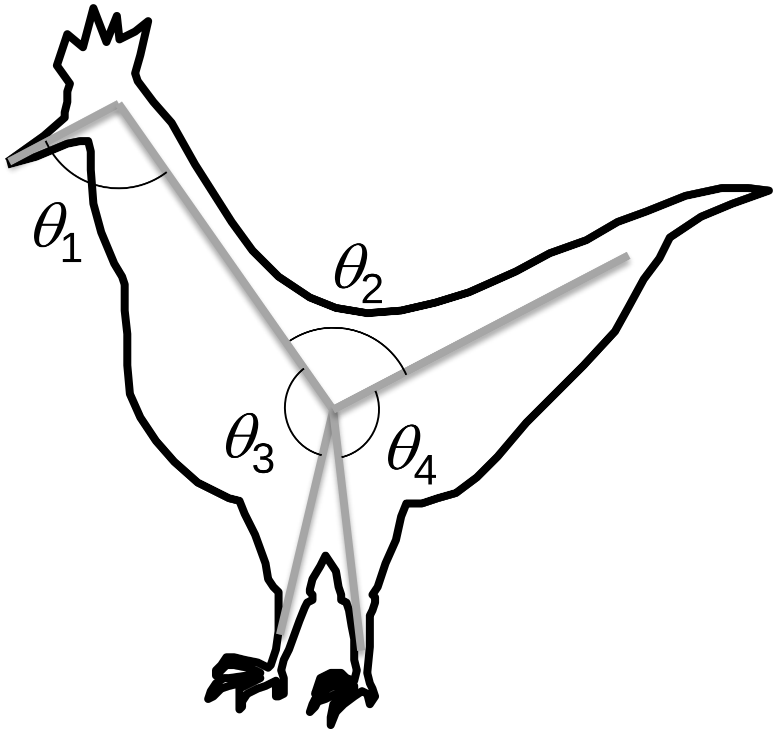
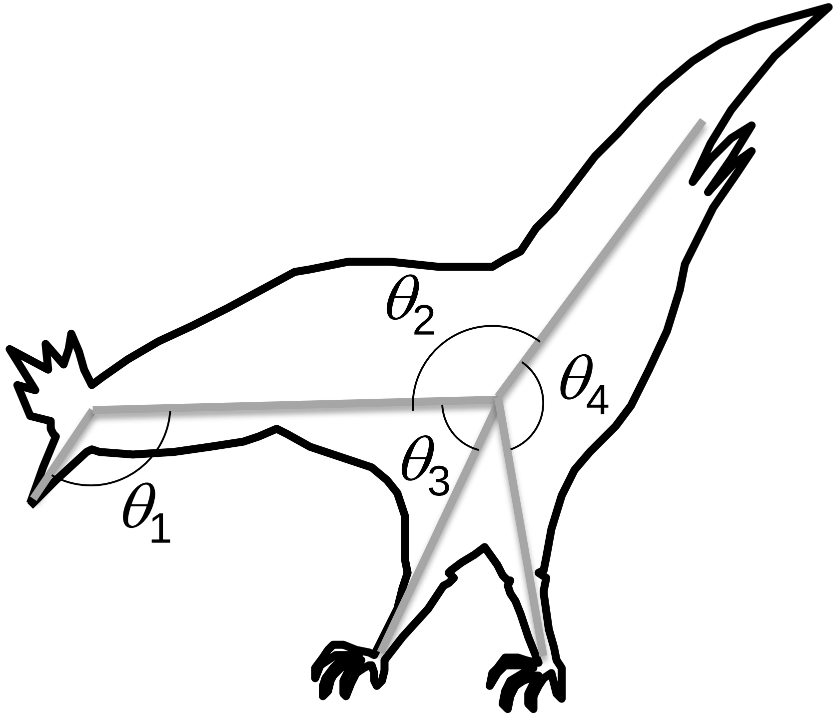
Skinning

