Skinning

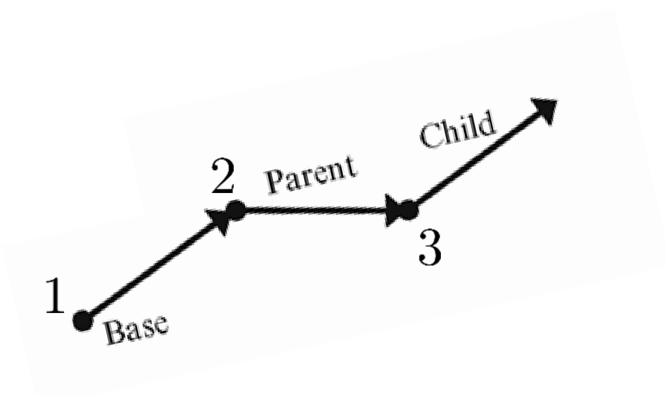
Nearest-Bone Skinning

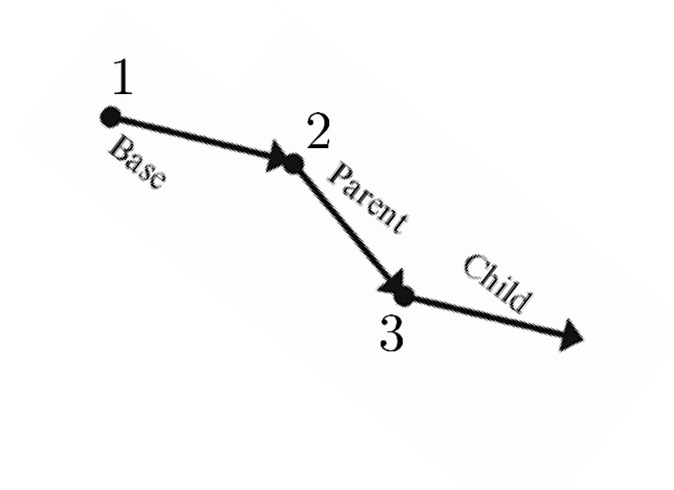
Recall **skinning:** given motion of skeleton, how does skin move?

Nearest-Bone Skinning

Recall **skinning:** given motion of skeleton, how does skin move?

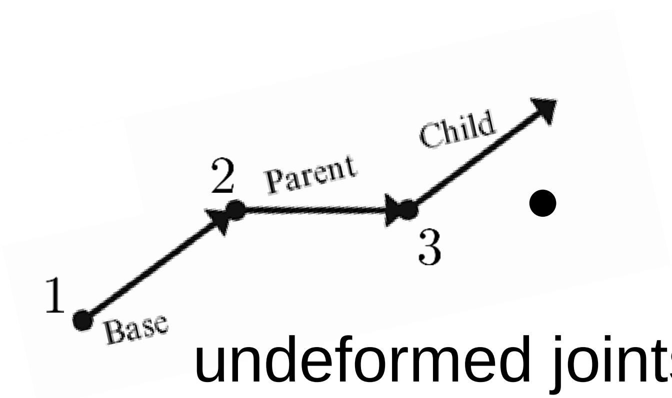
Given: **undeformed** (rest) skeleton and **deformed** skeleton



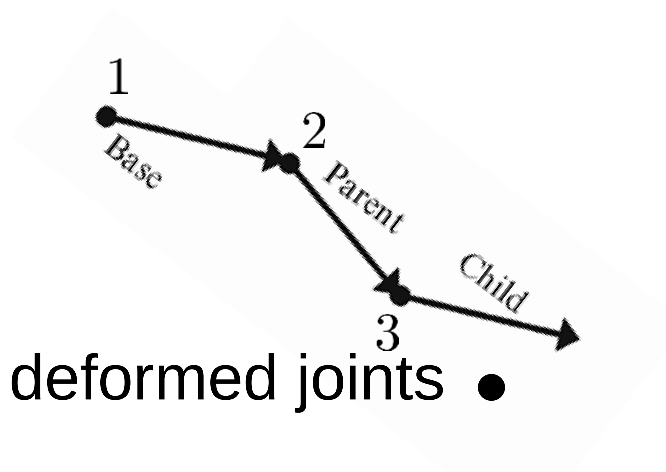


Coordinate Systems

world



undeformed joints



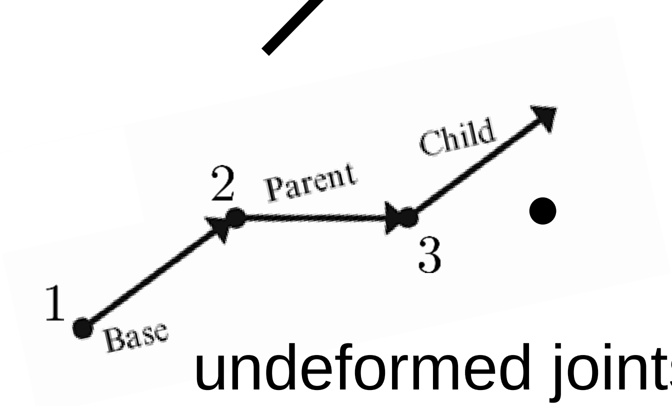
deformed joints

Coordinate Systems

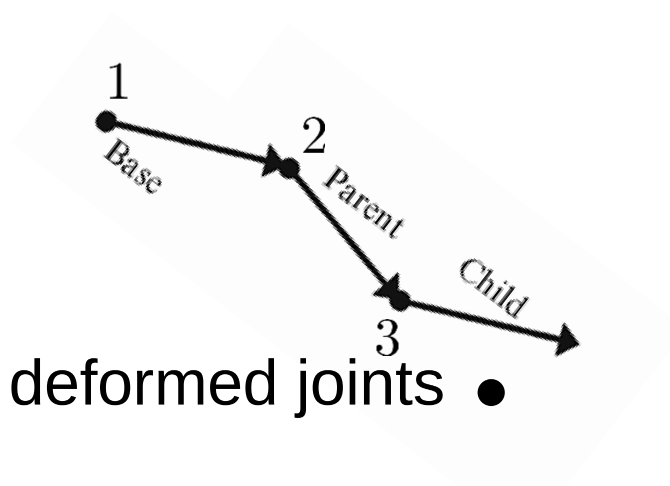
translation (offset) of
joint 3 wrt. joint 2

world

$$U_3 = U_2 B_{23}$$



undeformed joints



deformed joints

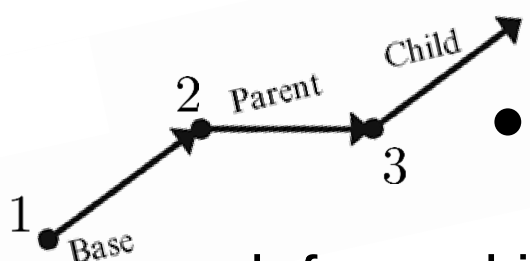
Coordinate Systems

translation (offset) of
joint 3 wrt. joint 2

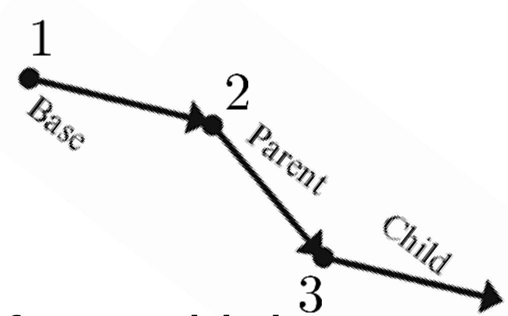
map from parent
coords to world
(recursive)

