Computer Animation and Visualisation

Lecture 4.

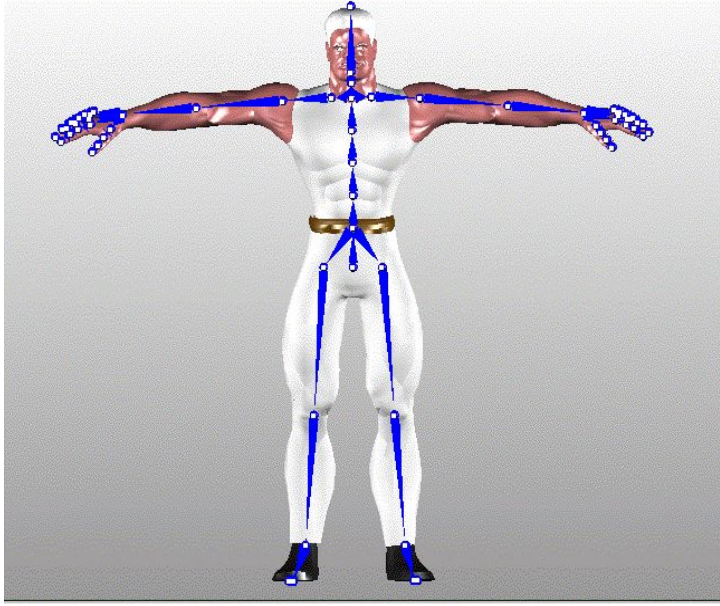
Skinning

Taku Komura

Overview

- Skinning
  - Background knowledge
  - Linear Blending
  - How to decide weights?
- Dual Quaternion Skinning
- Anatomical models

Skinning

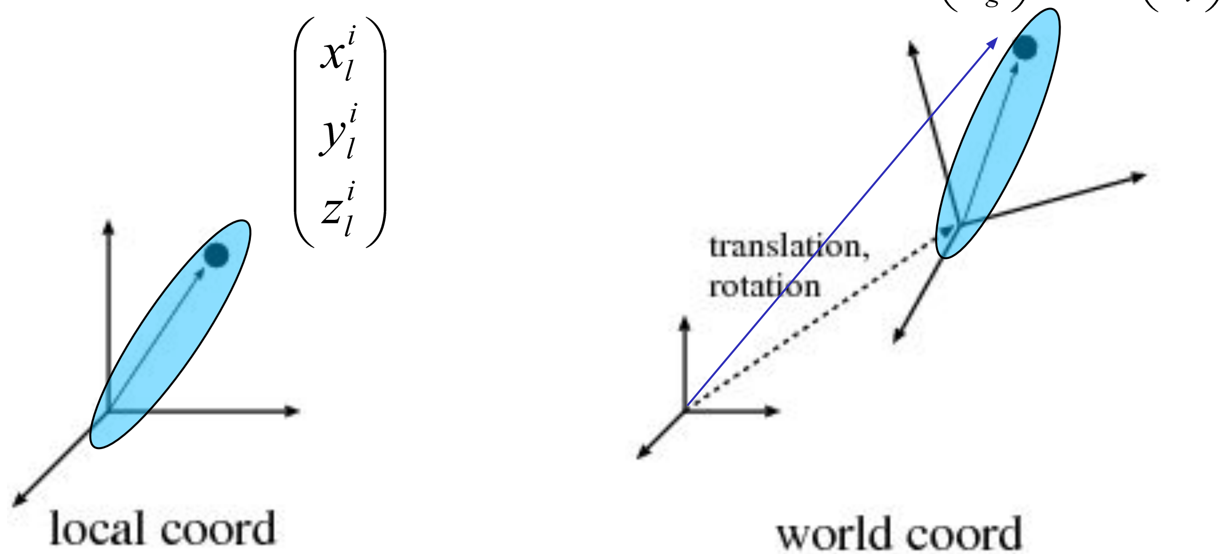


- Assume the movements of the skeleton is determined
- The character's skin must deform according to the motion of the skeleton
- This process is called skinning

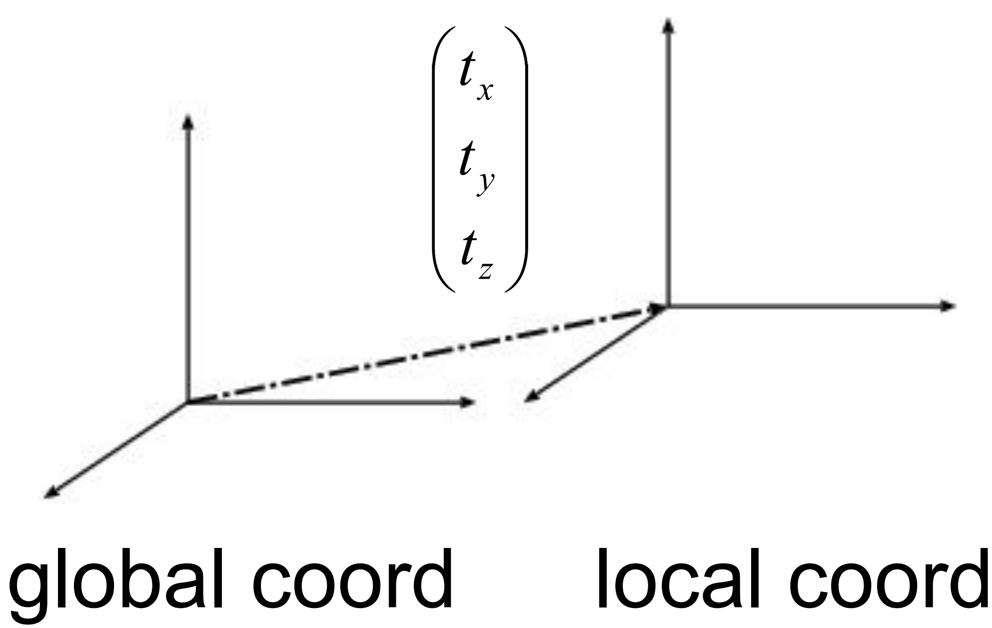
Background

- A homogeneous transformation matrix *M* (4x4) is defined per bone
- It represents the pose of the bone
- It converts local coordinates to world coordinates

$$\begin{pmatrix} x_g^i \\ y_g^i \\ z_g^i \end{pmatrix} = M_i \begin{pmatrix} x_l^i \\ y_l^i \\ z_l^i \end{pmatrix}$$

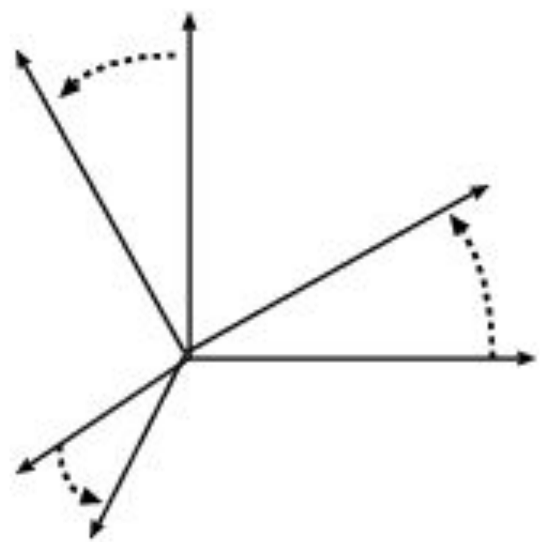


Translation matrix



$$M = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation matrix



Rot x (θ)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

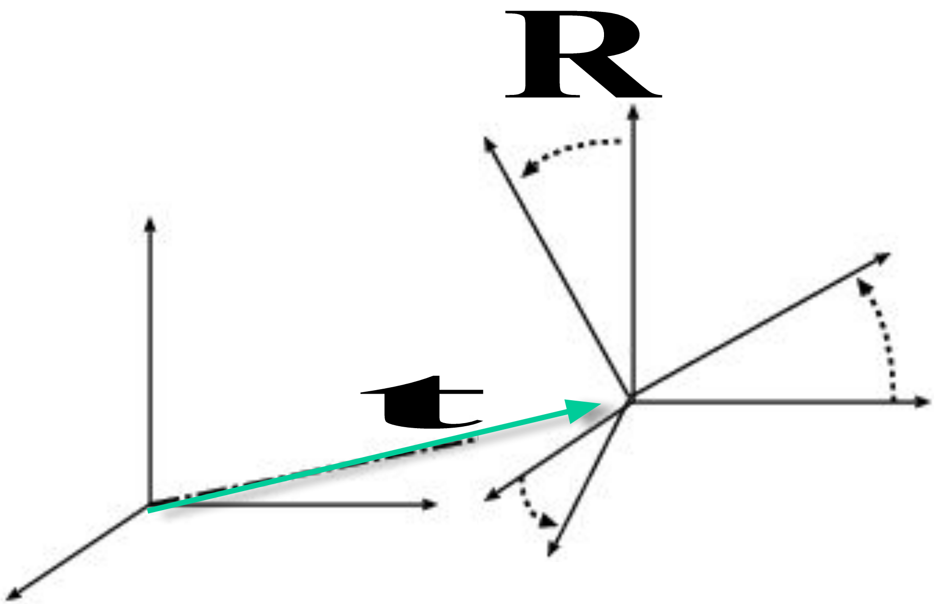
Rot y (θ)

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For Euler angles in the order of x, y, z, the matrix can be computed by

$$Rot_x(\phi) \bullet Rot_y(\varphi) \bullet Rot_z(\xi)$$

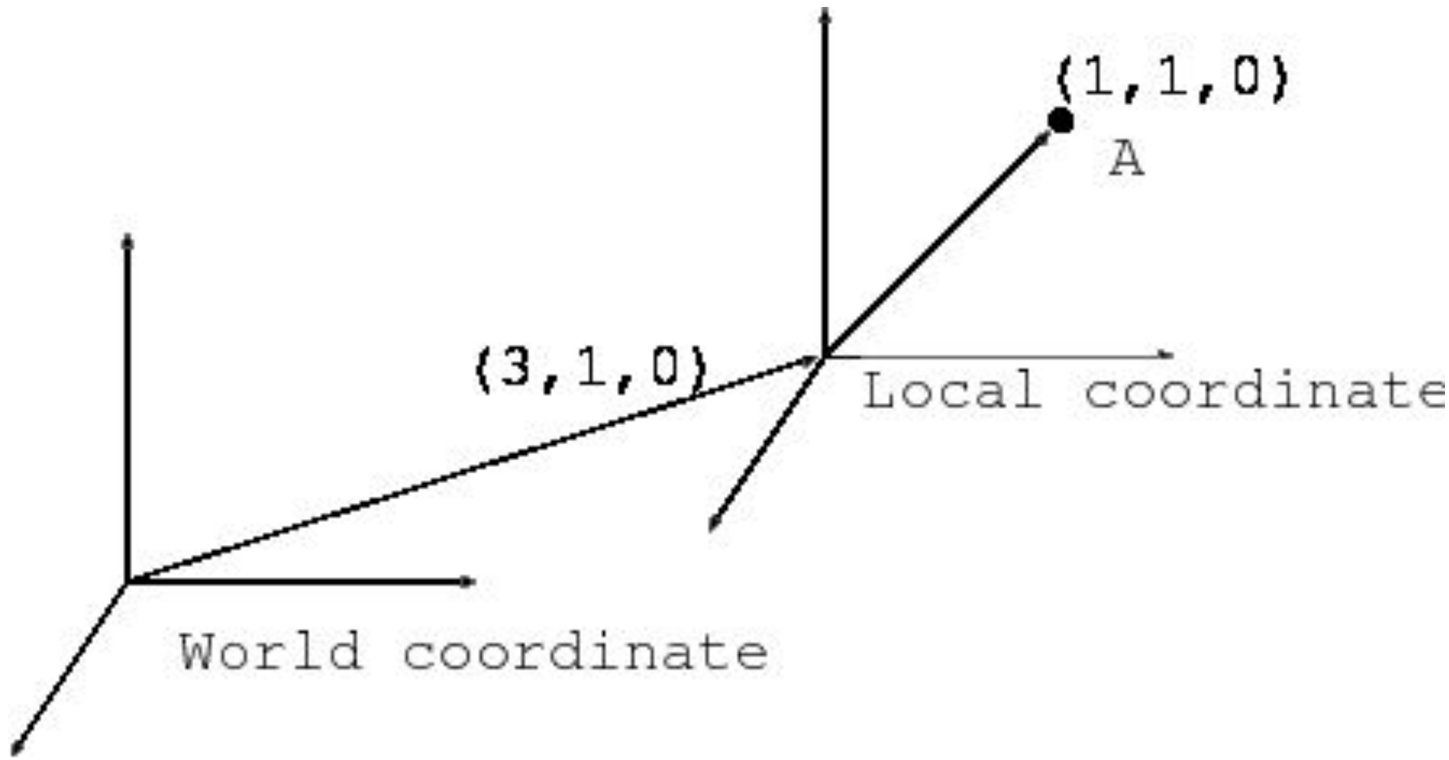
Translation & Rotation matrix



$$M = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

$\mathbf{R}$: rotation part (3x3)

