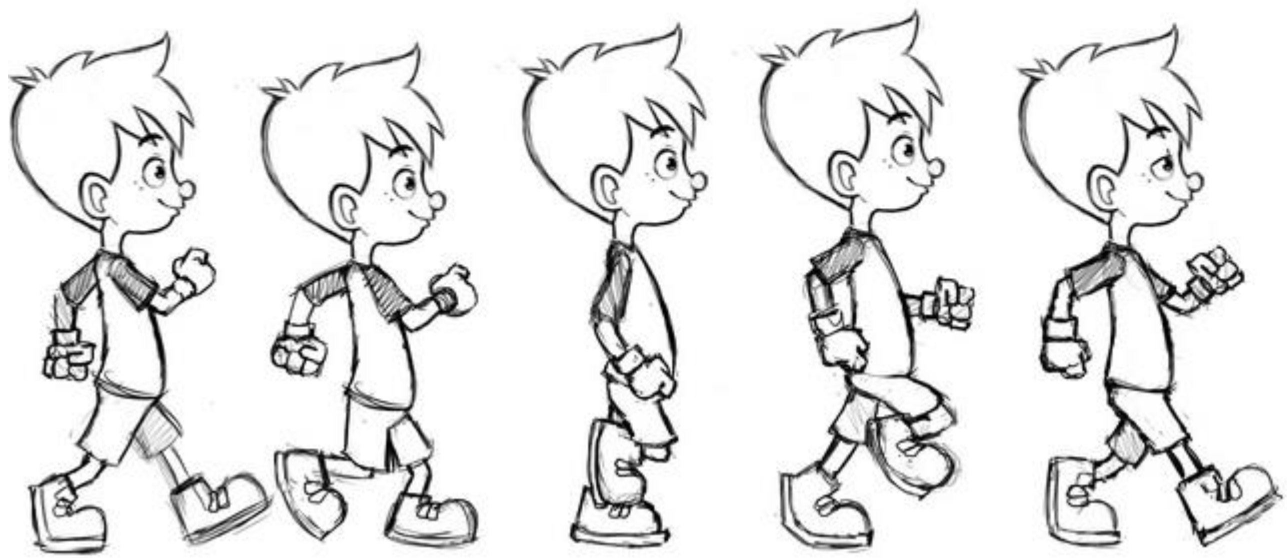
Animation

Animation

Motion over time



Animation

Motion over time

Usually focus on **character animation**
but environment is often also animated

- trees, water, fire, explosions, …

Animation

Motion over time

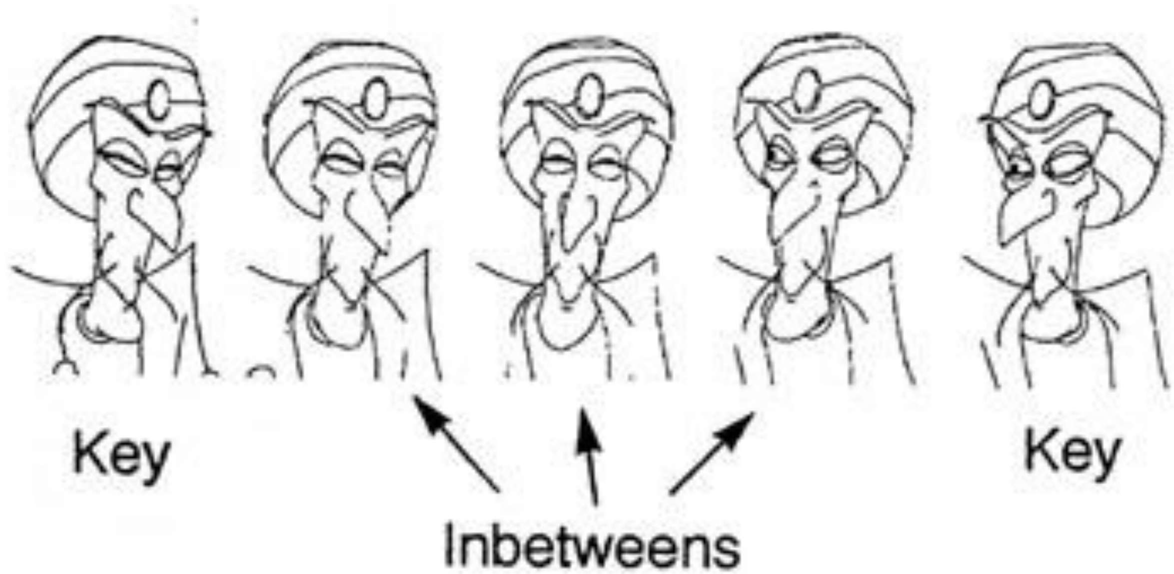
Usually focus on **character animation**
but environment is often also animated

- trees, water, fire, explosions, …

Could be physically realistic, or stylized

Traditional Character Animation

Lead animator draws sparse **key frames**



Secondary artists fill in (by hand) the intermediate frames: **inbetweening**

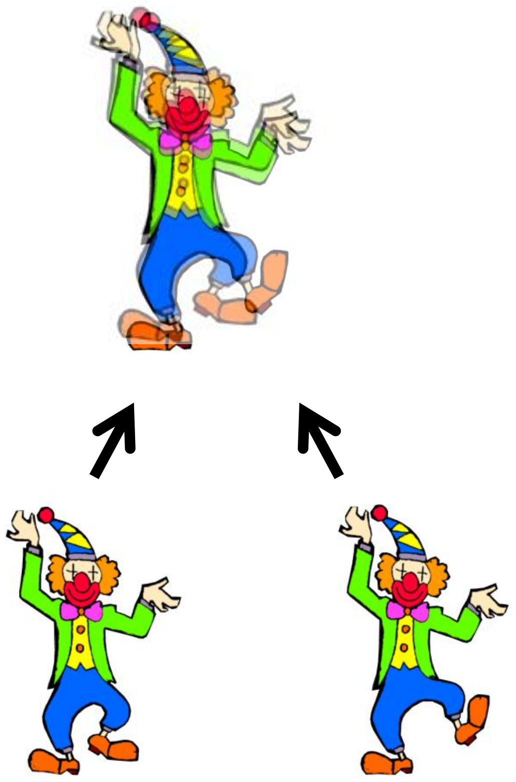
Computer Character Animation

How to inbetween automatically?

Linear interpolation?

Computer Character Animation

How to inbetween automatically?



Linear interpolation?

- No

Computer Character Animation

How to inbetween automatically?

Need way to **parameterize motion**

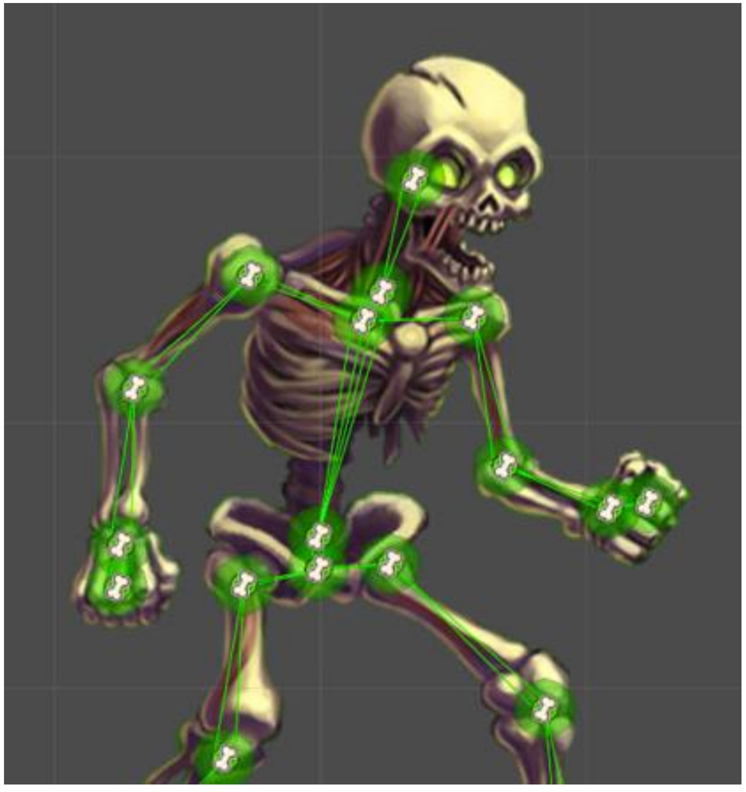
Handle-Based Animation

Pick special points (**handles**) on object

Moving handles moves nearby points

Joint-Based Animation

Extend this idea to line segments:

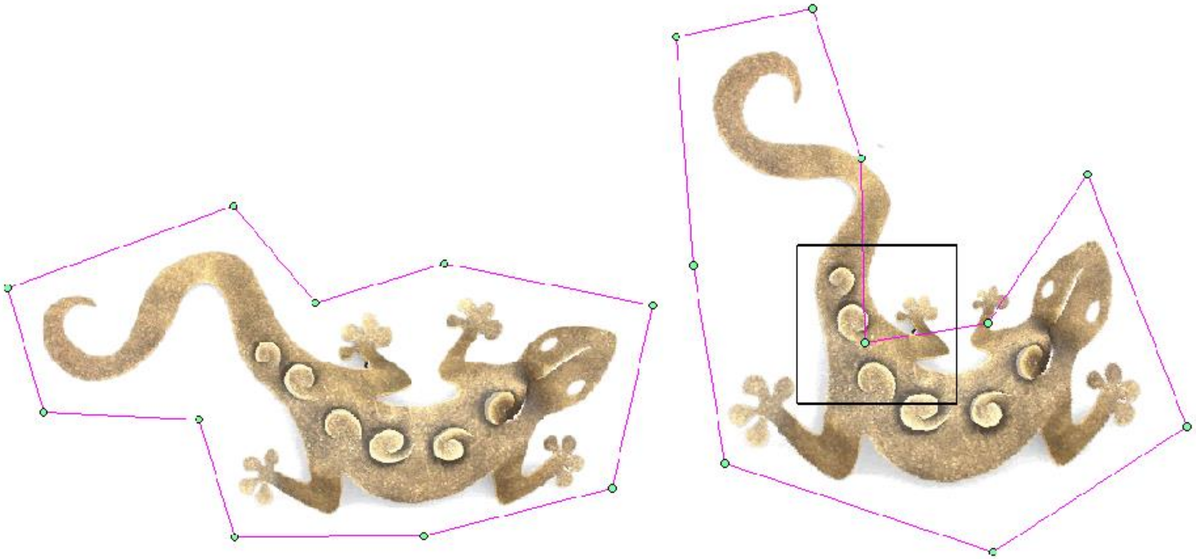


set of bones inside a character is called a **rig**

we will discussing rigging in detail next lecture

Cage-Based Animation

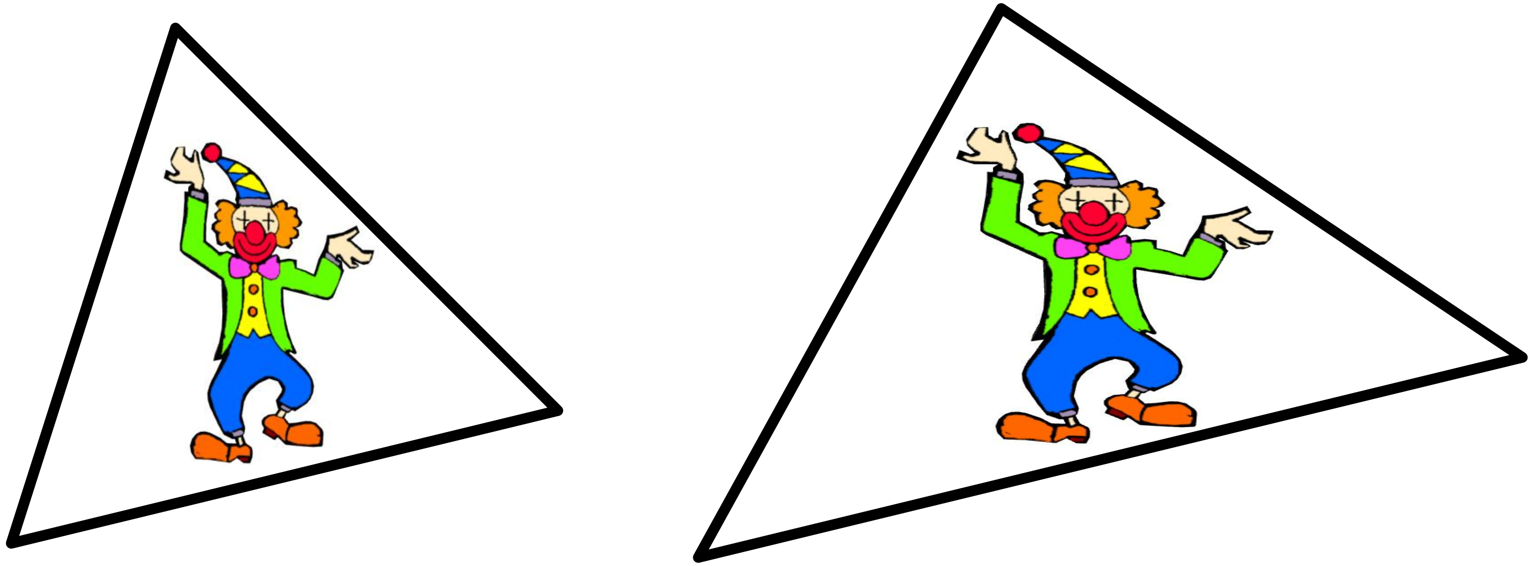
Surround object with **animation cage**



Moving the cage moves interior points

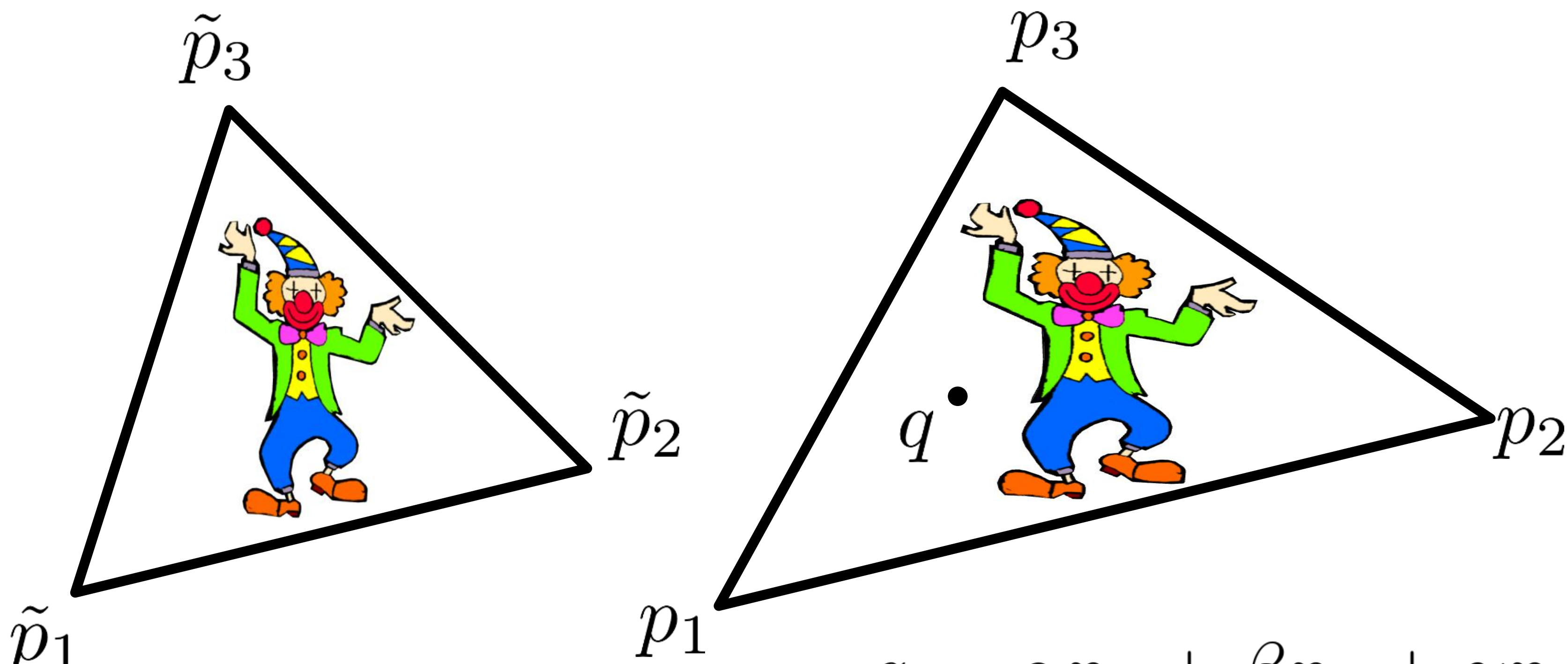
Simplest Cage: Triangle

Use **barycentric interpolation**



Simplest Cage: Triangle

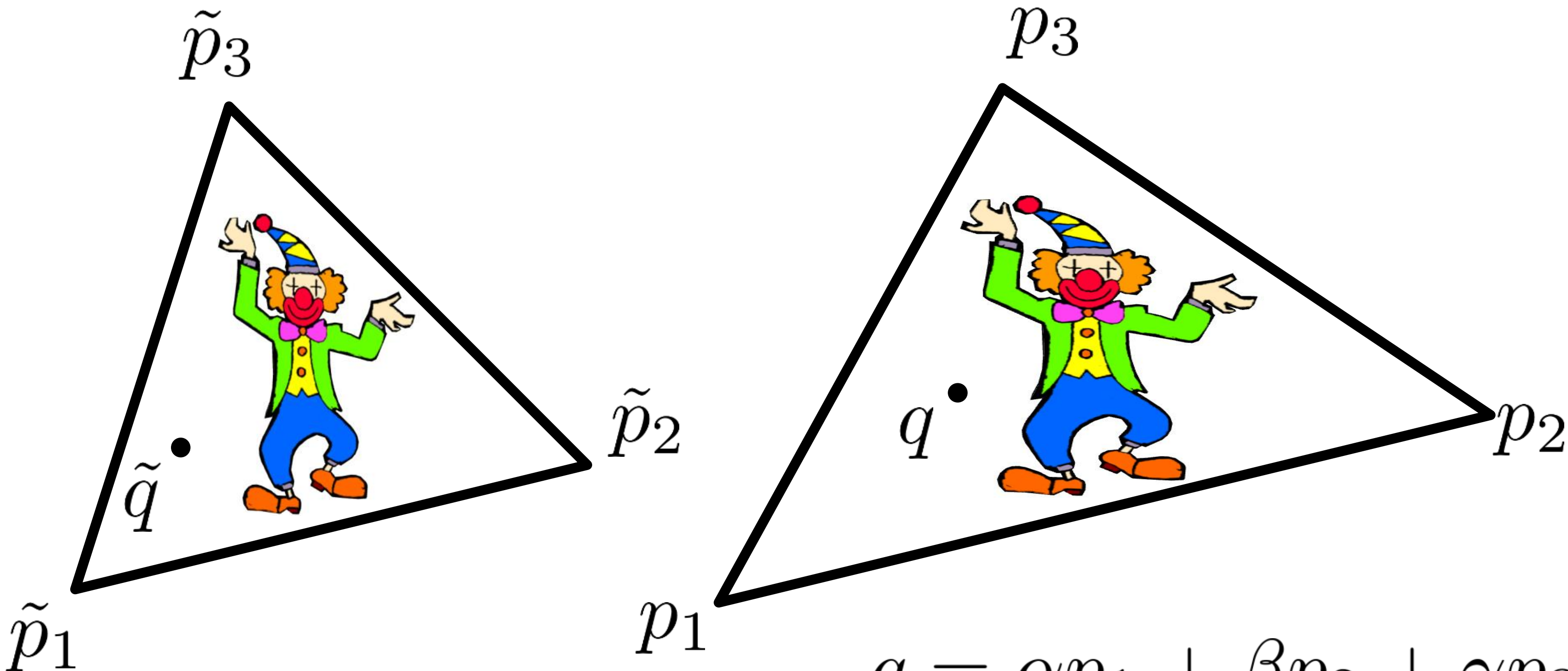
Use **barycentric interpolation**



$$q = \alpha p_1 + \beta p_2 + \gamma p_3$$

Simplest Cage: Triangle

Use **barycentric interpolation**



$$q = \alpha p_1 + \beta p_2 + \gamma p_3$$
$$\tilde{q} = \alpha \tilde{p}_1 + \beta \tilde{p}_2 + \gamma \tilde{p}_3$$