$$U_3 = U_2 B_{23}$$

world

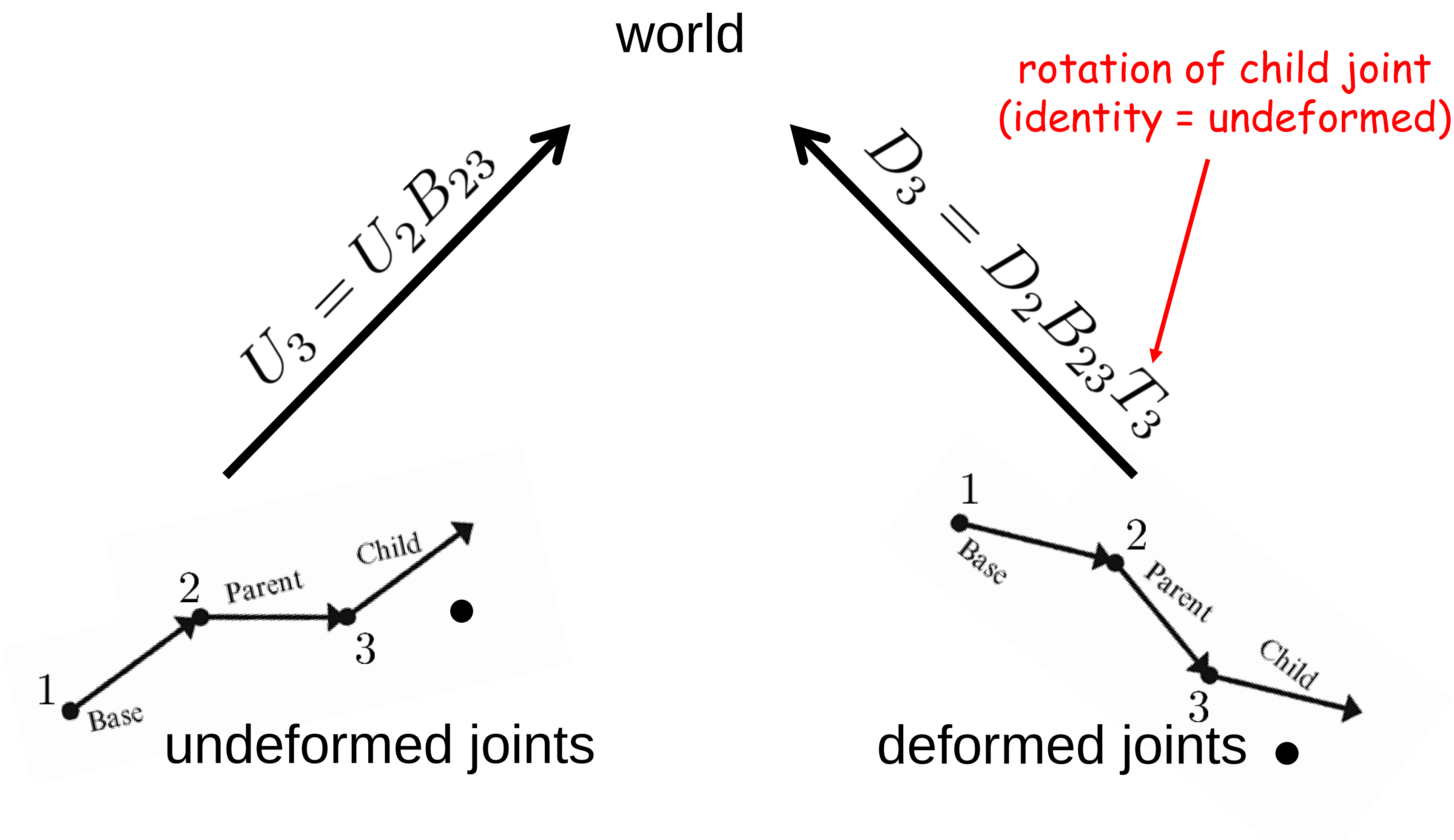


undeformed joints

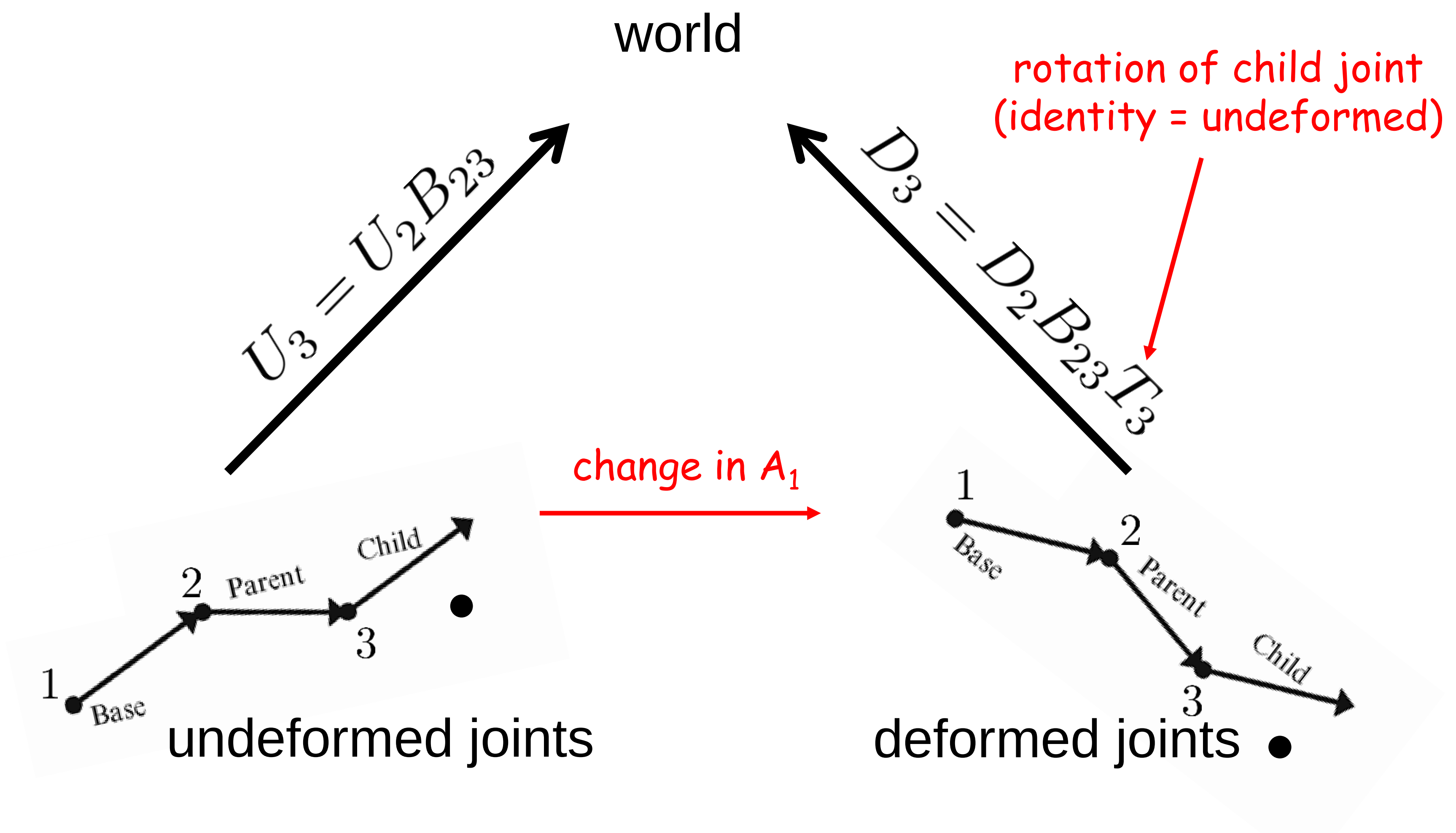


deformed joints

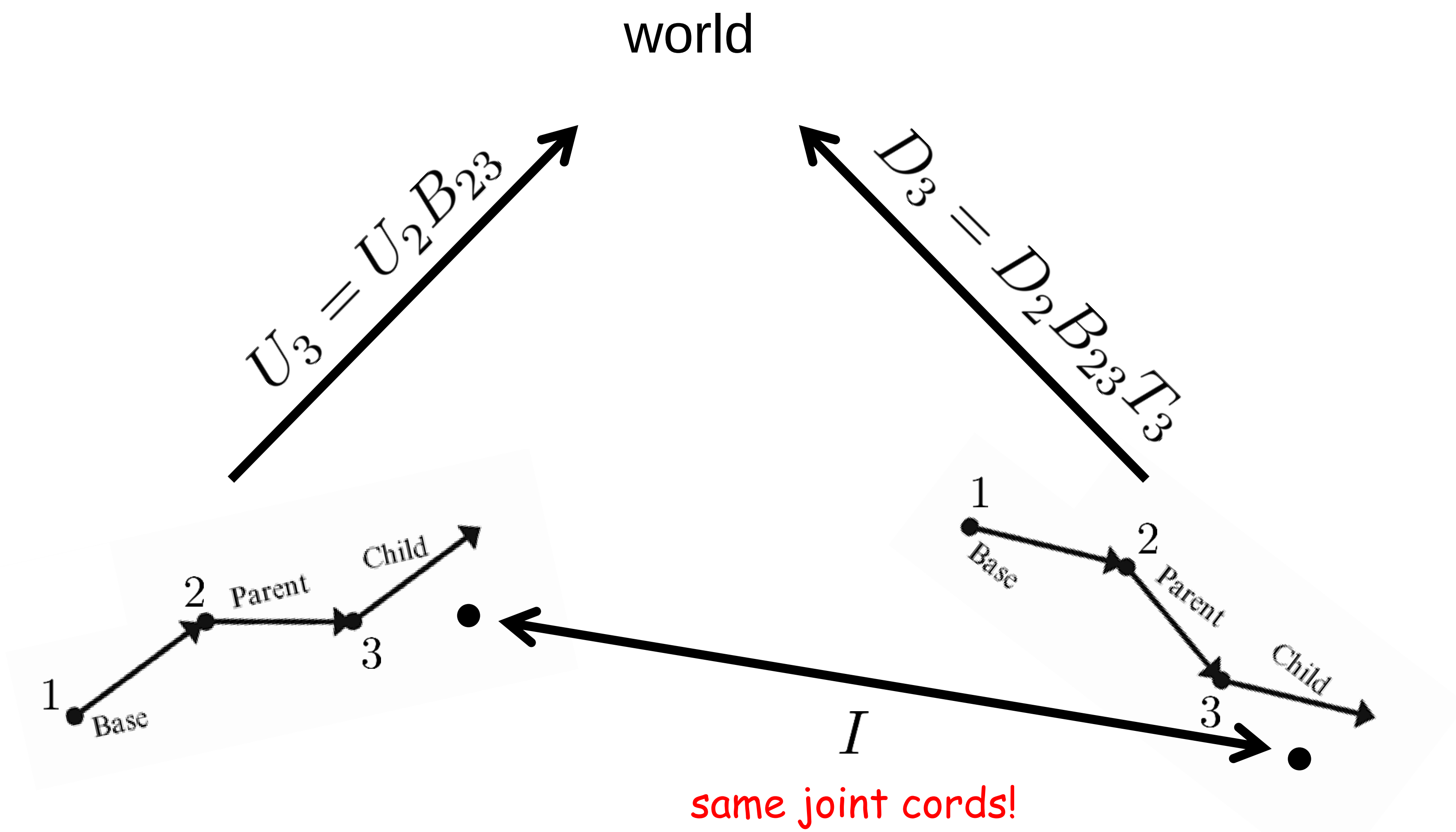
Coordinate Systems



Coordinate Systems



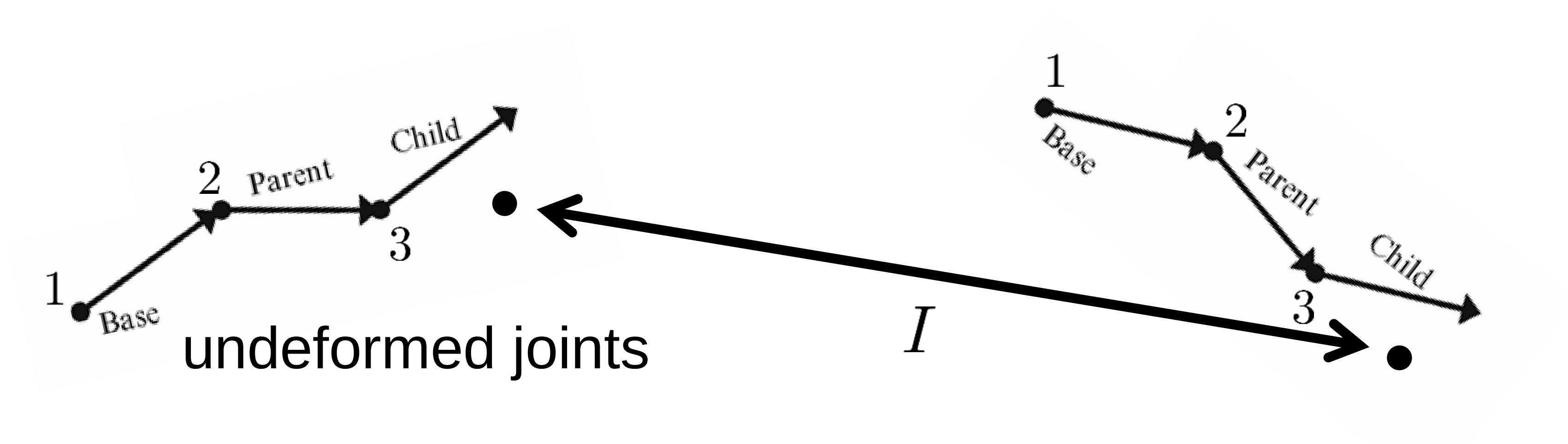
Coordinate Systems



Coordinate Systems

Key (and confusing) point:

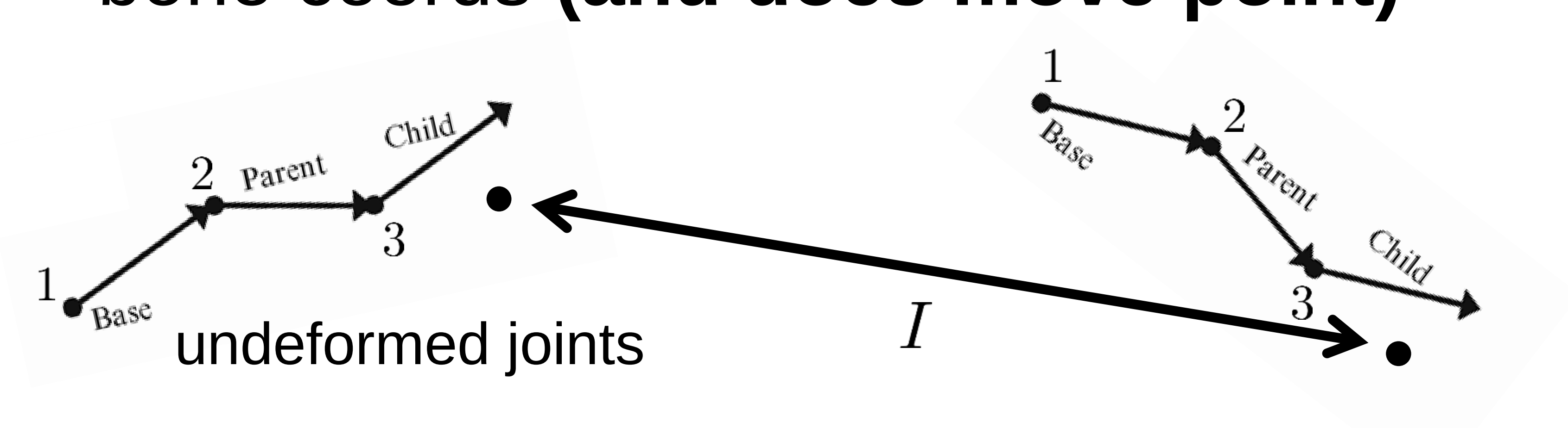
