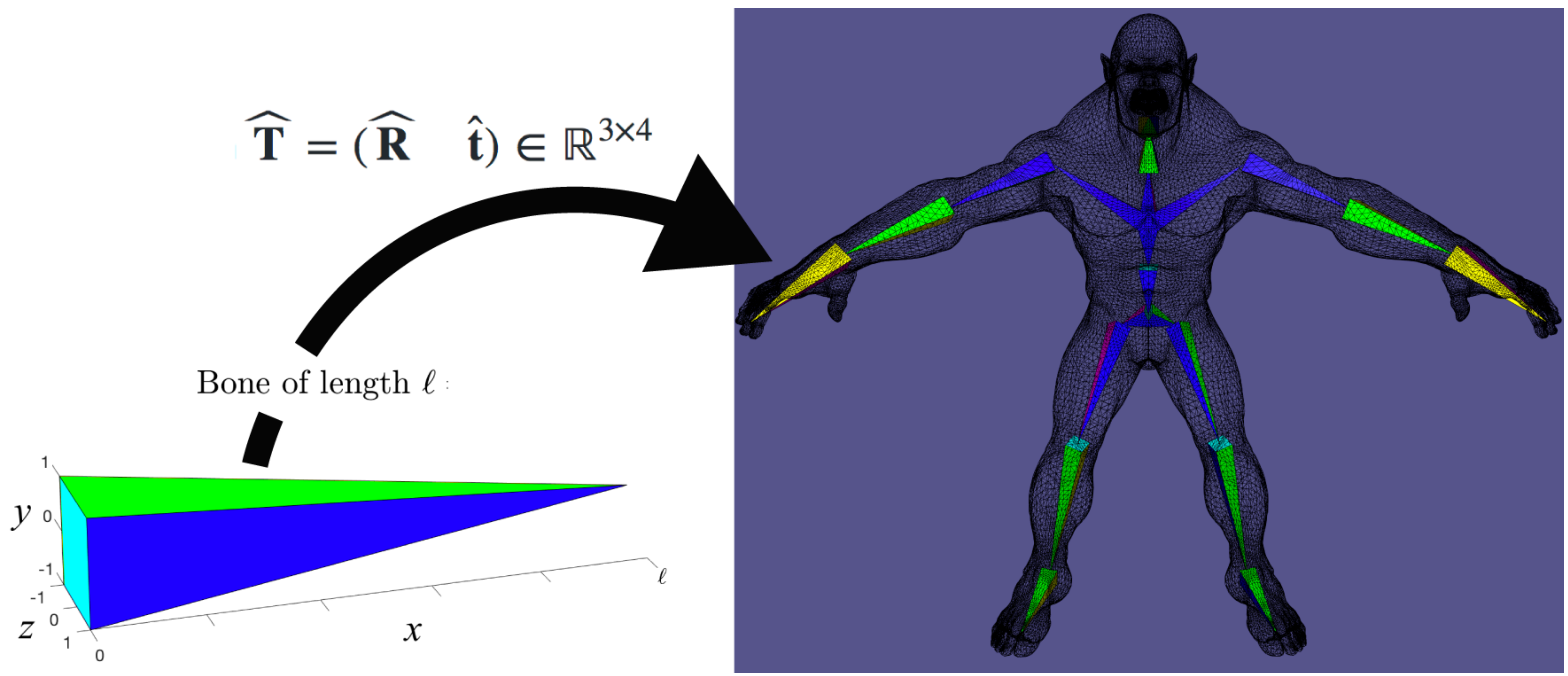


Figure courtesy of Alec Jacobson

Compute Absolute Transformations

Apply the relative transform, then apply the absolute transform of the parent.

Tricky! Need to apply the relative rotation in each bone's "canonical" coordinate space, then apply the rest transformation

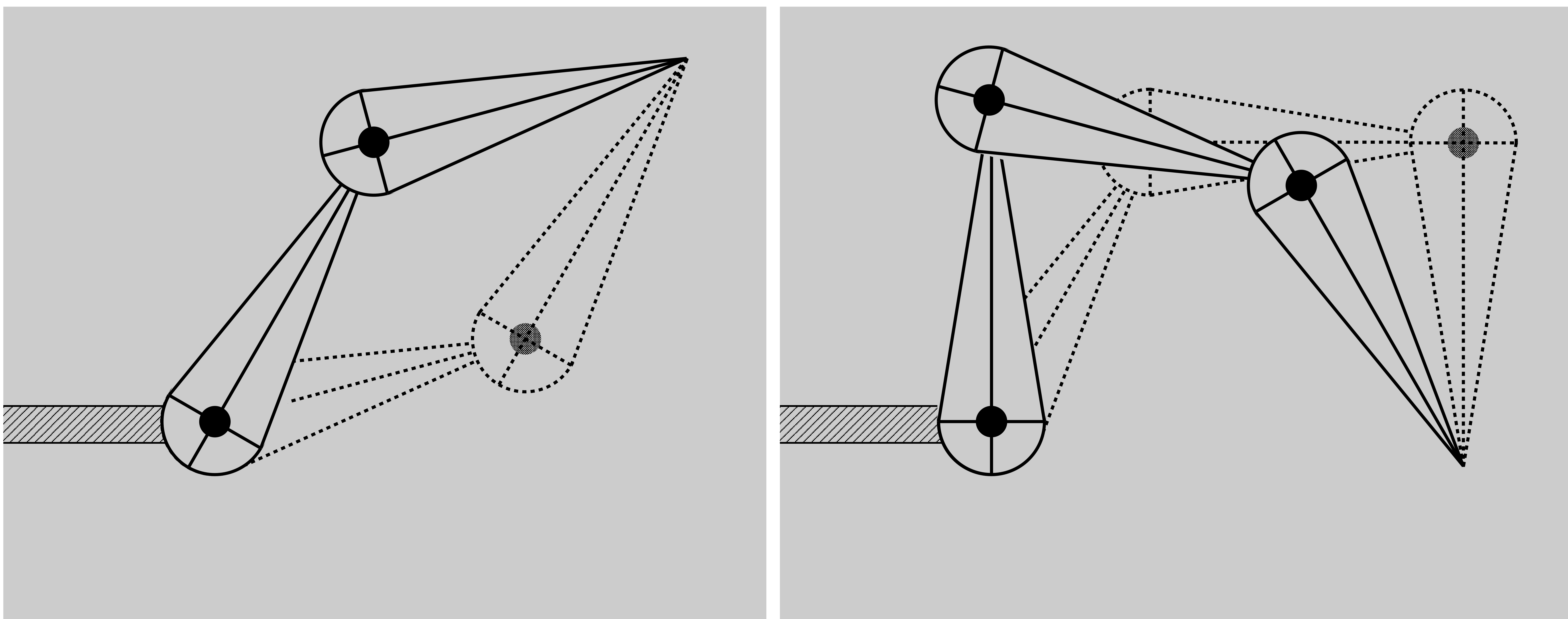
$$T_i = T_{p_i} \hat{T}_i R_i \hat{T}_i^{-1}$$

https://www.alecjacobson.com/weblog/?tag=forward-kinematics

Inverse Kinematics

What is the joint angles given the target location?

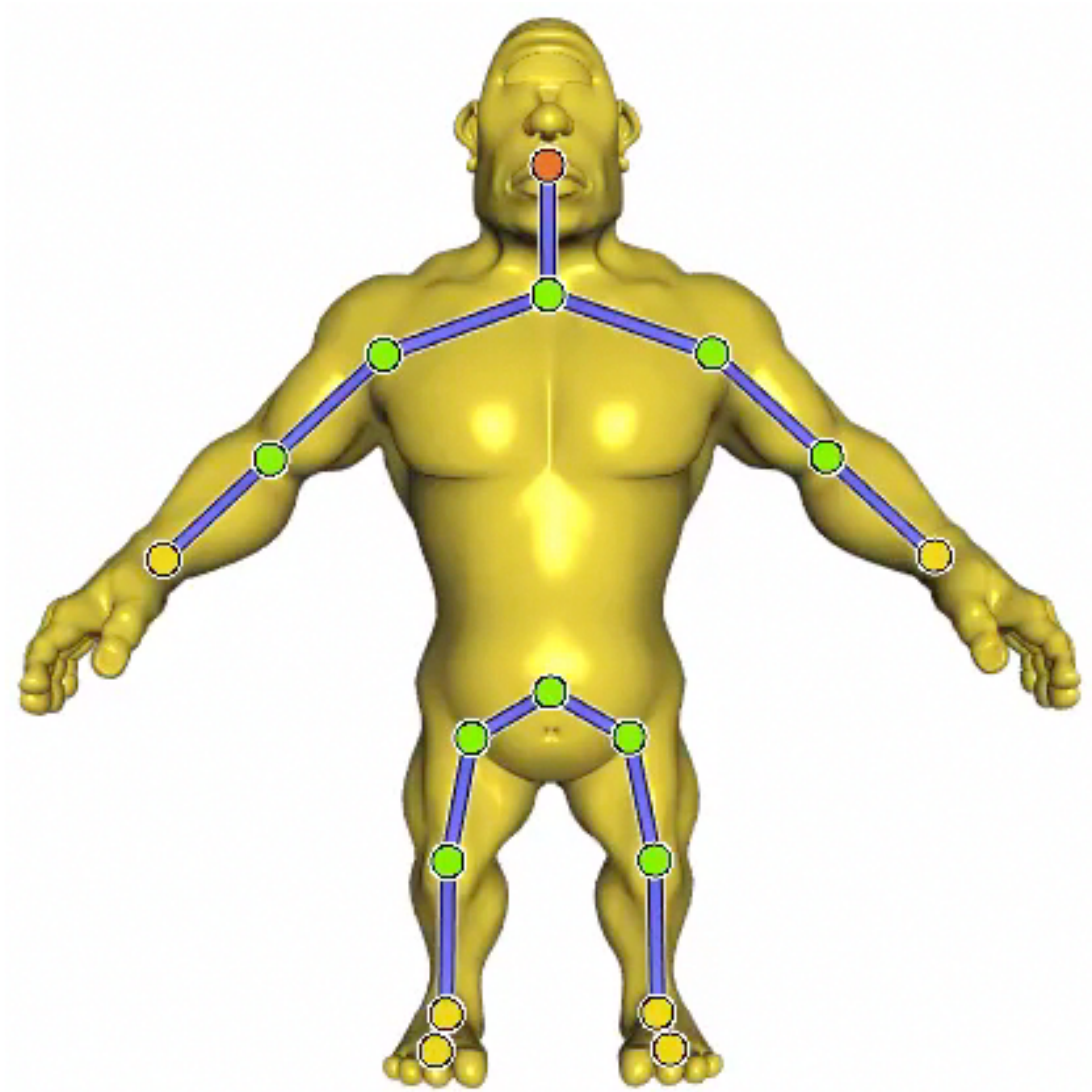
Inverse problem:



Skinning

Skinning

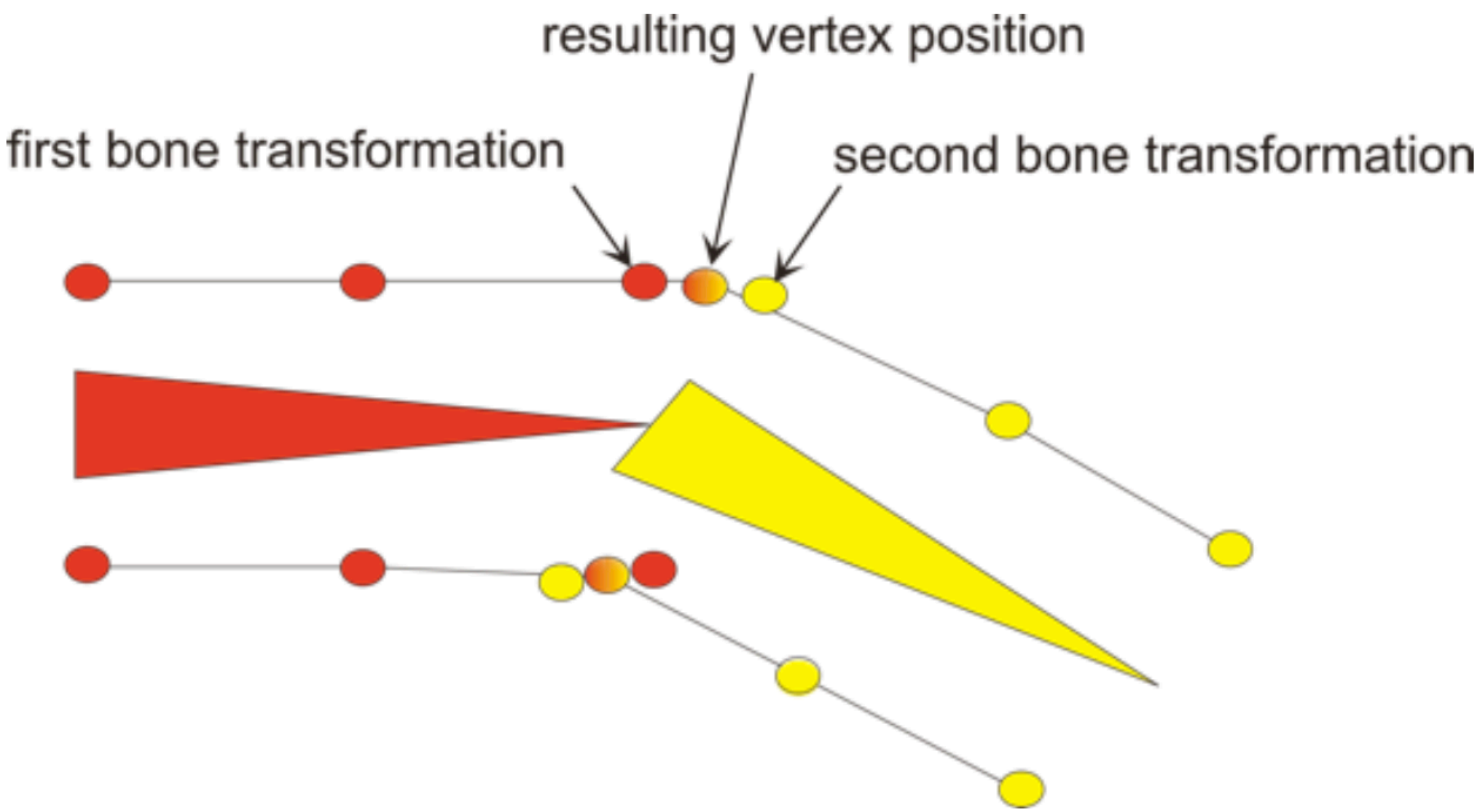
Goal: move the surface along with assigned bones or "handles"



Courtesy of Alec Jacobson

Basic Idea

1. Transform each vertex with each bone rigidly
2. Blend the results using weights, or assignments



Linear Blend Skinning (LBS)

Blend contribution linearly.

Super simple to implement. Great for real time.

