# Computer Animation and Visualisation

# Lecture 4.

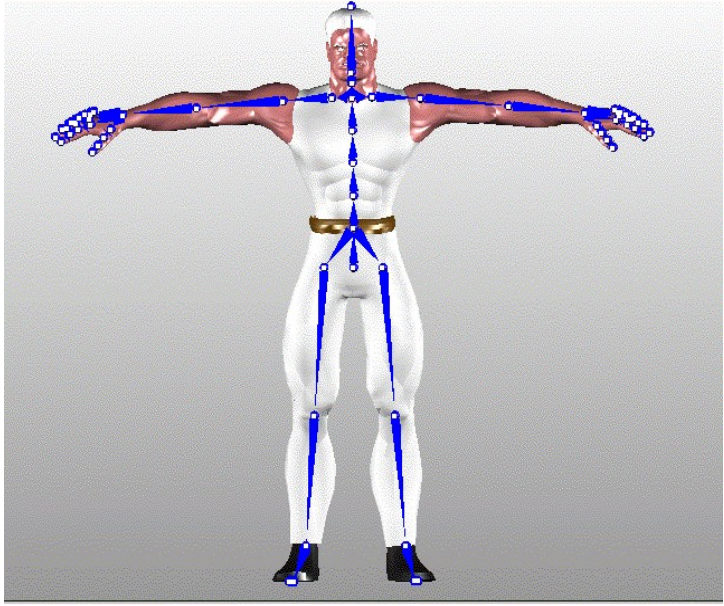
# Skinning

Taku Komura

# Overview

- Skinning
  - Background knowledge
  - Linear Blending
  - How to decide weights?
  - Example-based Method
- 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

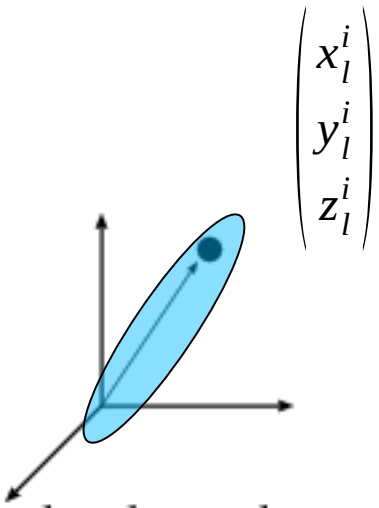
http://www.youtube.com/watch?v=dniWVu55PEc&feature=related

http://www.youtube.com/watch?v=z0QWBdk8MCA&feature=related

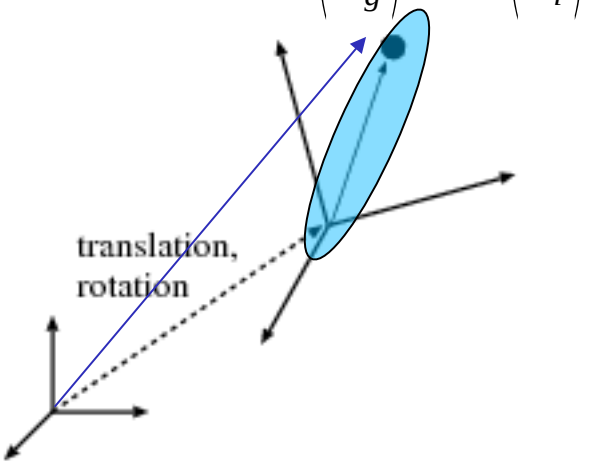
# Background

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

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

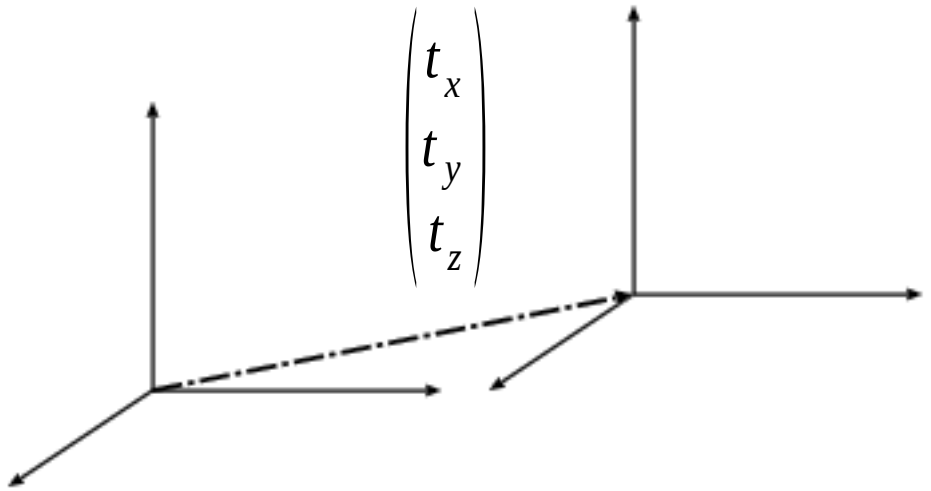


local coord




world coord

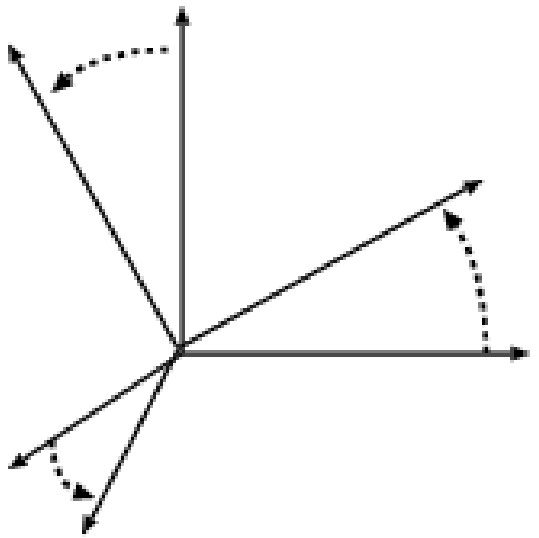
# Translation matrix




global coord local coord

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

# Rotation matrix



$$\text{Rot x}(\theta) \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

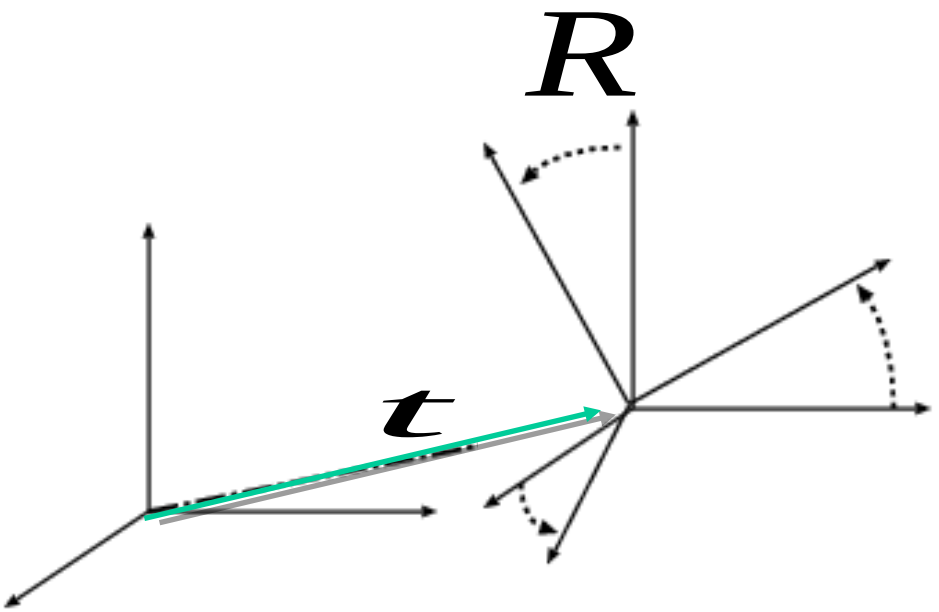
$$\text{Rot y}(\theta) \quad \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot z}(\theta) \quad \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