- U_3 maps from undeformed local to world coords **(doesn't move point)**



Coordinate Systems

Key (and confusing) point:

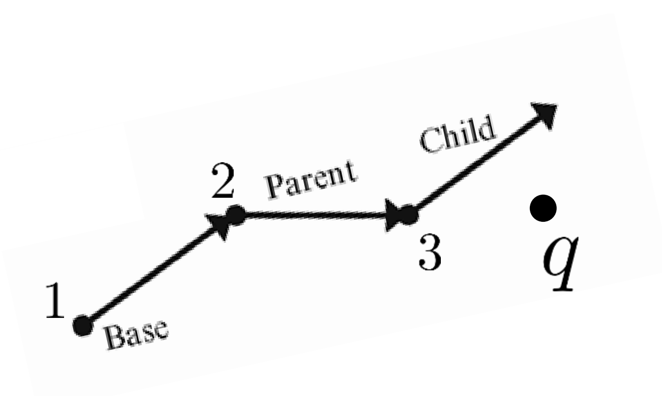
- U_3 maps from undeformed local to world coords **(doesn't move point)**
- **Identity** maps undeformed to deformed bone coords **(and does move point)**

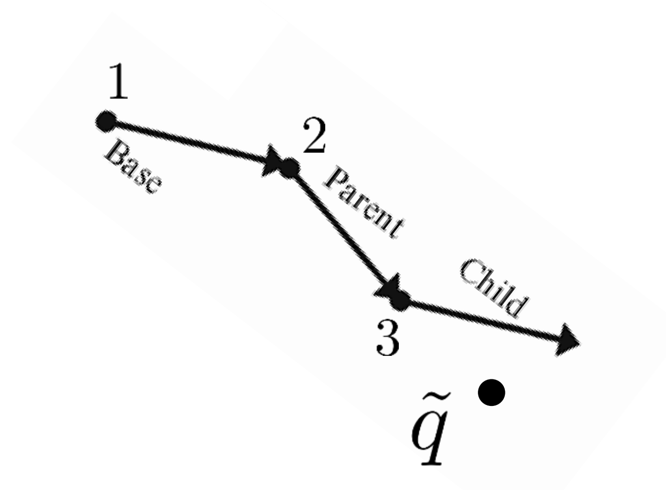


Nearest-Bone Skinning

Undeformed to deformed skin position (world coordinates):