Simplest Cage: Triangle

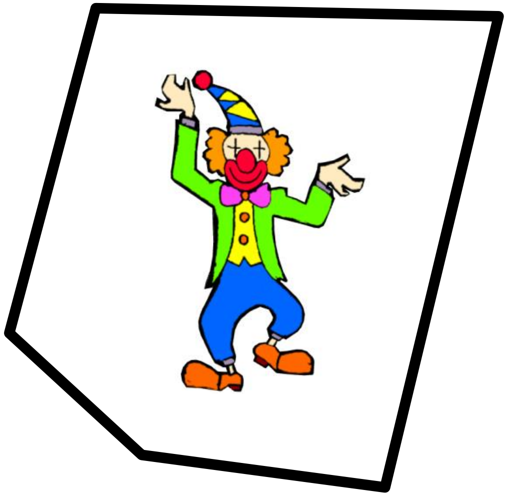
Use **barycentric interpolation**

For every pixel:

- find its barycentric coordinates
- look up point with same coordinates on **undeformed** shape
- copy pixel at that point

Polygonal Cages

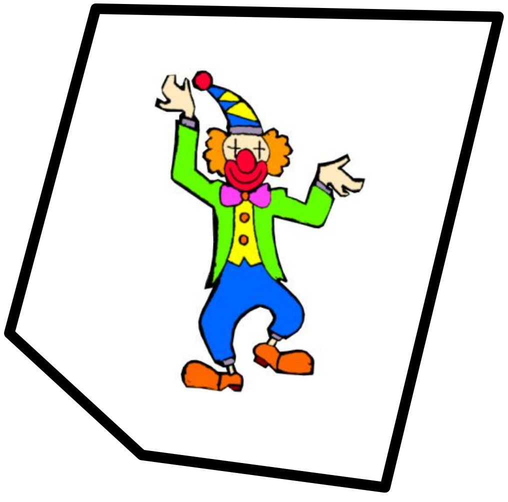
Must generalize barycentric coordinates to arbitrary polygons



$$q = \alpha p_1 + \beta p_2 + \gamma p_3 + \delta p_4 + \epsilon p_5$$

Polygonal Cages

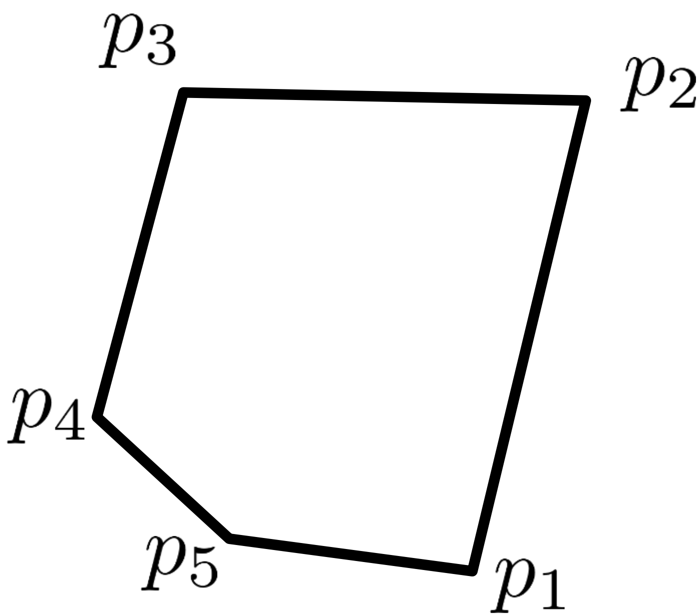
Must generalize barycentric coordinates to arbitrary polygons



Many ways to do this: generalized barycentric coordinates **not** unique

$$q = \alpha p_1 + \beta p_2 + \gamma p_3 + \delta p_4 + \epsilon p_5$$

Generalized Barycentric Coords



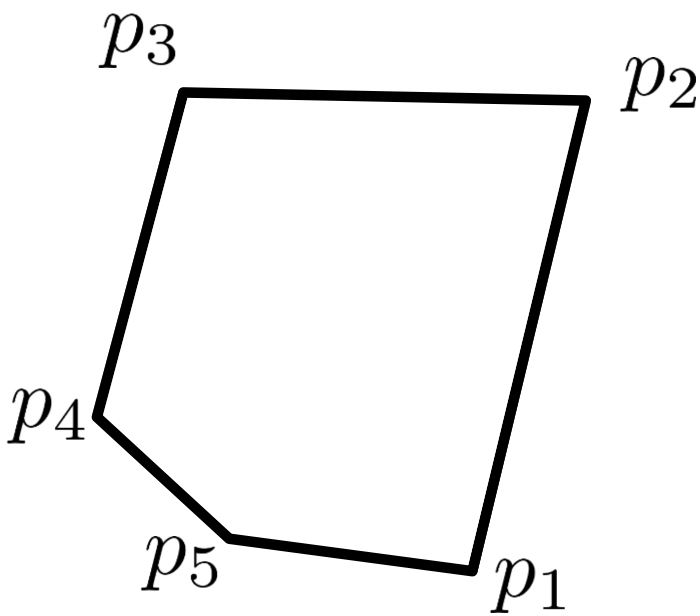
$$q = \sum \alpha_i p_i$$

Partition of unity:

$$1 = \sum \alpha_i$$

why important?

Generalized Barycentric Coords



$$q = \sum \alpha_i p_i$$

Partition of unity:

$$1 = \sum \alpha_i$$

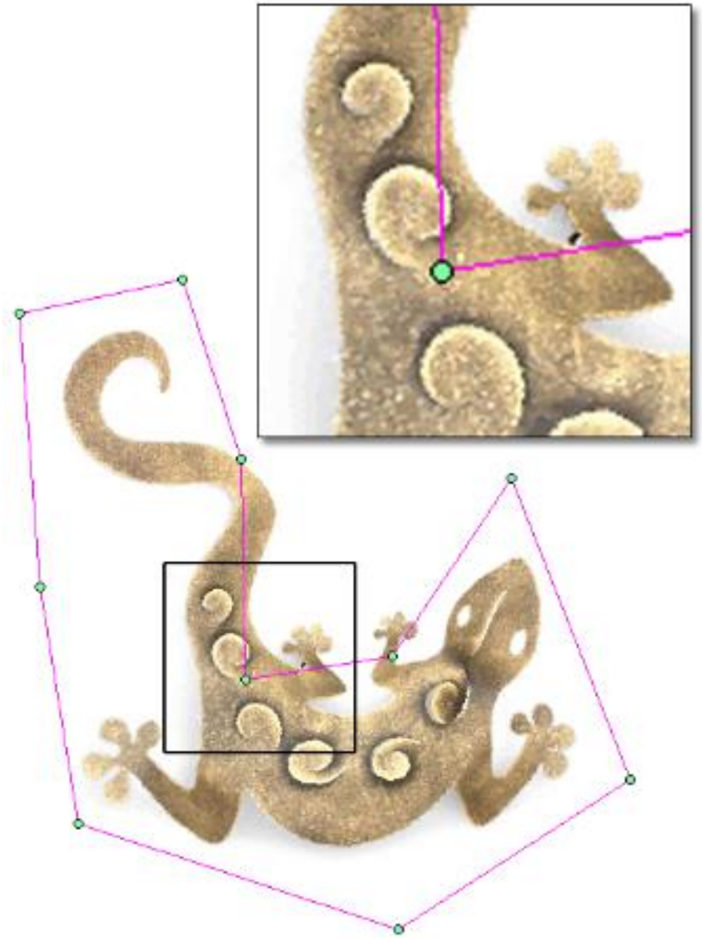
Reproduces the verts:

$$\alpha_i(p_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

Polygonal Cages

Other properties:

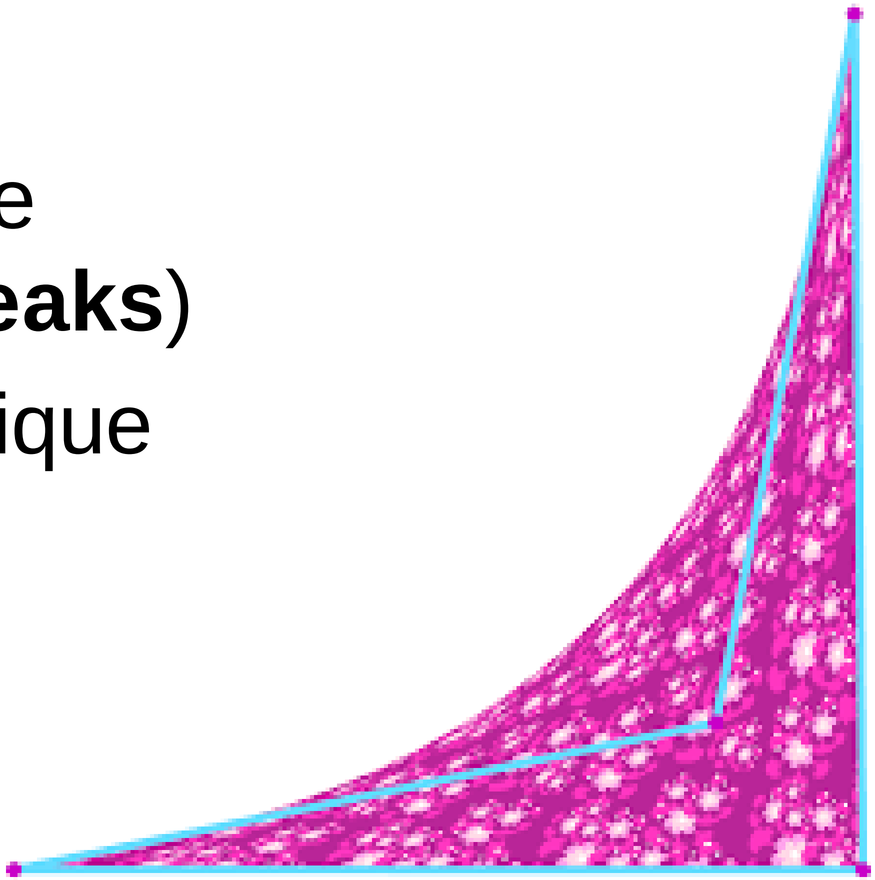
1. Weights must be positive inside the polygon (or get **leaks**)



Polygonal Cages

Other properties:

1. Weights must be positive inside the polygon (or get **leaks**)
2. Weights must unique (or get **flips**)



Polygonal Cages

Other properties:

1. Weights must be positive inside the polygon (or get **leaks**)
2. Weights must unique (or get **flips**)
3. Smooth
4. Easy to compute