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

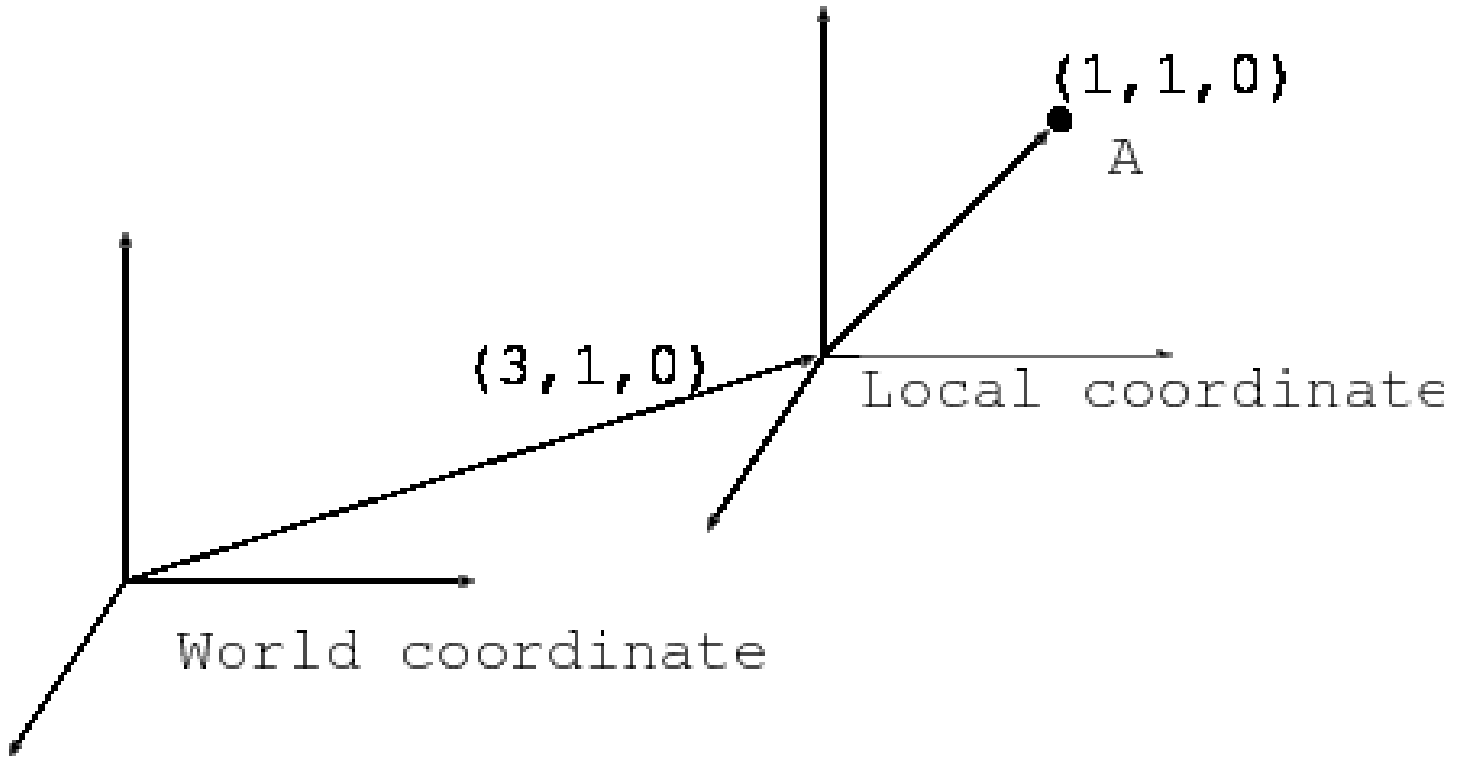
# Translation & Rotation matrix




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

$R$ : rotation part $(3x3)$

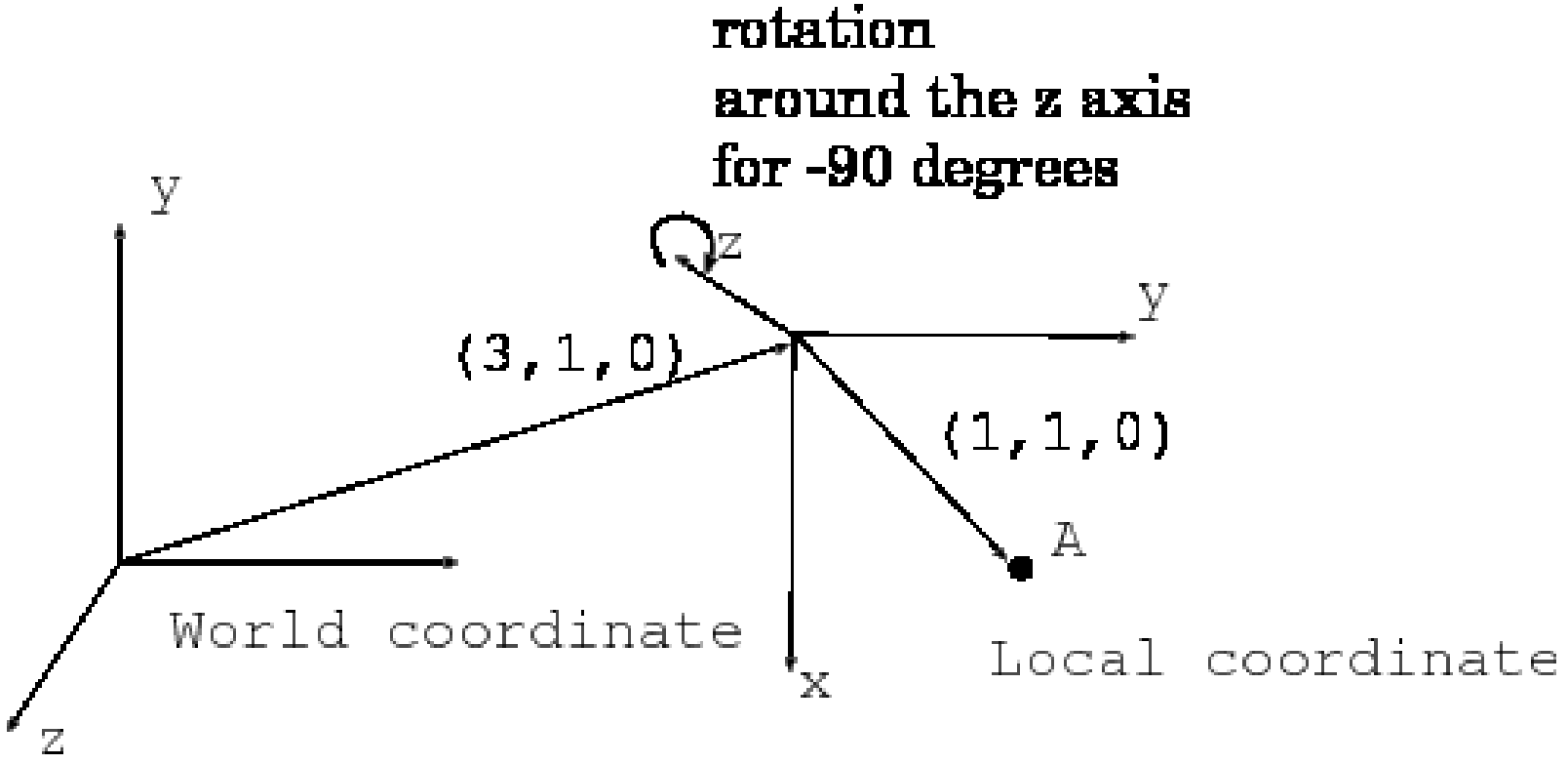
$t$ : translation vector $(3x1)$




- 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)