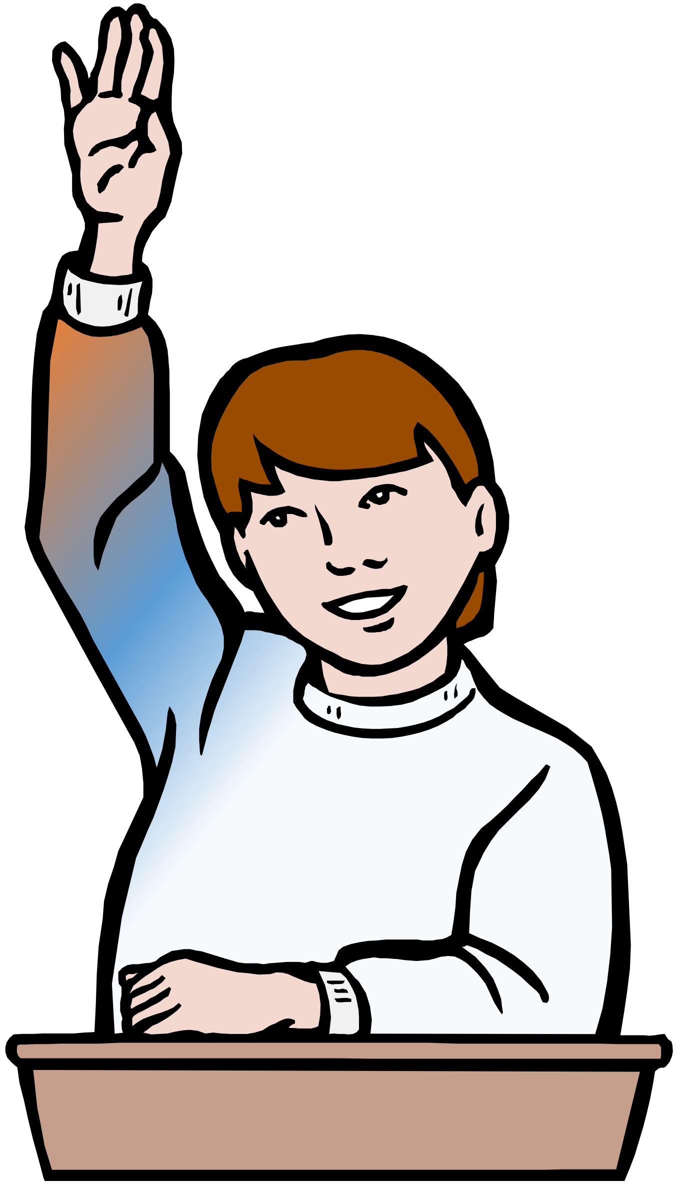
CS418 Interactive Computer Graphics

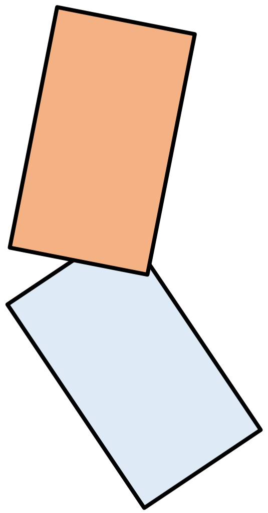
John C. Hart

Modeling Posed Shapes from Bones





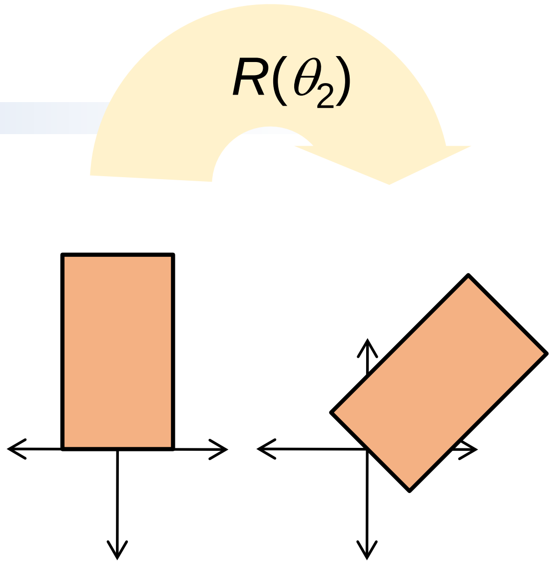




“bones” are coordinate frames

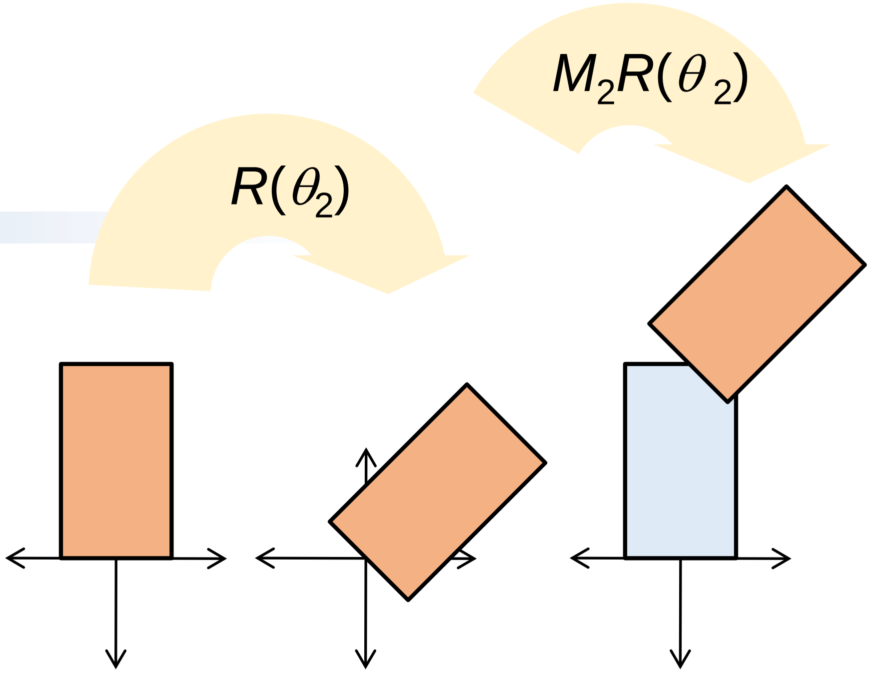
Skinning

- $R(\theta_2)$ rotates forearm cylinder about its elbow at the origin