How much influence this bone has on v (often sparse)

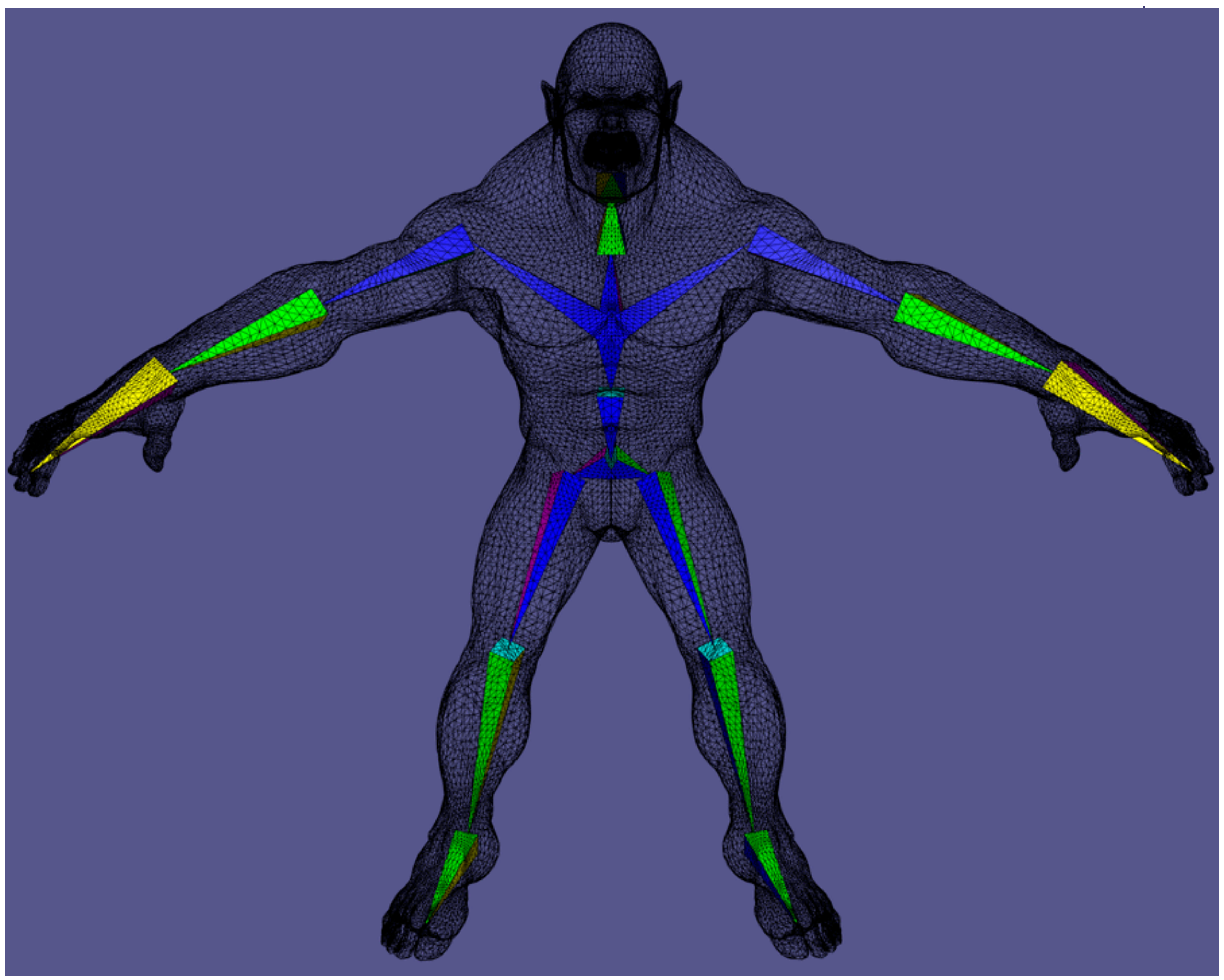
Bone j transformation

$$\mathbf{v}' = \sum_{j \in H} w_j(\mathbf{v}) \mathbf{T}_j \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}$$

New vertex

Original vertex

Illustration of rig & skinning weights

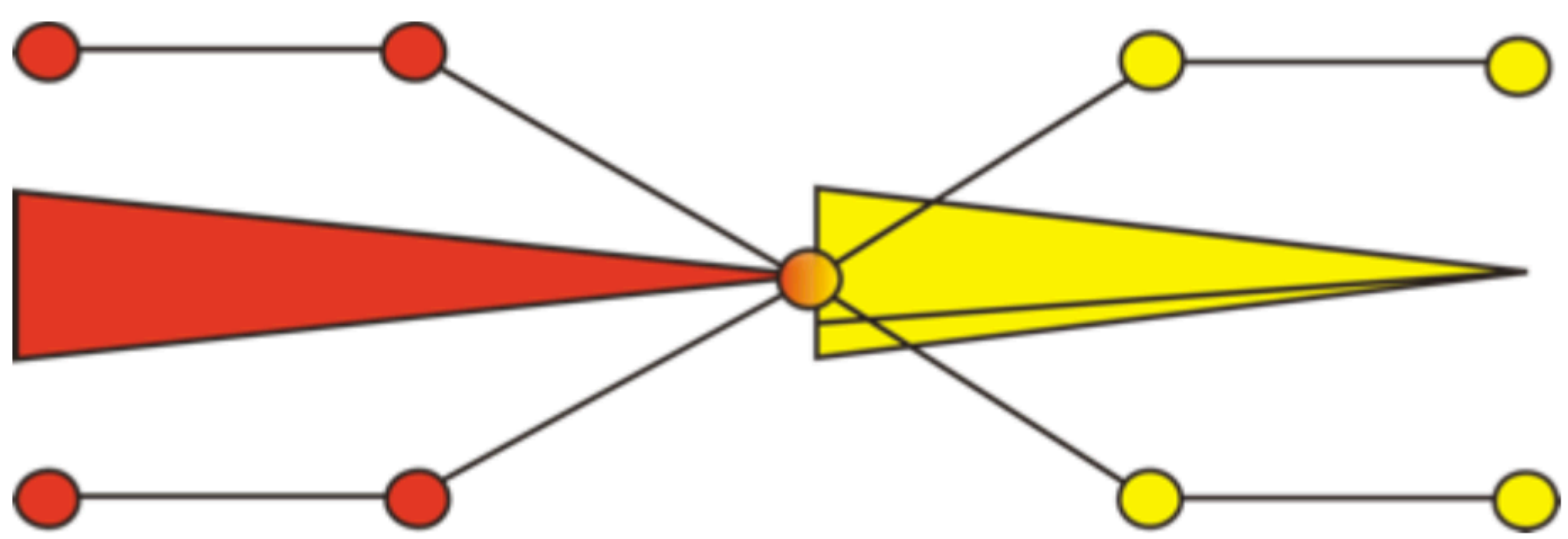


Bone transformations

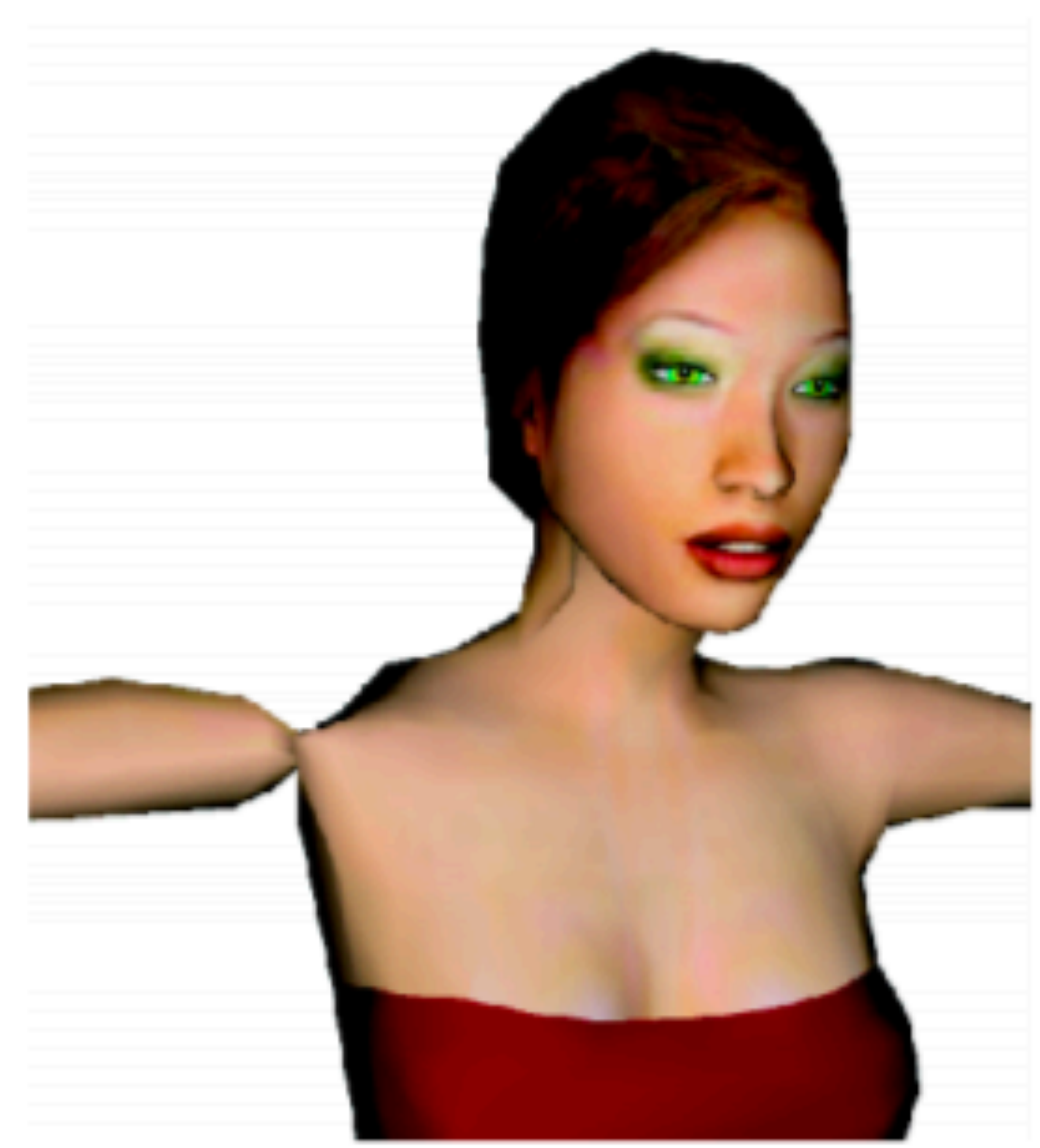
Skinning Weights

Problems with LBS

When joint rotates 180 degrees



"Candy Wrapper" artifact

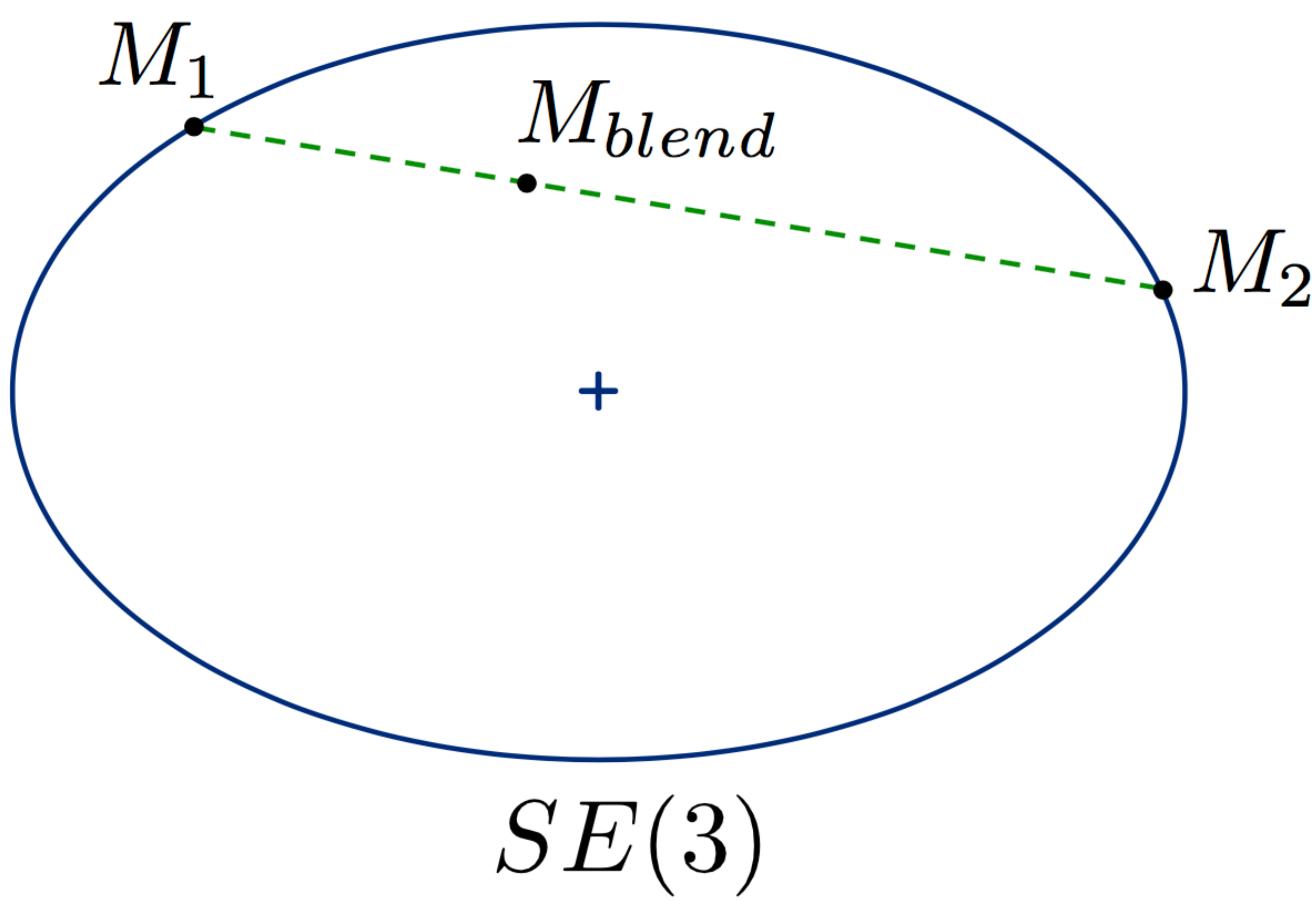


Why?

Can't linearly combine rigid transformations!