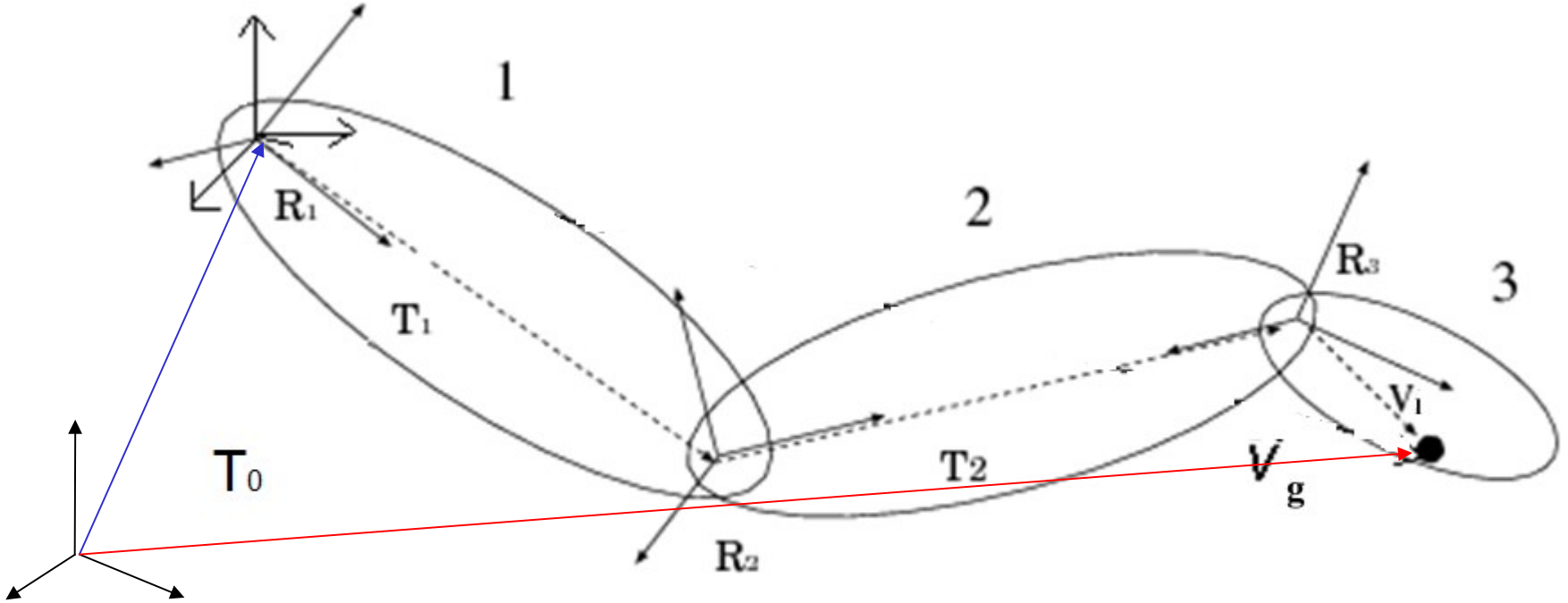



- What is the local-to-world transformation matrix?
- What is the position of point A in the world coordinate system?

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

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

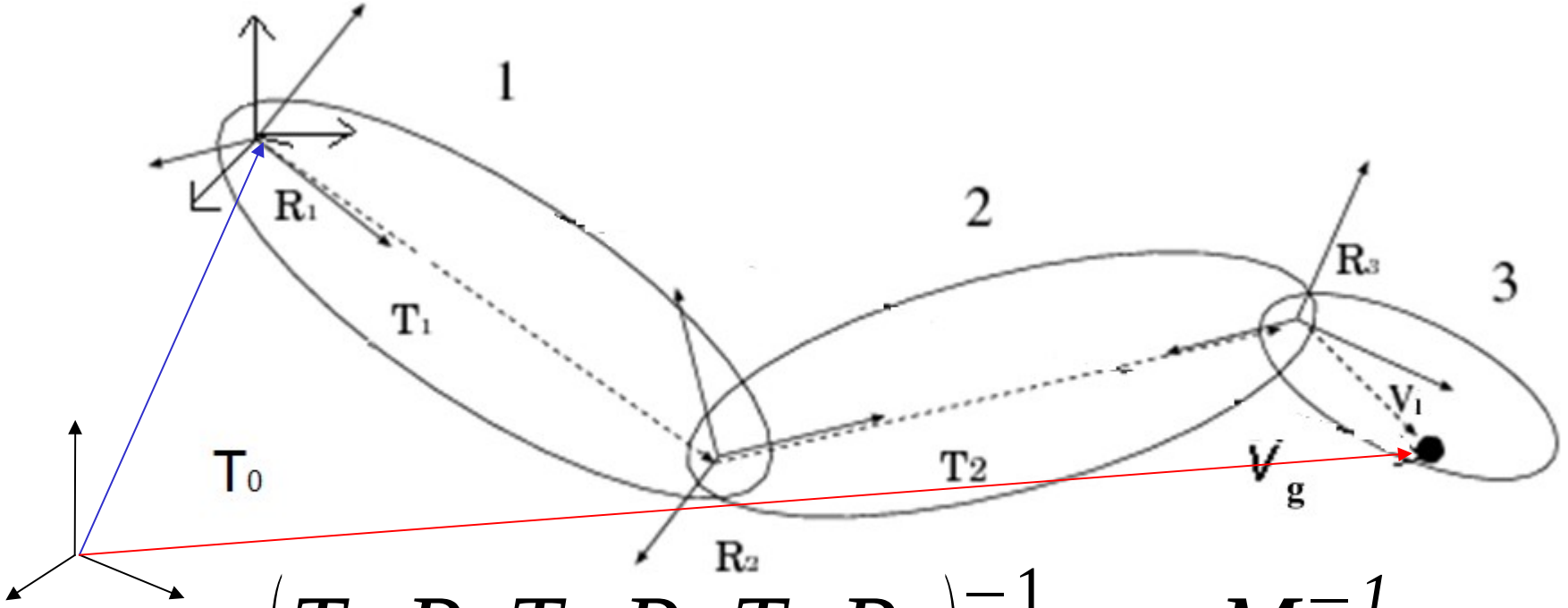
# A local-to-global transformation for a hand




$$v_g = T_0 R_1 T_1 R_2 T_2 R_3 v_l = M_3 v_l$$

- A body composed of three segments
- A vertex ***v3*** defined in a local coordinate system of segment 3
- Its global position $v_g$ is computed by the above equation

# A global-to-local transformation for a hand




$$v_l = \left(T_0 R_1 T_1 R_2 T_2 R_3\right)^{-1} v_g = M_3^{-1} v_g$$

- Given the global location , the local coordinate of the point can be the above equation