$$\tilde{q} = D_3 U_3^{-1} q$$



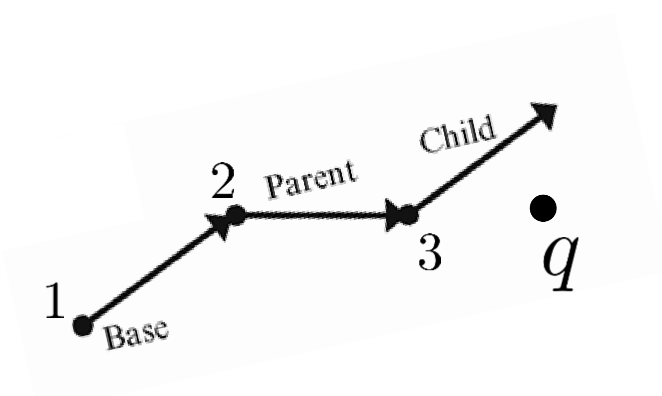


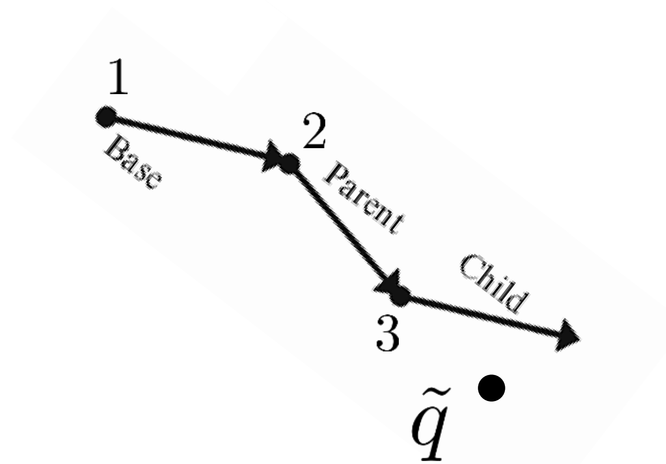
Nearest-Bone Skinning

Undeformed to deformed skin position
(world coordinates):

$$\tilde{q} = D_3 U_3^{-1} q$$

joint controlling
nearest bone





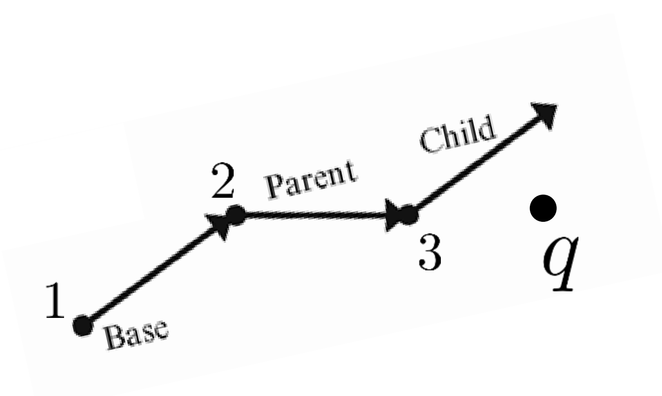
Nearest-Bone Skinning

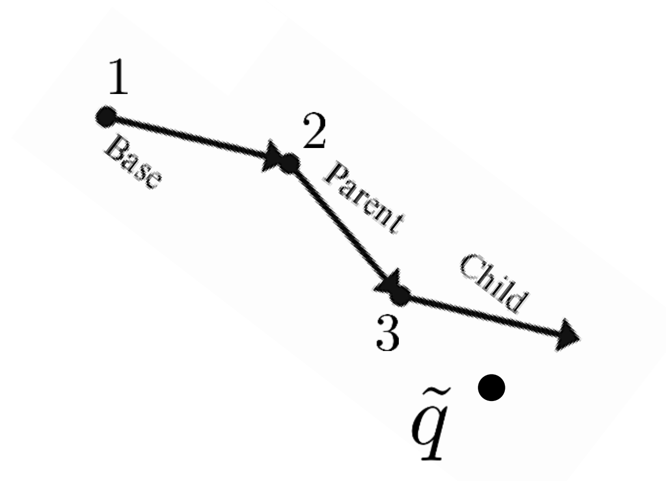
Undeformed to deformed skin position (world coordinates):

$$\tilde{q} = D_3 U_3^{-1} q$$

changes during animation

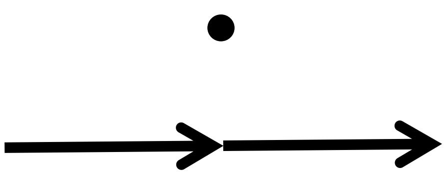
joint controlling nearest bone

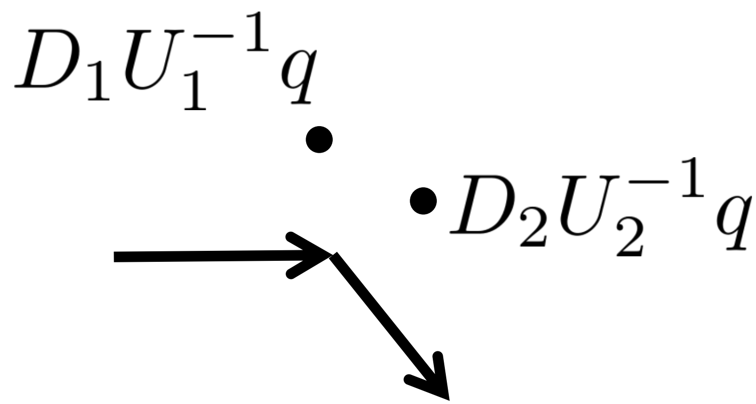




Problems with Nearest-Bone

Which bone does point belong to?

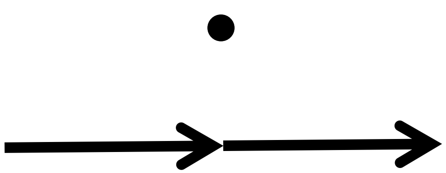


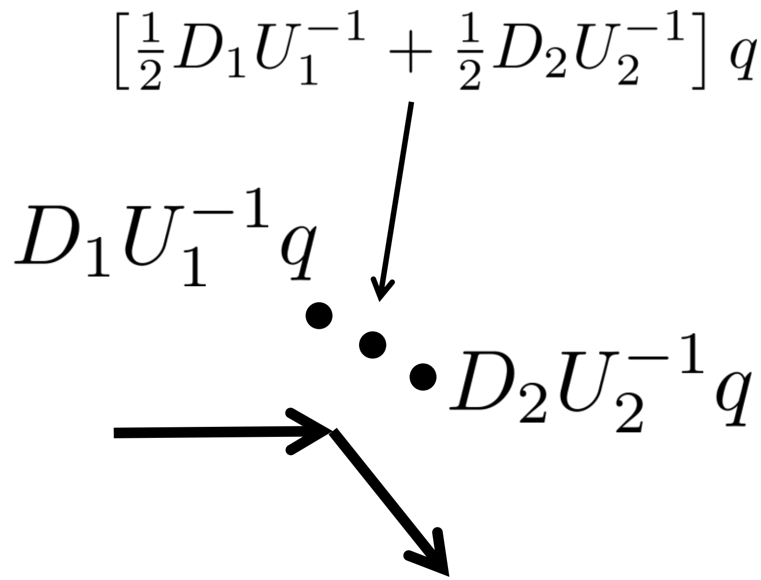


Problems with Nearest-Bone

Which bone does point belong to?