Better methods exist like Dual Quaternion Skinning [Kavan et al. TOG 2008]

Still LBS is most popular for simplicity and is the de facto industry standard



Process

$$\mathbf{v}' = \sum_{j \in H} w_j(\mathbf{v}) \mathbf{T}_j \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}$$

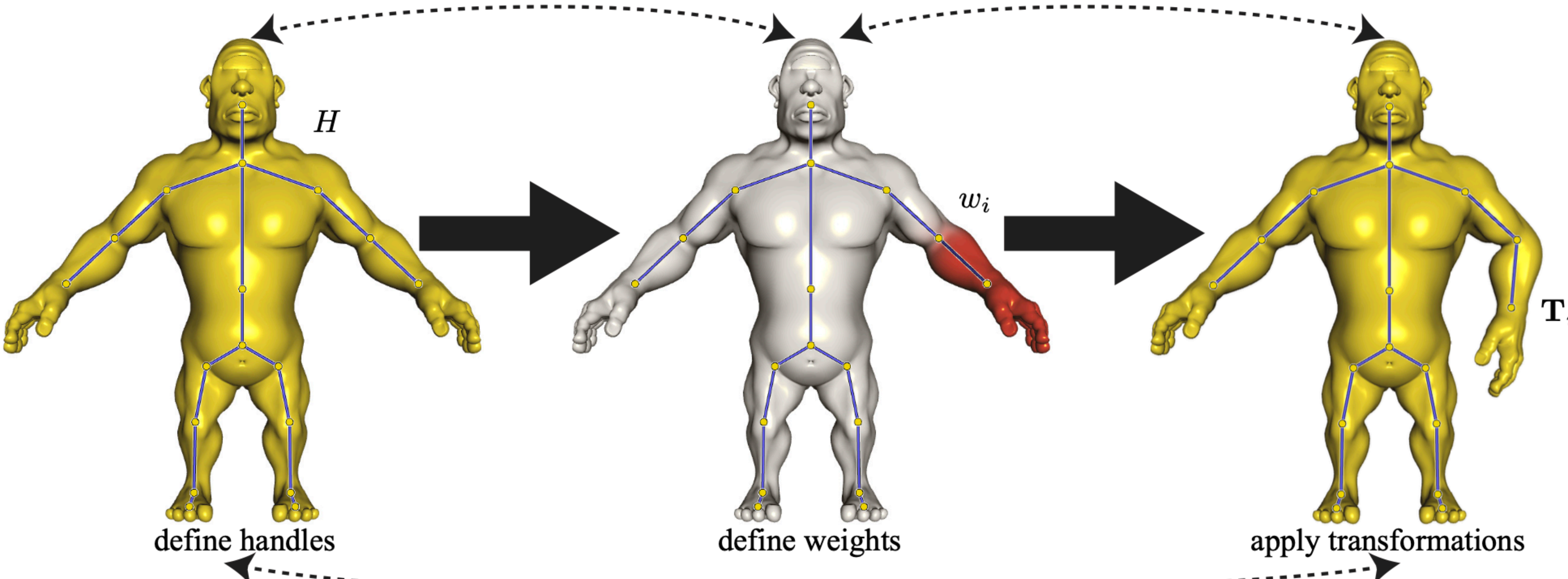
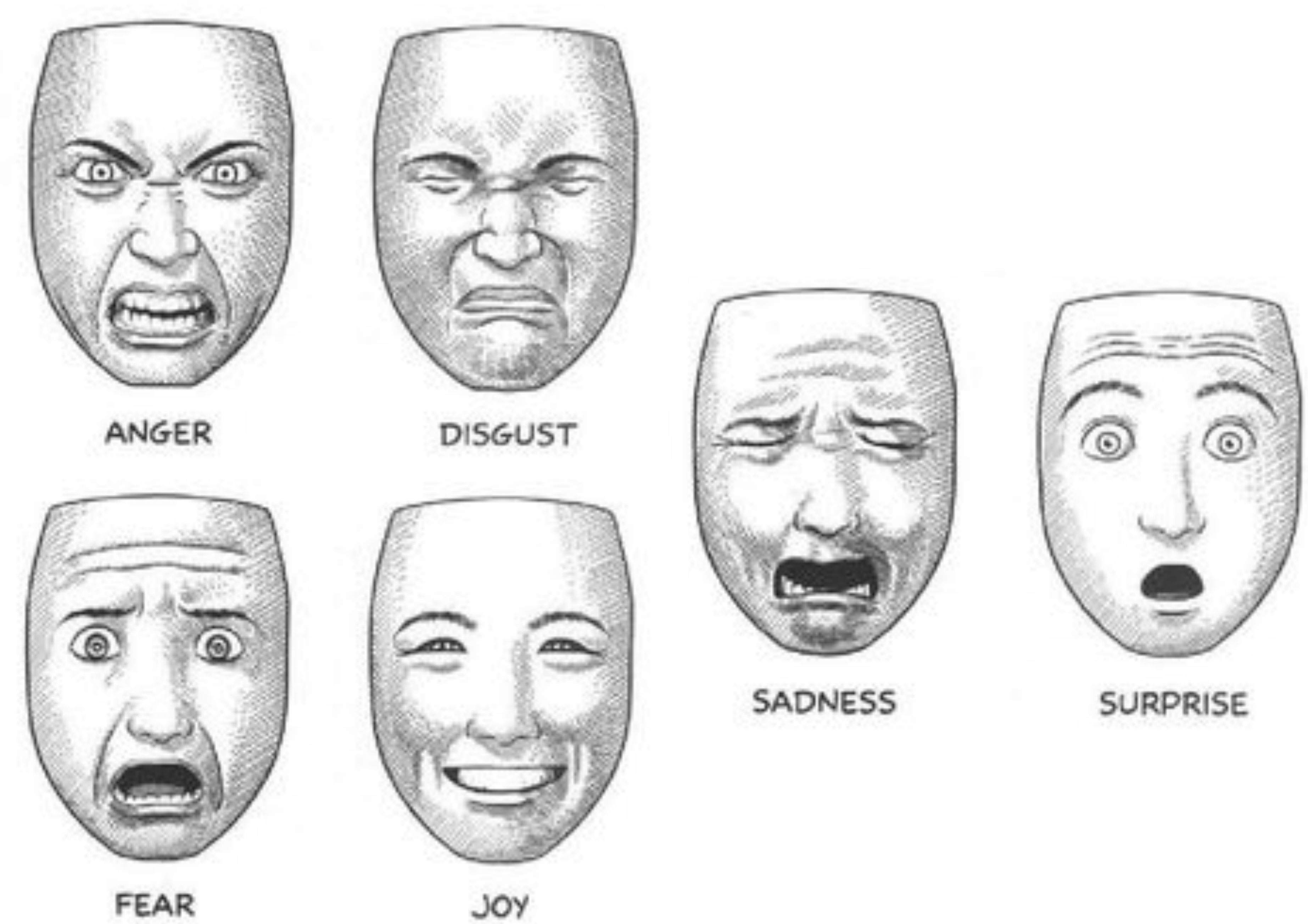


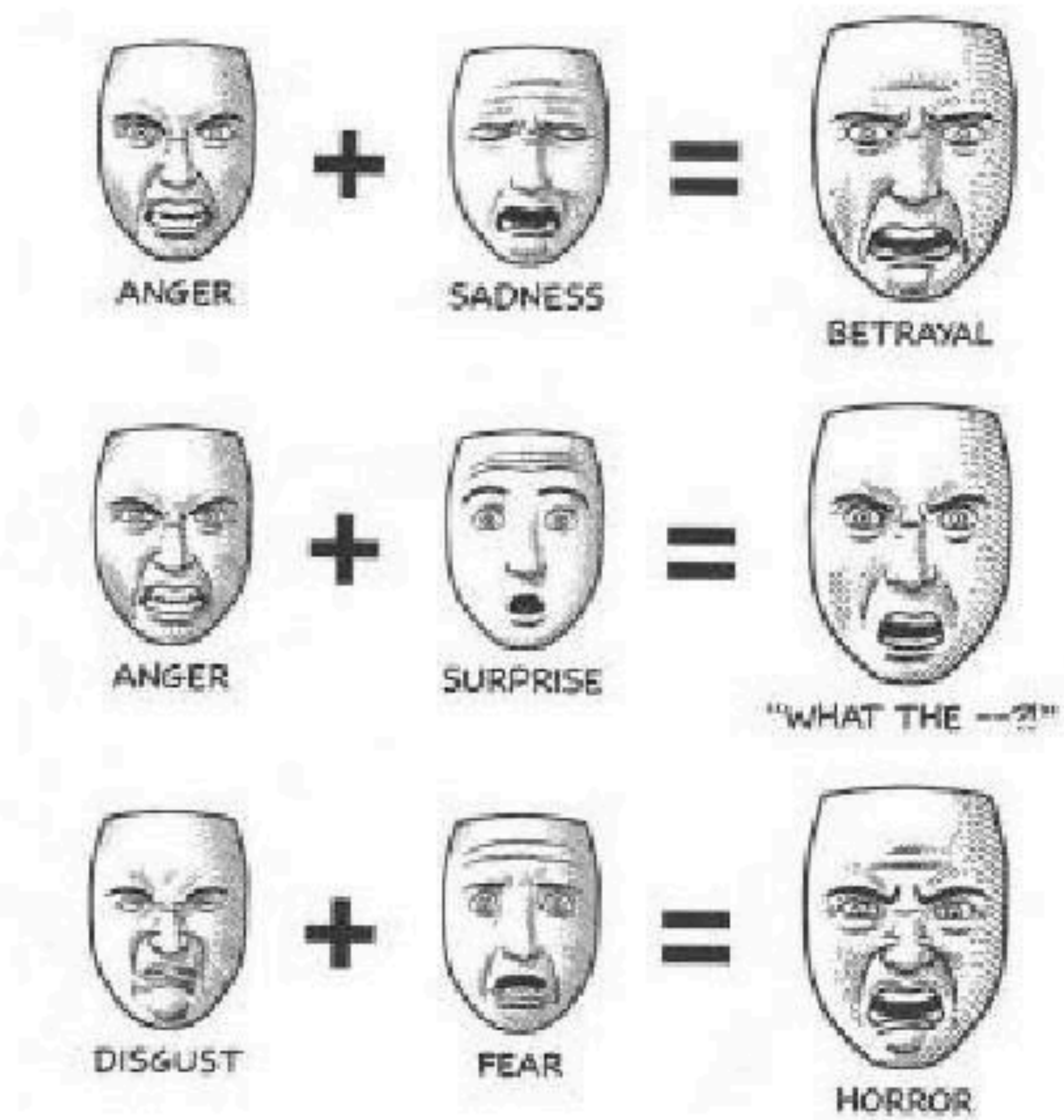
Figure credit: Alec Jacobson

Blend Shapes

Blend Shapes

Not all deformation is from bones.
Interpolate surfaces between key shapes



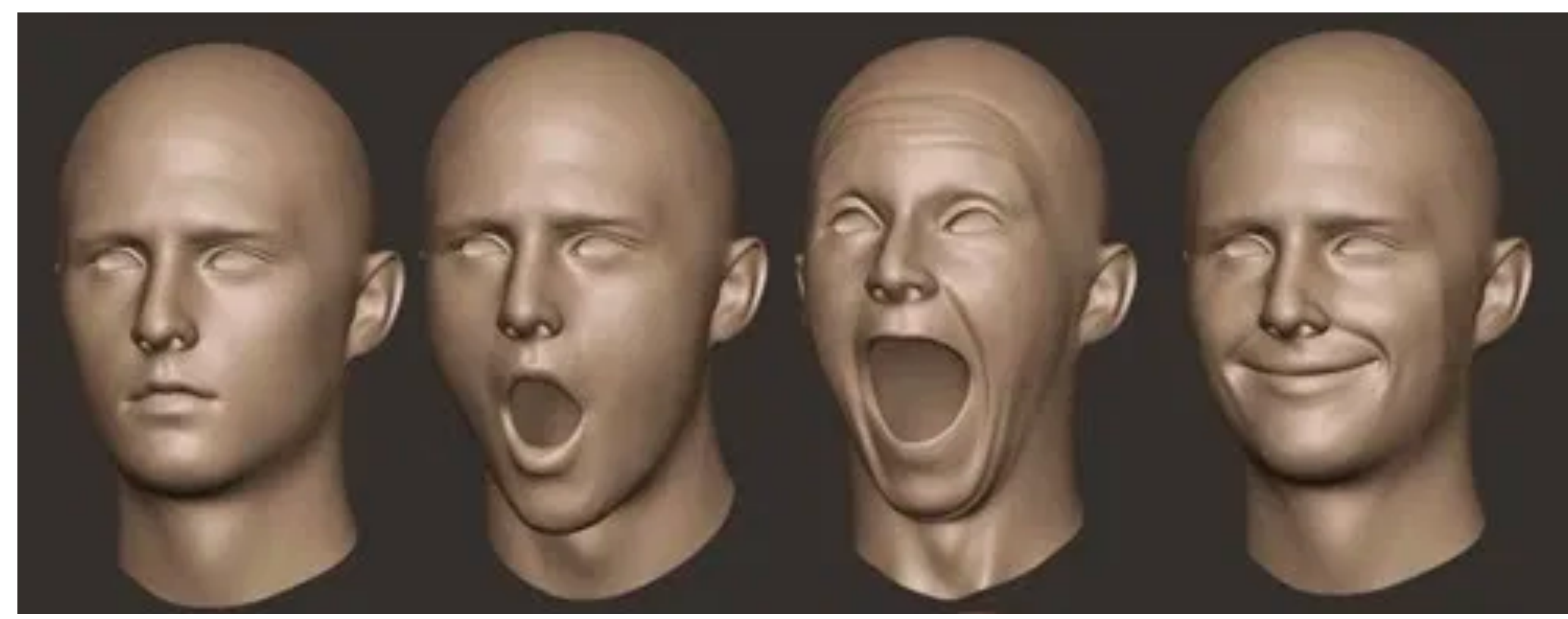


Making Comics: Storytelling Secrets Of Comics, Manga, and Graphics Novels by Scott McCloud

Blend Shapes

- A set of vertex offsets to neutral shape
- Linearly interpolate these key blend shapes for control
- Often used for expressions
- Works for deformations that are linear, i.e. the average of two shapes is a valid shape

$$B = \mathrm{vec}\left(\begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta z_1 \\ & \vdots & \\ & \vdots & \\ \Delta x_N & \Delta y_N & \Delta z_N \end{bmatrix}\right)$$



$$V = \sum_i \beta_i B_i$$

Blend Shapes

Courtesy Félix Ferrand

Rigging

Rigging

Augment character with controls to
easily change its pose, create facial expressions, bulge muscles, etc.

Rigging is like the strings on a marionette.

Capture space of meaningful deformations.

Varies from character to character.