# 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 :

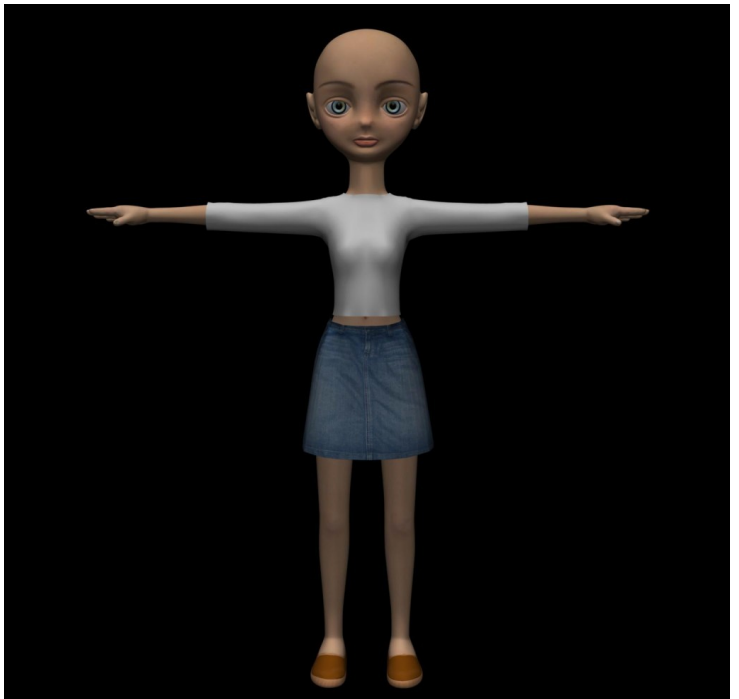
$$v_g = M_i v_i$$

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

# What about polygon characters?

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

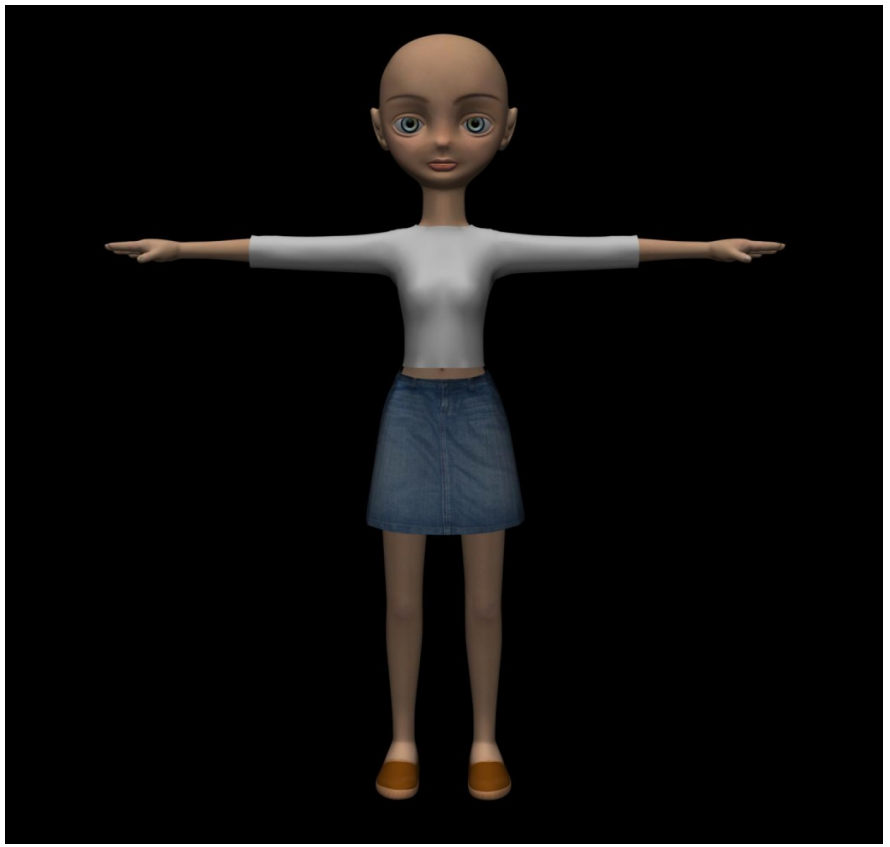
Problem 2. Some points on the body do not belong to a single bone but to multiple bones close to it

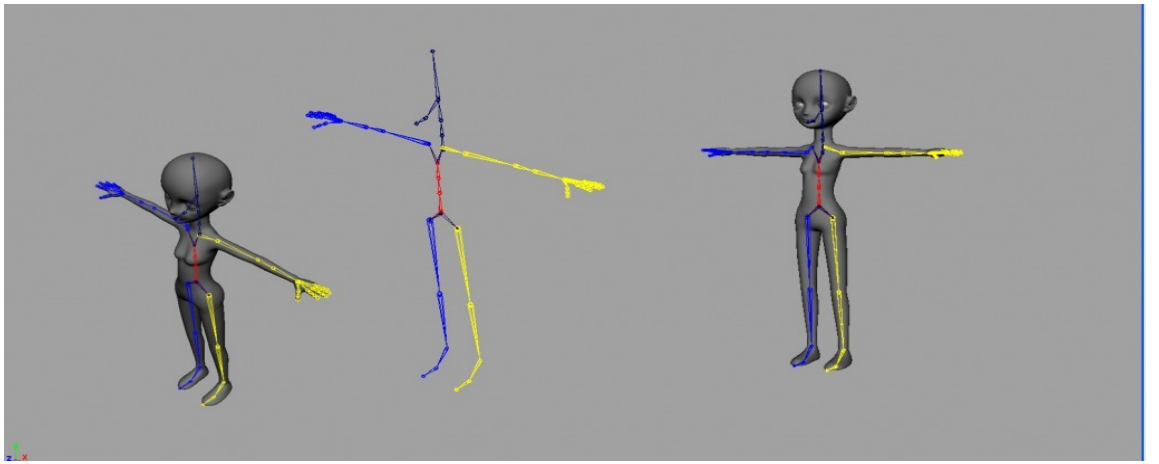


https://www.youtube.com/watch?v=04d_Pp6qjsw

# Solution:  Skinning

Fitting the skeleton into the polygon model (done manually)

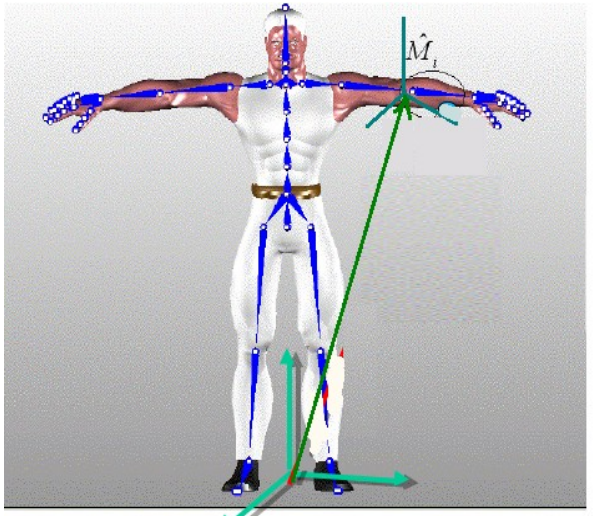


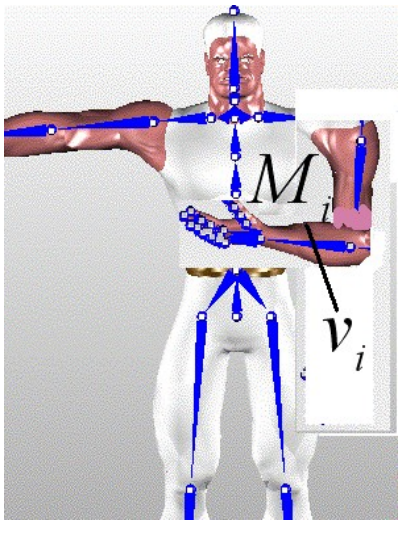


http://www.youtube.com/watch?v=dniWVu55PEc&feature=related

# Solution: Overview

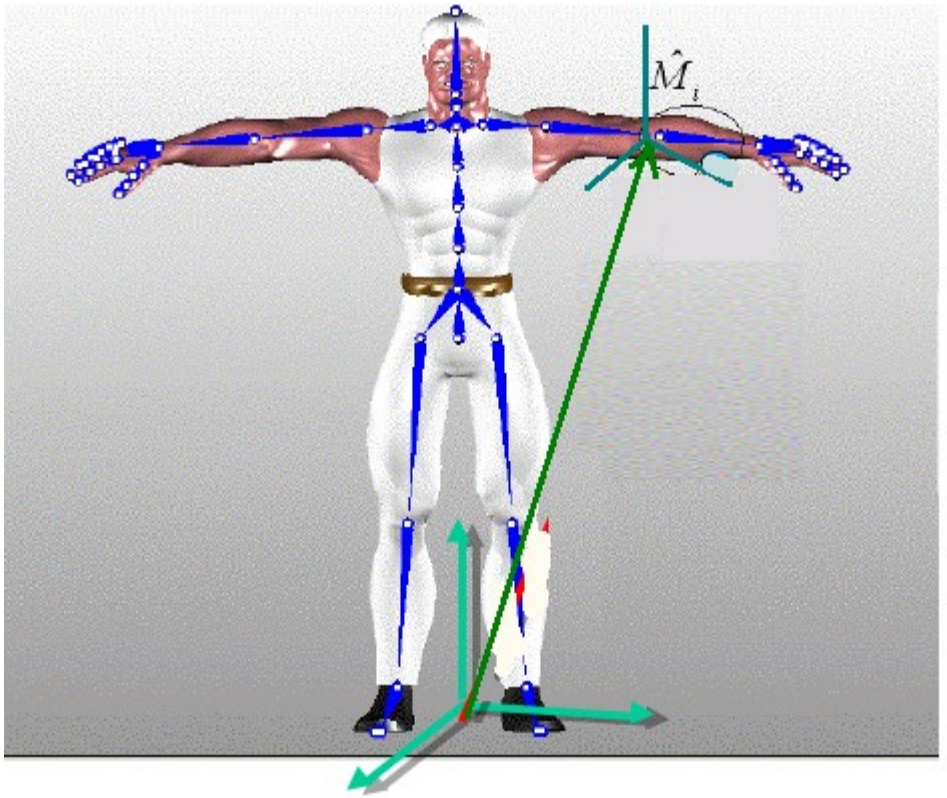
1. Convert the position of vertices to the local coordinates of each bone
2. Compute their global positions in the new pose using the local-to-global matrices
3. Blend the results between bones