Many Possible Schemes

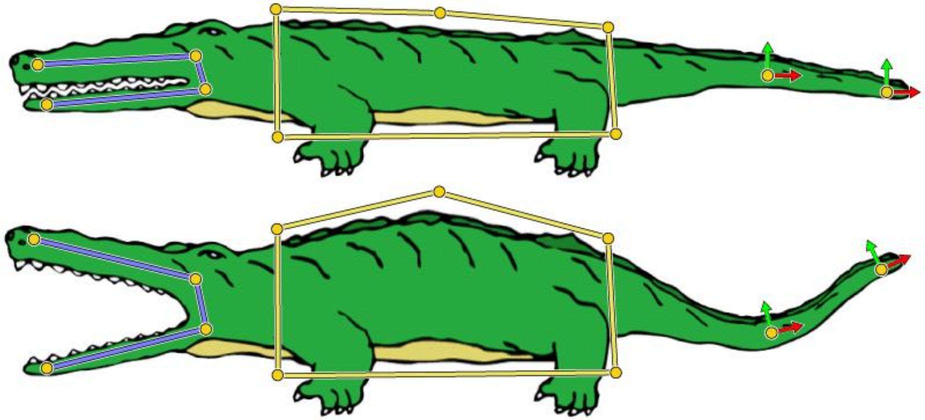
Wachspress Coordinates

Mean-value Coordinates

Green Coordinates

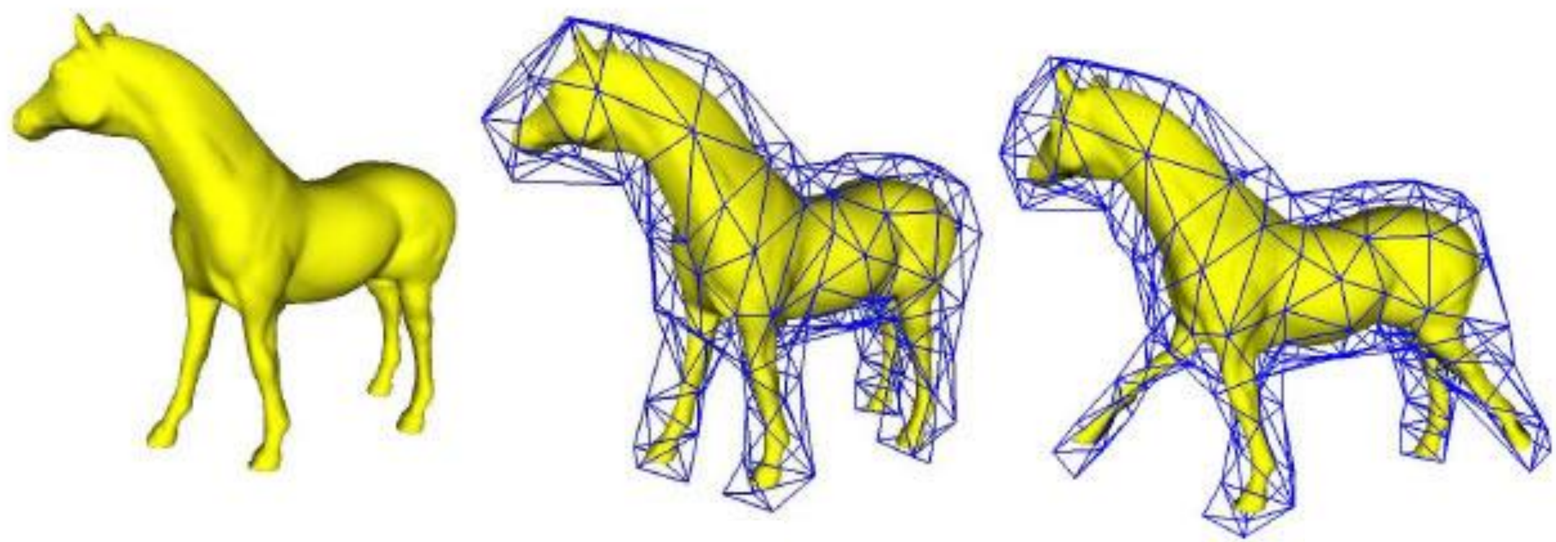
Bounded Biharmonic Weights

…



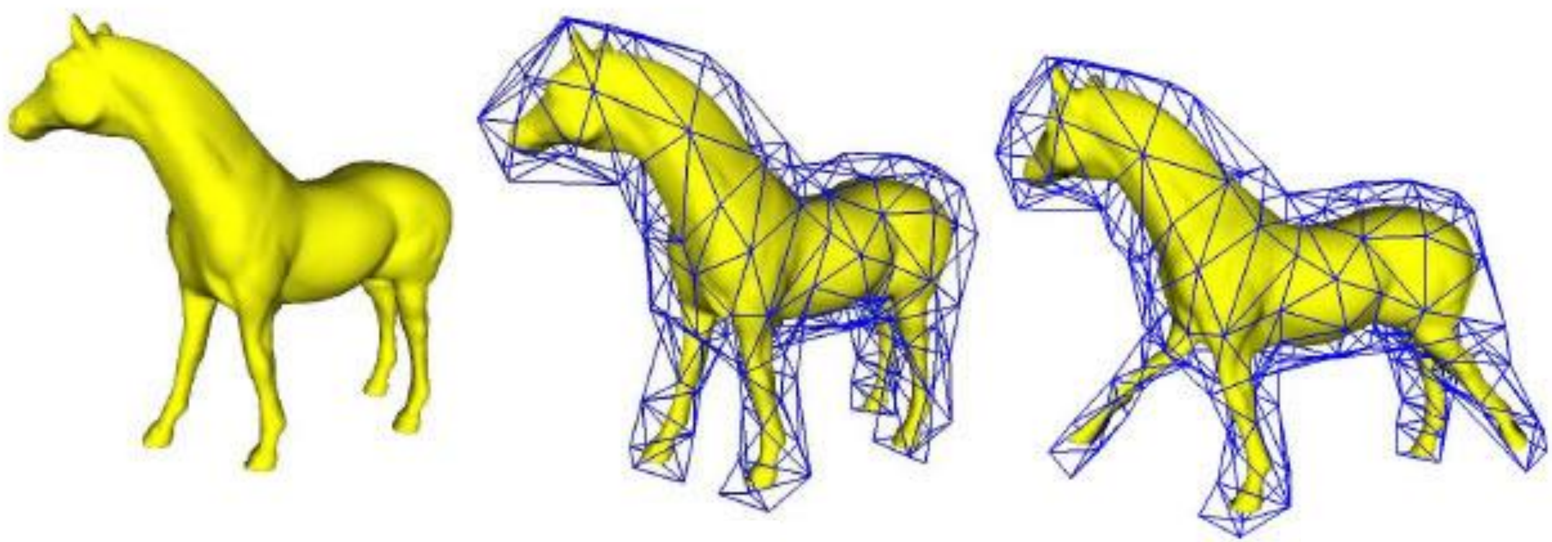
Cage-Based Animation

Extends to 3D from 2D naturally



Cage-Based Animation

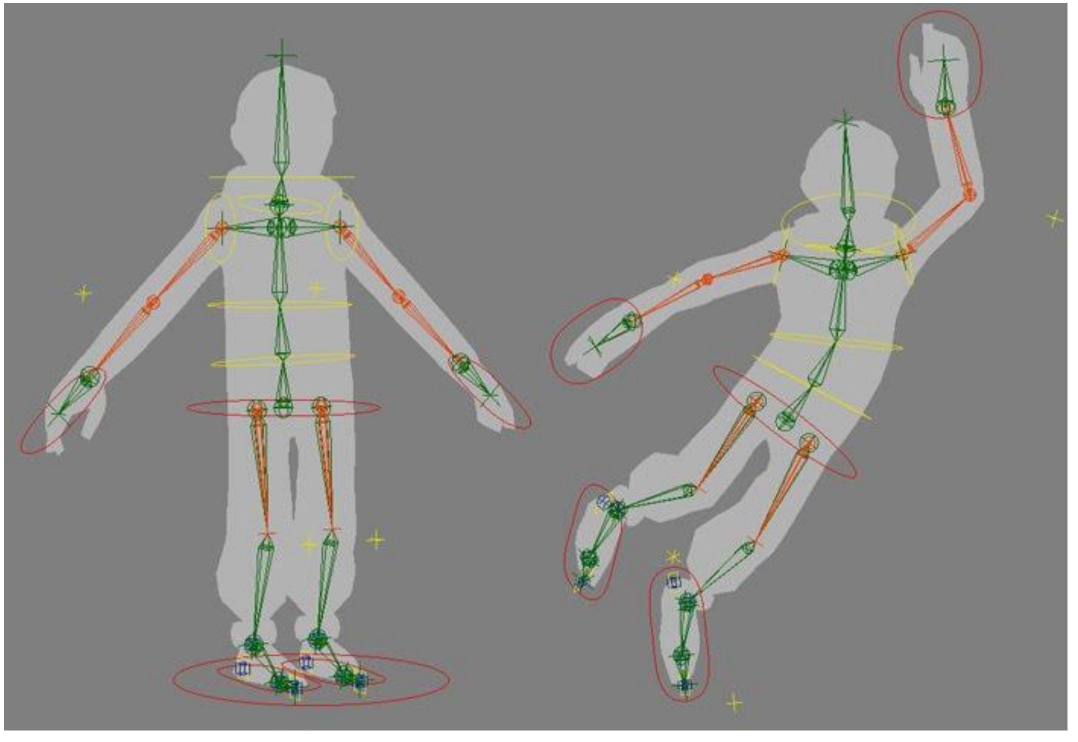
Extends to 3D from 2D naturally



Full control, but not intuitive

Character Rigs

Skeletons inside the geometry



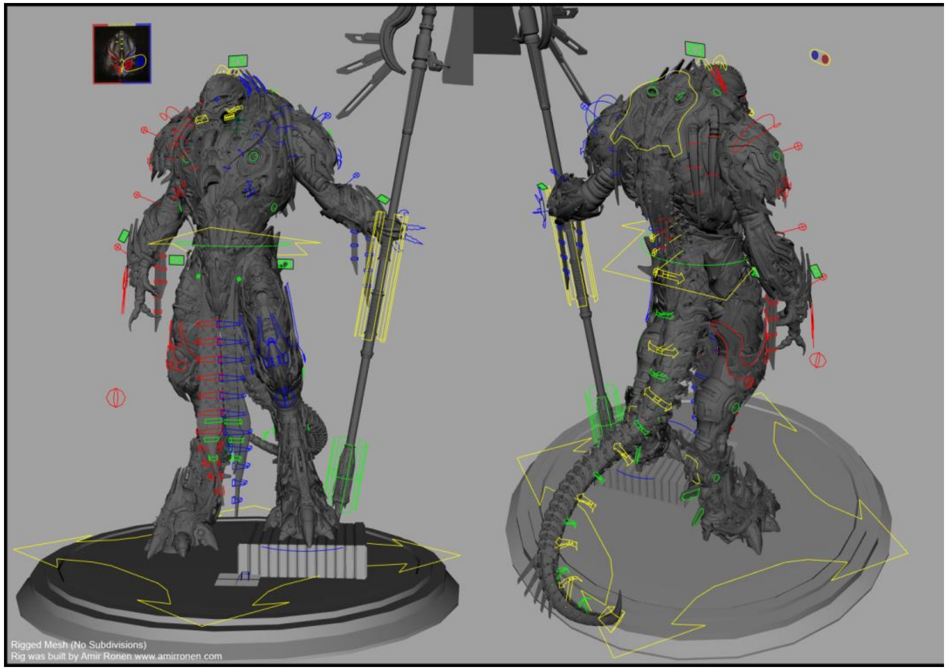
moving bones moves surrounding geometry

the industry standard for character animation

how to build rig?

Building a Rig

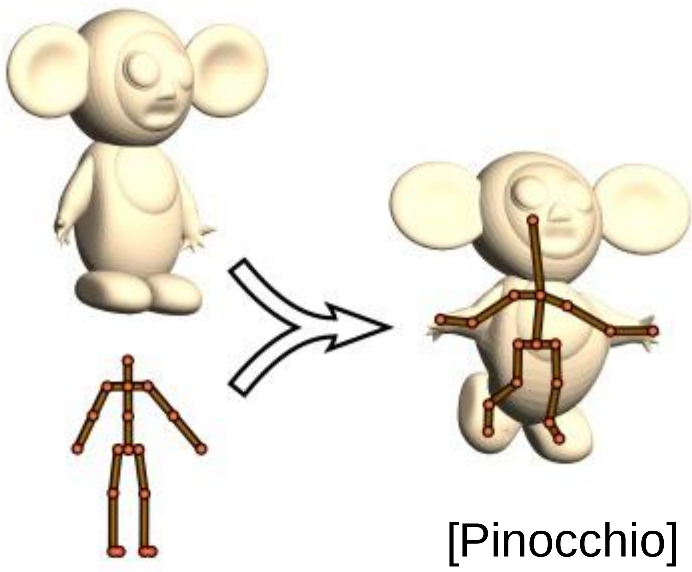
Usually done by hand using Maya etc.



Expressiveness/complexity tradeoff

Building a Rig

Some automatic tools exist

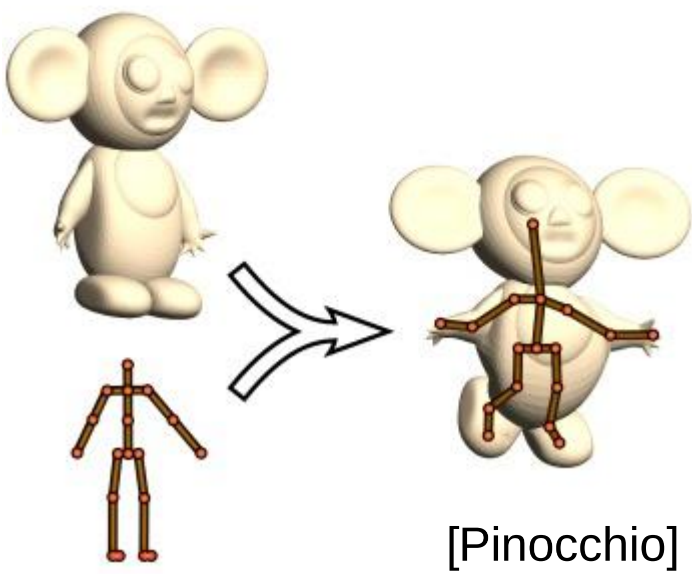


[Pinocchio]

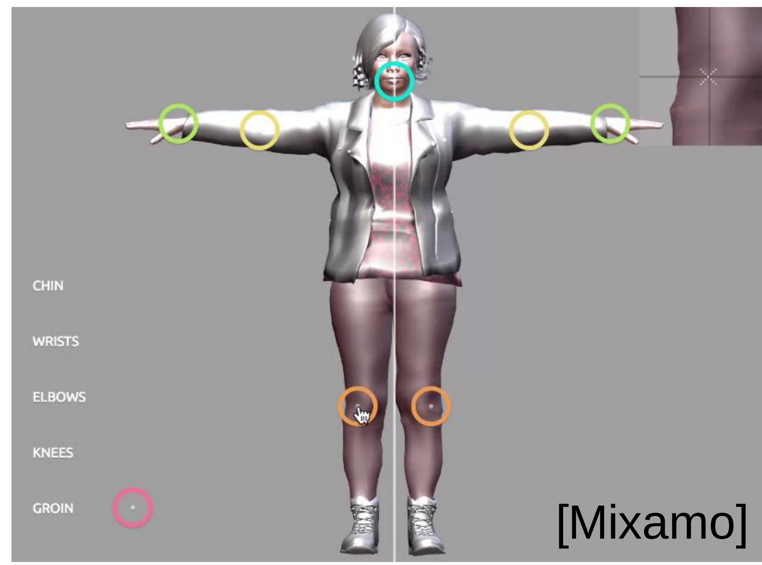
[Mixamo]

Building a Rig

Some automatic tools exist



[Pinocchio]



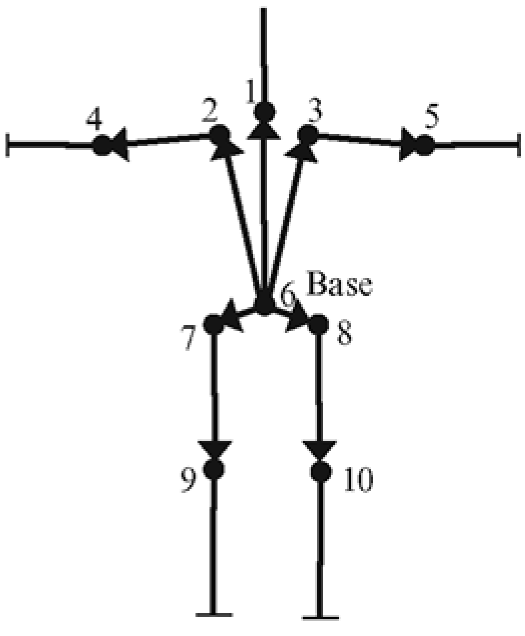
[Mixamo]