One solution: **average**



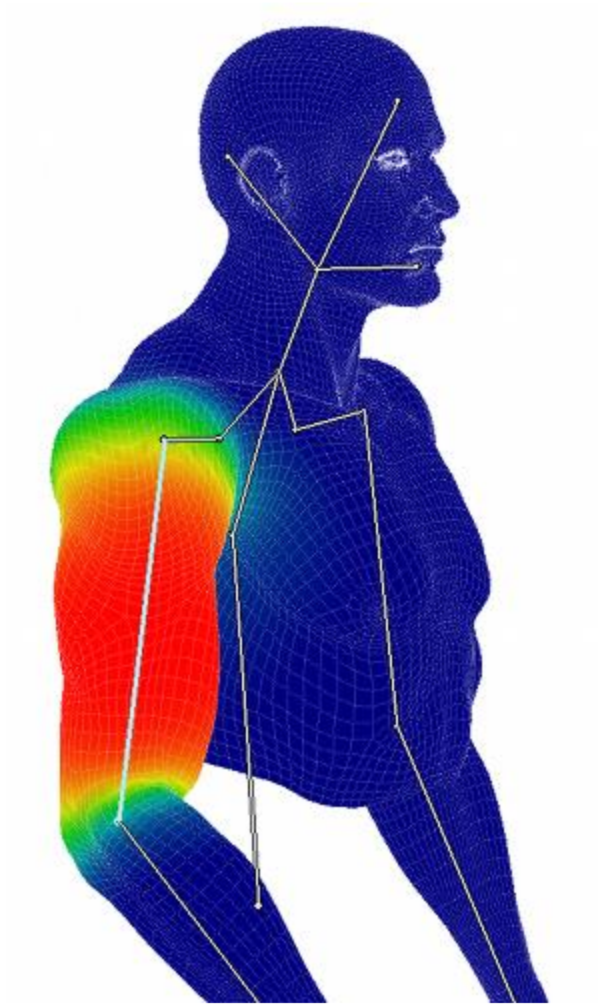


Linear-Blend Skinning

Each vertex feels **weighted average** of each joint's transformations

$$\tilde{q}_i = \sum\nolimits_{\text{joints } j} w_{ij} D_j U_j^{-1} q_i$$

Joints controlling nearby bones have higher weight



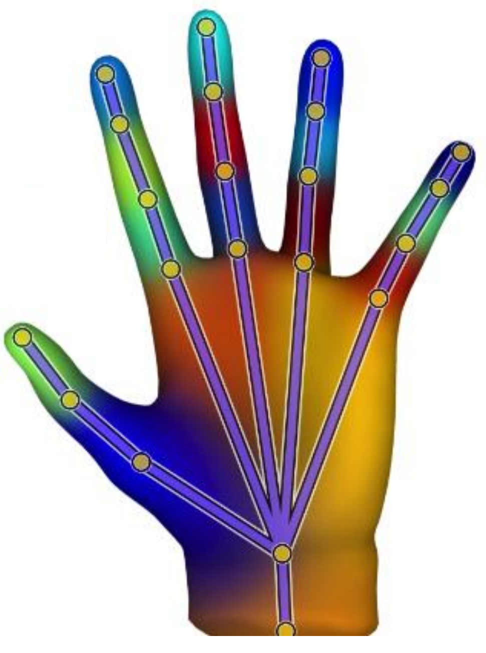
Linear-Blend Skinning

How to determine **skinning weights** w?

Use only nearest bone

Linear-Blend Skinning

How to determine **skinning weights** w?

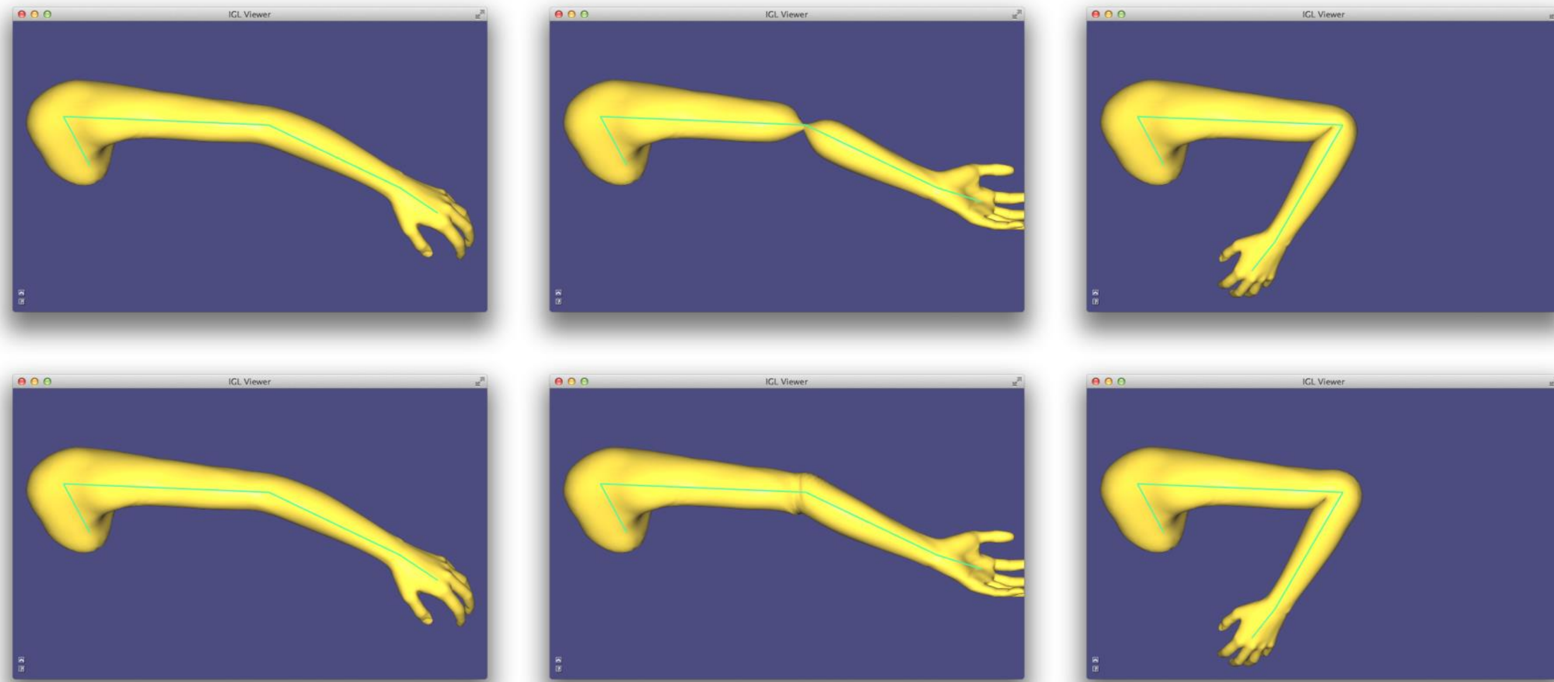


Use only nearest bone

Spatially blend the weights

(In practice: paint weights by hand)

The “Arm Twist” Problem



(Why does this happen?)

Blending Transformations

Each individual joint undergoes a **rigid transformation**

- combination rotation and translation

Blending Transformations

Each individual joint undergoes a **rigid transformation**

- combination rotation and translation

Arm twist / candy wrapper problem:

- linear blend of rigid motions not rigid

Blending Transformations

Translations alone: trivial to blend

$$\{\mathbf{t}_1, \ldots, \mathbf{t}_n\} \mapsto \sum_i \alpha_i \mathbf{t}_i$$

Blending Transformations

Translations alone: trivial to blend

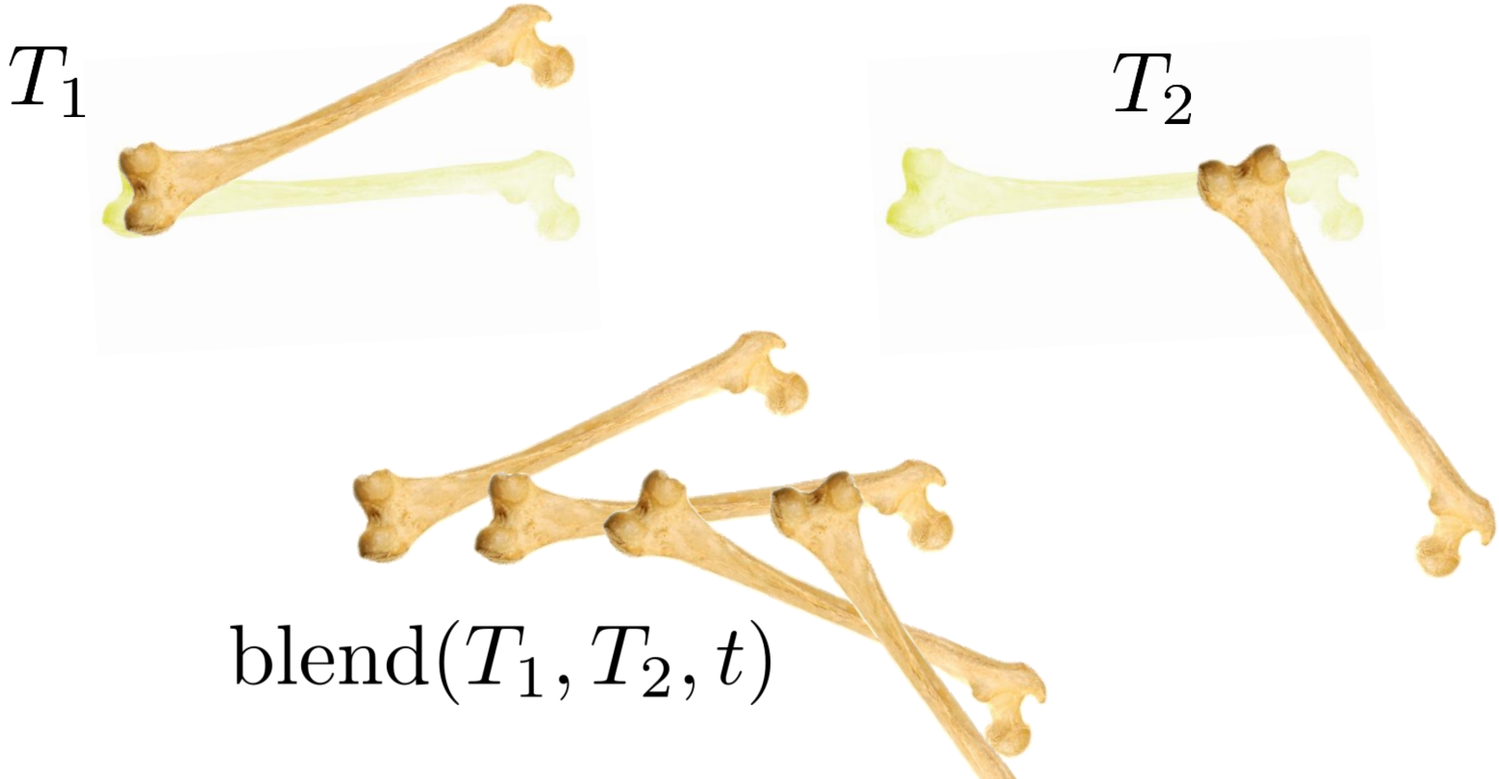
$$\{\mathbf{t}_1, \ldots, \mathbf{t}_n\} \mapsto \sum_i \alpha_i \mathbf{t}_i$$