Skeleton is ONE type of rigging

Other examples of handles

$$\mathbf{v}' = \sum_{j \in H} w_j(\mathbf{v}) \mathbf{T}_j \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}$$

Example of a sophisticated rig

Courtesy of Matthew Lailler

Modeling Pose and Shape

In general:

Pose (articulation) is by forward kinematics

Shape ("identity") is by linearly blending shapes

Data Driven Models

People are special

Perceiving Systems, Max Planck Institute

We can learn a body model from data

Parameters to learn:

- Skinning weights $\mathbf{W}$
- Statistical model of body shapes (surfaces & joints)
  - Data-driven shape blend shapes

$$V_s = \Sigma \beta_i B_i$$

Vertex positions in rest pose

- Pose blend shapes

$$V_p(\theta)$$

Training Data

* This requires non-trivial amount of pre-processing,

- CAESER dataset: ~2000 meshes per gender
- All *aligned* to a template mesh (same number of vertex, vertex-to-vertex corresp)
- Normalized to the same rest pose

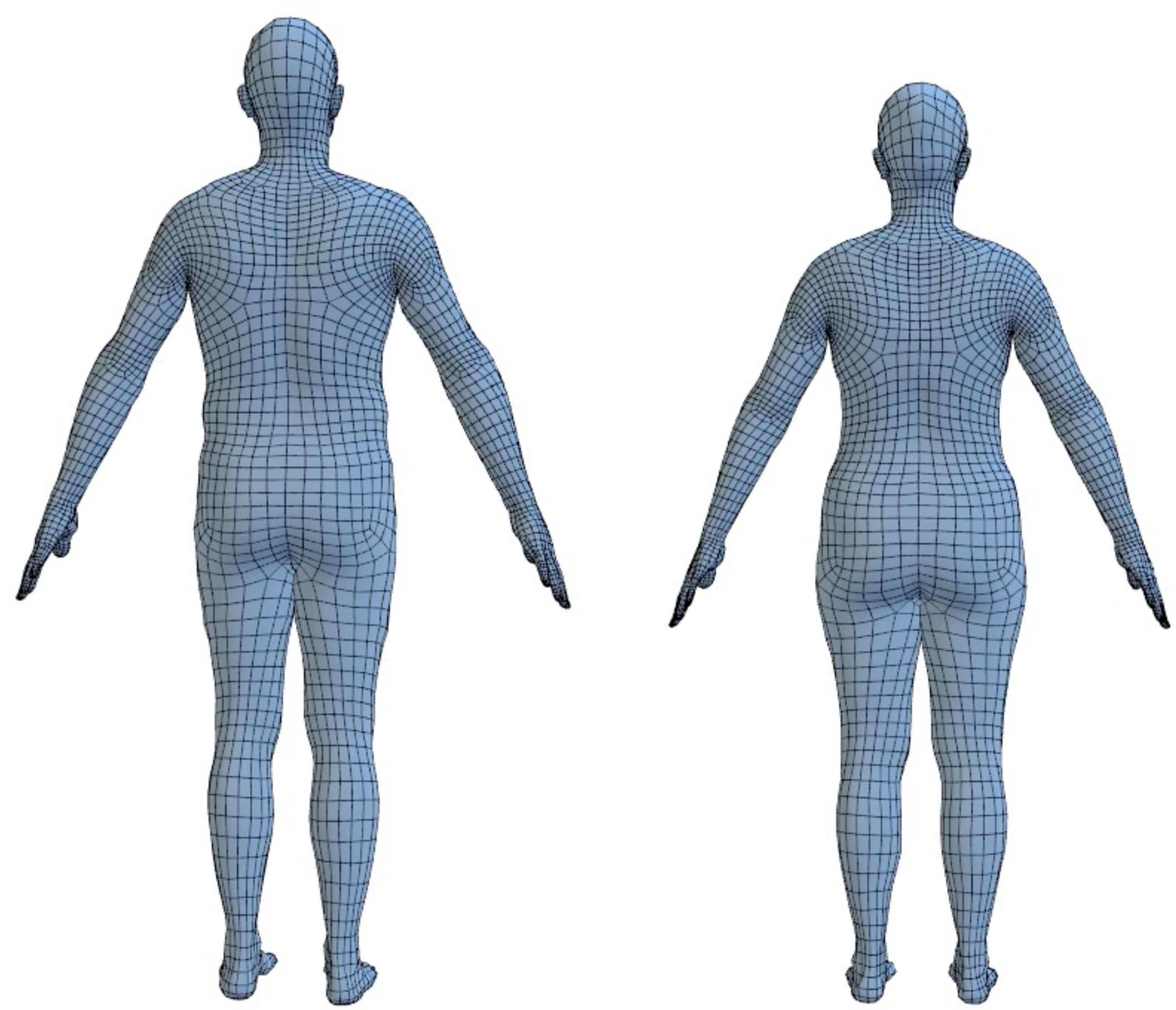
Statistical Analysis of Body Shapes

We have a data of aligned people in the exact same pose.

$$\{V_1, V_2, \cdots, V_n\}, V \in \mathbf{R}^{3\times|V|}$$

Now we can ask questions like: what is the average?

Average shape of ~2000 men and women in US and Europe



Figures Courtesy of Michael Black

Principal Component Analysis

For learning "Shape" Blend Shapes, apply PCA!

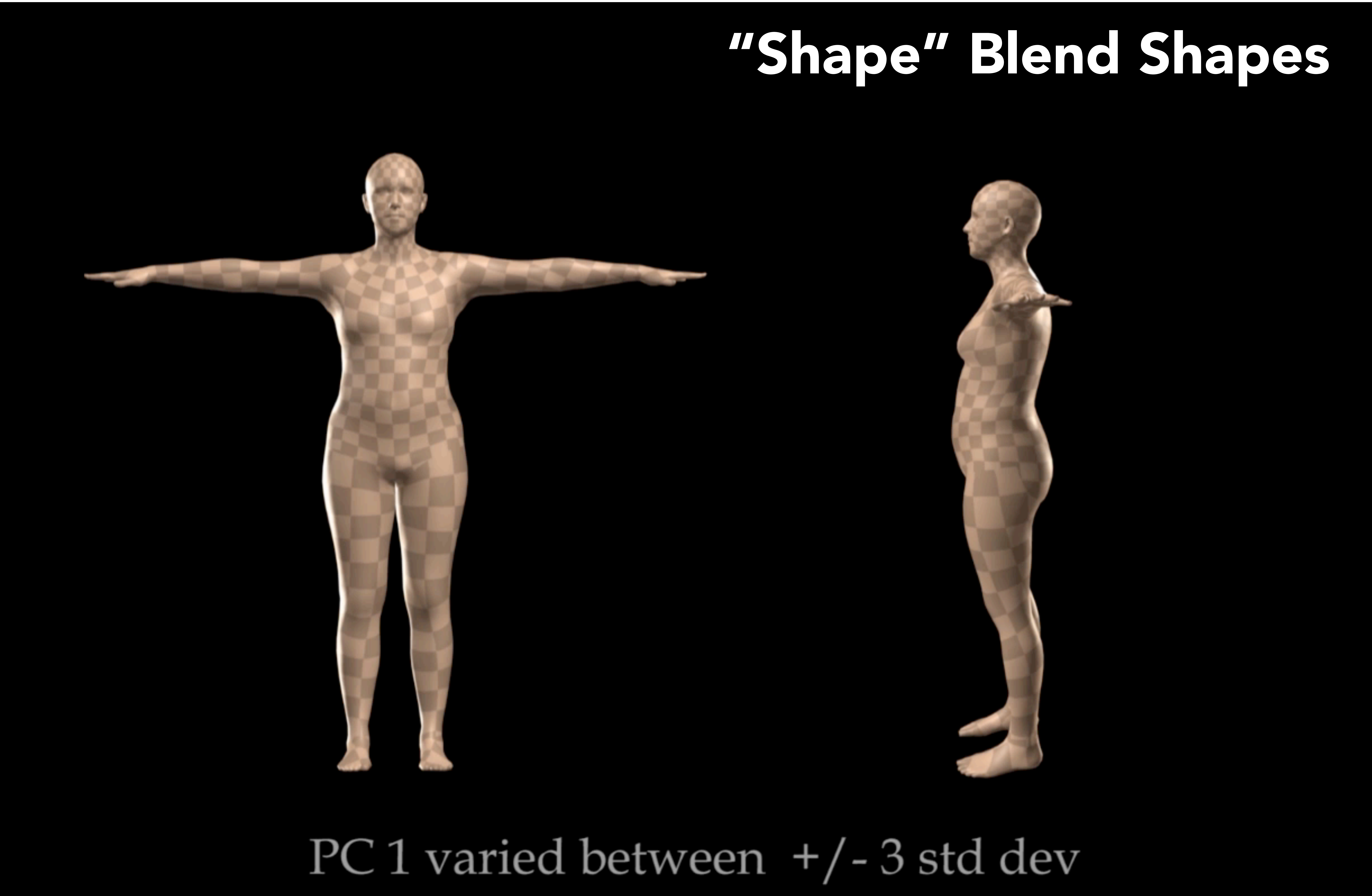
$$\{V_1, V_2, \cdots, V_n\}, V \in \mathbf{R}^{3\times|V|}$$