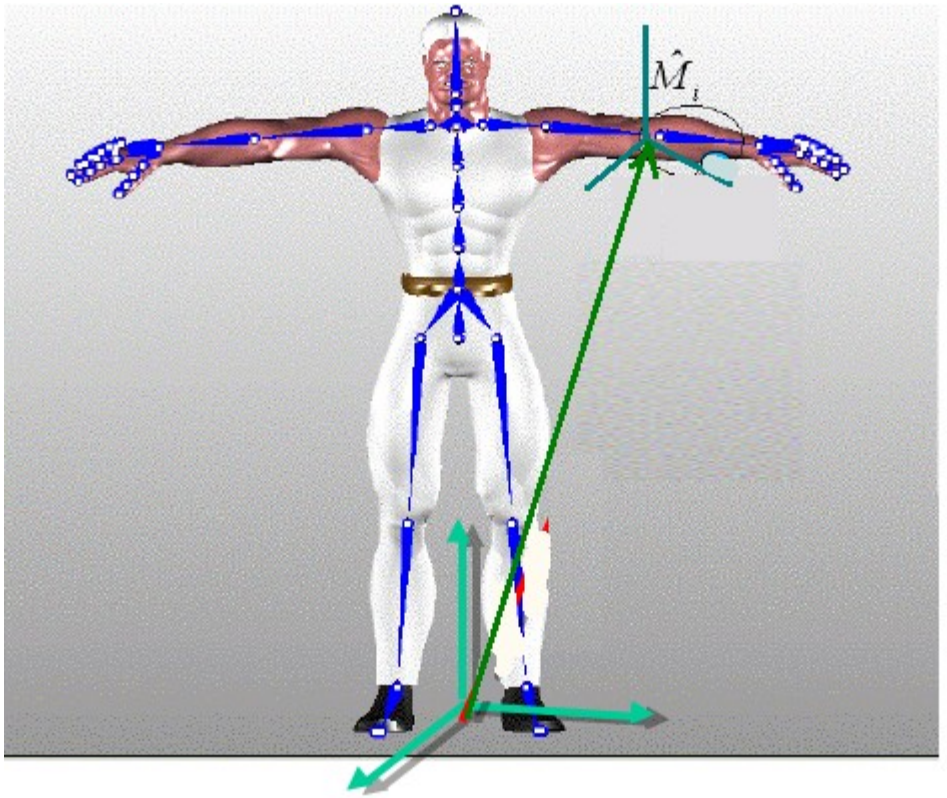
# Rest Pose to Bone Coordinate

The transformation matrix (local to global) associated with bone *i* in the rest pose is defined by $M_i$



# Rest Pose to Bone Coordinate

The transformation matrix (local to global) associated with bone *i* in the rest pose is defined by $M_i$