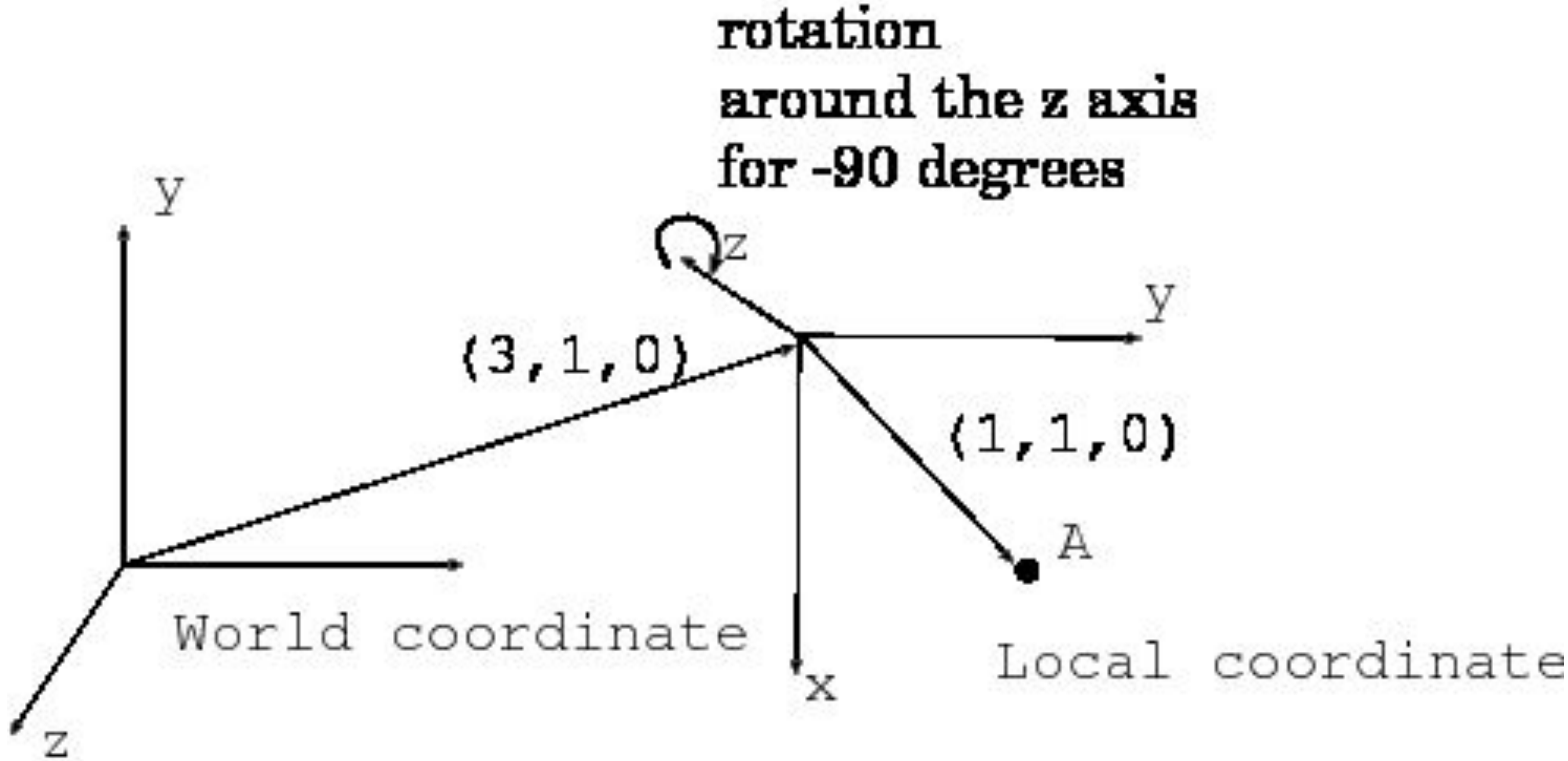
$\mathbf{t}$: translation vector (3x1)



- What is the local-to-world transformation matrix?
- What is the position of point A in the world coordinate system?

$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

- So the location of the point in the world is (4,2,0)



- What is the local-to-world transformation matrix?
- What is the position of point A in the world coordinate system?

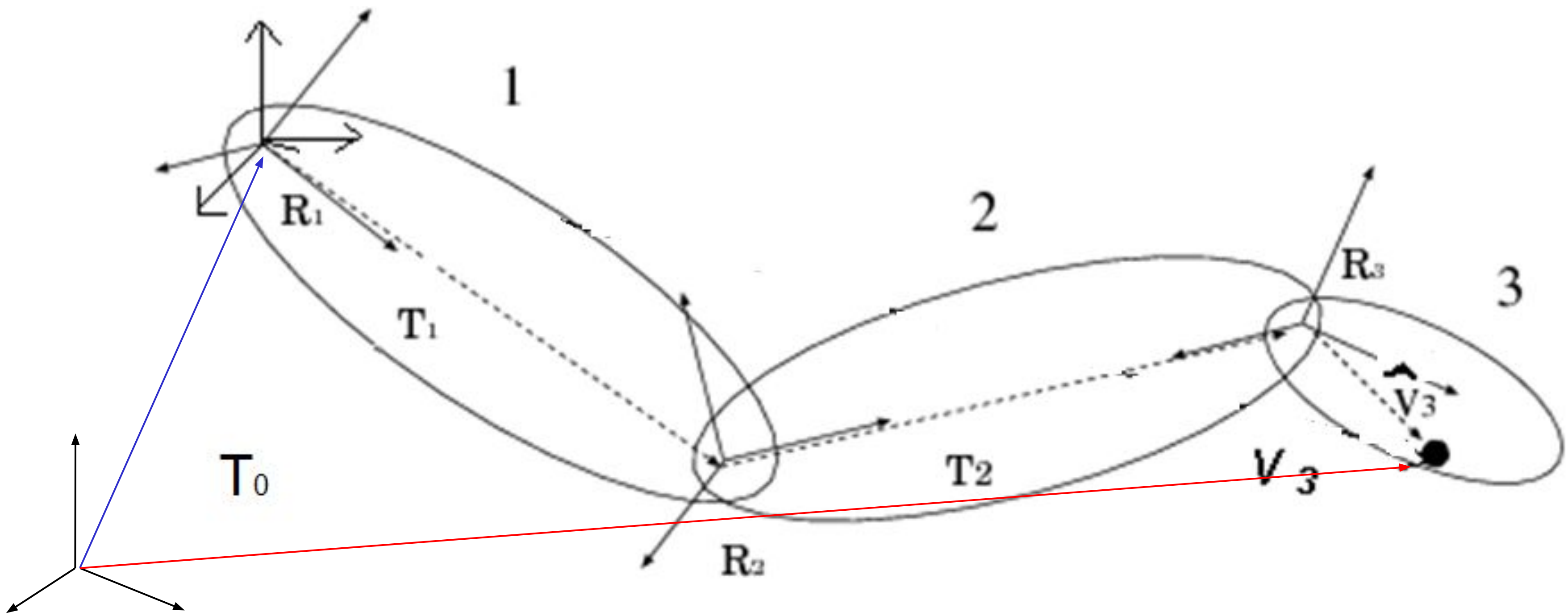
$$\begin{bmatrix} 0 & 1 & 0 & 3 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- So the location of the point in the world is (4,0,0)

Animating Robots

What is the position of the points composing the segments of the robots?

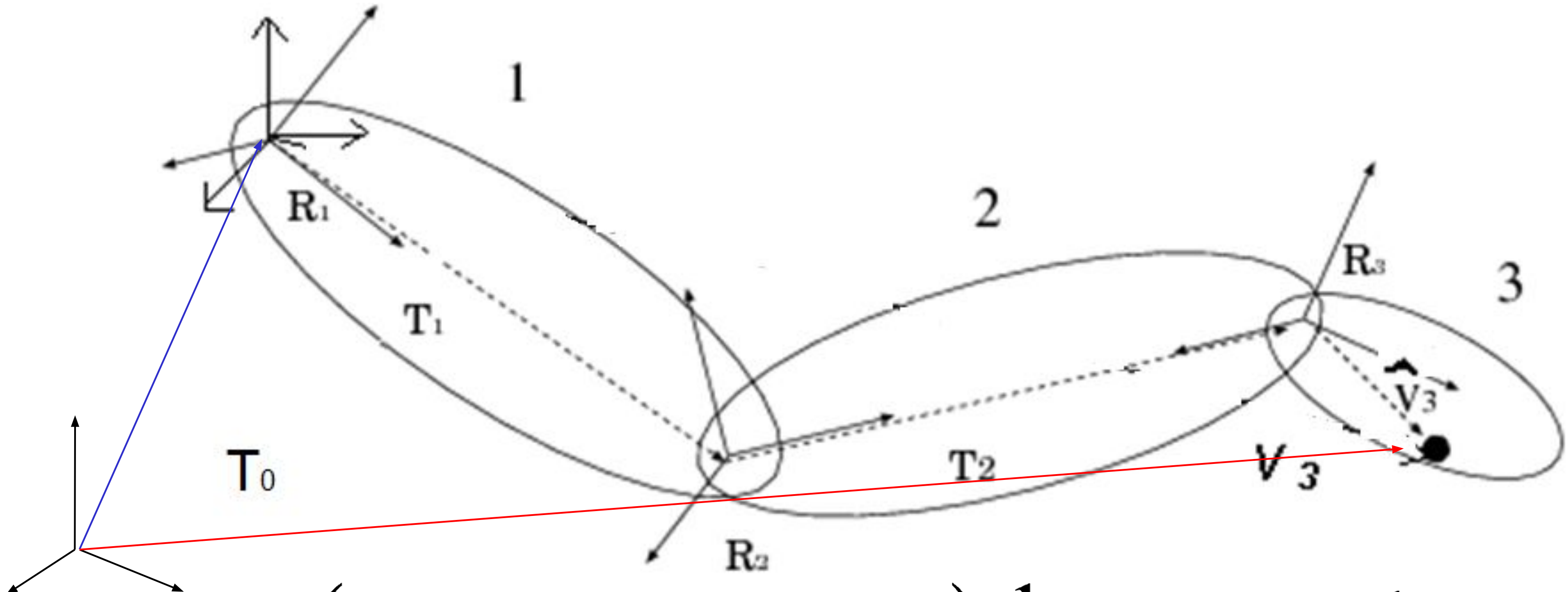
A local-to-global transformation for a hand



$$\mathbf{v}_3 = \mathbf{T}_0\mathbf{R}_1\mathbf{T}_1\mathbf{R}_2\mathbf{T}_2\mathbf{R}_3\hat{\mathbf{v}}_3 = \mathbf{M}_3\hat{\mathbf{v}}_3$$

- A body composed of three segments
- A vertex $\boldsymbol{v_3}$ defined in a local coordinate system of segment 3
- Its global position v is computed by the above equation

A global-to-local transformation for a hand



$$\hat{\mathbf{v}}_3 = \left(\mathbf{T}_0\mathbf{R}_1\mathbf{T}_1\mathbf{R}_2\mathbf{T}_2\mathbf{R}_3\right)^{-1}\mathbf{v}_3 = \mathbf{M}_3^{-1}\mathbf{v}_3$$

- Given the global location, the local coordinate of the point can be computed as above.