$$\cong \sum_i \beta_i B_i + \bar{V}$$

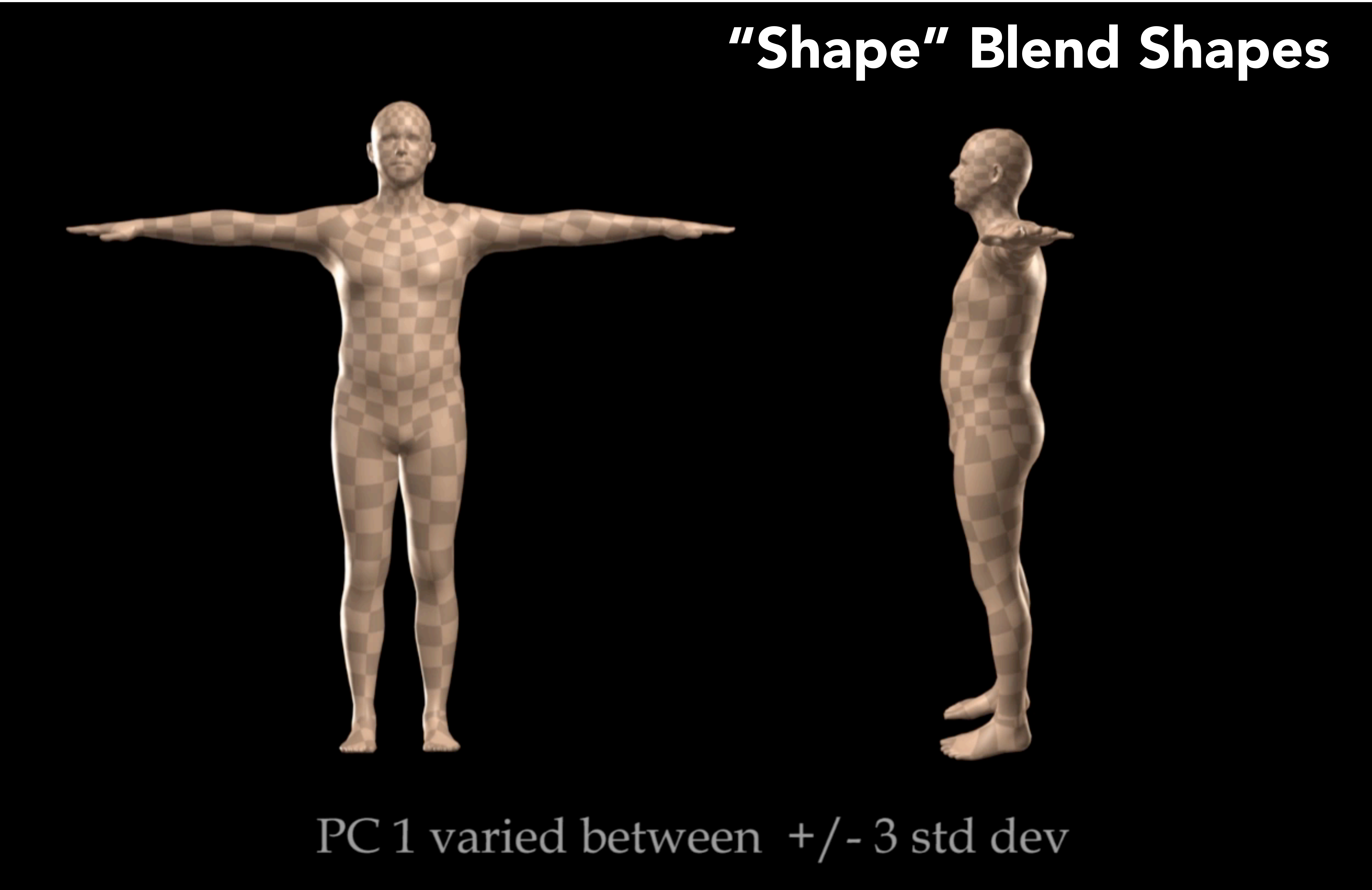
Eigenvalues

Eigenvectors / Blend Shapes

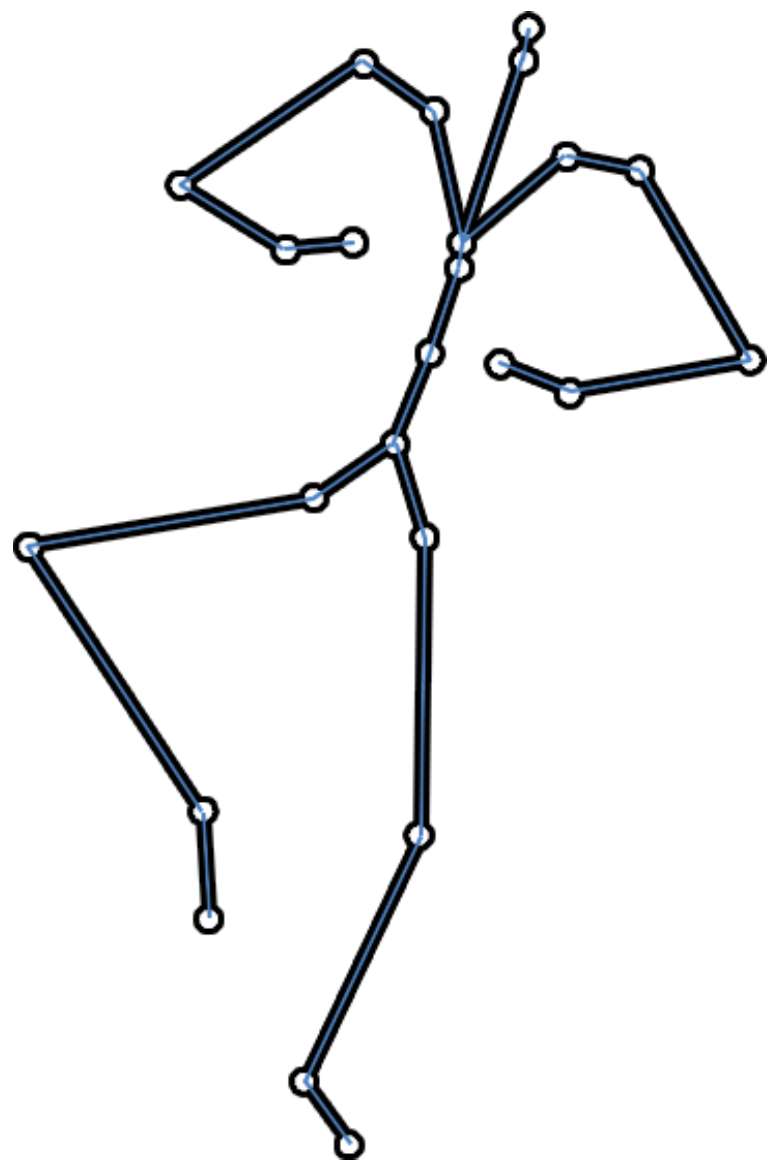
Principal components: Female



Principal components: Male

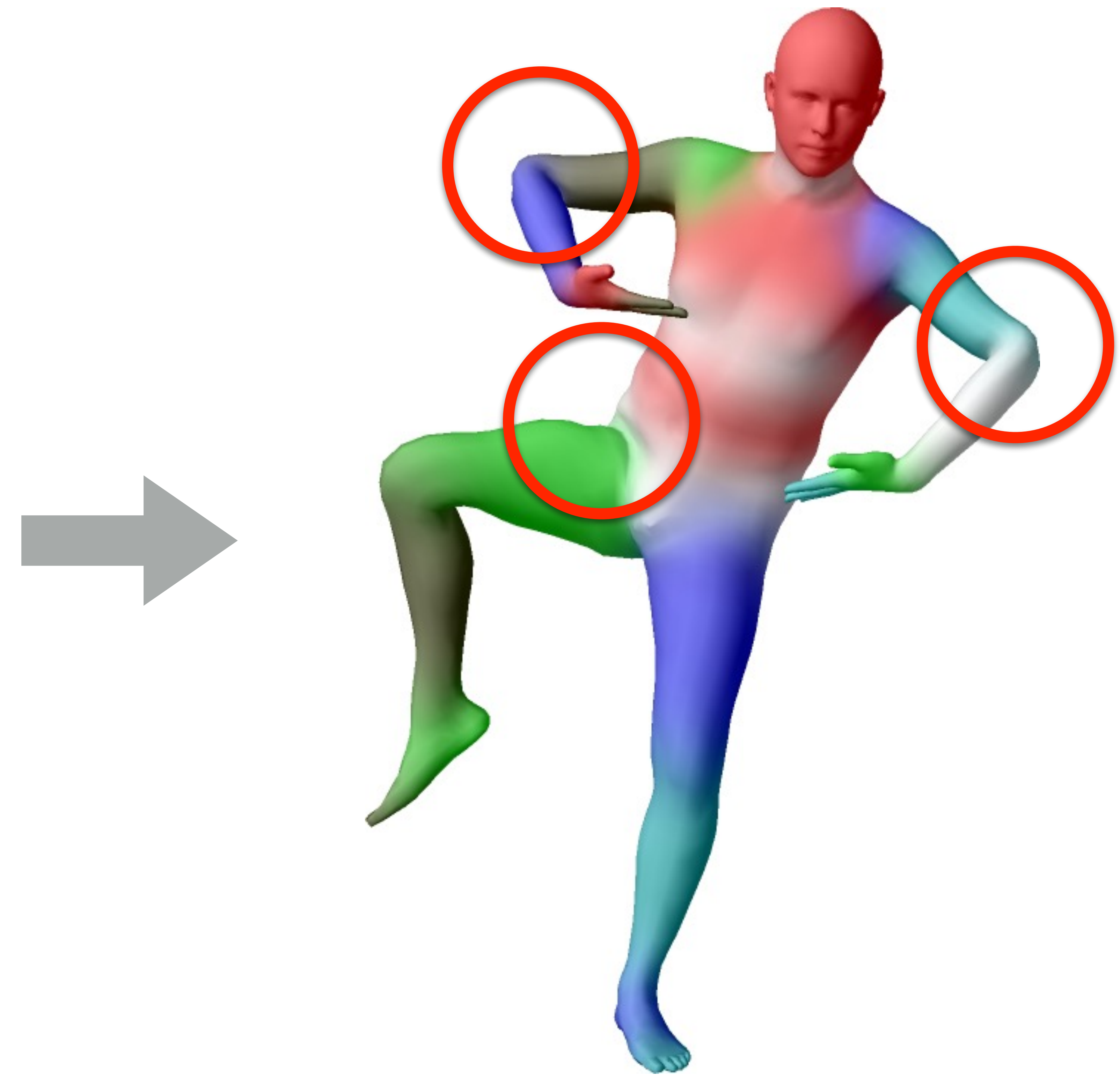


Pose with kinematics + Linear Blend Skinning



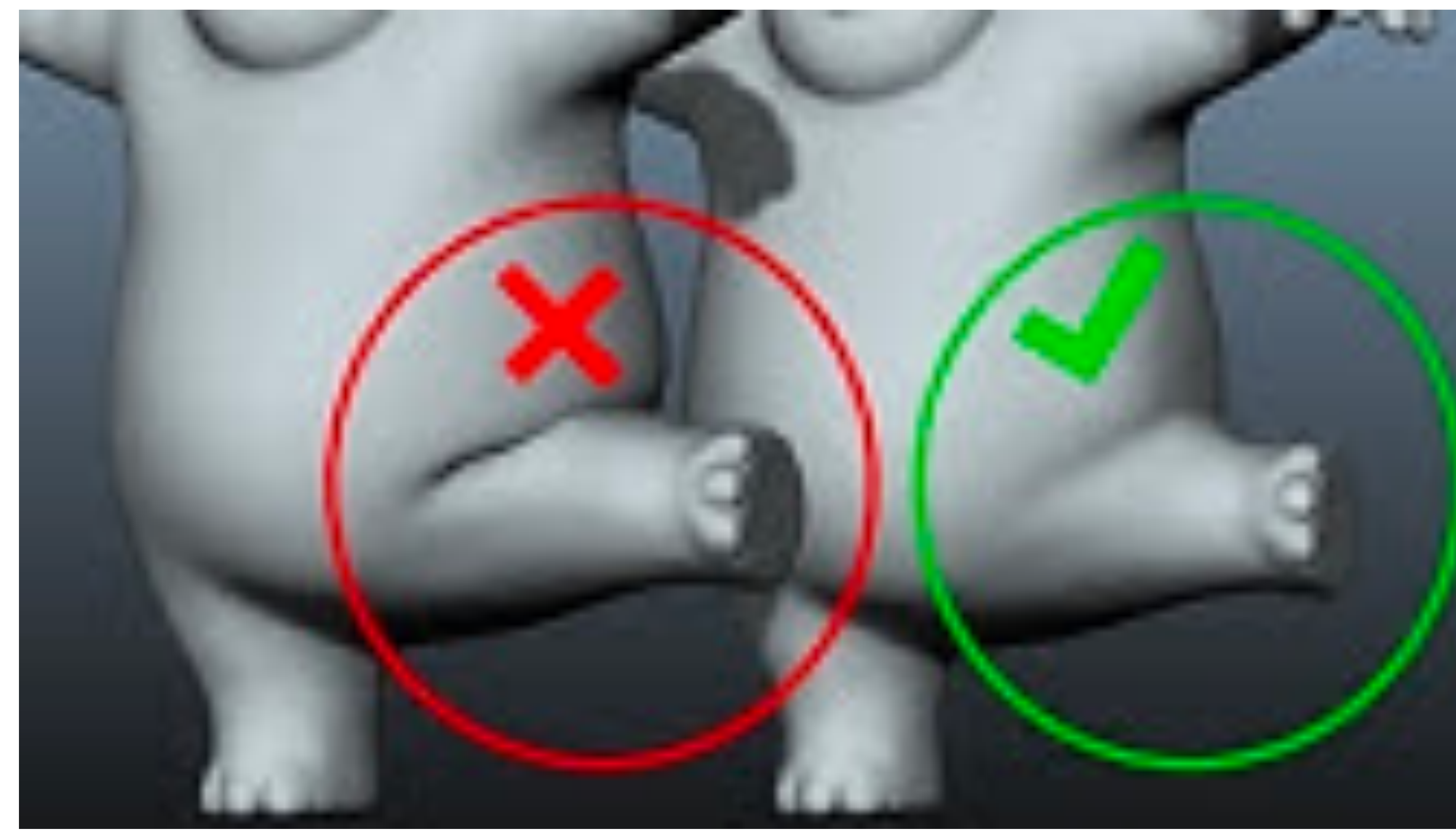
User Input:
pose as relative rotation

$$\theta \in \mathbb{R}^{3\times|B|}$$



Recall the issues of LBS

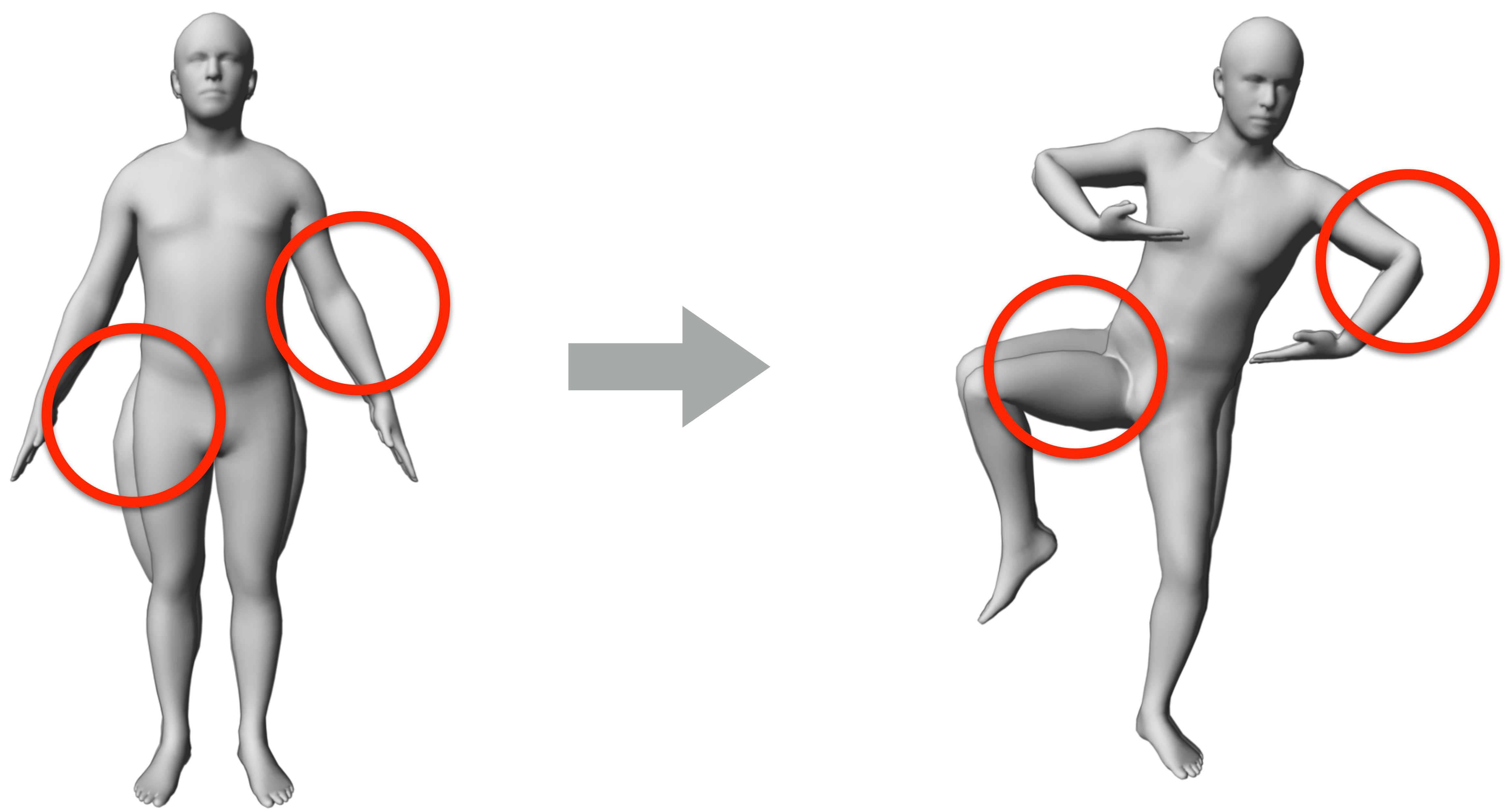
Blend shapes to the rescue!



Corrective Blend Shapes

Blend Shape: a set of vertex displacements in rest pose.

Corrective Blend Shape: Offsets as a function of pose, so after LBS it looks good.

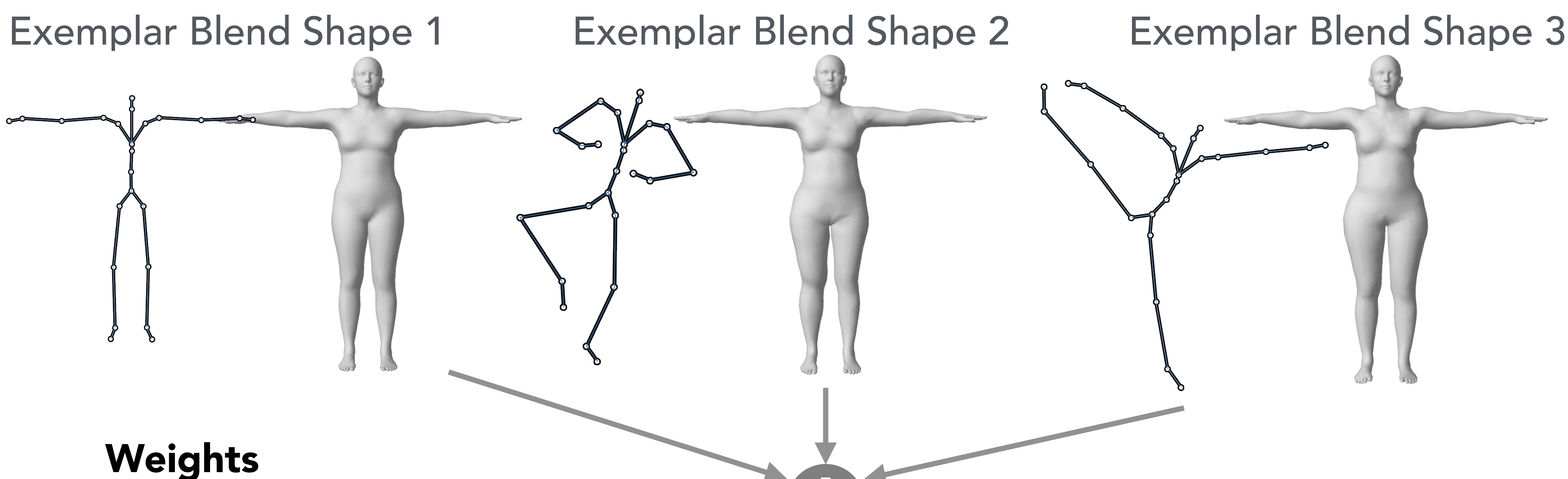


"Pose" blend shapes, bc if fixes issues with posing

Corrective Blend Shapes

Traditionally artists had to create this per pose…!