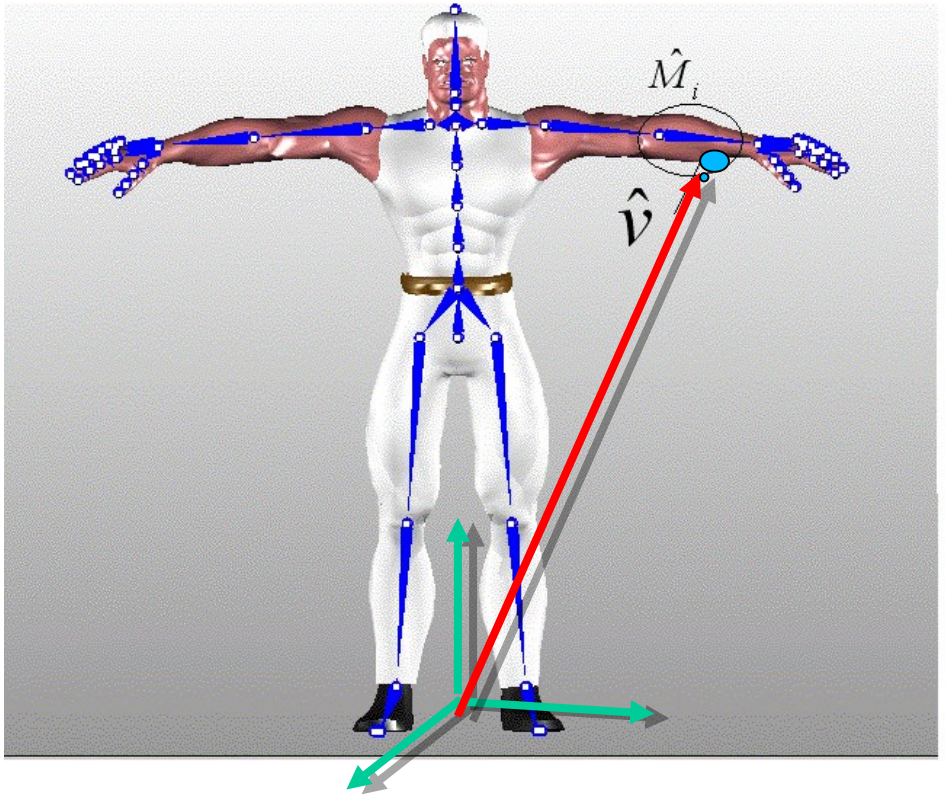

center of elbow joint

$$x = (0.7, 1.5, 0)$$

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

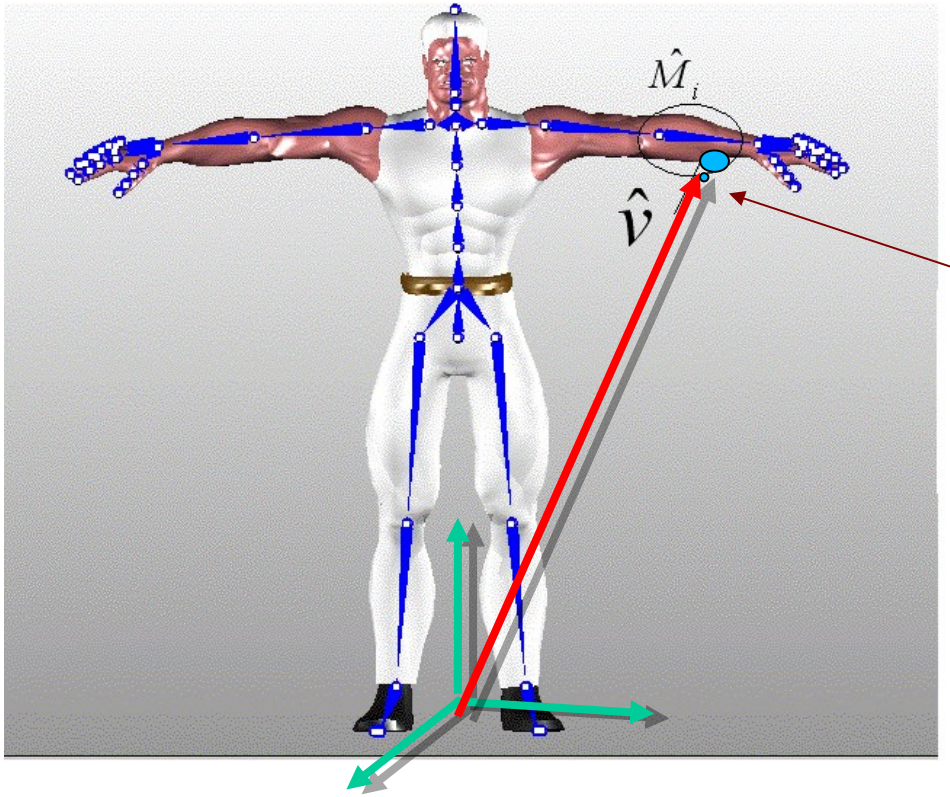
# Rest Pose to Bone Coordinate

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



# Rest Pose to Bone Coordinate

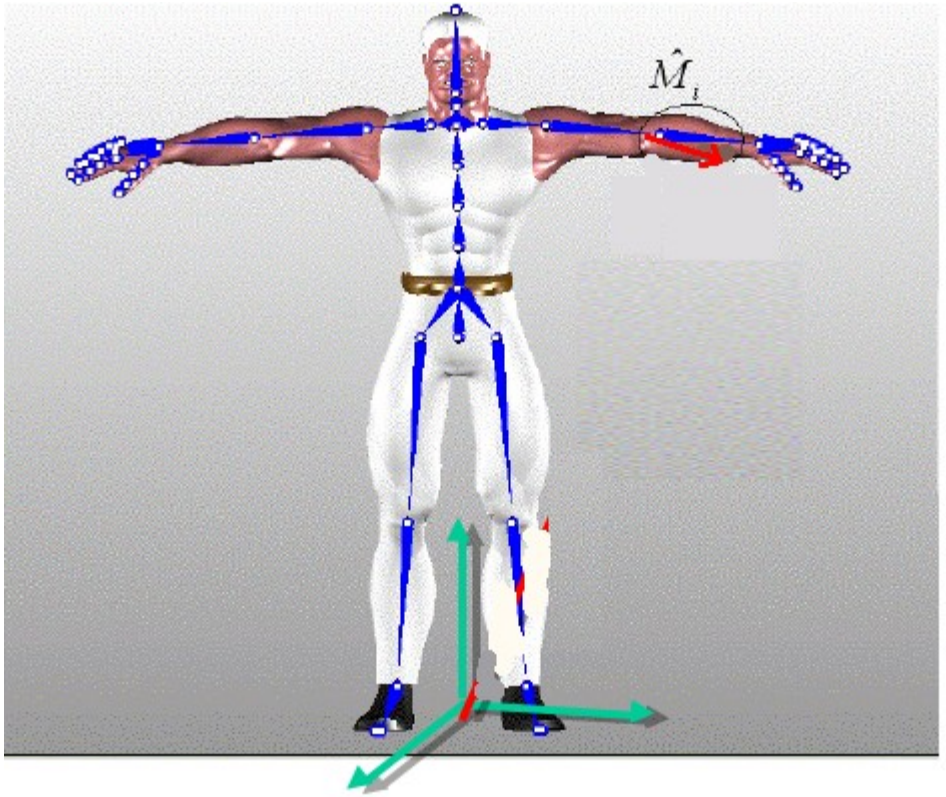
the global position of a particular vertex, ***v***, in the rest pose is defined $v_g$




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

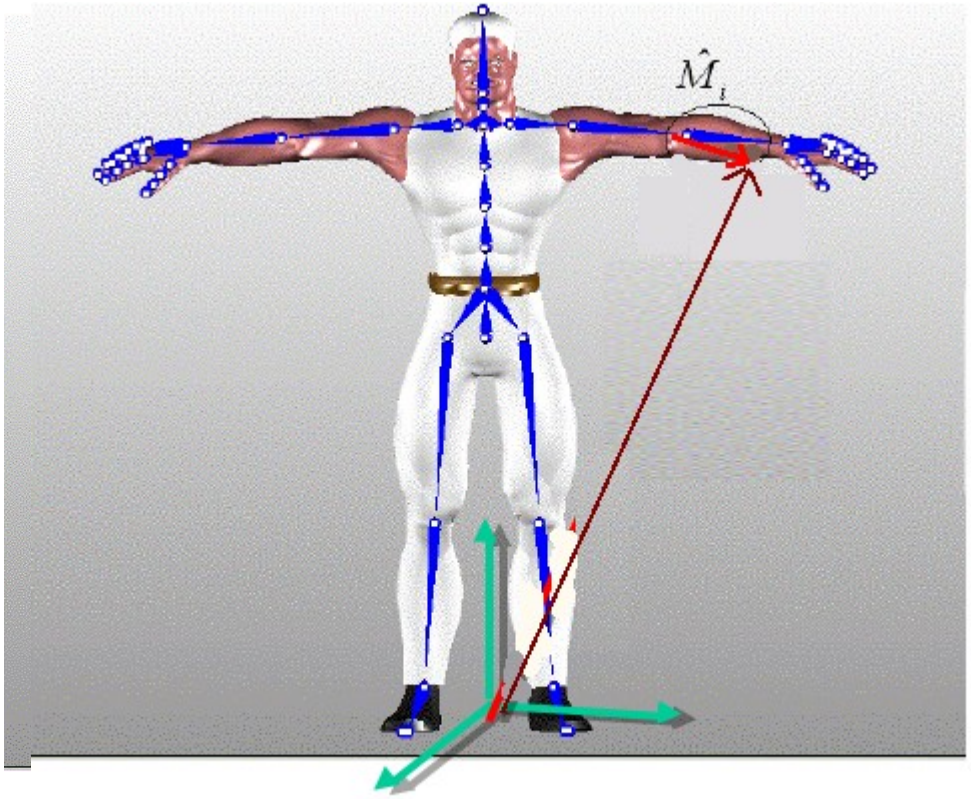
# Rest Pose to Bone Coordinate

- We want to know where this point $v_g$ is in the local coordinates



# Rest Pose to Bone Coordinate

The transformation matrix $M_i^{-1}$ transforms the global coordinates to the local coordinates.