Rotations alone: blend using SLERP

- use quaternions
- **do not use Euler angles!**

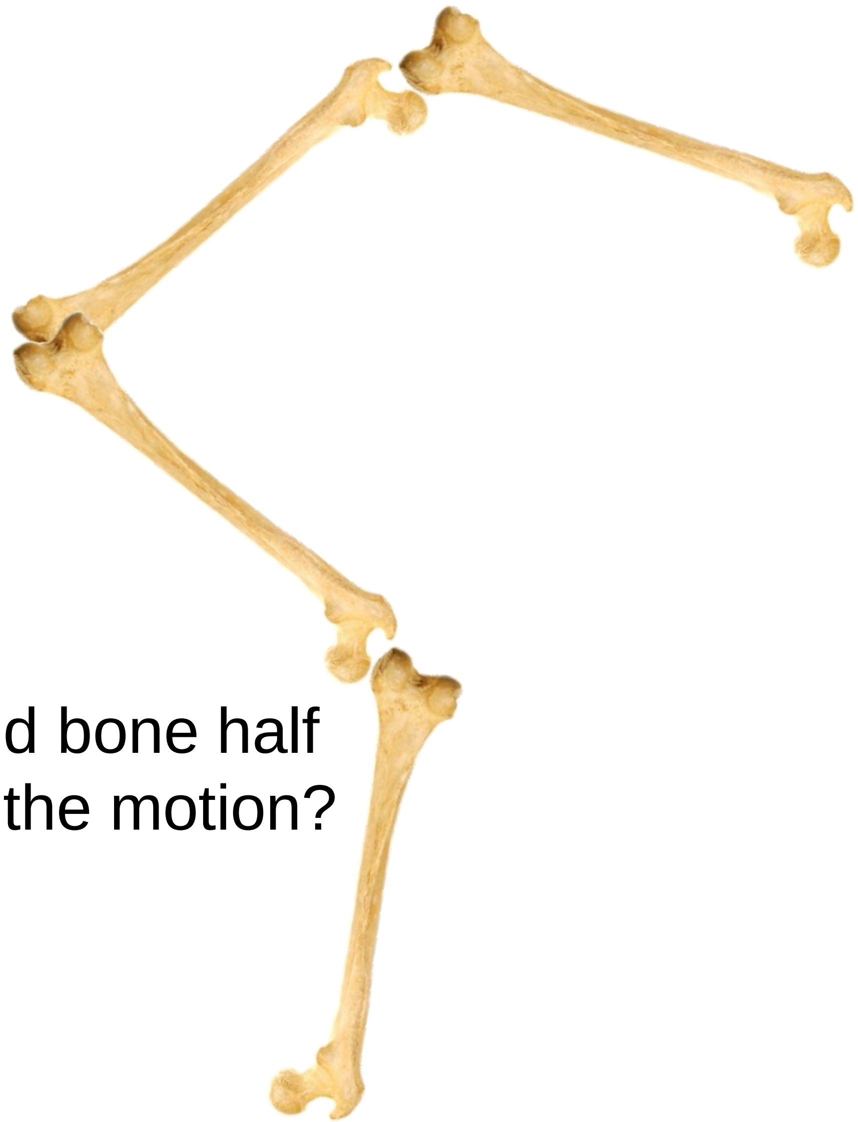
Blending Transformations

Idea: separately blend translation and rotation components of rigid motion



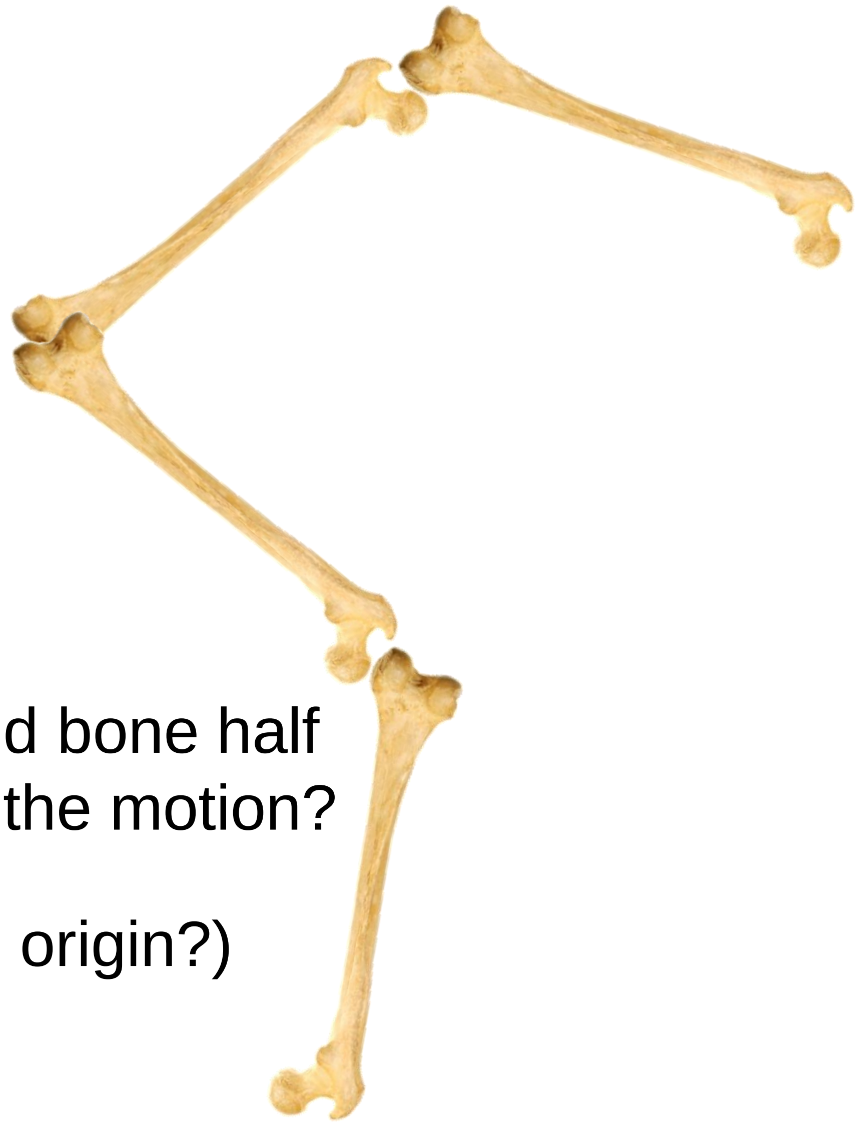
$\mathrm{blend}(T_1, T_2, t)$

Separate Transforms: Problem



where is the child bone half way in between the motion?