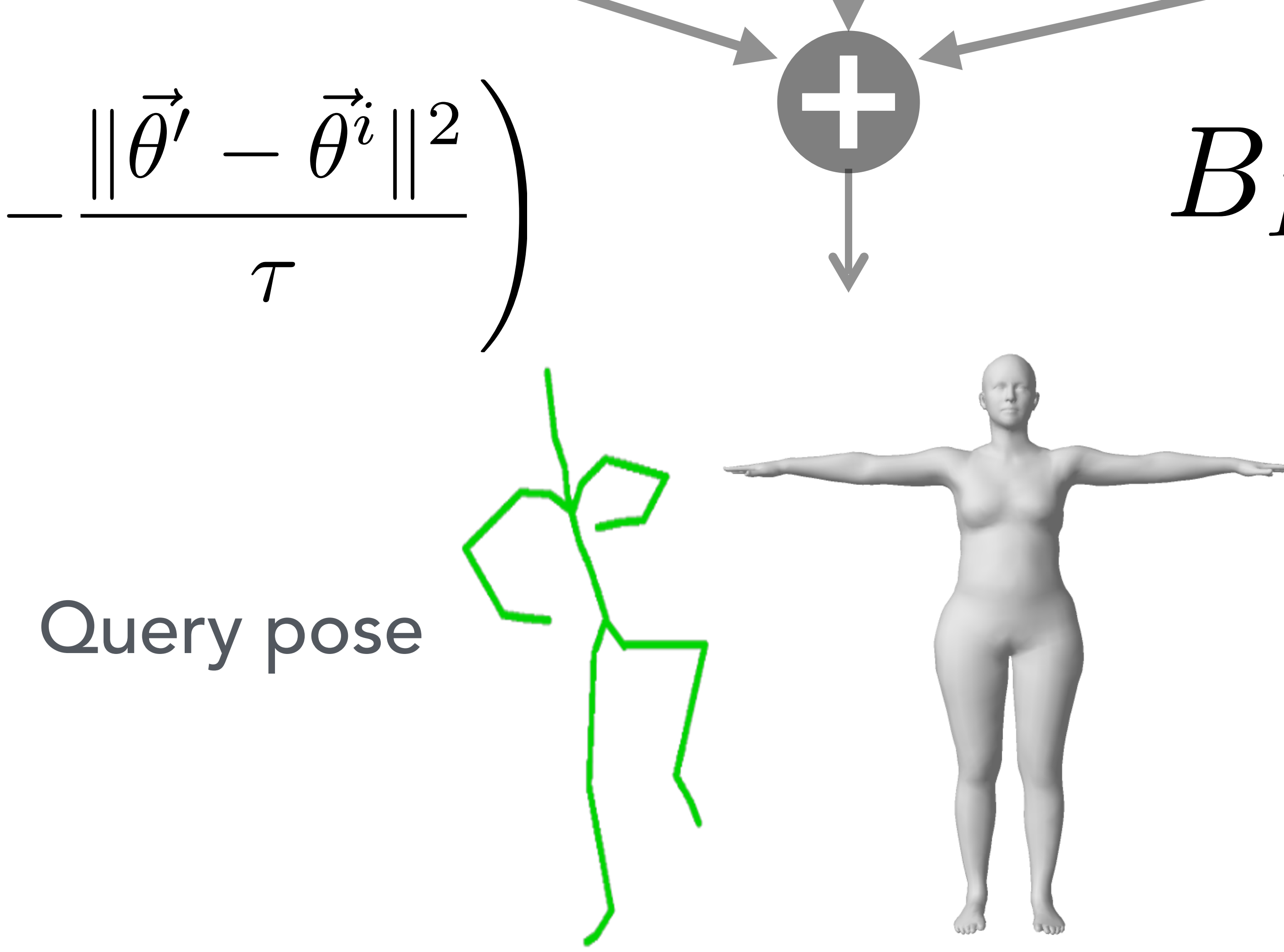
Exemplar interpolation introduced in 2000



Weights

$$\lambda_i \propto K(\vec{\theta'}, \vec{\theta^i}) = \exp\left(-\frac{\|\vec{\theta'} - \vec{\theta^i}\|^2}{\tau}\right)$$

$$B_P(\vec{\theta'}) = \sum_i \lambda_i(\vec{\theta'}) B_i$$



J.P.Lewis et.al. 2000

Corrective "Pose" Blend Shapes

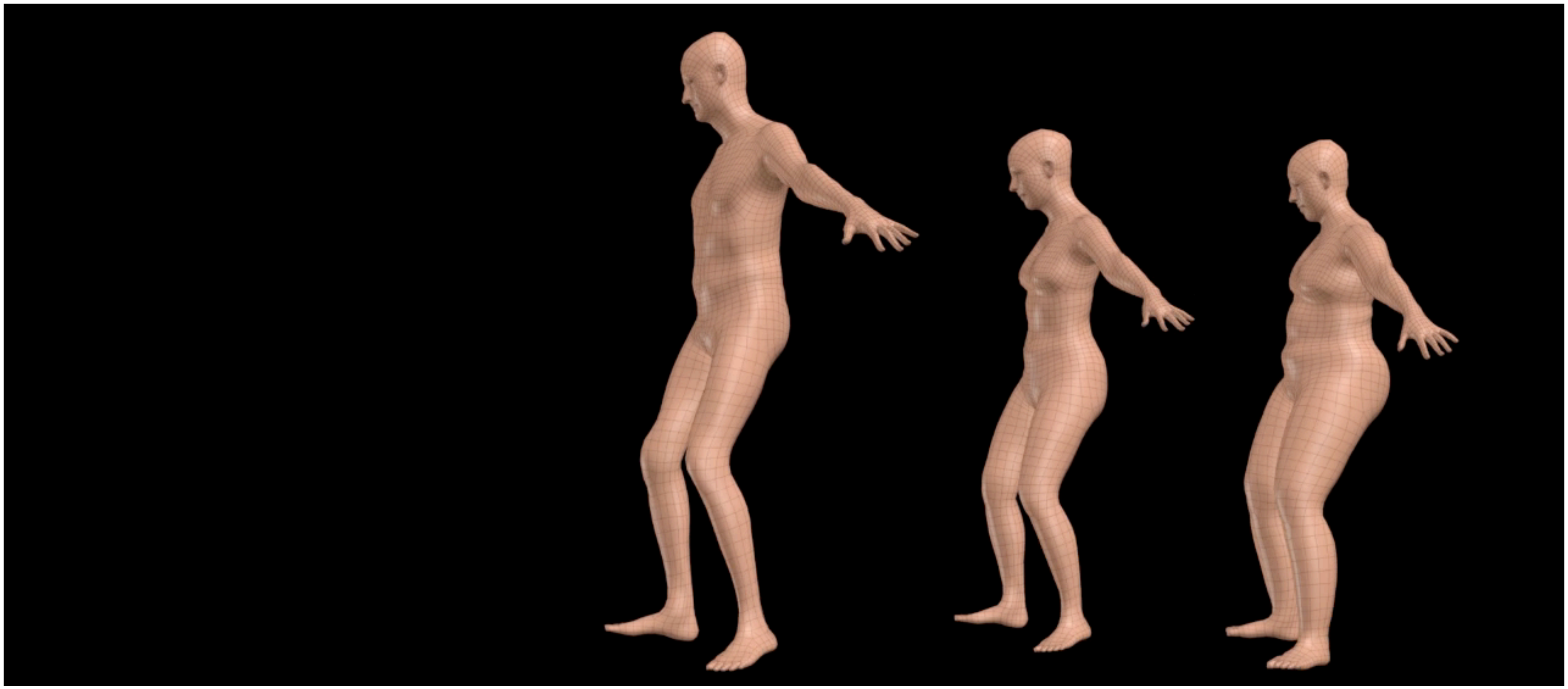
Still artists had to make the exemplars.

With training data we can learn the exemplar pose blend shapes for each bone rotation:

$$B_P(\vec{\theta}; \mathcal{P}) = \sum_{n=1}^{9|B|} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{B}_n$$

Putting it all together

Skinned Multi-person Linear Model (SMPL) [Loper et al. 2015]

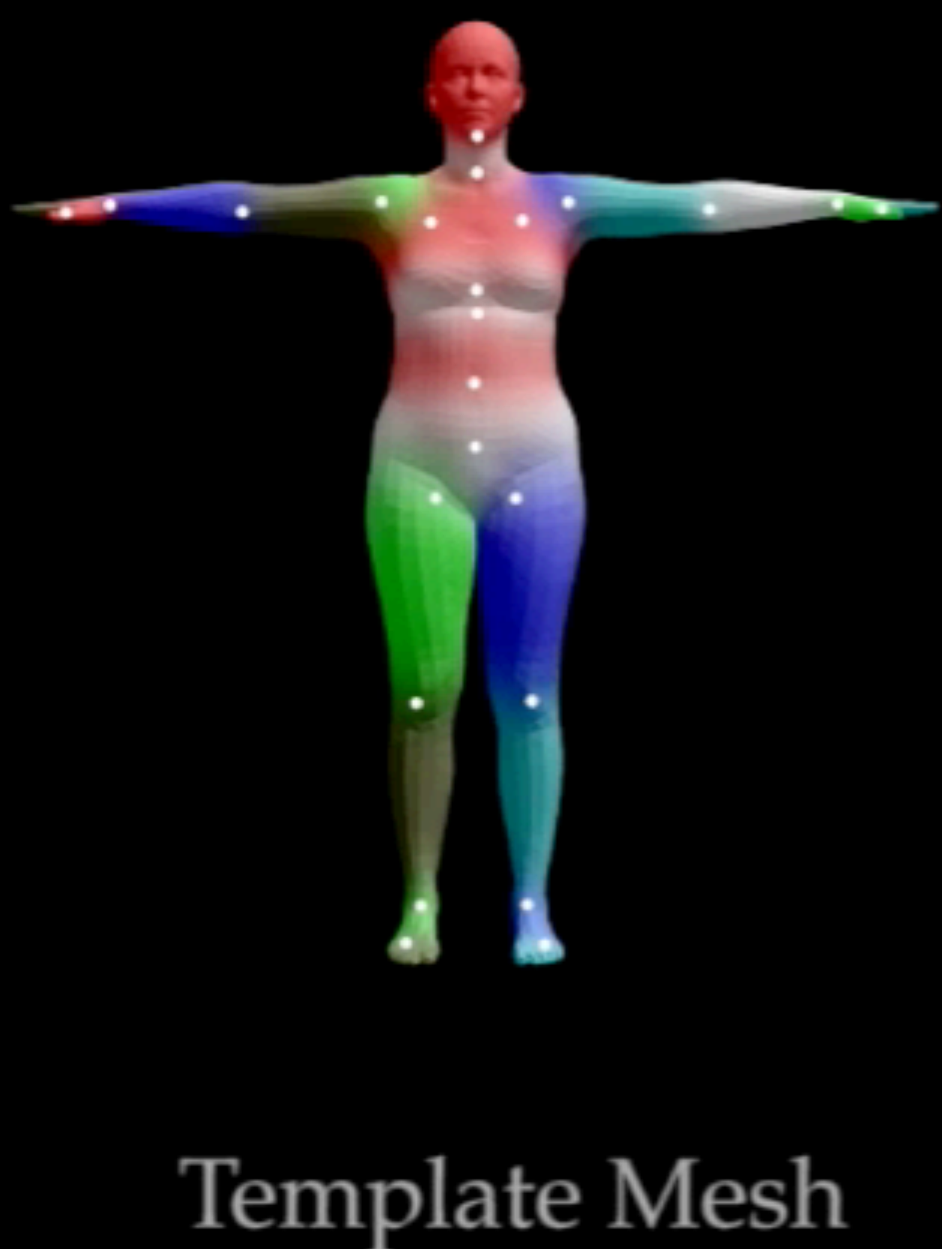


There are prior models like: SCAPE [Anguelov et al. 2005]

SMPL is linear and easy to differentiate, fast.

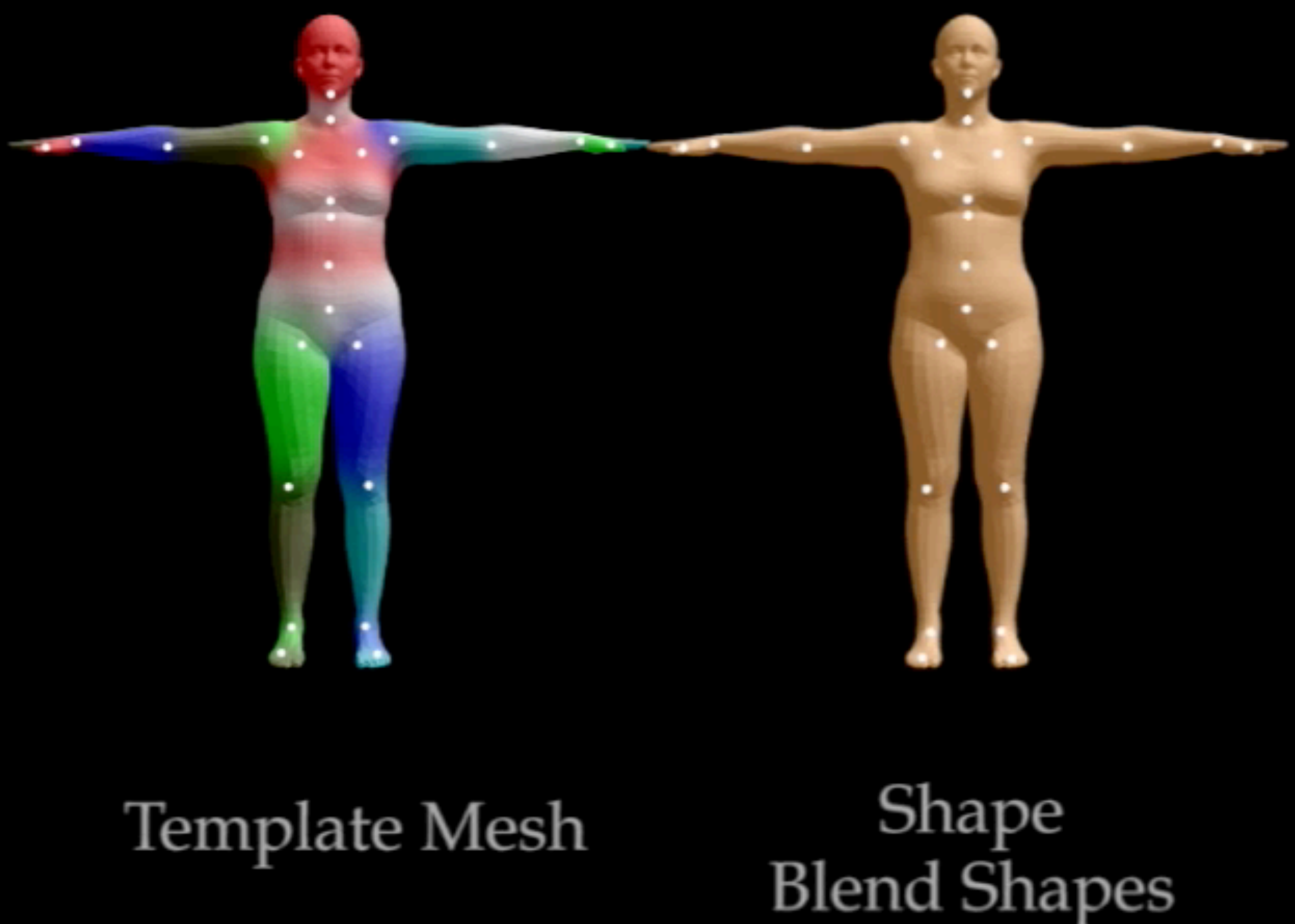
SMPL Summary

SMPL is a blend skinned model with blend shapes.
Model parameters are learned from data.