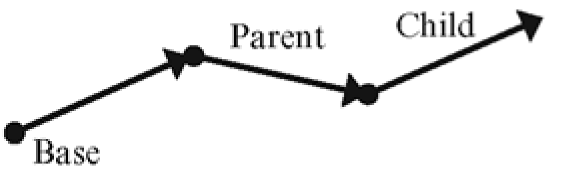
Work well for humans/humanlike objects

Not so impressive for arbitrary characters

Representing a Rig

Tree of **bones** connected by **joints**





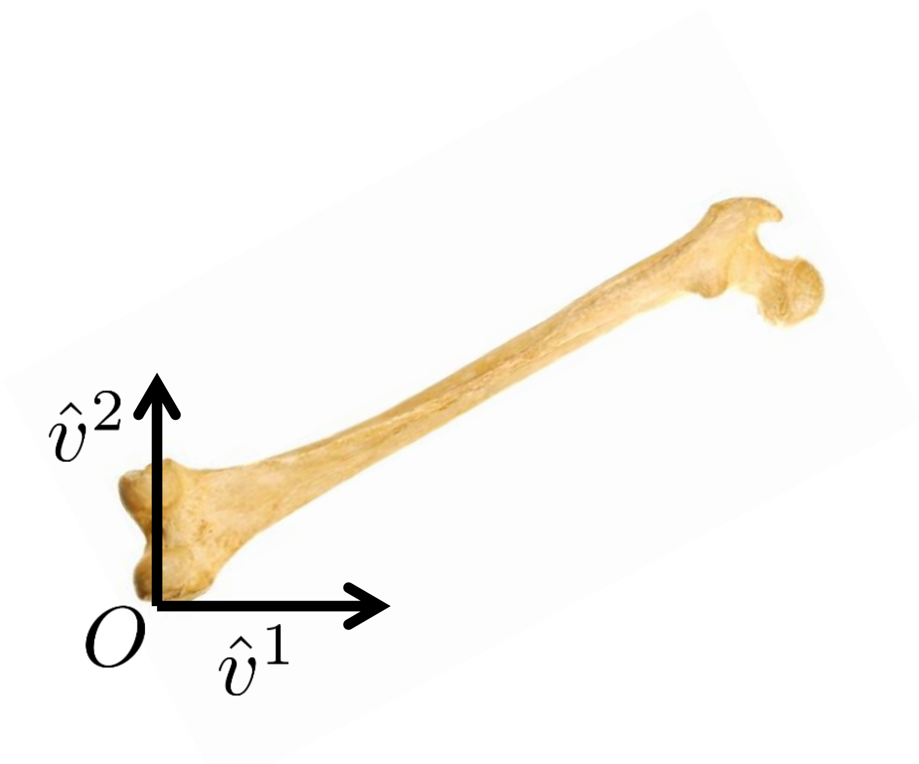
bones have two endpoints

- first attached to **parent**

Joint Local Coordinates

Origin O at joint location

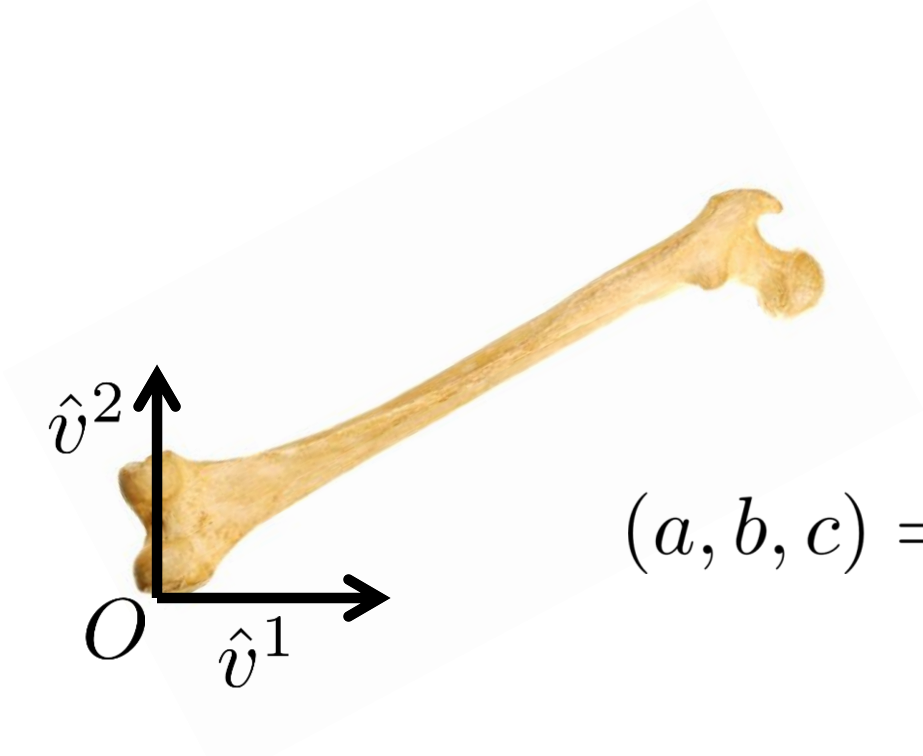
Coordinate axes: initially, same as world



Joint Local Coordinates

Origin O at joint location

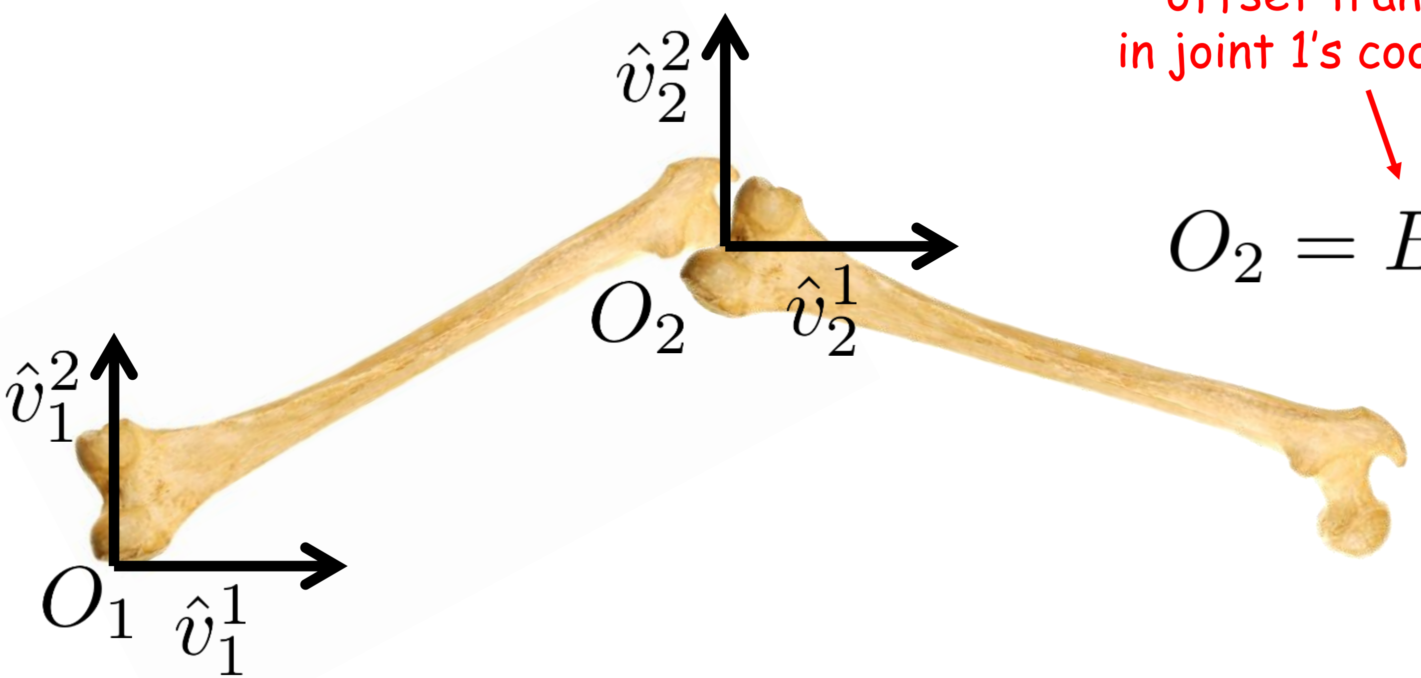
Coordinate axes: initially, same as world



$$(a, b, c) = O + a\hat{v}^1 + b\hat{v}^2 + c\hat{v}^3$$

Joint Local Coordinates

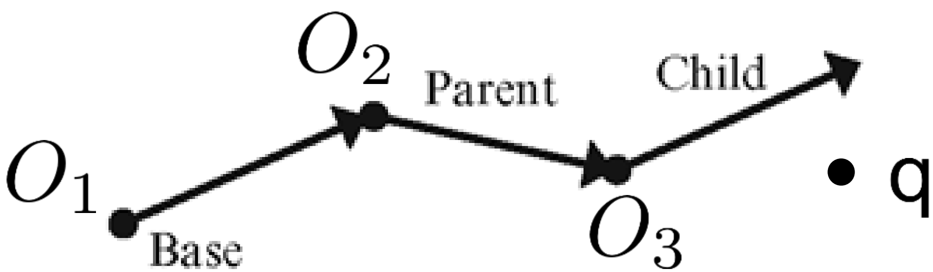
Child bone can be expressed in terms of **parent** coordinate system



offset translation in joint 1's coordinates

$$O_2 = B_{12} O_1$$

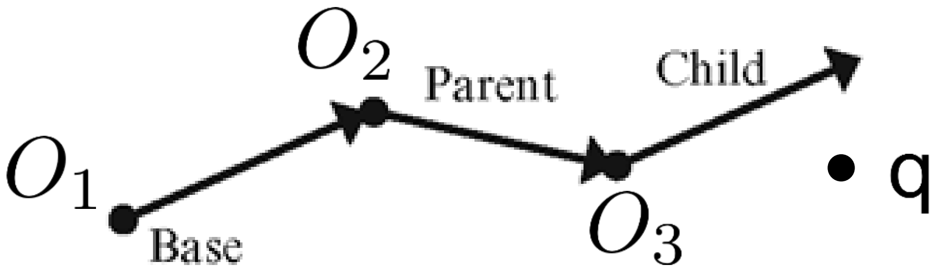
Joint to World Coordinates



In local coordinates:

$$q = (x, y, z)$$

Joint to World Coordinates



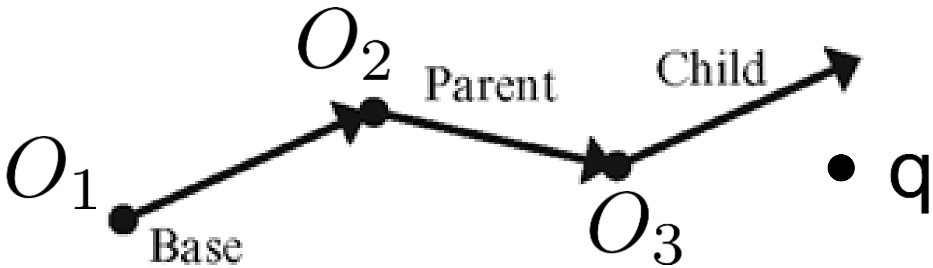
In local coordinates:

$$q = (x, y, z)$$

In world coordinates:

$$q = O_3 + x\hat{v}_1^3 + y\hat{v}_2^3 + z\hat{v}_3^3$$

Joint to World Coordinates



In local coordinates:

$$q = (x, y, z)$$

In world coordinates:

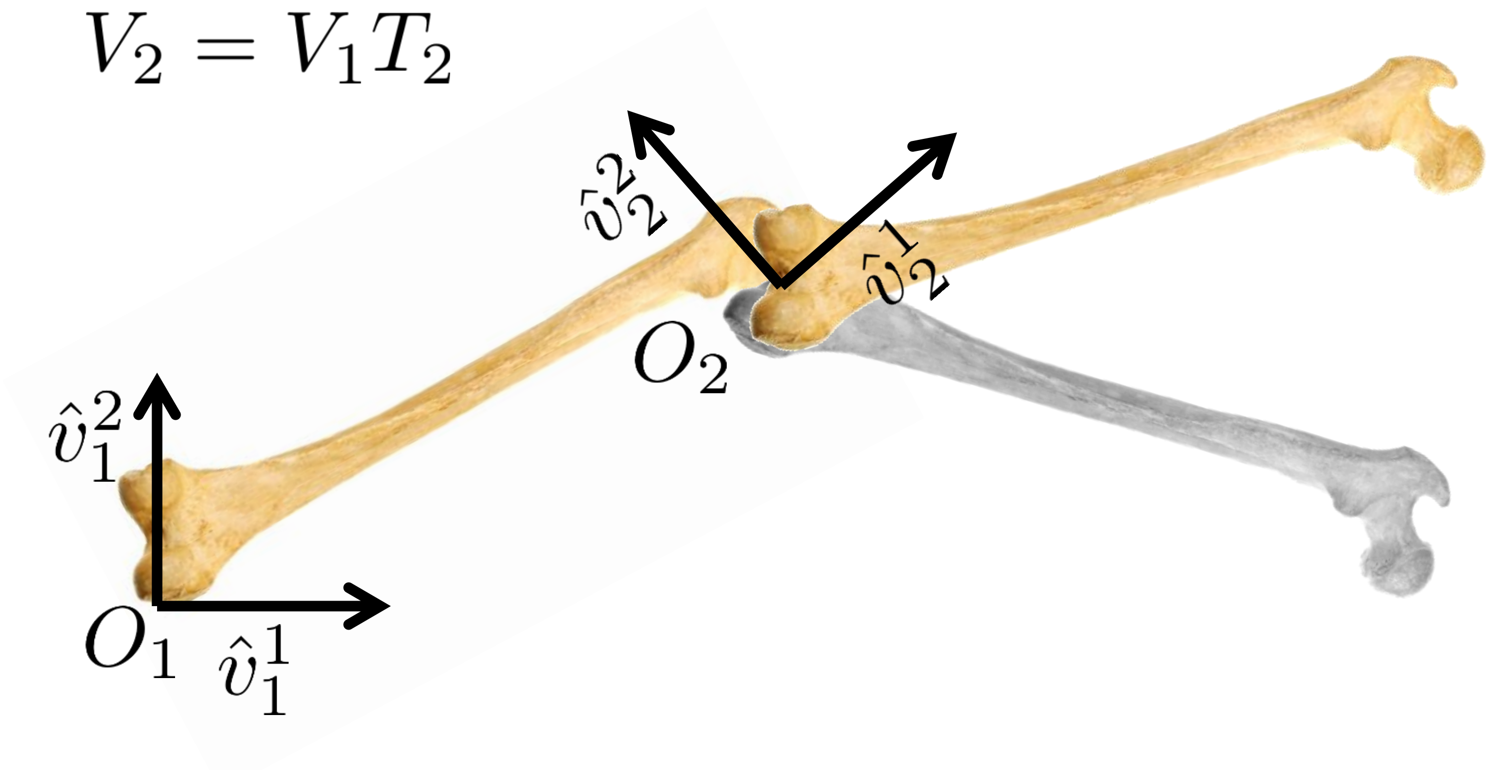
$$q = O_3 + x\hat{v}_1^3 + y\hat{v}_2^3 + z\hat{v}_3^3 = U_3 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$U_{i+1} = U_i B_{i(i+1)}$$

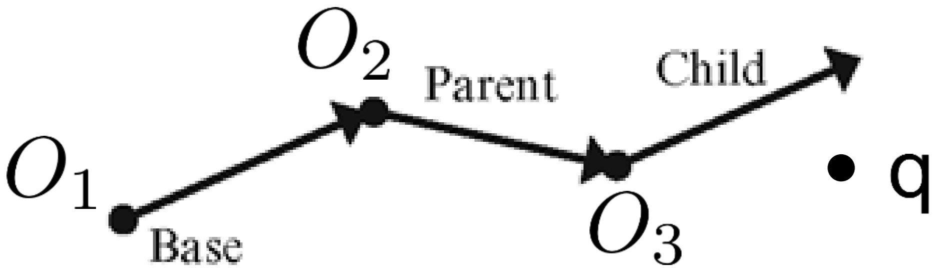
Articulating the Joints

Represent motion by a rotation T_i per joint

$$O_2 = B_{12} O_1$$
$$V_2 = V_1 T_2$$



Deformed Joint to World



In local coordinates:

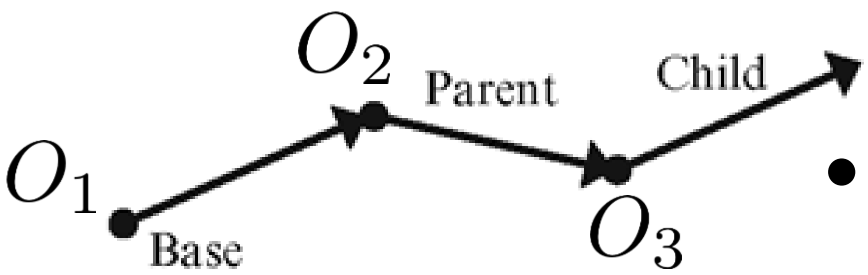
$$q = (x, y, z)$$

In world coordinates:

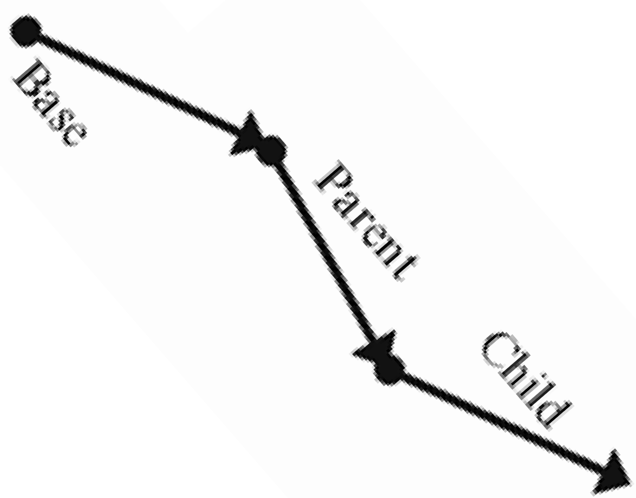
$$q = O_3 + x\hat{v}_1^3 + y\hat{v}_2^3 + z\hat{v}_3^3 = D_3 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$D_{i+1} = D_i B_{i(i+1)} T_{i+1}$$

Forward Kinematics



- $q_{\text{world}} = U_3 q_{\text{local}}$



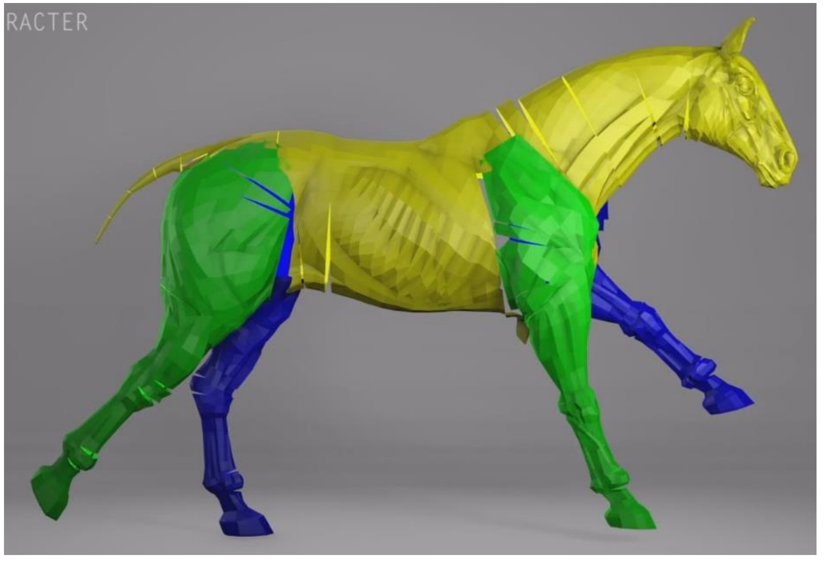
changing T_1 **also** changes child coordinate systems