center of elbow joint

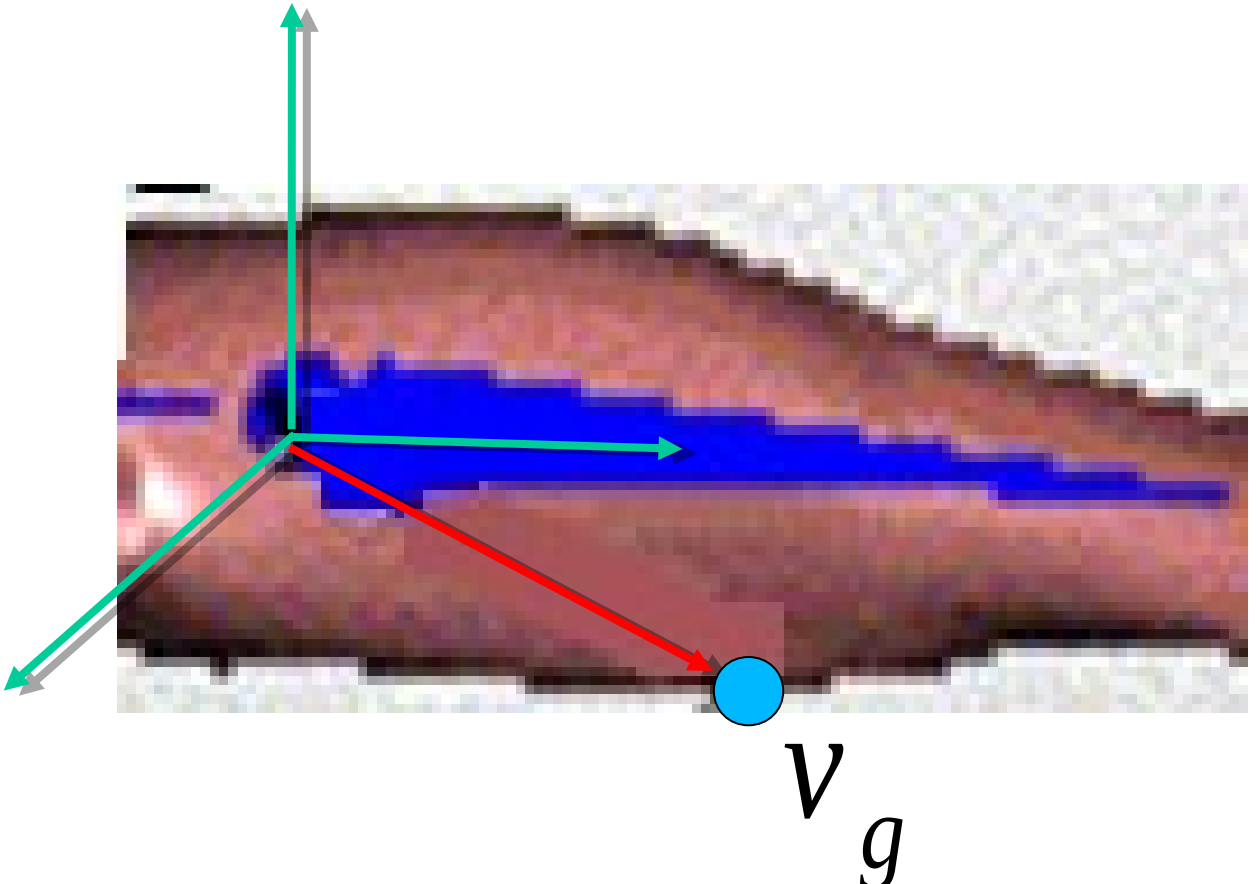
$$x = (0.7, 1.5, 0)$$

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

# 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:

$$v_i = M_i^{-1} v_g$$



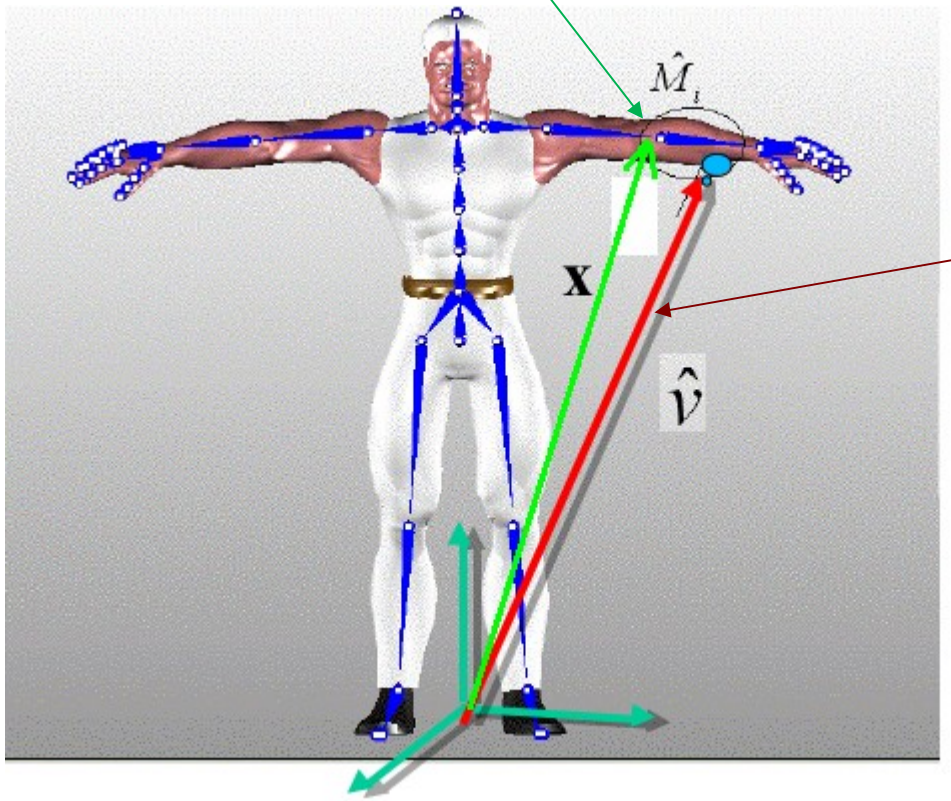
# Rest Pose to Bone Coordinate: Example

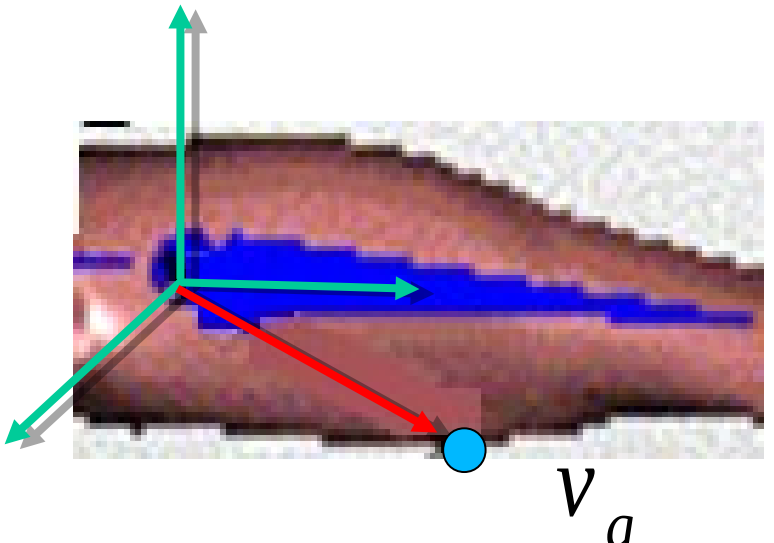
center of elbow joint

$x = (0.7, 1.5, 0)$

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

$v_g = (0.8, 1.45, 0)$

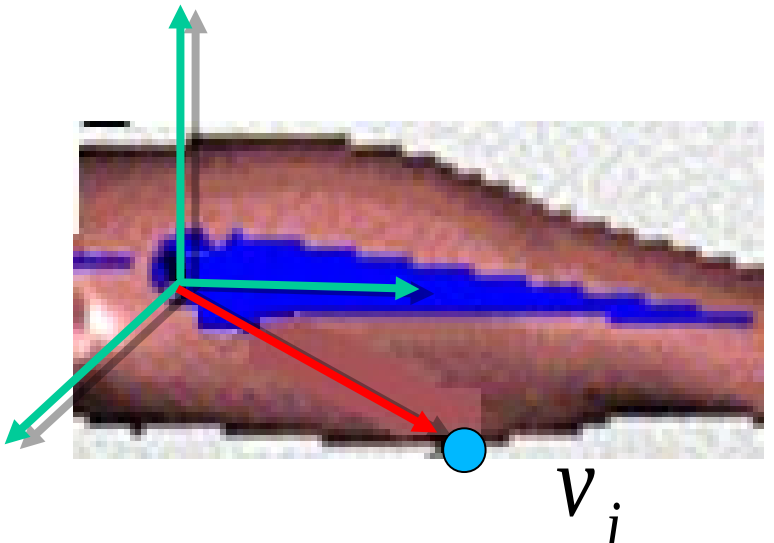


# Rest Pose to Bone Coordinate

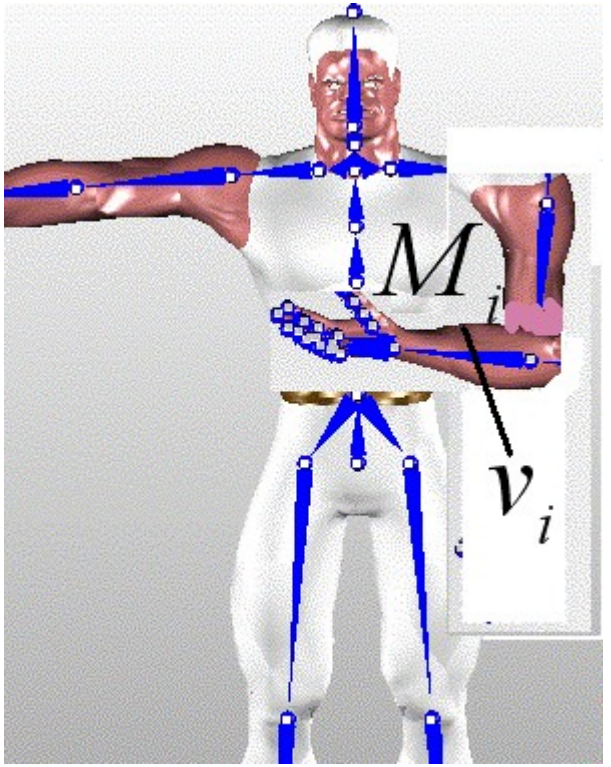
$$v_i = M_i^{-1} v_g = \begin{bmatrix} 1 & 0 & 0 & 0.7 \\ 0 & 1 & 0 & 1.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.8 \\ 1.45 \\ 0 \\ 1 \end{bmatrix}$$

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



# Bone Coordinate to World Coordinate

- The vertex in bone coordinates, $V_i$ is then transformed back into world coordinates by applying the transformation of the bone in the new pose