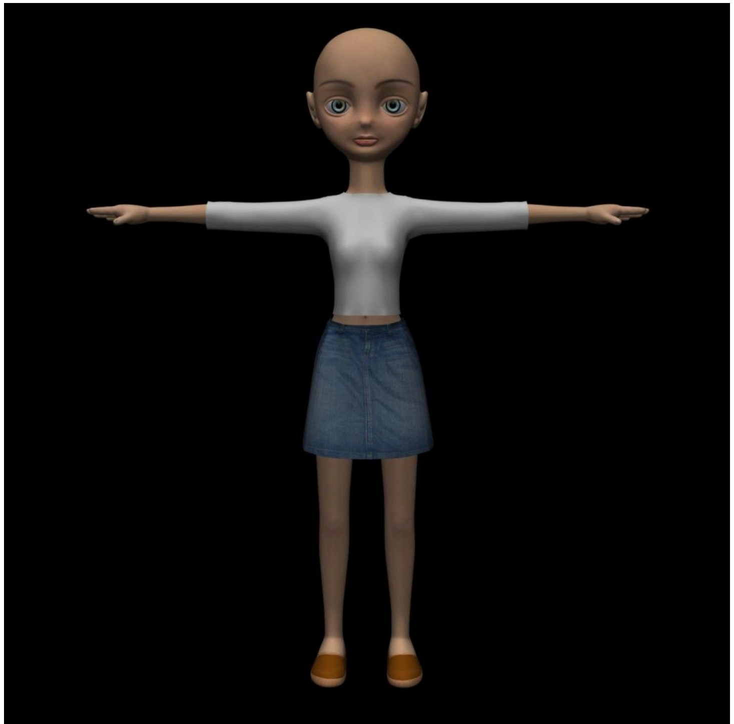
For robots, this is fine

- Compute the local-to-global matrix for all body segments : $\mathbf{M_i}$
- Multiply the local coordinates $\mathbf{v_i}$ of every body segment to this matrix to calculate its global position :

$$\mathbf{v_i} = \mathbf{M_i}\hat{\mathbf{v}}_\mathbf{i}$$

- Then, the global position of all the points can be computed

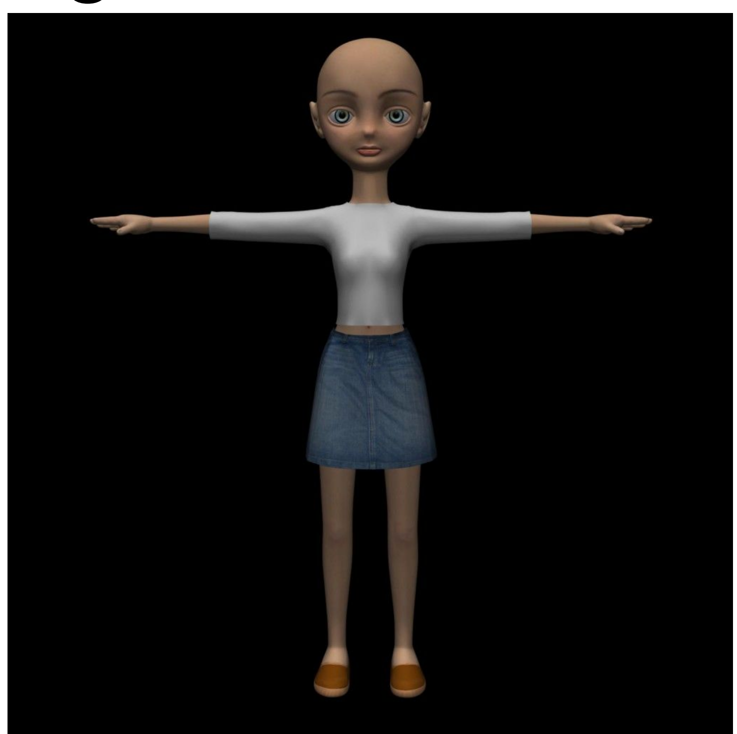
What about polygon characters?



What about polygon characters?

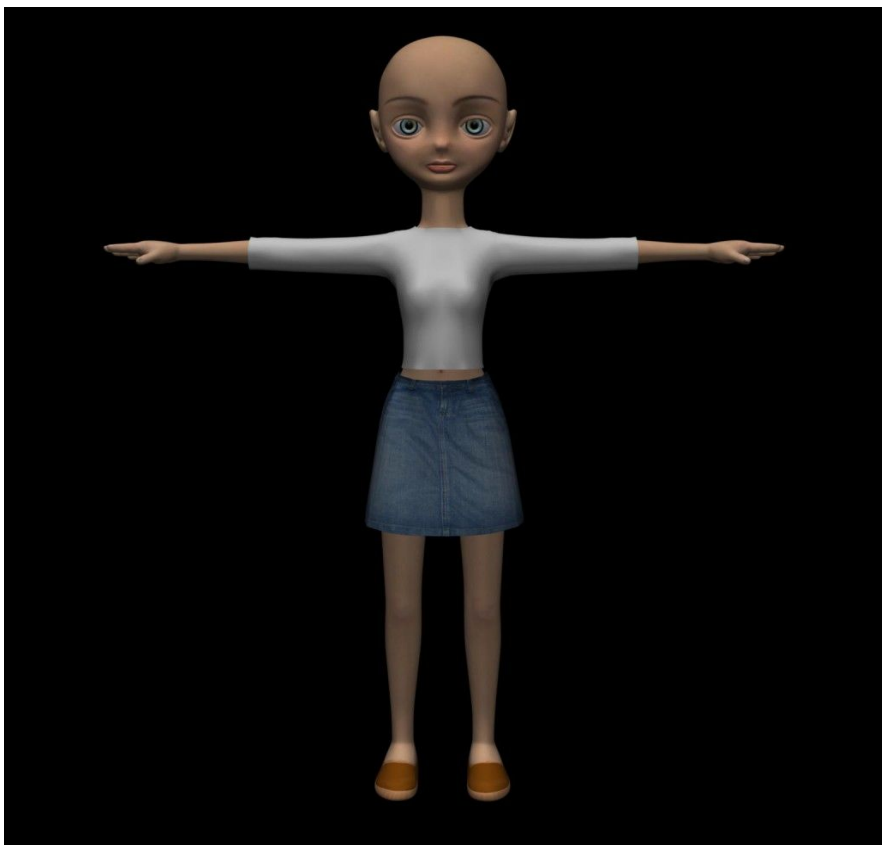
Problem 1. We are only given the polygon – no skeleton structure

Problem 2. Some points on the body do not belong to a single bone but to multiple bones close to it

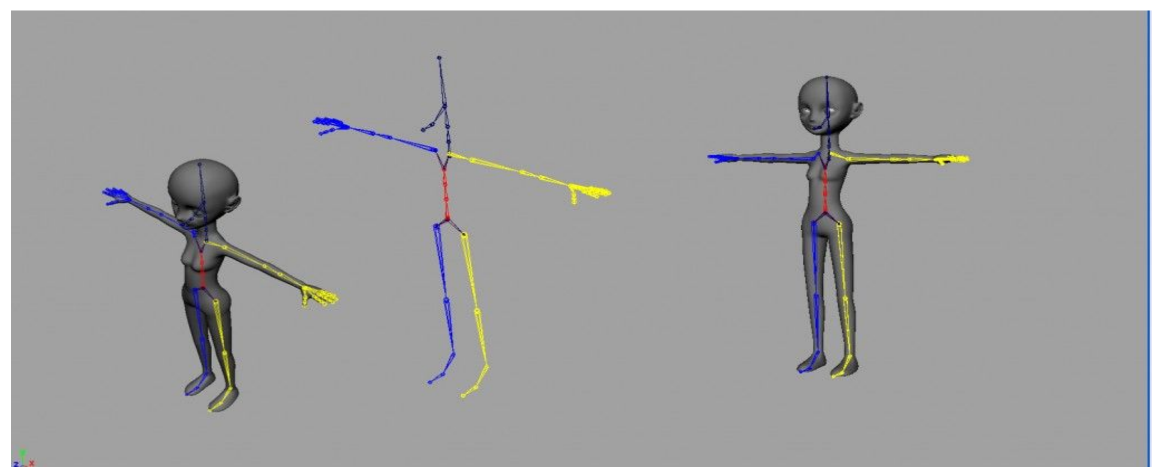


Solution: Skinning

Fitting the skeleton into the polygon model (done manually)