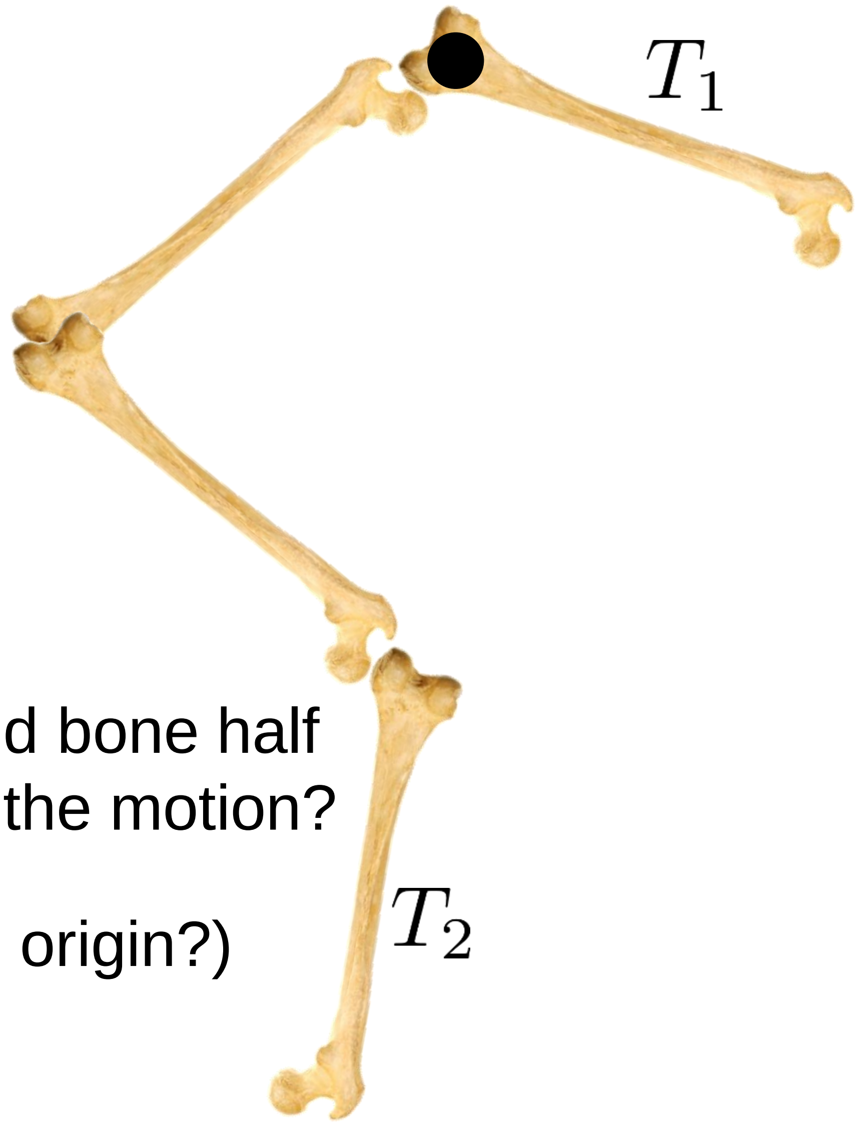
Separate Transforms: Problem



where is the child bone half way in between the motion?

(where it the origin?)

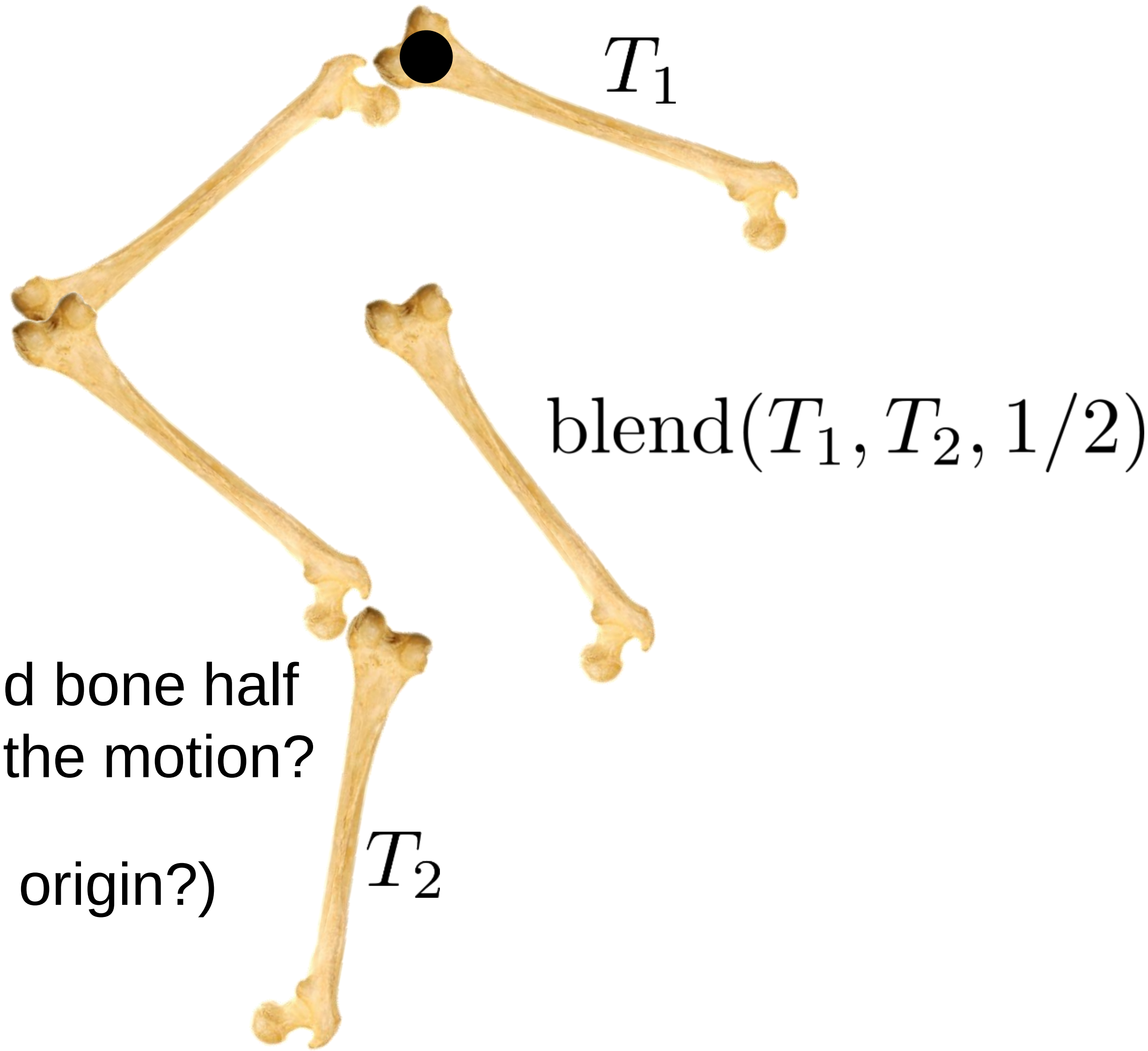
Separate Transforms: Problem



where is the child bone half way in between the motion?

(where it the origin?)

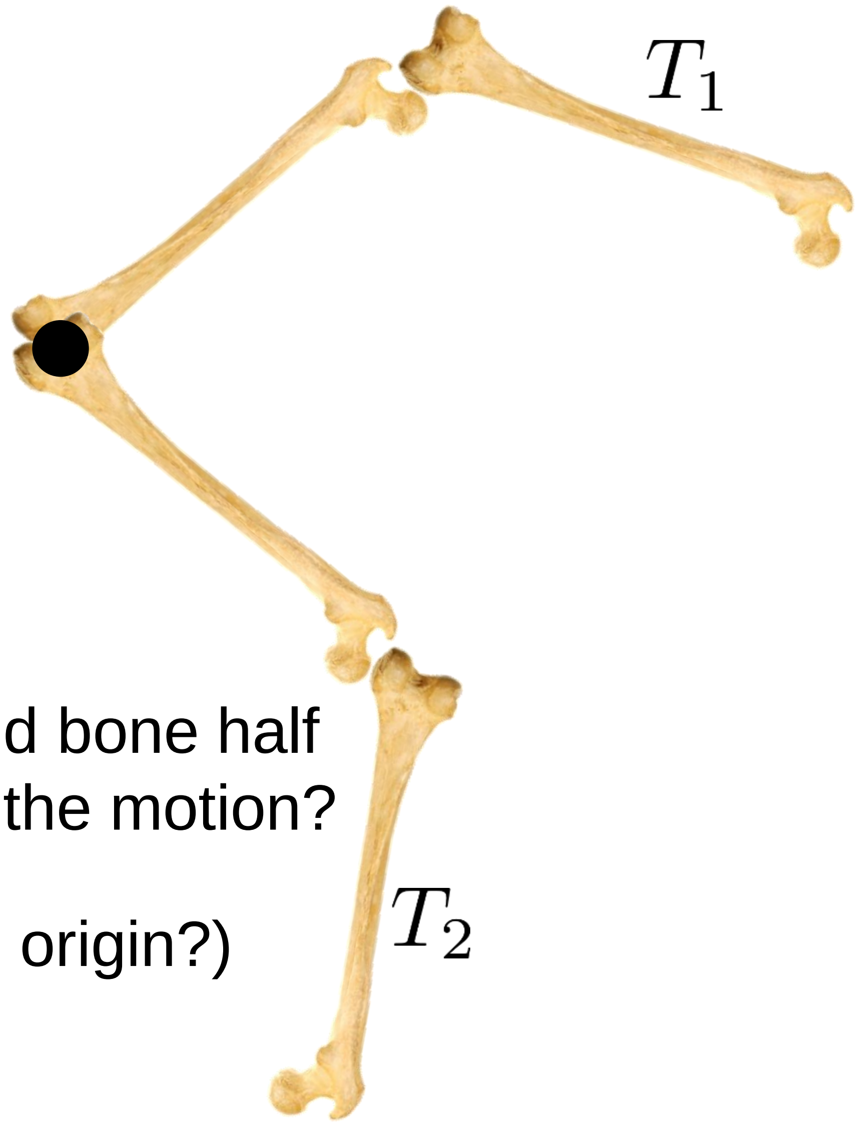
Separate Transforms: Problem



where is the child bone half way in between the motion?

(where it the origin?)