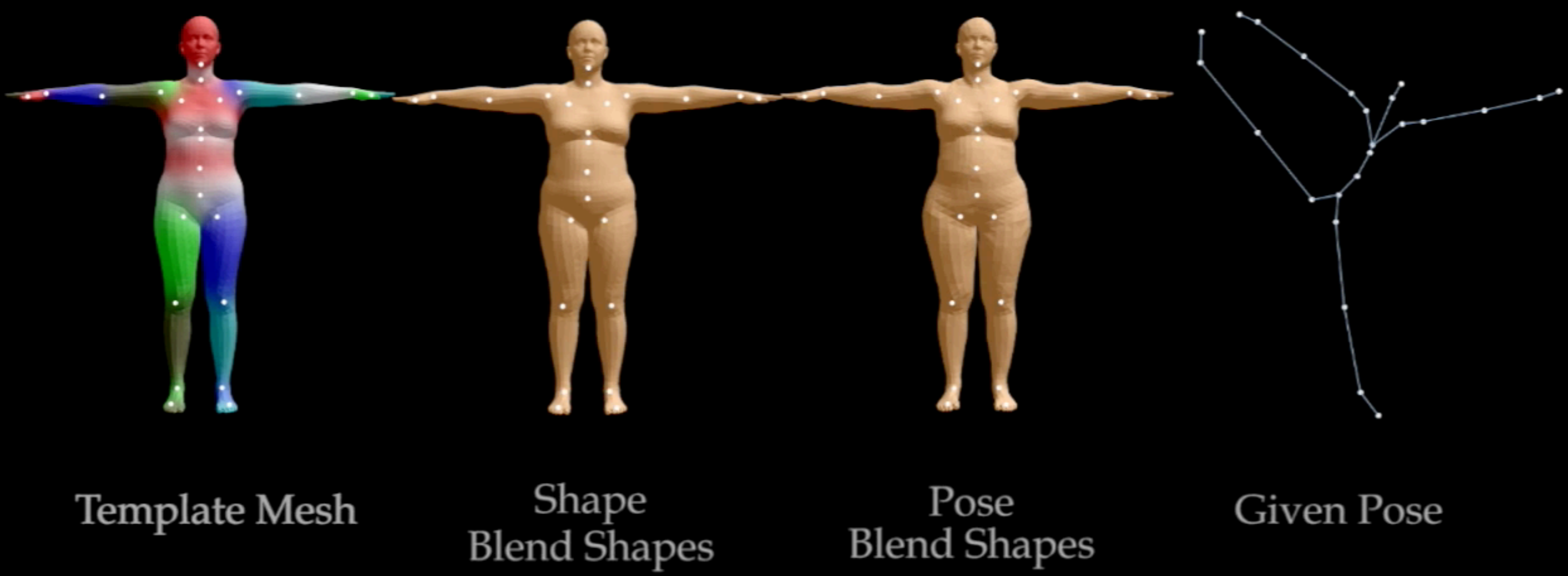
SMPL Summary

SMPL is a blend skinned model with blend shapes.
Model parameters are learned from data.



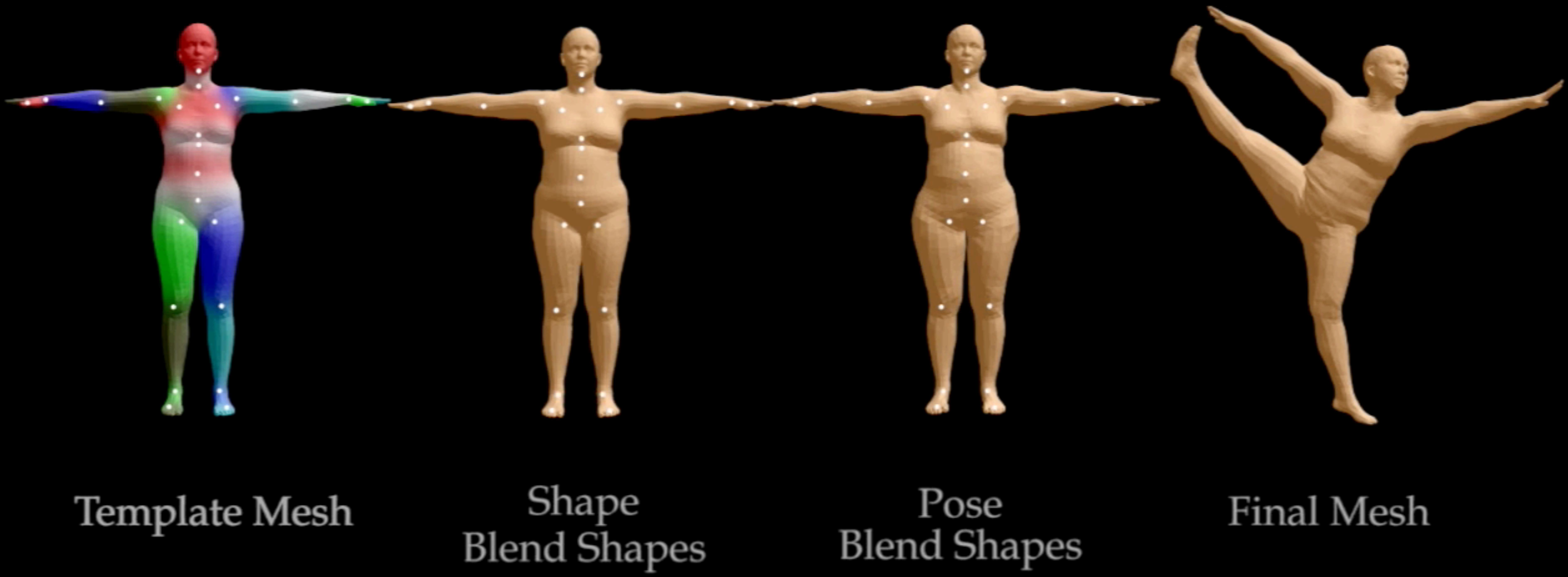
SMPL Summary

SMPL is a blend skinned model with blend shapes.
Model parameters are learned from data.



SMPL Summary

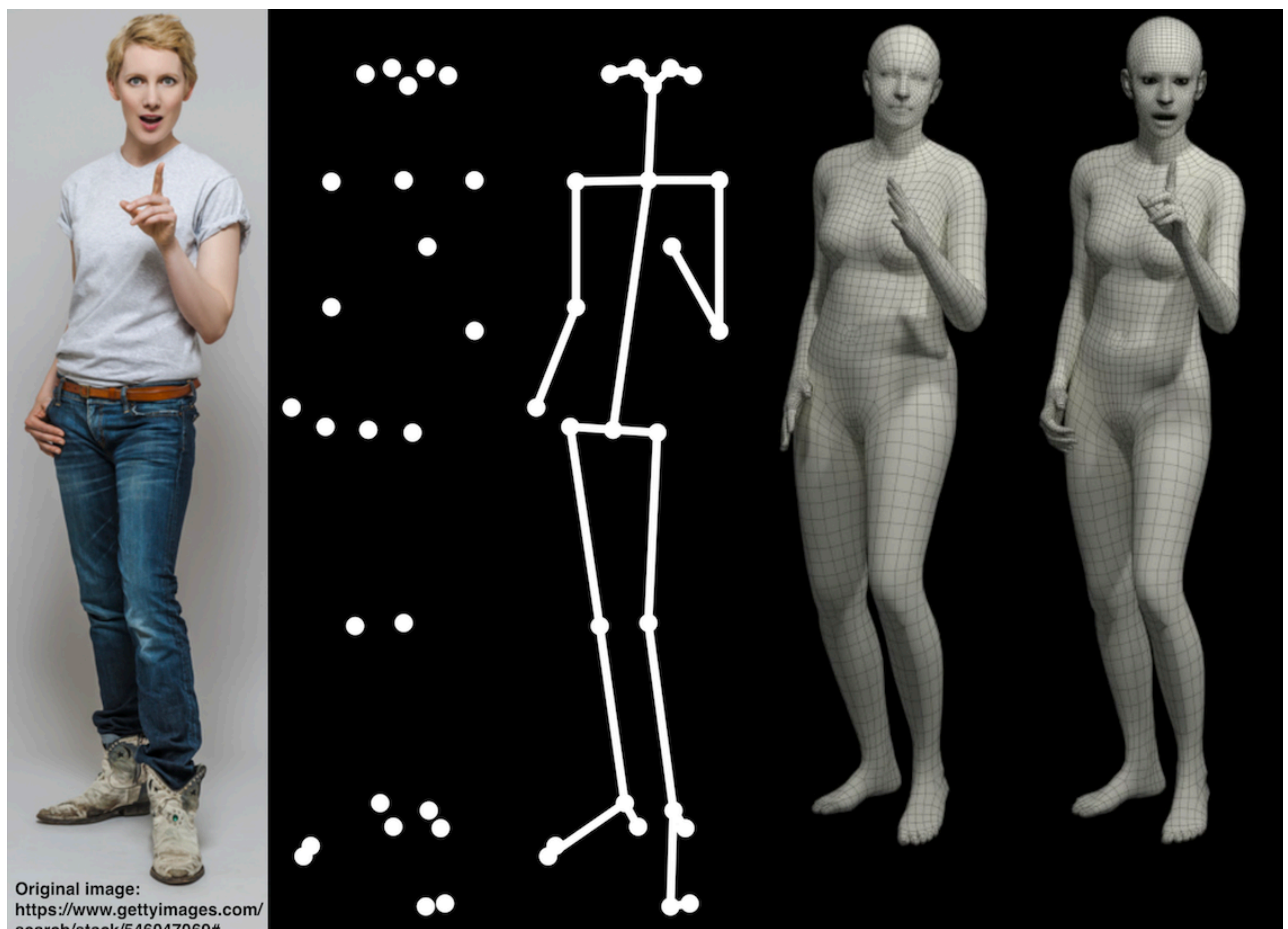
SMPL is a blend skinned model with blend shapes.
Model parameters are learned from data.



There's a lot more for realism

This does not model jiggles, breathing, muscle tension, other non-linear skin deformations. In practice artists adds a lot more handles + design blend shapes.

Subsequent work (SMPL-X) adds face + fingers



Pavlakos et al. CVPR 2019

Motion Capture

Motion Capture

Data-driven approach to creating animation sequences

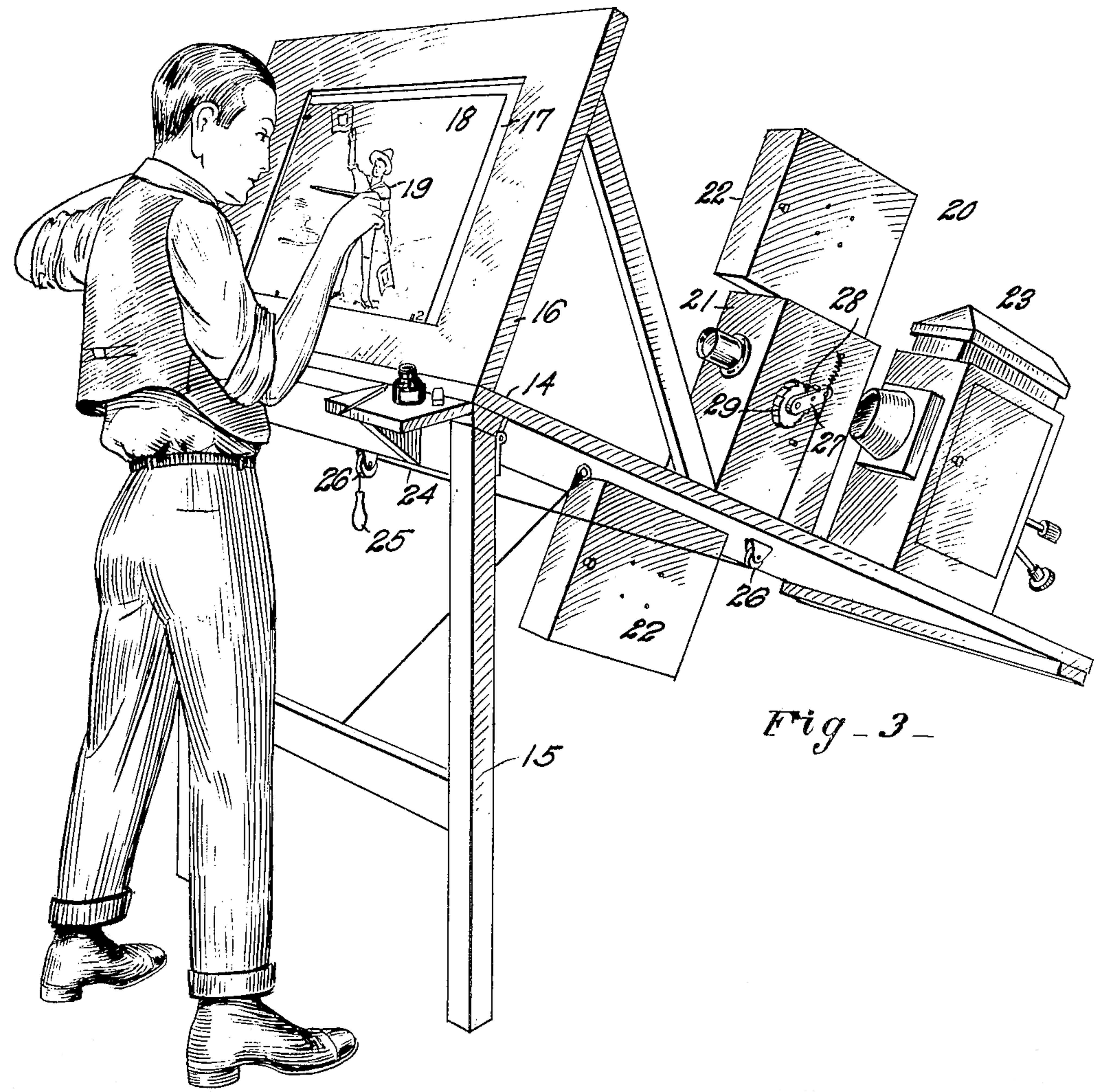
- Record real-world performances
- Extract pose as a function of time from raw data