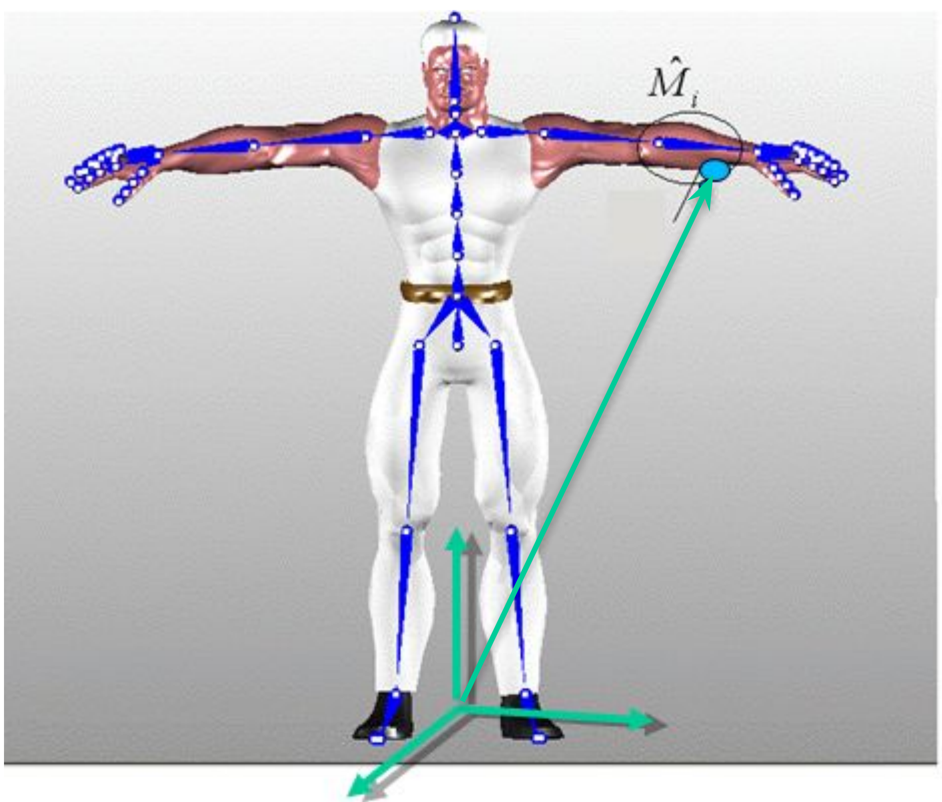


https://www.youtube.com/watch?v=dniWVu55PEc&feature=related



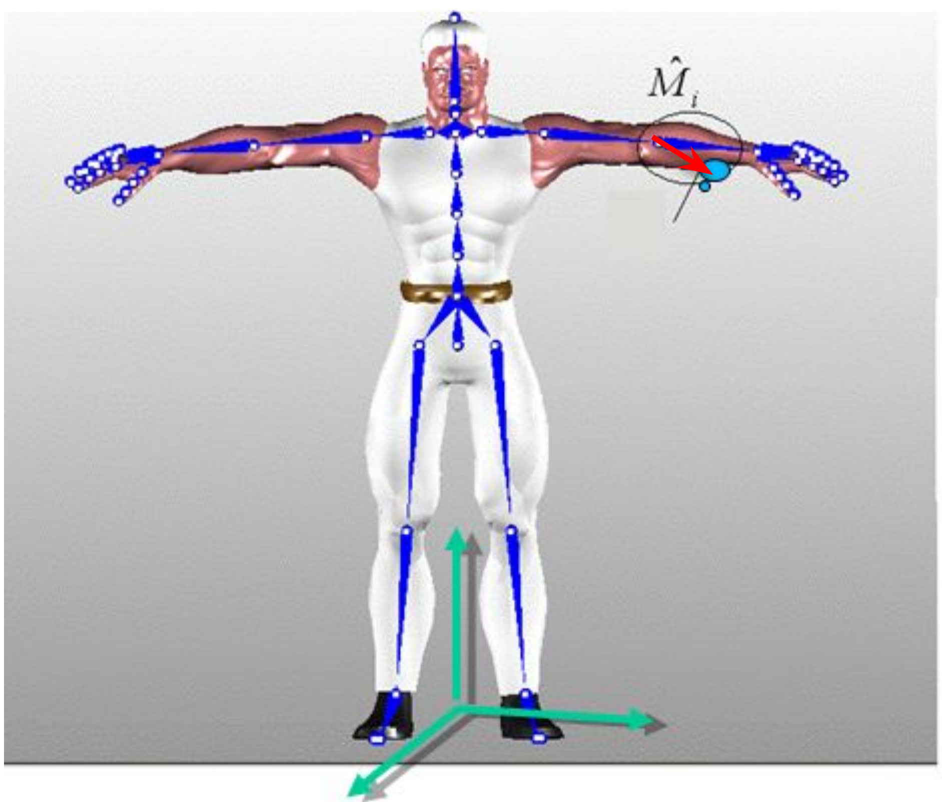
Rest Pose to Bone Coordinate

- the global position of a particular vertex, ***v***, in the rest pose is defined $\mathbf{v_g}$



Rest Pose to Bone Coordinate

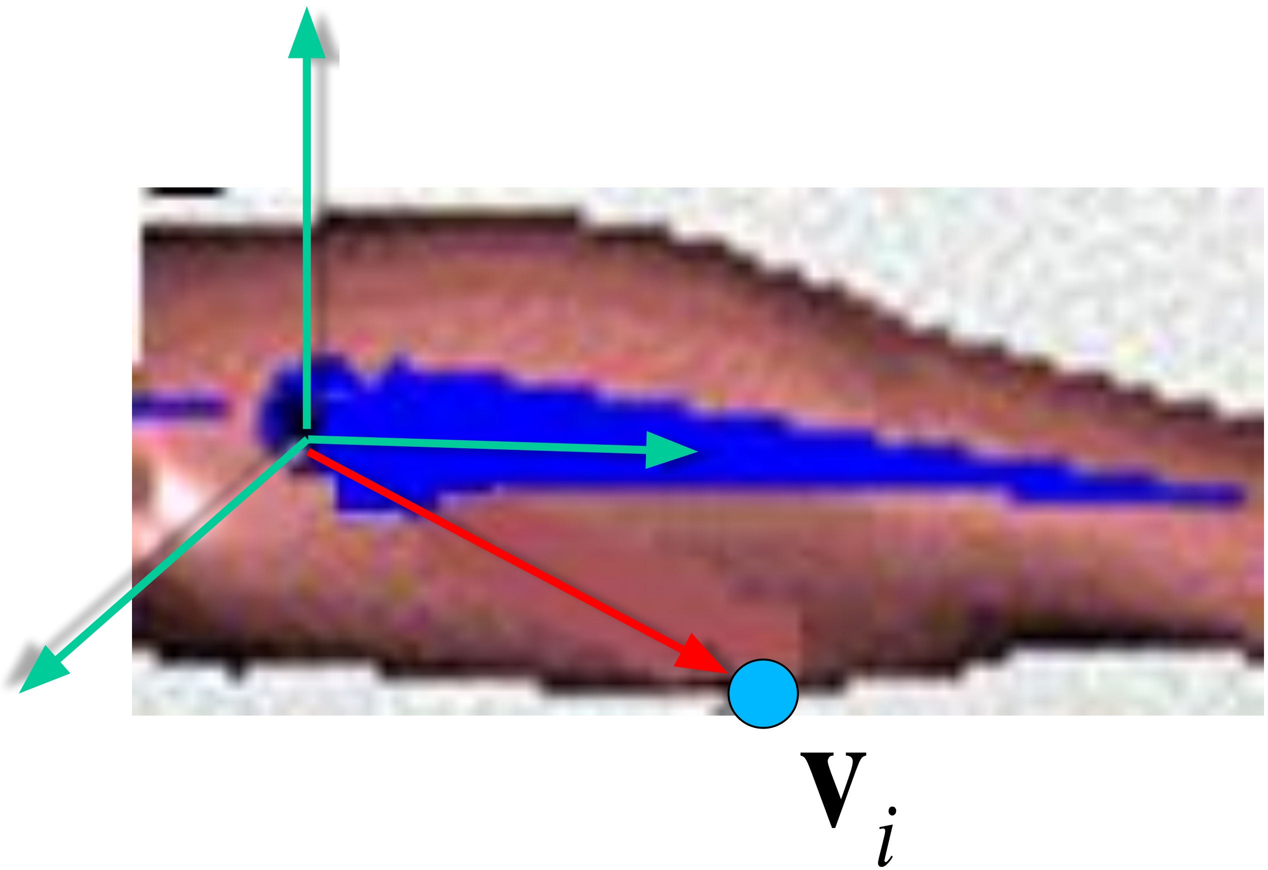
- We want to know where this point $\mathbf{v}_g$ is in the local coordinates



Rest Pose to Bone Coordinate

- for each bone, i, the position of the vertex in the rest pose is first transformed from model coordinates (v_g) to bone coordinates (v_i) by applying the inverse of the rest pose bone transformation:

$$\mathbf{v}_\mathbf{i} = \mathbf{M}_\mathbf{i}^{-1}\mathbf{v}_\mathbf{g}$$

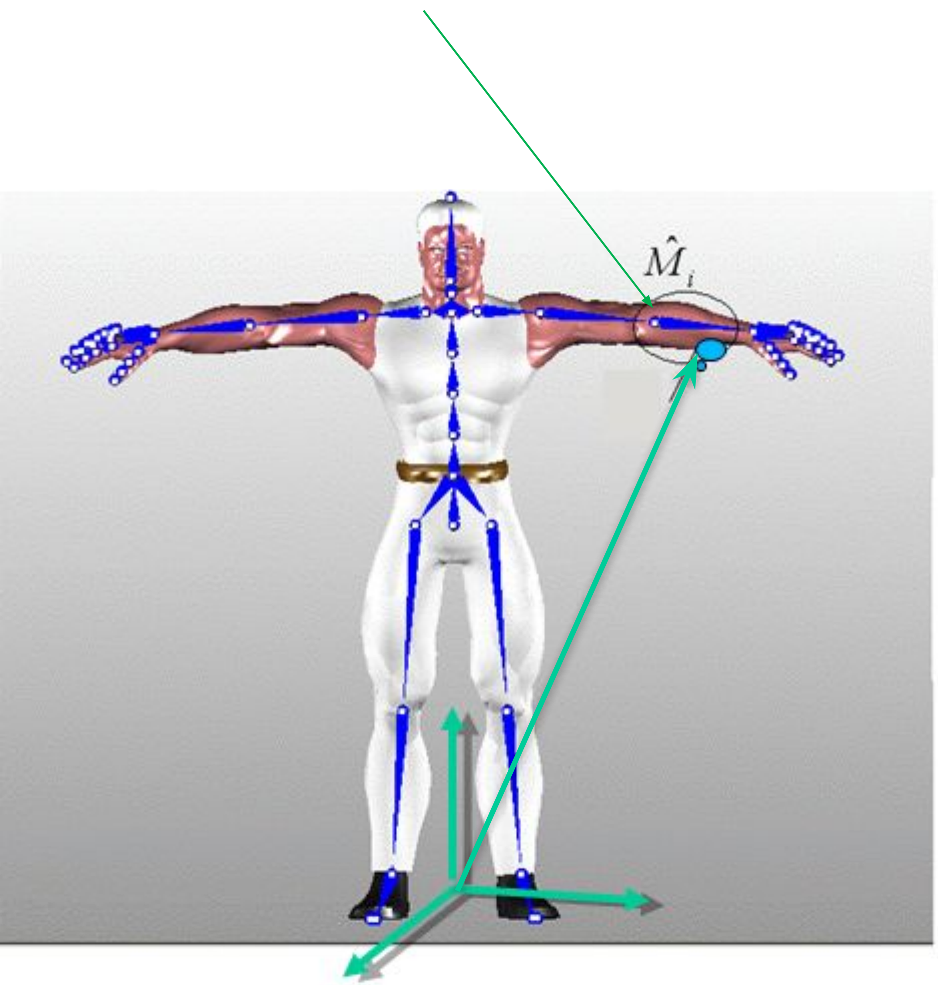


Rest Pose to Bone Coordinate: Example

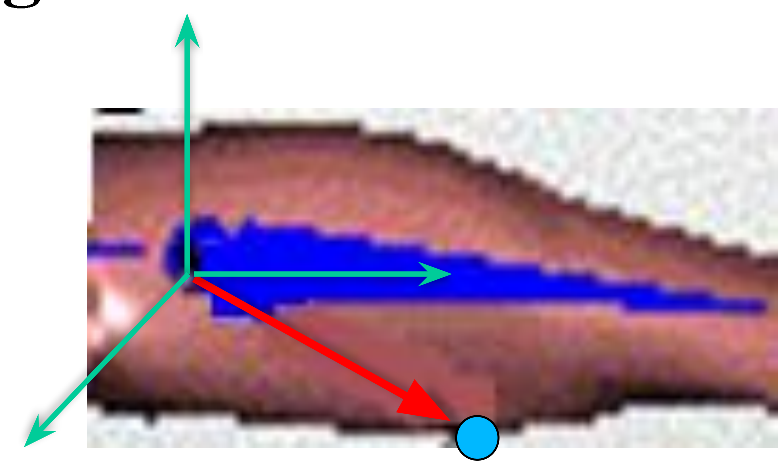
center of elbow joint

(0.7,1.5,0)

$$\begin{bmatrix} 1 & 0 & 0 & 0.7 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$\mathbf{v_g} = (0.8, 1.45, 0)$$