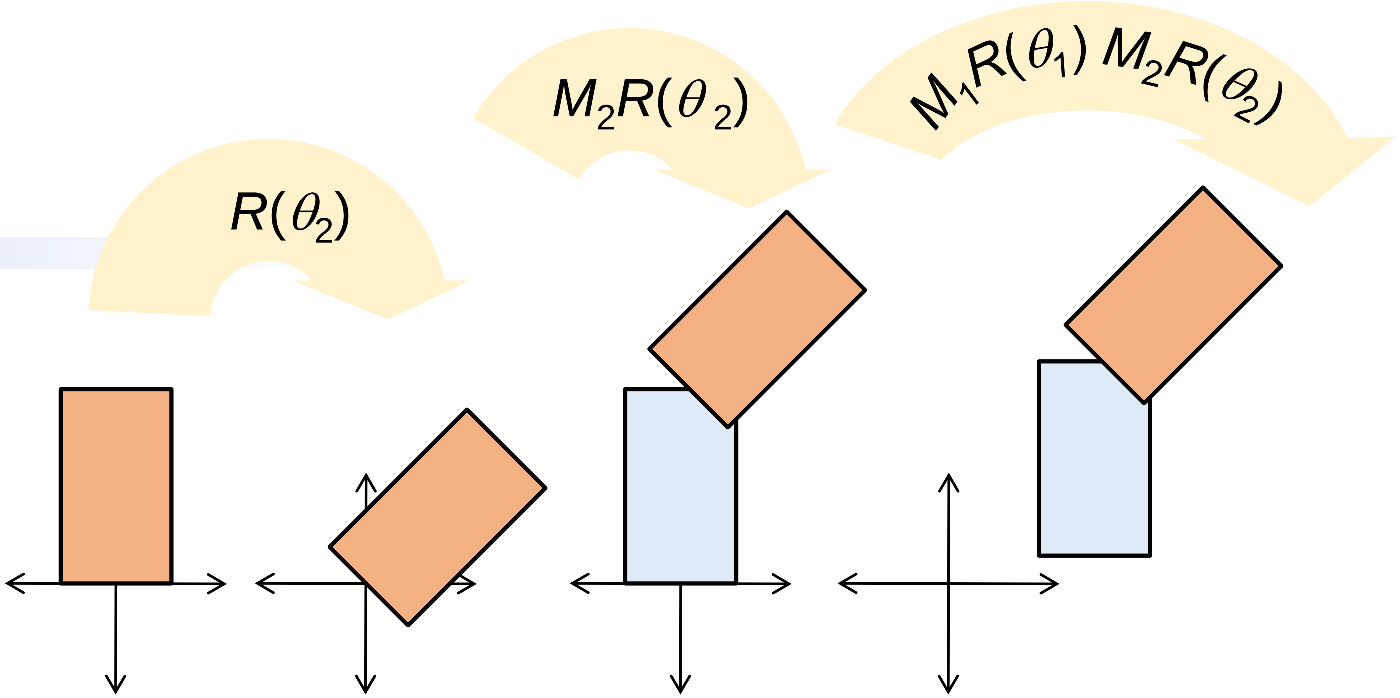
Skinning

- $R(\theta_2)$ rotates forearm cylinder about its elbow at the origin
- M_2 moves forearm elbow from the origin to the end of the upper-arm cylinder when its shoulder is based at the origin



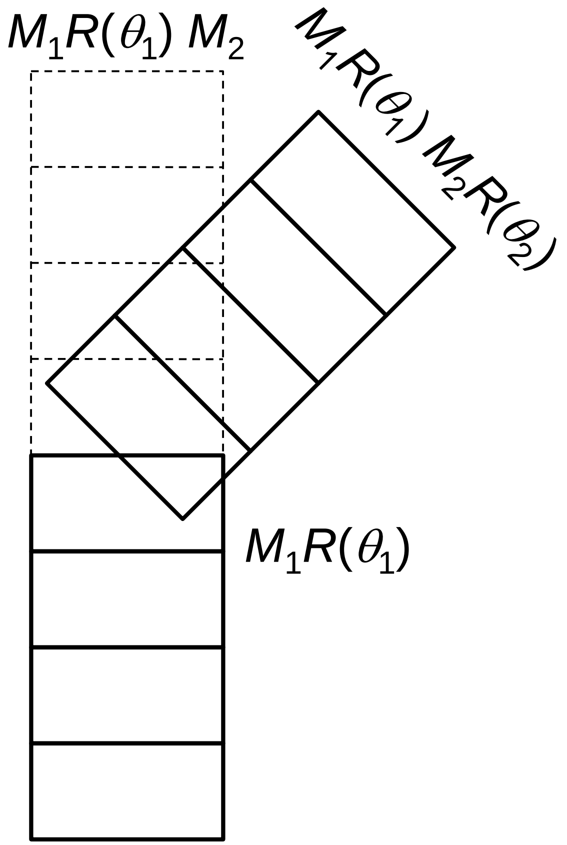
Skinning

- $R(\theta_2)$ rotates forearm cylinder about its elbow at the origin
- M_2 moves forearm elbow from the origin to the end of the upper-arm cylinder when its shoulder is based at the origin
- $R(\theta_1)$ rotates upper-arm cylinder about its shoulder at the origin
- M_1 moves upper-arm cylinder from the origin to its position in world coordinates