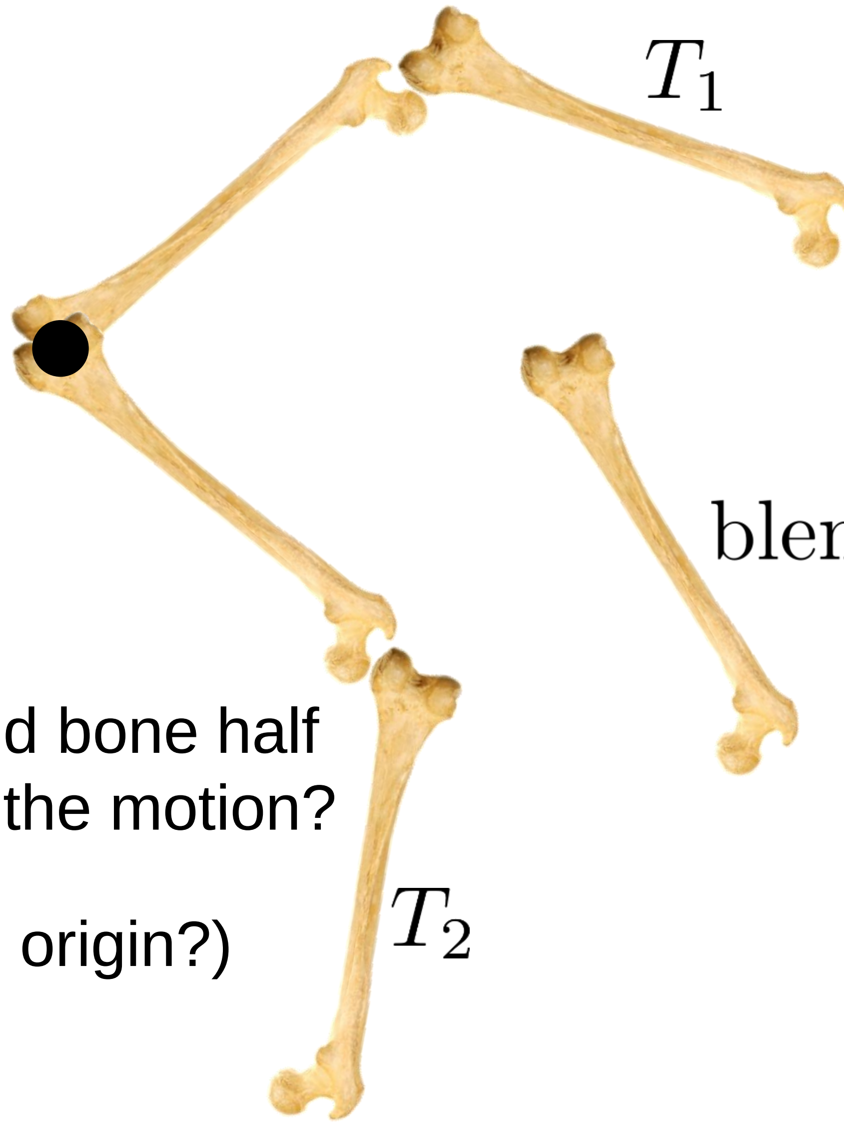
Separate Transforms: Problem



where is the child bone half way in between the motion?

(where it the origin?)

Separate Transforms: Problem



where is the child bone half way in between the motion?

(where it the origin?)

Separate Transforms: Problem

Blended transformation **not** coordinate-independent

- different origins --> totally different blends

Separate Transforms: Problem

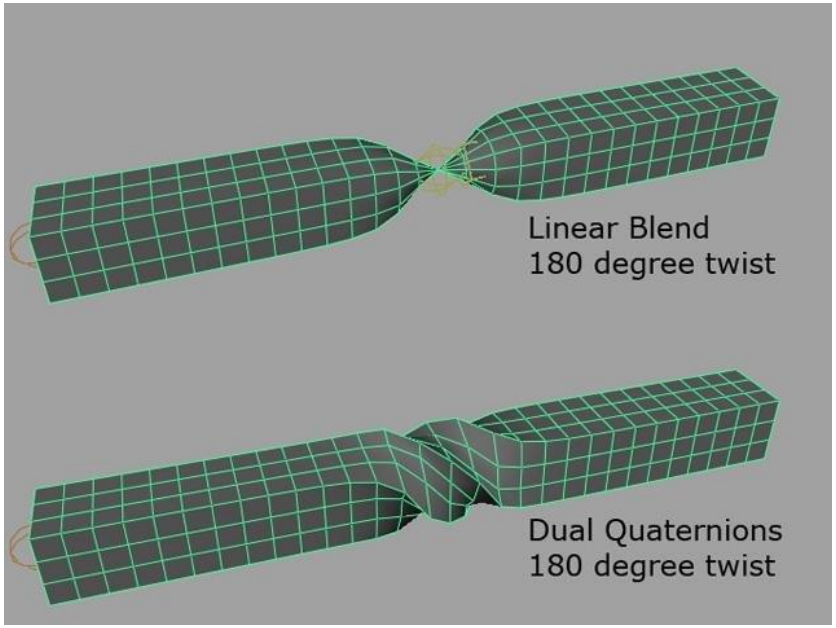
Blended transformation **not** coordinate-independent

- different origins --> totally different blends

- must also blend **centers of rotation**

Dual Quaternion Skinning

Represents rigid motion as pair of quaternions, solving all these problems

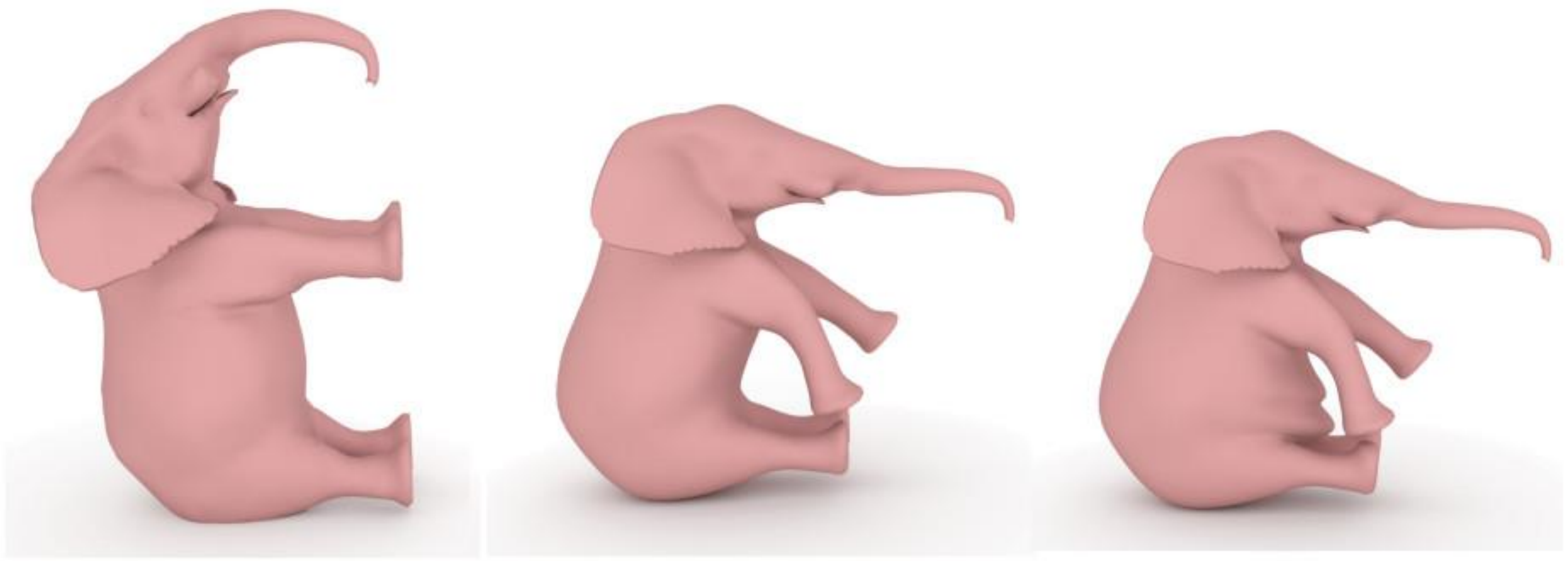


no more arm twisting issues

used in Maya, etc.

Other Skinning Challenges

Volume Conservation



Avoiding Animation Altogether

Motion capture ("mocap")

Avoiding Animation Altogether

Simulation ("inverse kinematics")

Animation Recap

Most common pipeline:

- build a 3D model of the character
- **rig** the 3D model (build a skeleton inside)
- **skin** the model (determine joint-skin weights)
- animate the skeleton by specifying **keyframes**; skin moves with it

Animation Recap

Most common pipeline:

- model, rig, skin, animate

Automatic approaches exist for each step

- not great, but getting better

Animation Recap

Most common pipeline:

- model, rig, skin, animate

Automatic approaches exist for each step

- not great, but getting better

Still a grand challenge: use Kinect to build a fully rigged and skinned digital avatar