$\mathbf{v_i}$

Rest Pose to Bone Coordinate

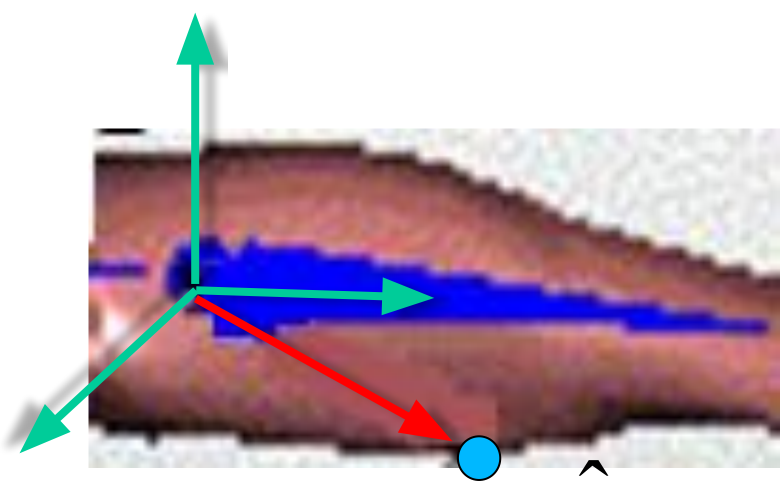
$$\mathbf{v}_i = \mathbf{M}_i^{-1}\mathbf{v}_g$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0.7 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.8 \\ 1.45 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & -0.7 \\ 0 & 1 & 0 & -1.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.8 \\ 1.45 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ -0.05 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{M}_i = \begin{bmatrix} 1 & 0 & 0 & 0.7 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

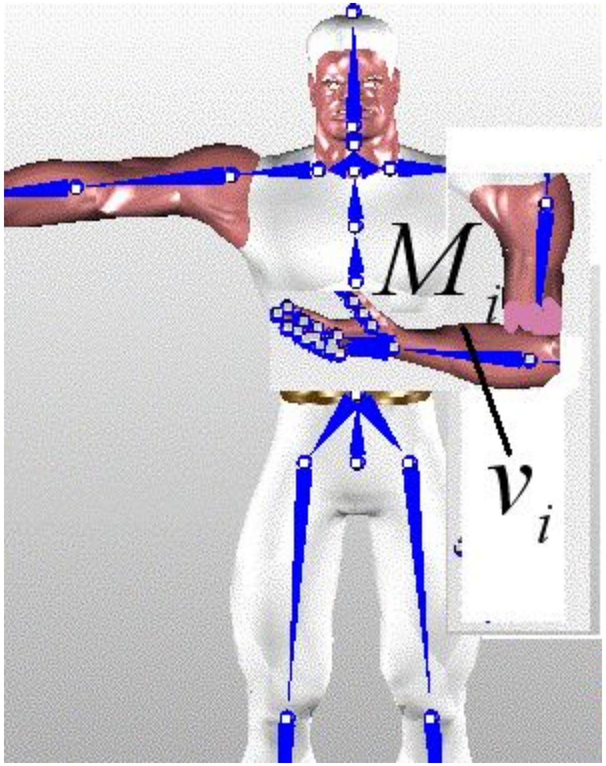
$$v_g = (0.8, 1.45, 0)$$



$\hat{\mathbf{v}}_i$

Bone Coordinate to World Coordinate

- The vertex in bone coordinates, $\mathbf{v_i}$ is then transformed back into world coordinates by applying the transformation of the bone in the new pose

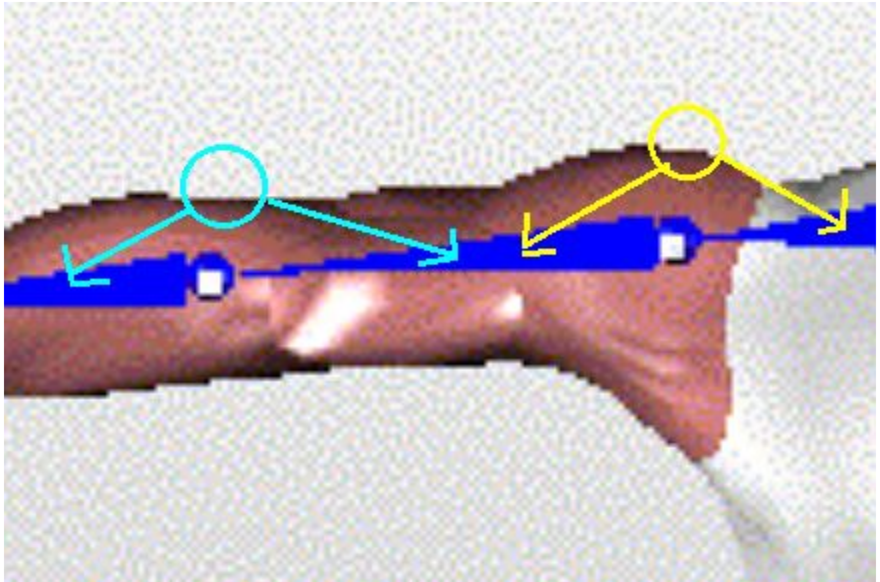


$$\mathbf{v}'_g = \mathbf{M_i v_g} = \mathbf{M'_i M_i^{-1} v_g}$$

$\mathbf{M_i}$: the local-to-global matrix in the new posture

Problem 2:

- For some points, we don’t know to which body segment it belongs to
- For the points near the elbow joint, it might belong to the upper arm, or the forearm or maybe both
- We want both segments to affect its movements

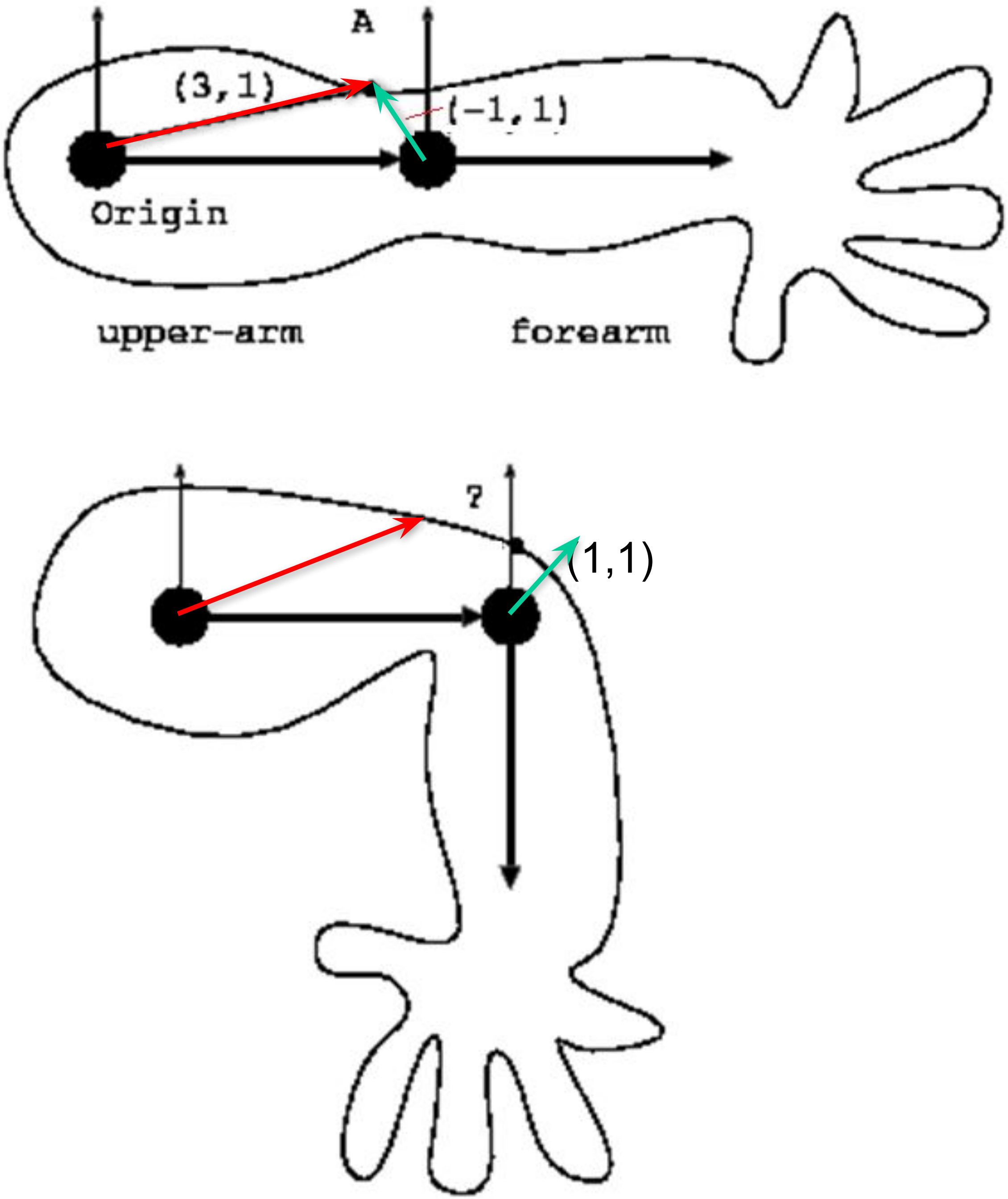


Solution: Linear Blend Skinning

- Linear Blend Skinning: linearly blend the results of the vertex transformed rigidly with each bone.
- A scalar weight, w_i, is given to each influencing bone and the weighted sum gives the vertex's position, v, in the new pose, as follows:

$$\mathbf{v} = \sum_{i=1}^{n} w_i \mathbf{M'_i} \mathbf{M_i^{-1}} \mathbf{v_g} = \left(\sum_{i=1}^{n} w_i \mathbf{T_i}\right) \mathbf{v_g} \qquad \sum_{i=1}^{n} w_i = 1$$

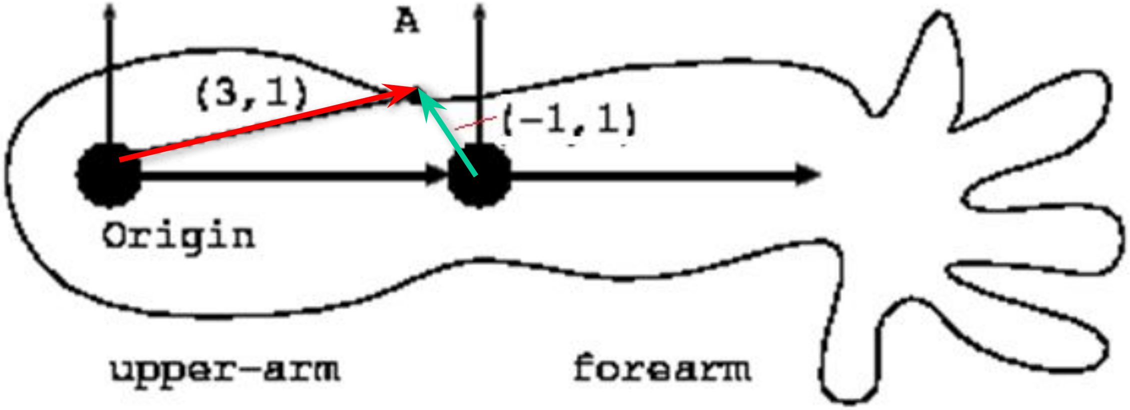
n is the number of bones influencing the position of v