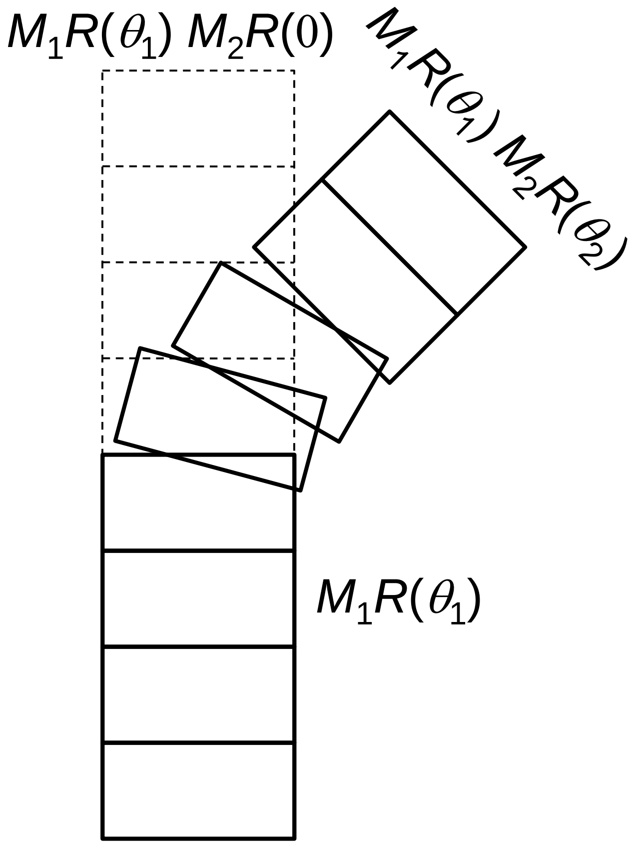
Skinning

- $R(\theta_2)$ rotates forearm cylinder about its elbow at the origin
- M_2 moves forearm elbow from the origin to the end of the upper-arm cylinder when its shoulder is based at the origin
- $R(\theta_1)$ rotates upper-arm cylinder about its shoulder at the origin
- M_1 moves upper-arm cylinder from the origin to its position in world coordinates