Andy Serkis, The Two Towers

History: Rotoscoping

Trace footage frame by frame

Invented in 1915 by Max Fleischer

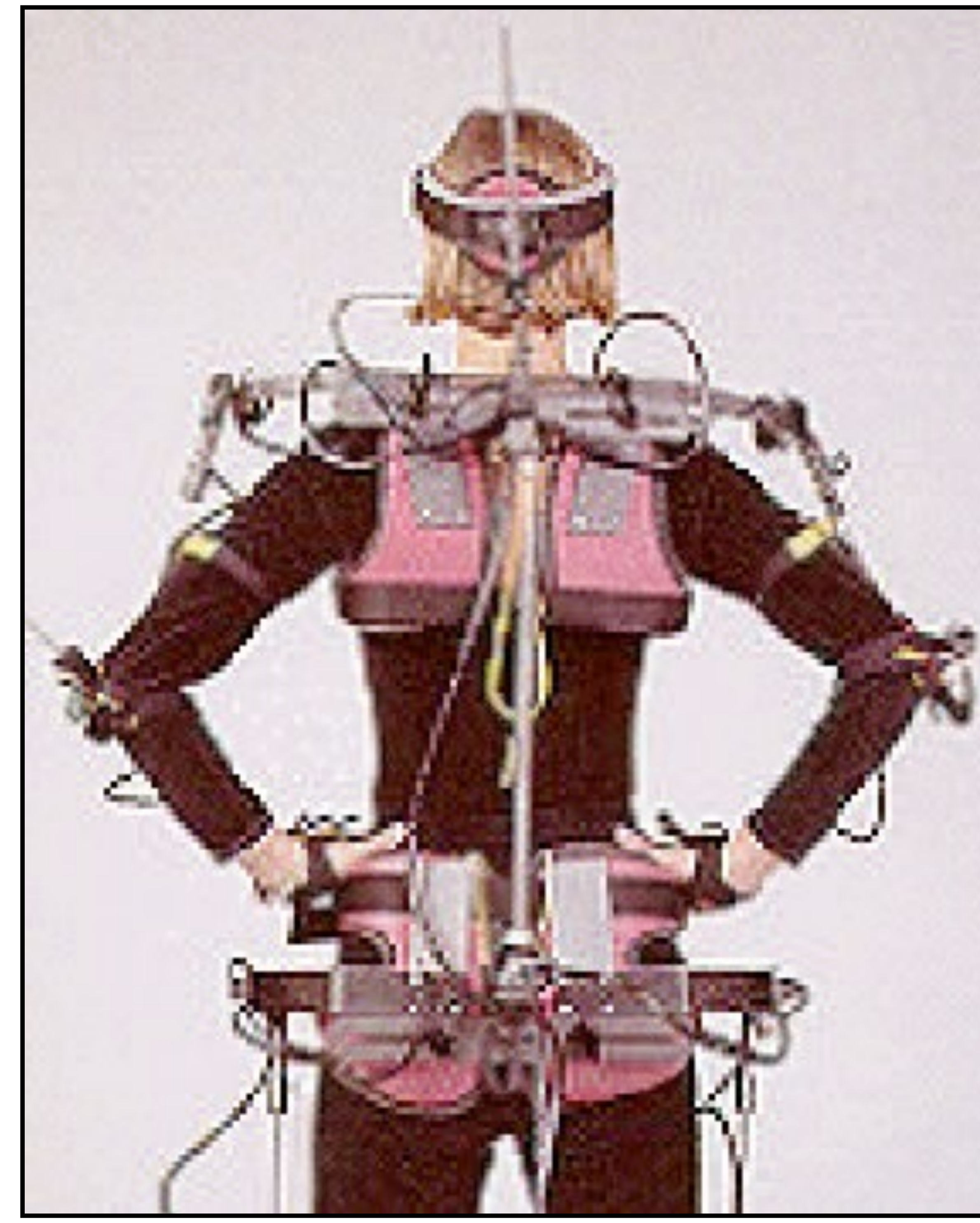


Rotoscoped animations

Motion Capture Equipment

Optical
(More on following slides)

Magnetic
Sense magnetic fields to infer position / orientation. Tethered.



Mechanical
Measure joint angles directly. Restricts motion.

Optical Motion Capture

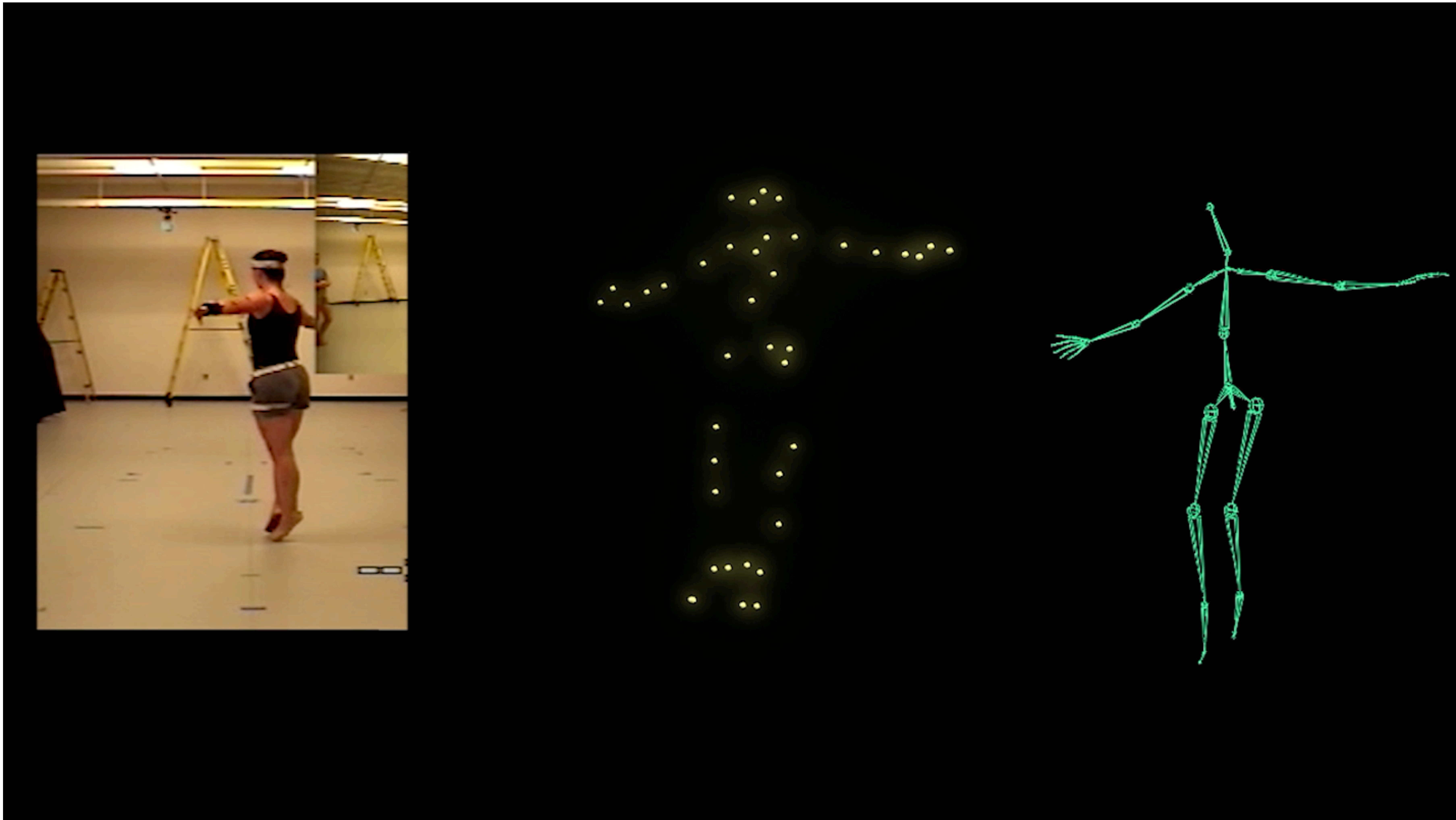
Retroflective markers attached to subject

IR illumination and cameras

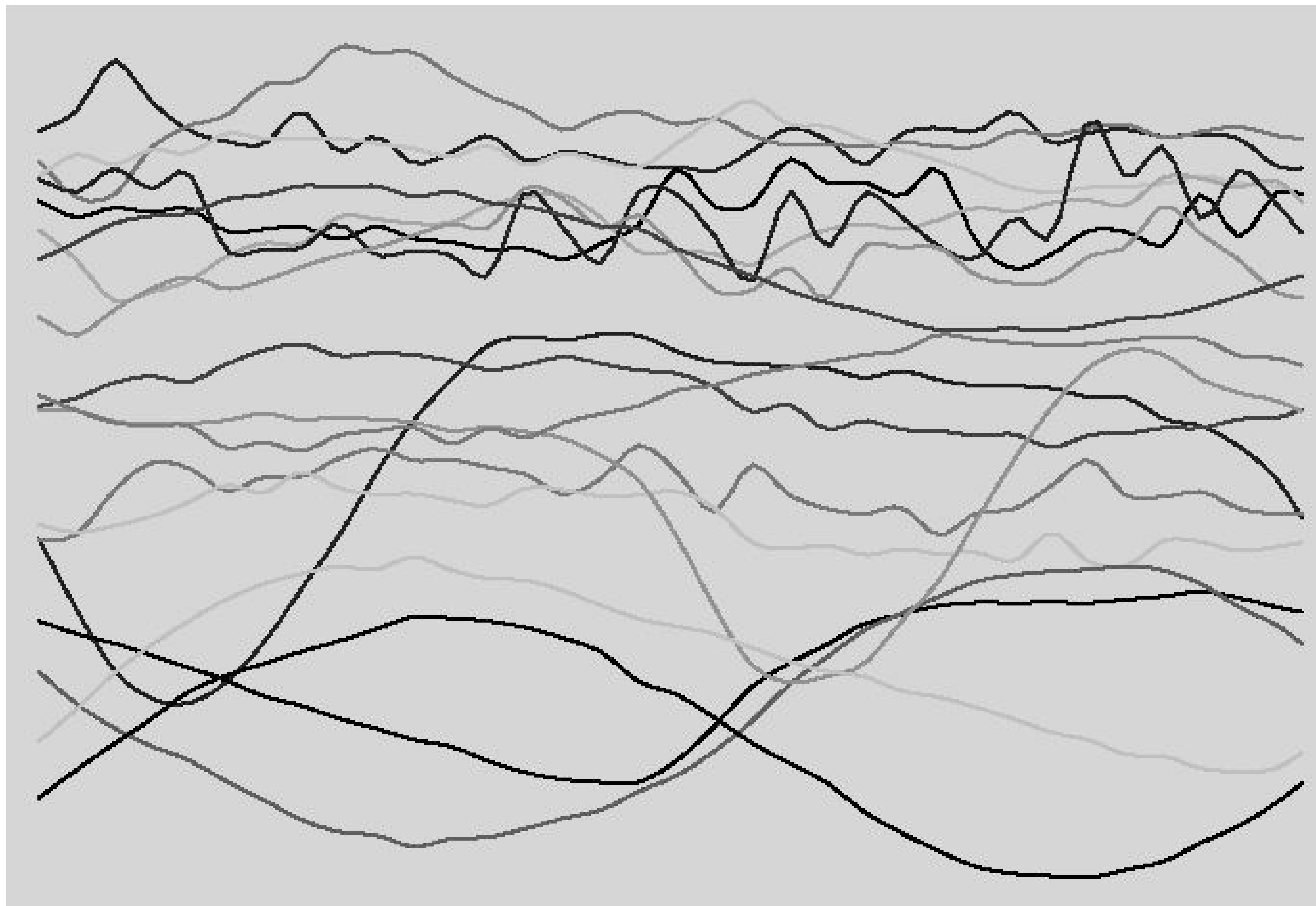
- Markers on subject
- Positions by triangulation from multiple cameras
- 8+ cameras, 240 Hz, occlusions are difficult

Slide credit: Steve Marschner

Motion Capture



Motion Data



Subset of motion curves from captured walking motion.

From Witkin and Popovic, 1995

Motion Capture Pros and Cons

Strengths

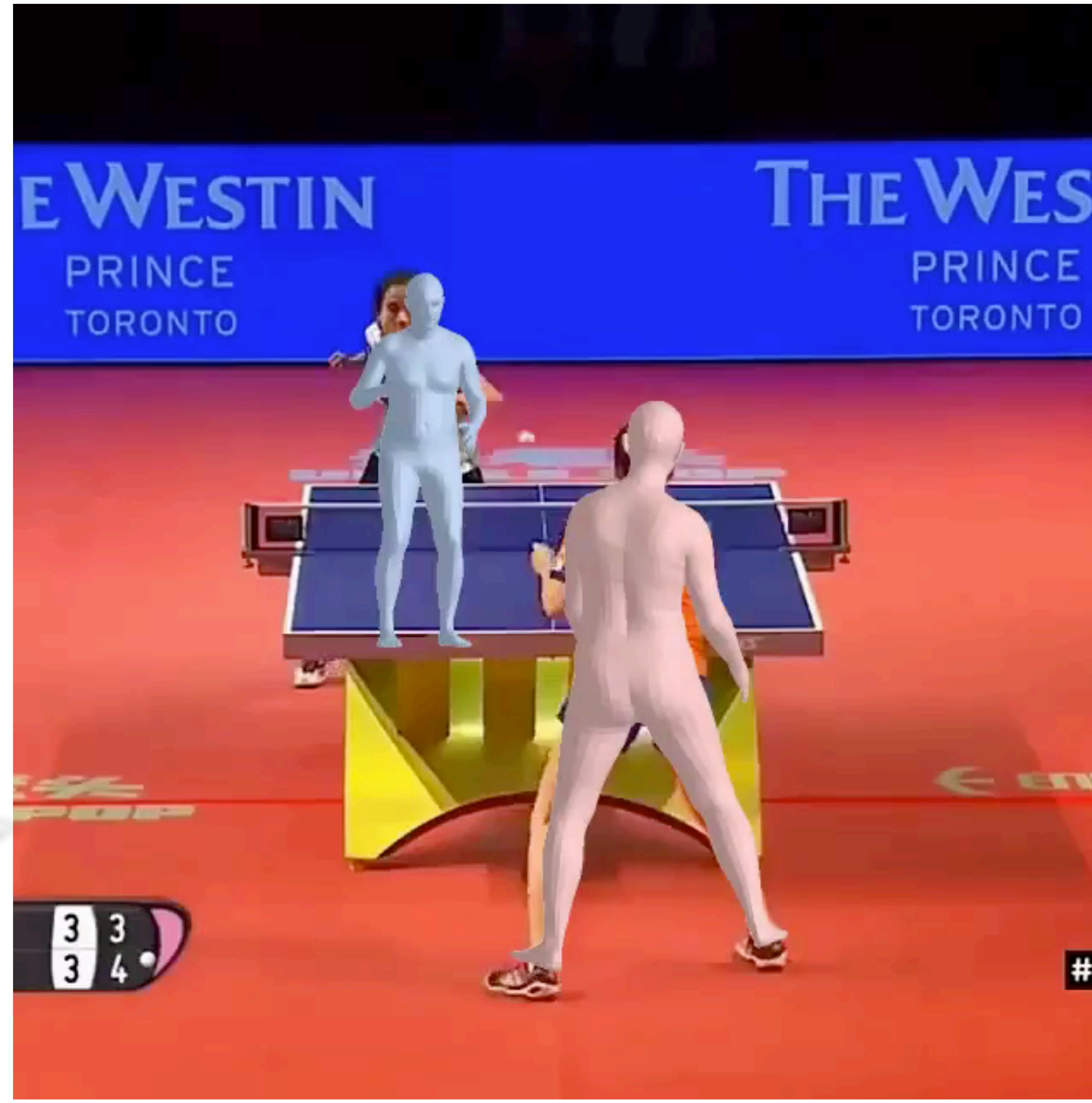
- Can capture large amounts of real data quickly
- Realism can be high

Weaknesses

- Complex and costly set-ups
- Captured animation may not meed artistic needs, requiring alterations

Markerless Motion Capture :)

Kanazawa et al. 2018



Kanazawa et al. 2019

TBC in later lectures

More resources

skinning.org

Skinning: Real-time Shape Deformation

SIGGRAPH 2014 Course

Playing around with SMPL is easy!

Acknowledgments

Many thanks to Michael Black, Gerard Pons-Moll, Ladislav Kavan, Olga Sorkine-Hornung, Alec Jacobson, and Leon Sigal for lecture resources.