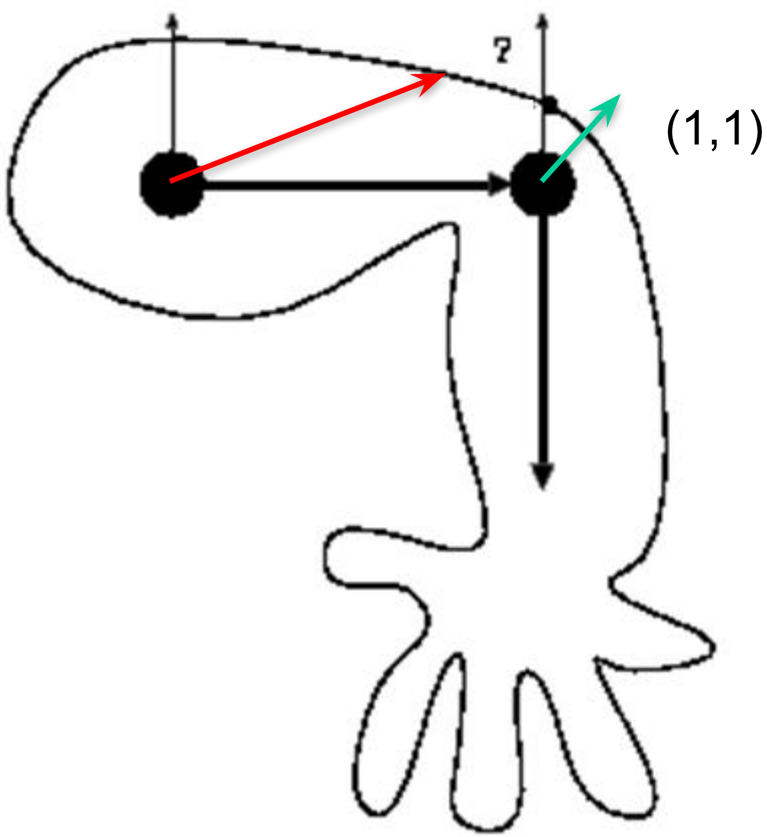
What is the position of point A after the elbow is bent 90 degrees?

- assuming it is a point of the upper arm
- assuming it is a point of the forearm

Assuming the weight is 0.8 for the upper-arm and 0.2 for the forearm



- Assuming it is a point of the upper arm, the position is (3,1)
- assuming it is a point of the forearm, it is (5,1)



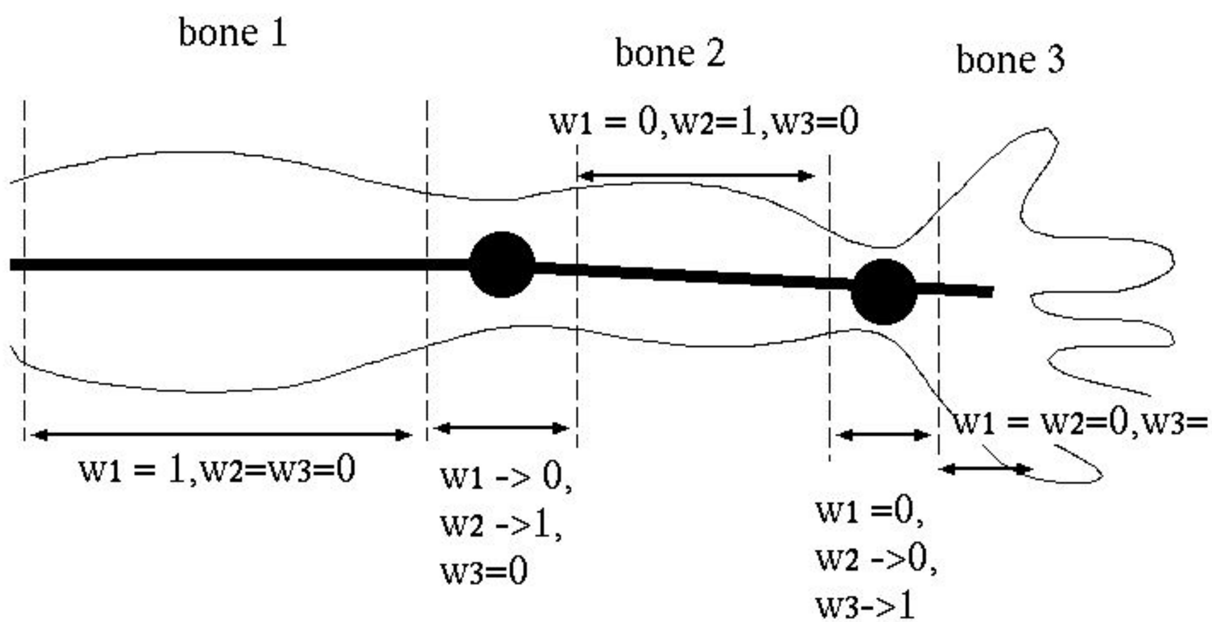
- Assuming the weight is 0.8 for the upper-arm and 0.2 for the forearm, the position is

$$0.8*(3,1)+0.2*(5,1) =(3.4, 1)$$

How to decide the weights?

Decide the mapping of the vertex to the bone

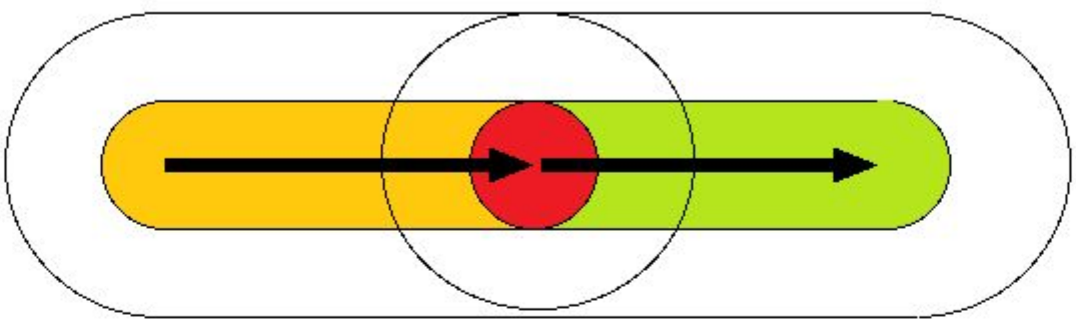
- If vertex v is in the middle of bone i, then $w_i = 1$ and for the rest $w_{j \neq i} = 0$
- If the vertex is near the border of bone i and i+1, w_i will gradually decrease to 0 and w_{i+1} will gradually increase to 1
- If the vertex is affected by more than three bones, the weight can be determined according to its distance to each bone



How to decide the weights?

Example: use the Euclidean distance

- Surround the bones by the inner and outer capsules
- If the vertex is inside only one inner capsule, the weight for the corresponding bone is 1, and the rest are 0
- If inside multiple inner capsules, compute the distance to each bone, and use that to decide the weights (longer distance, lower weight)

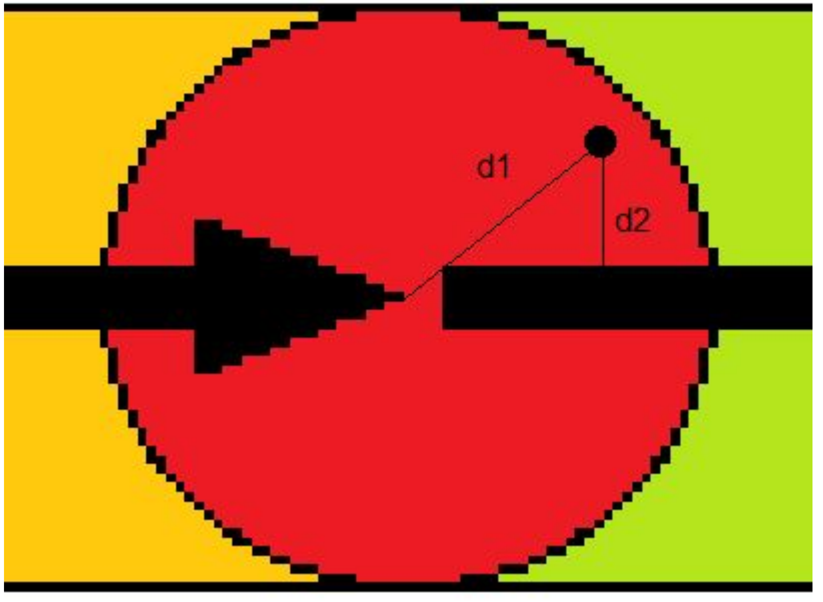


How to decide the weights?

Example: use the Euclidean distance

- Surround the bones by the inner and outer capsules
- If the vertex is inside only one inner capsule, the weight for the corresponding bone is 1, and the rest are 0
- If inside multiple inner capsules, compute the distance to each bone, and set the weights inverse proportional to the distance with normalization

$$w_1 = \sqrt{\frac{\frac{1}{d_1}}{\frac{1}{d_1} + \frac{1}{d_2}}}, w_2 = \sqrt{\frac{\frac{1}{d_2}}{\frac{1}{d_1} + \frac{1}{d_2}}}$$



How to decide the weights?

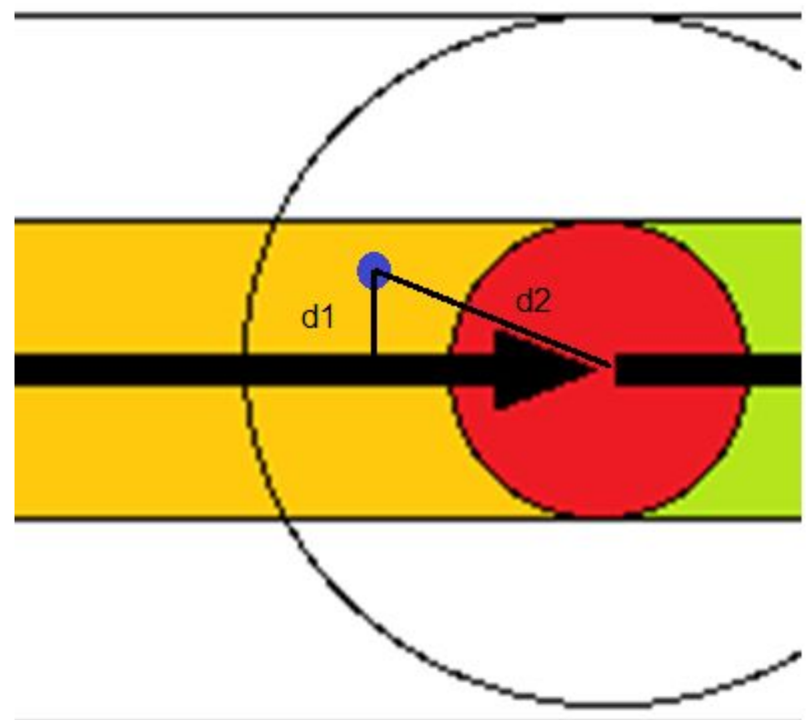
Example: use the Euclidean distance

- If inside the inner capsule of one and in outer capsule of the other, use the distance again but with a fall-off with the other

i.e.