Skinning

Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

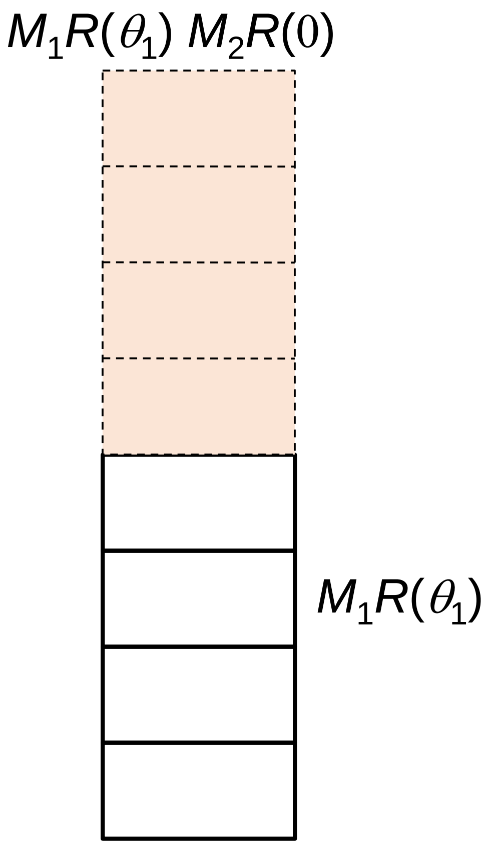


Skinning

Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

- Let

$$M_{\text{straight}} = M_1 R(\theta_1)\, M_2 R(0)$$

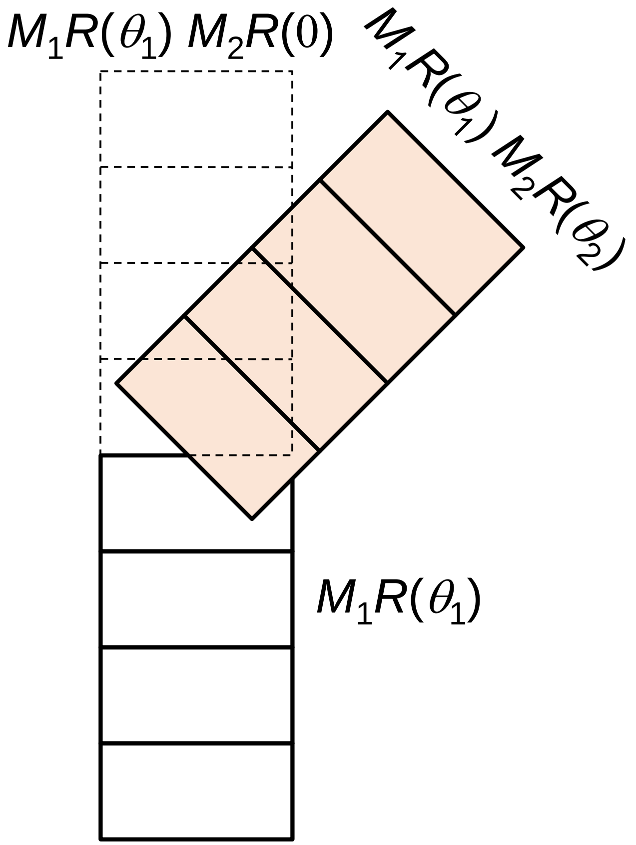


Skinning

Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

- Let

$$M_{\text{straight}} = M_1 R(\theta_1)\, M_2 R(0)$$
$$M_{\text{bent}} = M_1 R(\theta_1)\, M_2 R(\theta_2)$$



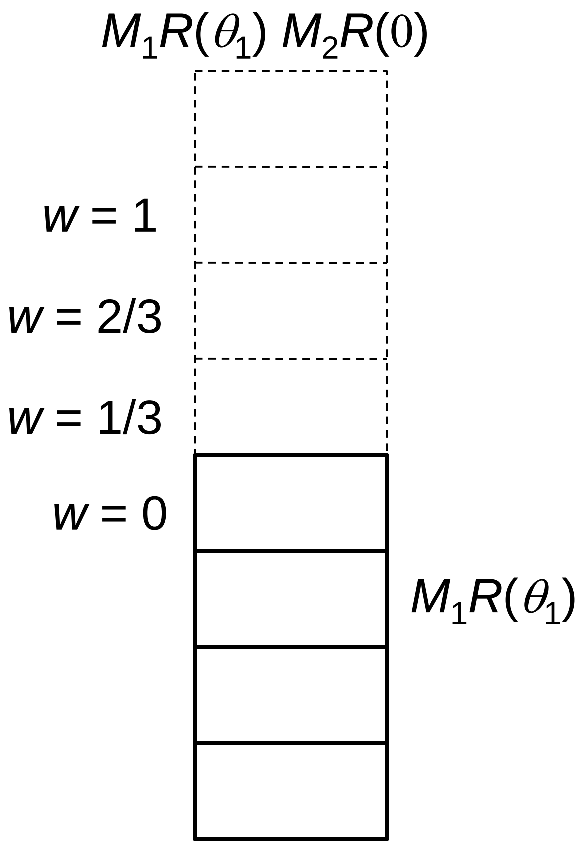
Skinning

Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

- Let

$$M_{\text{straight}} = M_1 R(\theta_1)\, M_2 R(0)$$
$$M_{\text{bent}} = M_1 R(\theta_1)\, M_2 R(\theta_2)$$

- Distribute (“paint”) weights w on vertices of forearm cylinder
  - $w = 0$ at elbow end
  - $w = 1$ after elbow