- $q_{\text{world}} = D_3 q_{\text{local}}$

Skinning

Moving bones moves the character

Closer bones have more influence

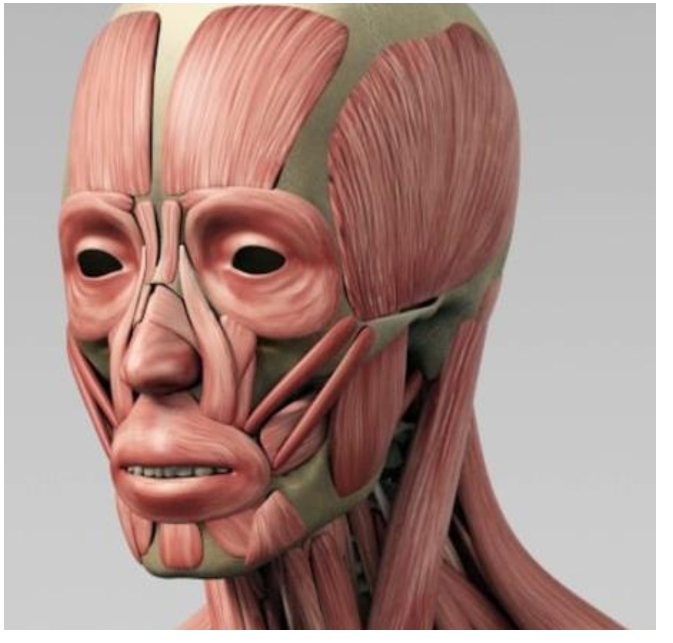


We will discuss the details on Thursday

Other 3D Animation Techniques

Anatomy-based animation

- simulate the tendons and muscles



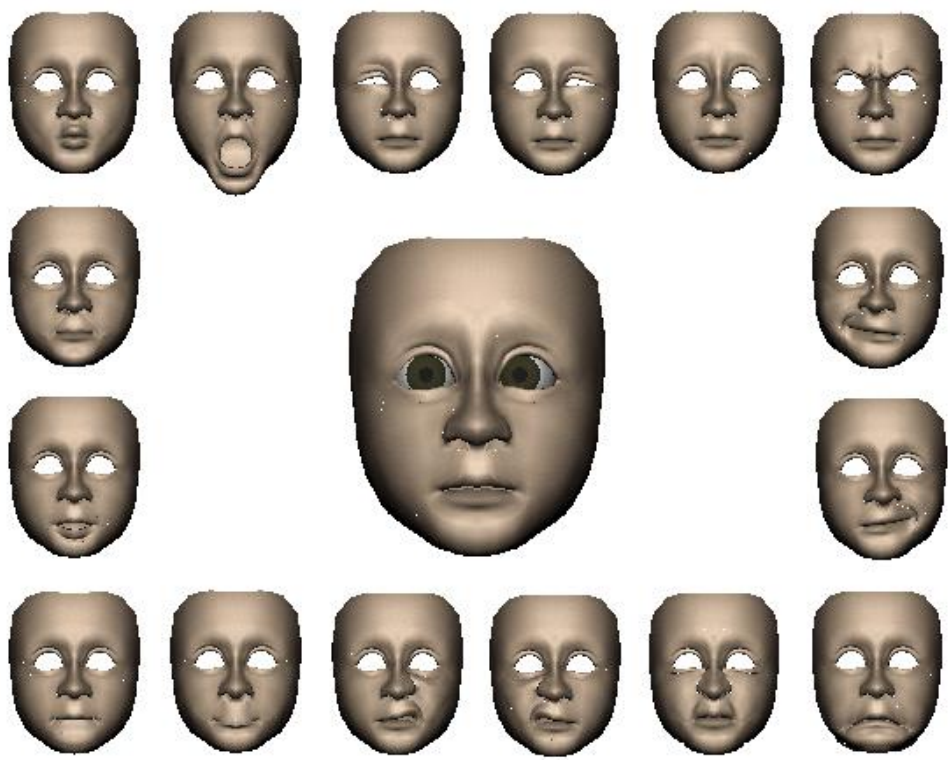
Most correct motion

Restricted to “real” animals

Slow

Other 3D Animation Techniques

Faces are hard to rig: many muscles

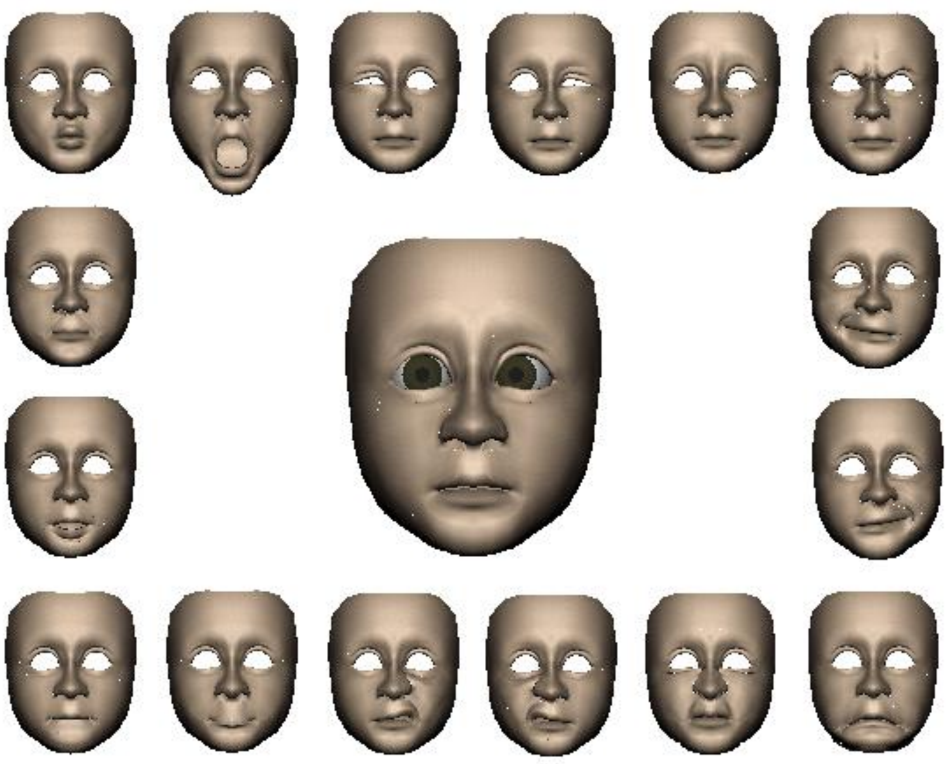


instead, precompute a small set of **basis deformations**

blend between them (**blendshapes**)

Other 3D Animation Techniques

Faces are hard to rig: many muscles



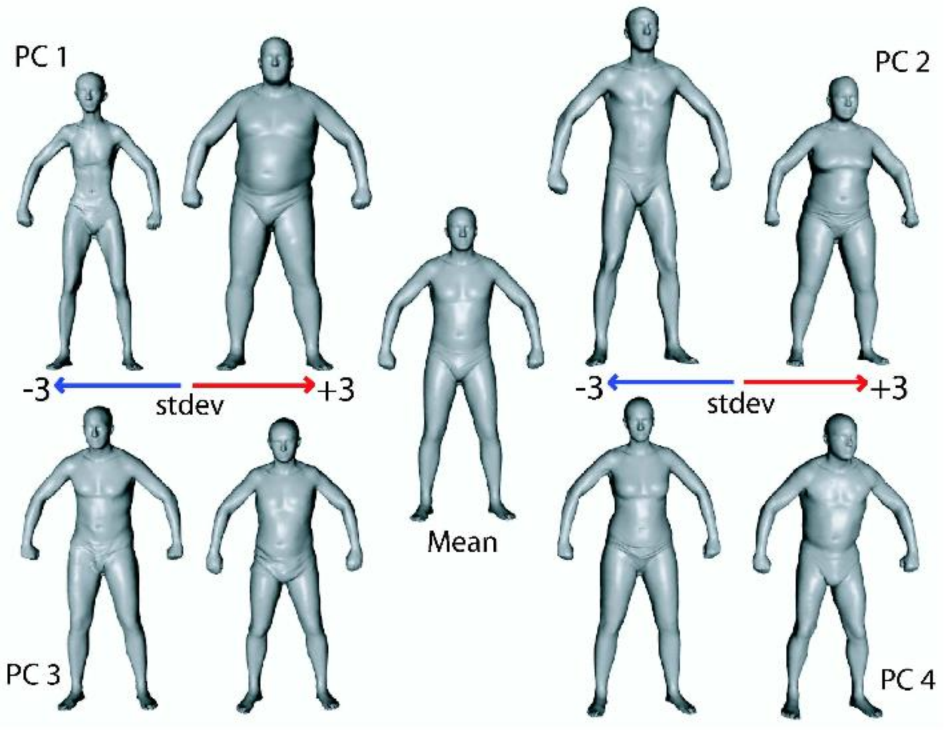
instead, precompute a small set of **basis deformations**

blend between them (**blendshapes**)

knobs not always intuitive

Other 3D Animation Techniques

SCAPE: data-driven statistical model of human body shapes



rig a “mean” template human once by hand

for an arbitrary human, transfer the rig by fitting to a SCAPE model

works well only for (nearly) naked humans