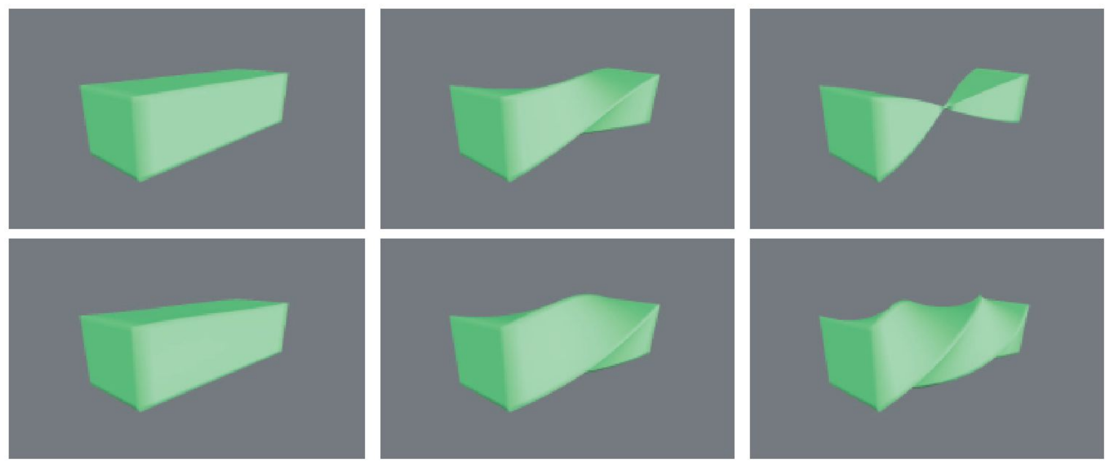
$$w_1 = \sqrt{\frac{\frac{1}{d_1}}{\frac{1}{d_1} + f(d_2)\frac{1}{d_2}}}, w_2 = \sqrt{\frac{f(d_2)\frac{1}{d_2}}{\frac{1}{d_1} + f(d_2)\frac{1}{d_2}}}$$

$$\begin{aligned}
&\mathrm{f(x)} \ : \text{fall-off function} \\
&f(x) = 1 \text{ if } \mathrm{x} < r_1 \\
&f(x) = 0 \text{ if } \mathrm{x} > r_2 \\
&f(x) = 1 - \frac{x - r_1}{r_2 - r_1}
\end{aligned}$$

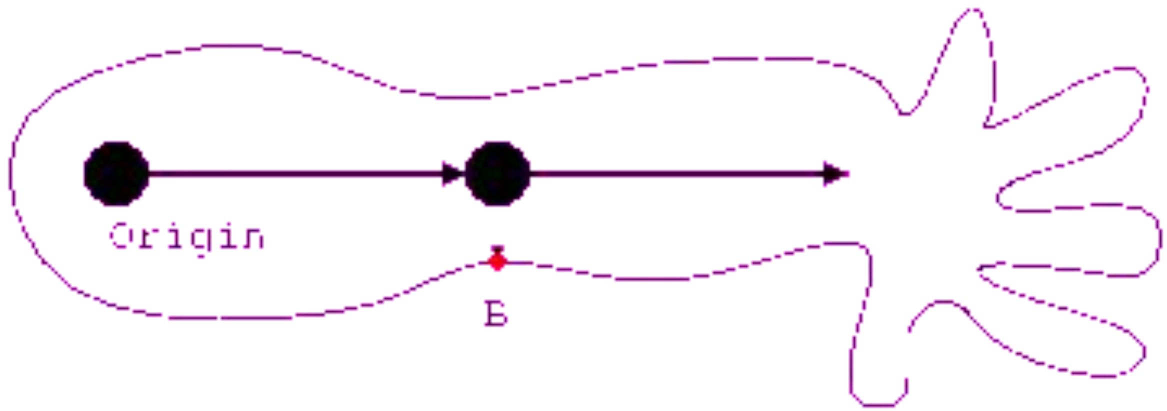


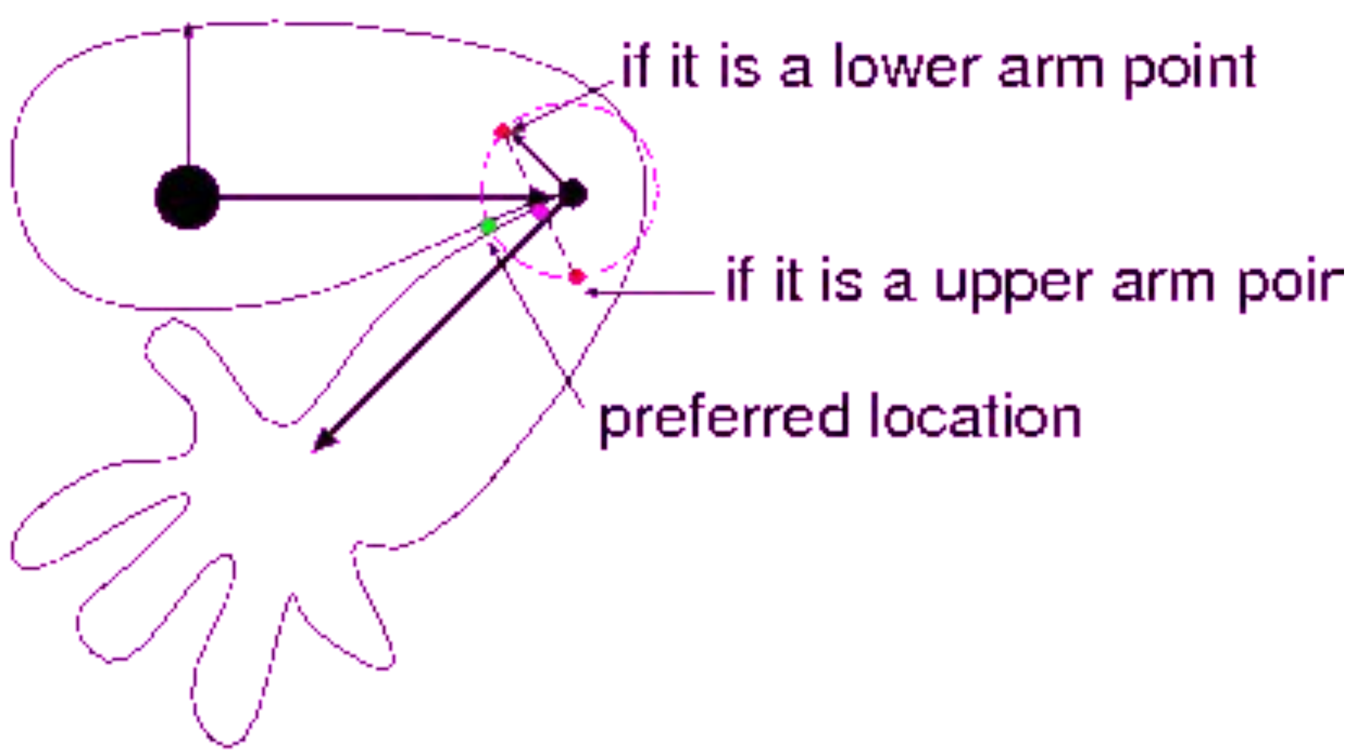
Problems with Linear Blending

- The meshes exhibit volume loss as joints are rotated to extreme angles.
- These are called “joint collapse” and “candy wrapper” effect
- Simple linear blending in the Cartesian space causes the artefacts

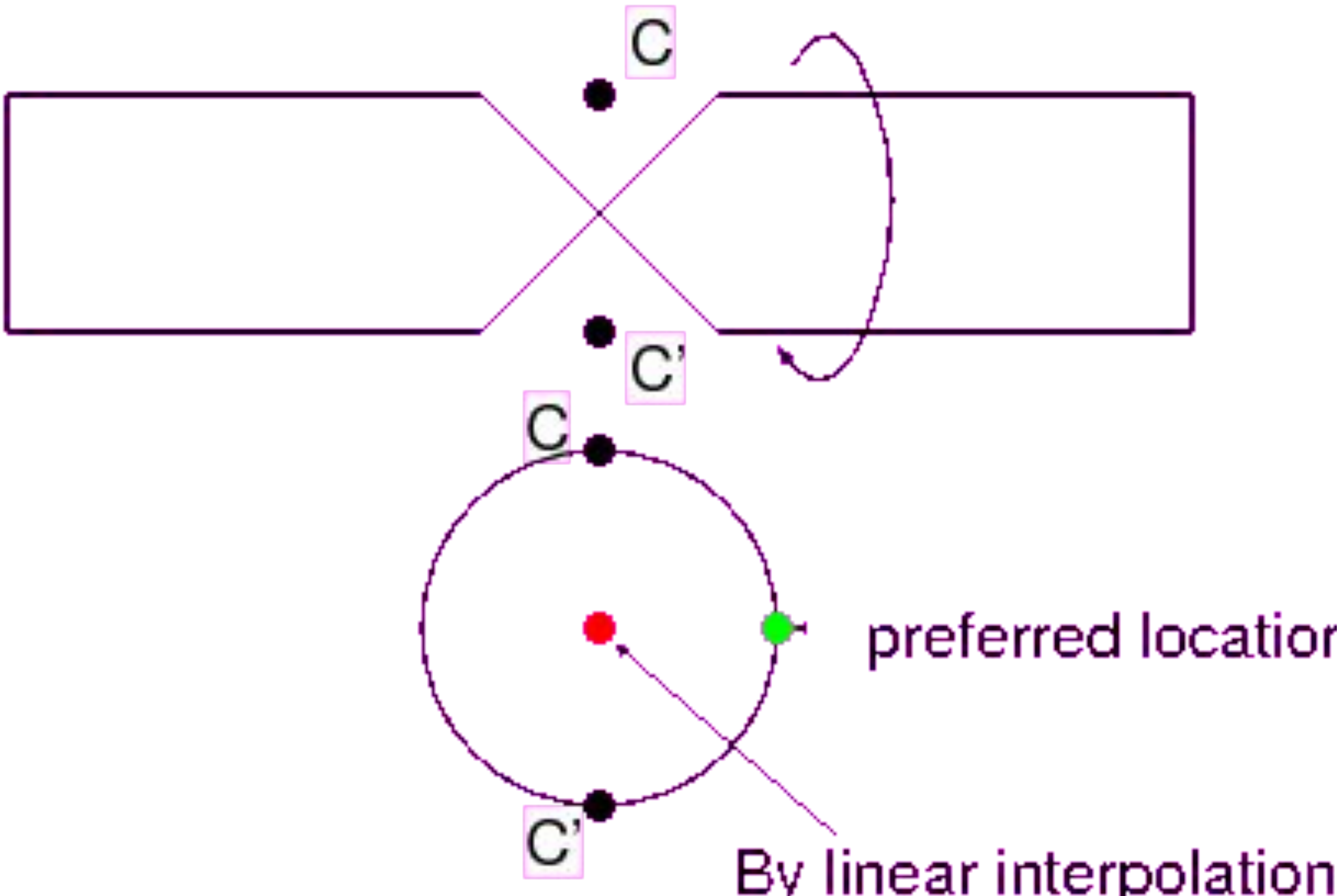


Why does it happen?