Skinning

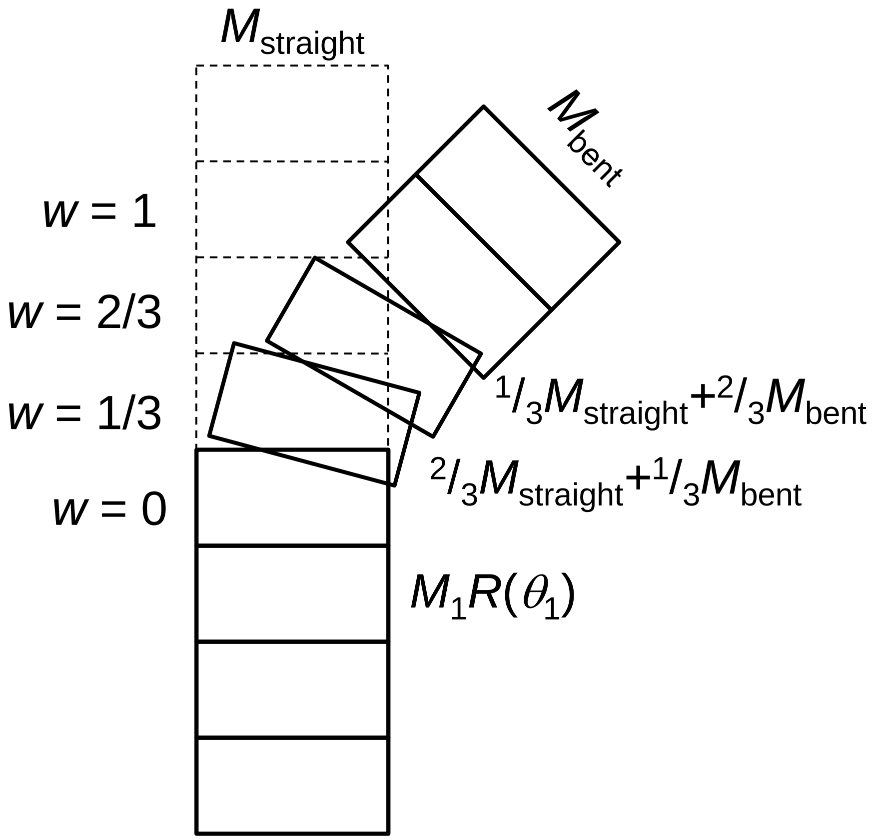
Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

- Let

$$M_{\text{straight}} = M_1 R(\theta_1)\, M_2 R(0)$$
$$M_{\text{bent}} = M_1 R(\theta_1)\, M_2 R(\theta_2)$$

- Distribute ("paint") weights w on elements of forearm cylinder
  - $w = 0$ at elbow end
  - $w = 1$ after elbow
- Transform elements using