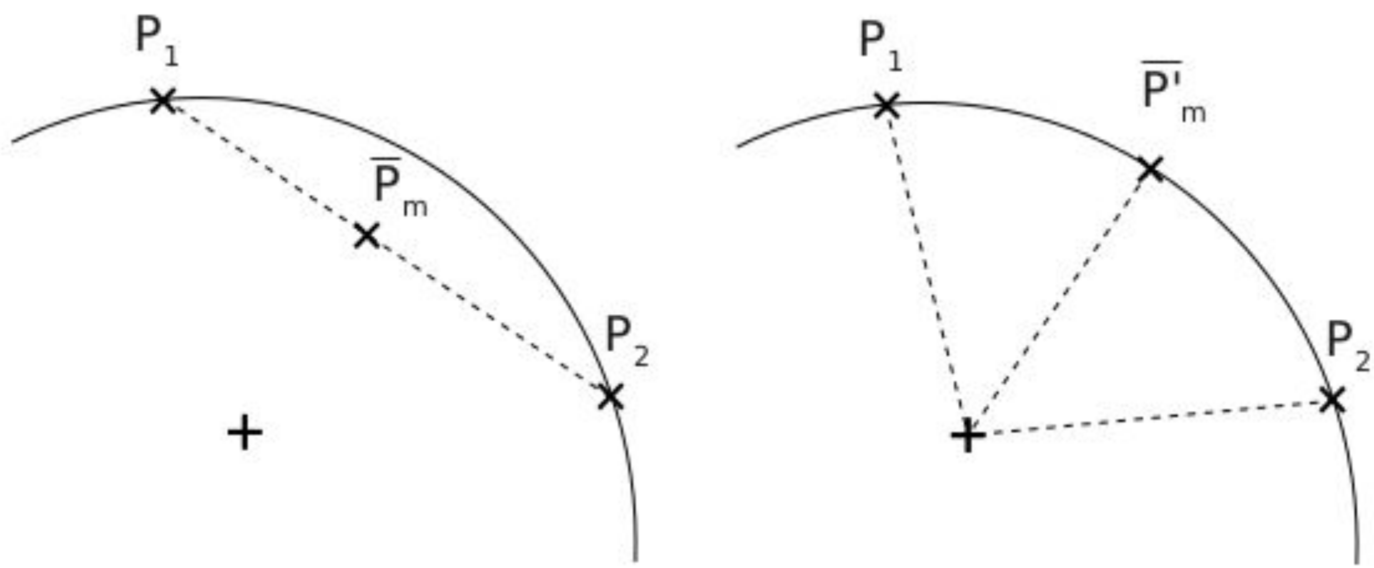




Why does it happen?



Dual Quaternion Skinning



Dual Quaternion Skinning

- Instead of using matrices to express the motions of the joints here we use
  Dual Quaternions.
- Dual quaternions is composed of two quaternions, one responsible for orientation, and the other responsible for translation

$$\mathbf{q} = \mathbf{q_0} + \varepsilon \cdot \mathbf{q}_e$$

- Linear blend skinning blends the matrices by the blending weights

$$\mathbf{T} = \sum_{i=1}^{n} w_i \mathbf{T_i}$$

- Dual quaternion skinning blends the dual quaternion of each bone by the blending weights

$$\mathbf{q} = \frac{\sum_{i=1}^{n} w_i \mathbf{q}_i}{\left\| \sum_{i=1}^{n} w_i \mathbf{q}_i \right\|}$$

- A dual quaternion representing a rotation :

$$\mathbf{q} = (\cos(\frac{\theta}{2}), \mathbf{n_x} \sin(\frac{\theta}{2}), \mathbf{n_y} \sin(\frac{\theta}{2}), \mathbf{n_z} \sin(\frac{\theta}{2})) + \varepsilon \bullet \mathbf{0}$$

- A dual quaternion representing a translation :

$$\mathbf{t} = (1,0,0,0) + \frac{\varepsilon}{2}(0, t_0, t_1, t_2)$$

- A dual quaternion representing a rotation $\mathbf{q_0}$ and a translation $\mathbf{t = (0, t_0, t_1, t_2)}$:

$$\mathbf{q}_0 + \frac{\varepsilon}{2}\mathbf{t} \bullet \mathbf{q}_0$$

- Applying the dual quaternion **q** to a vertex **v**

$$\mathbf{v}' = \mathbf{q}\mathbf{v}\mathbf{q}^*$$

where **q*** is a conjugate of **q,** i.e. $\mathbf{q} = \mathbf{q}_0 + \varepsilon \cdot \mathbf{q}_e$

$$\mathbf{q}^* = \mathbf{q}_0^* + \varepsilon \cdot \mathbf{q}_e^*$$

Procedure

1. Computing the matrices of each bone by FK
2. Convert each matrix to dual quaternion
3. For each vertex compute the weighted dual quaternion by

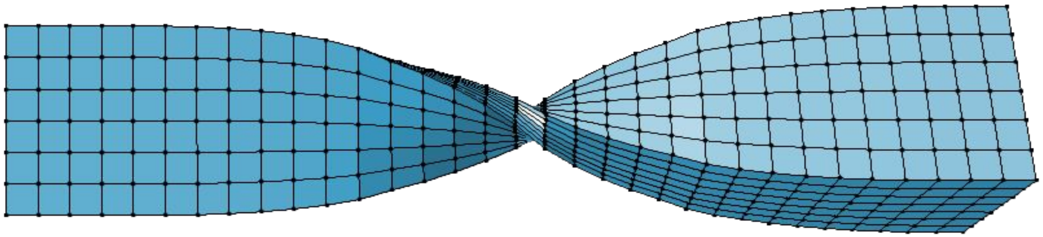
$$\mathbf{q} = \frac{\sum_{i=1}^{n} w_i \mathbf{q}_i}{\left\| \sum_{i=1}^{n} w_i \mathbf{q}_i \right\|}$$

4. Compute the global position of the vertex by

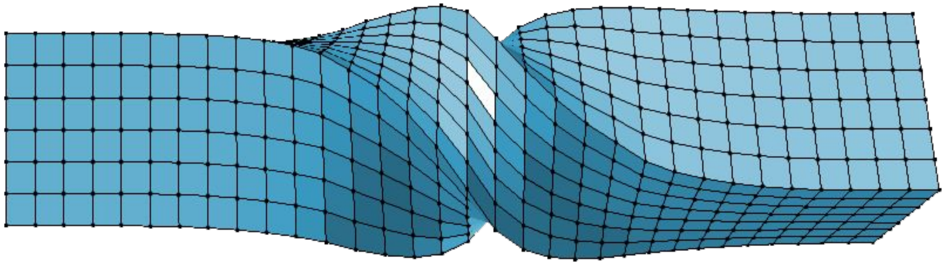
$$\mathbf{v}' = \mathbf{q}\mathbf{v}\mathbf{q}^*$$

- Joint collapse and candy wrap can be avoided using dual quaternion skinning.

https://www.youtube.com/watch?v=4e_ToPH-I5o



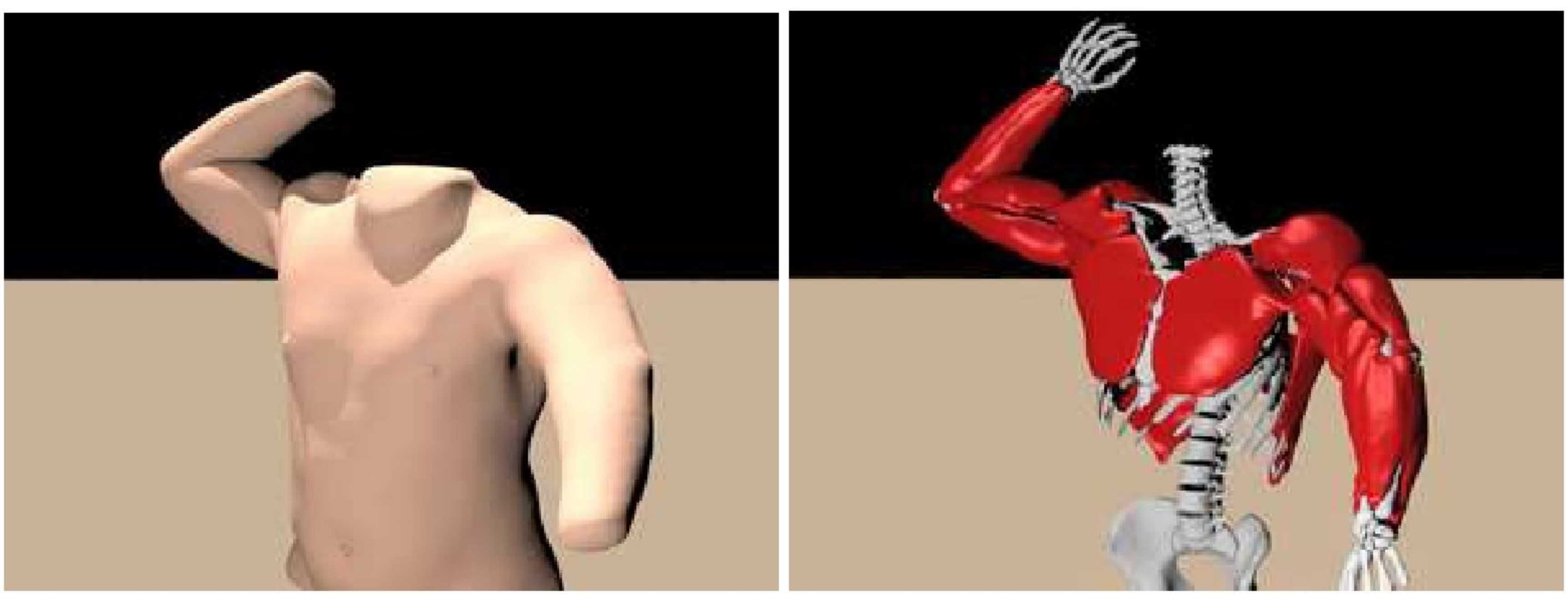
Linear Blending Skinning



Dual Quaternion Skinning

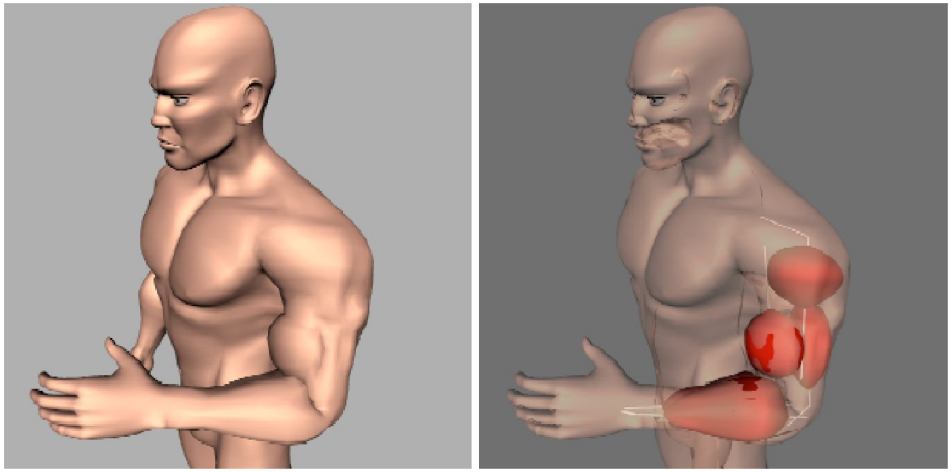
Anatomical models

- Model the body by
  - Muscles
  - Fat
  - Skin



Method

1. When the joints are bent, the muscles contract
2. The distance between the origin and insertion point decreases
3. The volume of the muscles are kept the same, so they pump up
4. The skin is deformed to cover the muscles



Visible Human Project

Many anatomical models are based the **Visible Human Project** dataset

Two cadavers sliced at 1milimeter intervals from the top to the bottom and photographed by cameras

The CT scans and MRI images were also taken

http://www.youtube.com/watch?v=iWP2HnPSMyo

Summary

- Skinning
  - Linear Blending
  - Dual Quaternion Skinning
  - Anatomical models

Readings

Skinning

- **A Comparison of Linear Skinning Techniques for Character Animation Afrigraph 2007**
- **Tutorial for Dual Quaternion Skinning** http://rodolphe-vaillant.fr/?e=29
- **Automatic Rigging and Animation of 3D Characters** Ilya Baran Jovan Popovi´c, SIGGRAPH 2007
- http://www.mit.edu/~ibaran/autorig/pinocchio.html
- Geometric Skinning with Approximate Dual Quaternion Blending, SIGGRAPH 2008
- Visible Human project http://www.nlm.nih.gov/research/visible/visible_human.html