$$M(w) = (1 - w)\, M_{\text{straight}} + w\, M_{\text{bent}}$$



Skinning

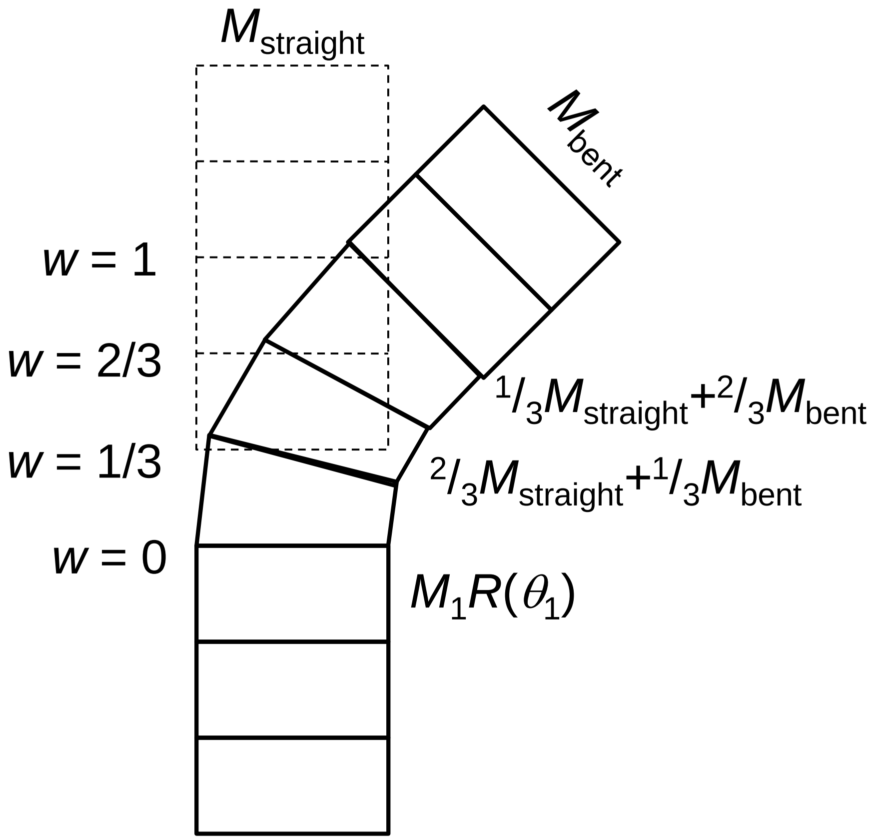
Solution: interpolate matrices from straight coordinate frame into the correctly oriented coordinate frame per-vertex

- Let

$$M_{\text{straight}} = M_1 R(\theta_1)\, M_2 R(0)$$
$$M_{\text{bent}} = M_1 R(\theta_1)\, M_2 R(\theta_2)$$

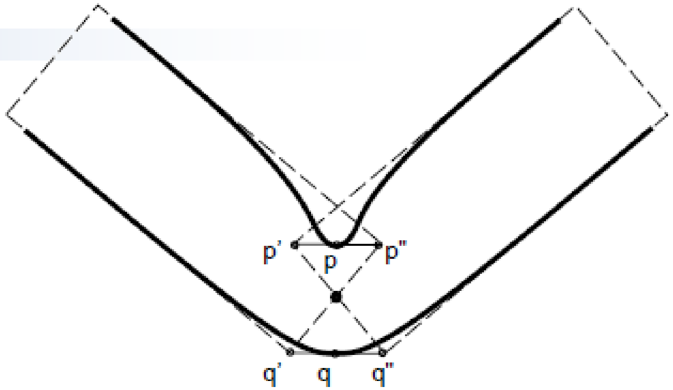
- Distribute ("paint") weights w on **vertices** of forearm cylinder
  - $w = 0$ at elbow end
  - $w = 1$ after elbow
- Transform **vertices** using

$$M(w) = (1 - w)\, M_{\text{straight}} + w\, M_{\text{bent}}$$



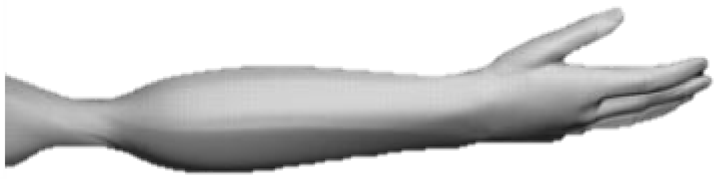
Interpolating Matrices

- Skinning interpolates matrices by interpolating their elements
- Identical to interpolating vertex positions after transformation
- We've already seen problems with interpolating rotation matrices
- Works well enough for rotations with small angles
- Rotations with large angles needs additional processing (e.g. polar decomposition)
- Quaternions provide a better way to interpolate rotations…



$$(aA + bB)p = a(Ap) + b(Bp)$$

a,b = weights
A,B = matrices
p = vertex position



From: J. P. Lewis, Matt Cordner, and Nickson Fong. "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation."
Proc. SIGGRAPH 2000