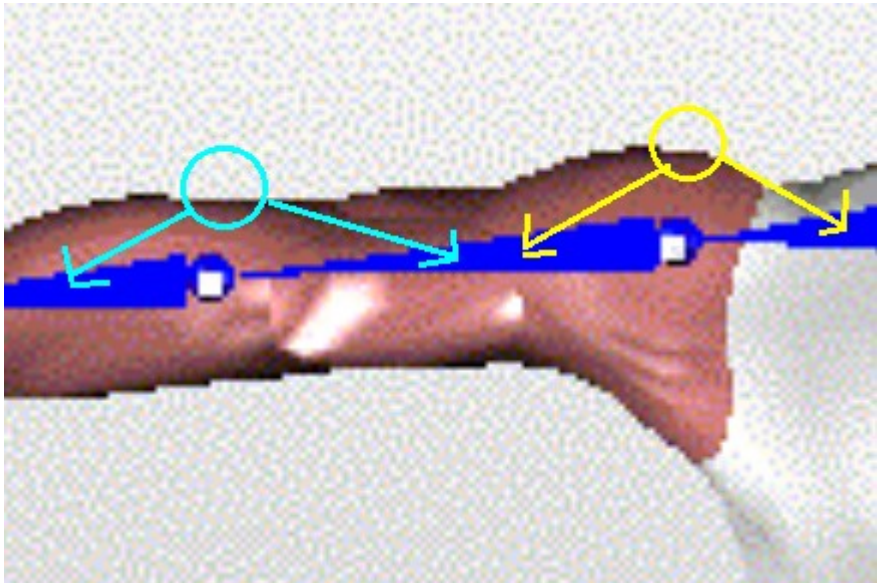



$$v_g = M'_i v_i = M'_i M_i^{-1} v_i$$

$M'_i$: 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

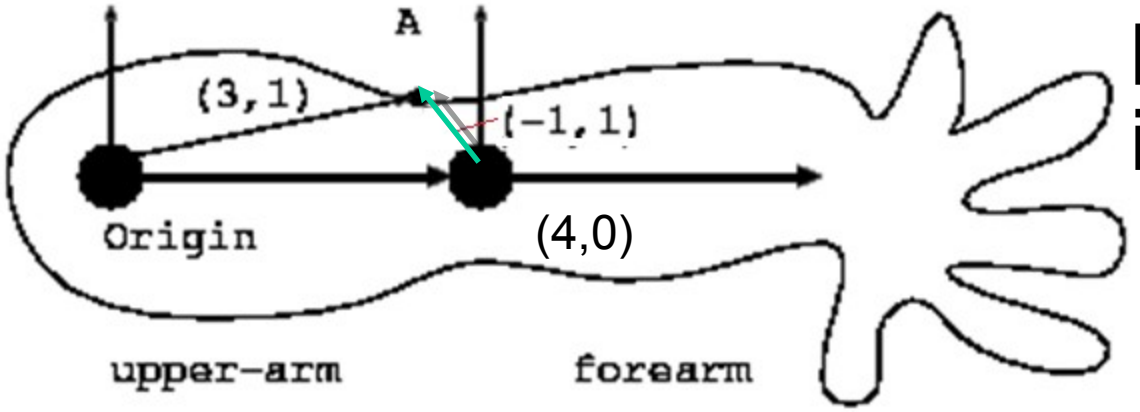


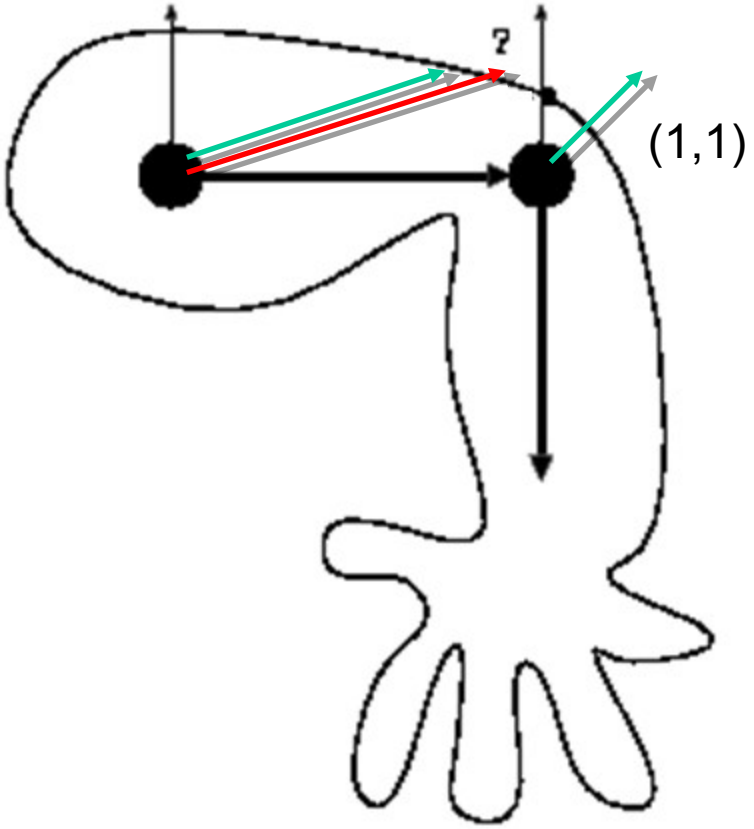
# Solution: Linear Blending

- Linear Blending determines the new position of a vertex by linearly combining 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:

$$v_i = \sum_{i=1}^{b} w_i M'_i M_i^{-1} v_g \qquad \sum_{i=1}^{b} w_i = 1$$

b is the number of bones influencing the position of v

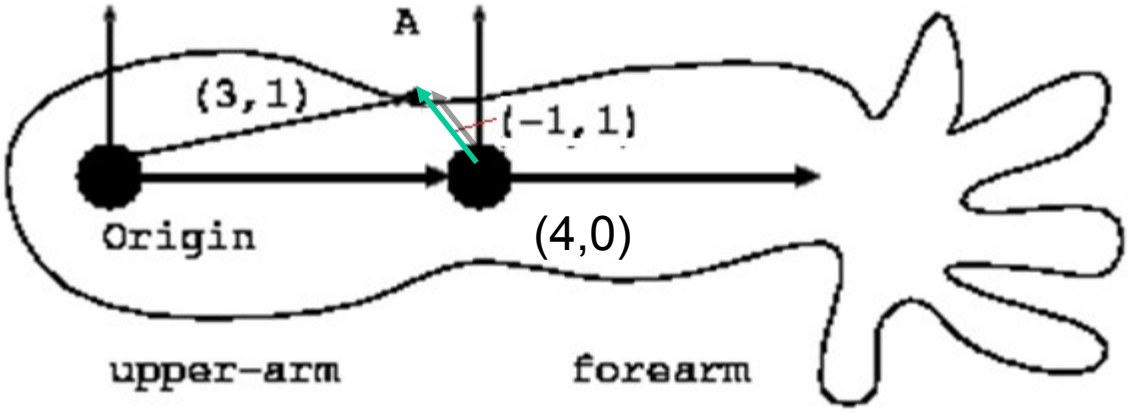


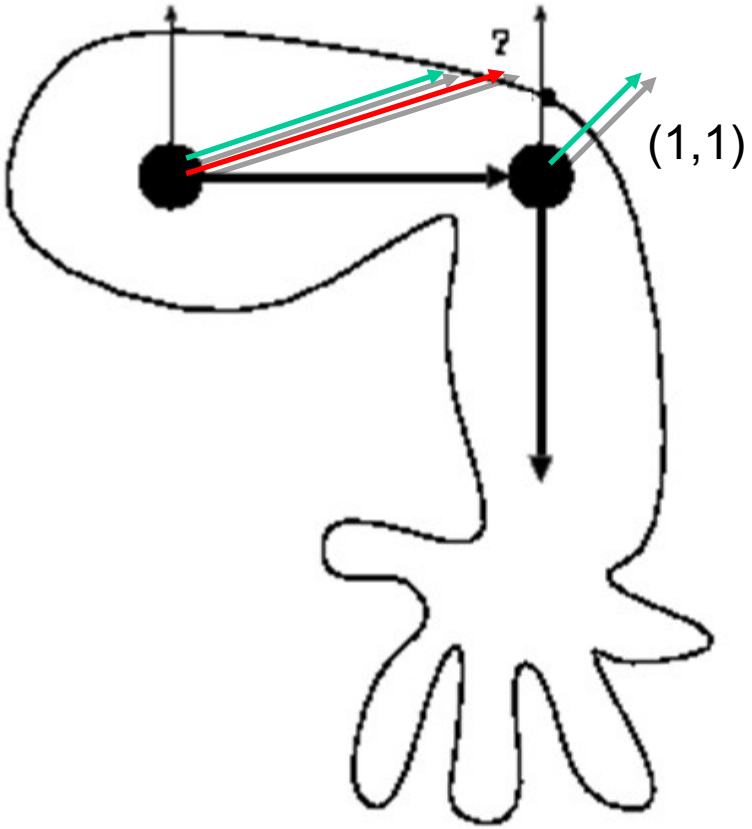




What is the position of point A after the elbow is bent 90 degrees?

- 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



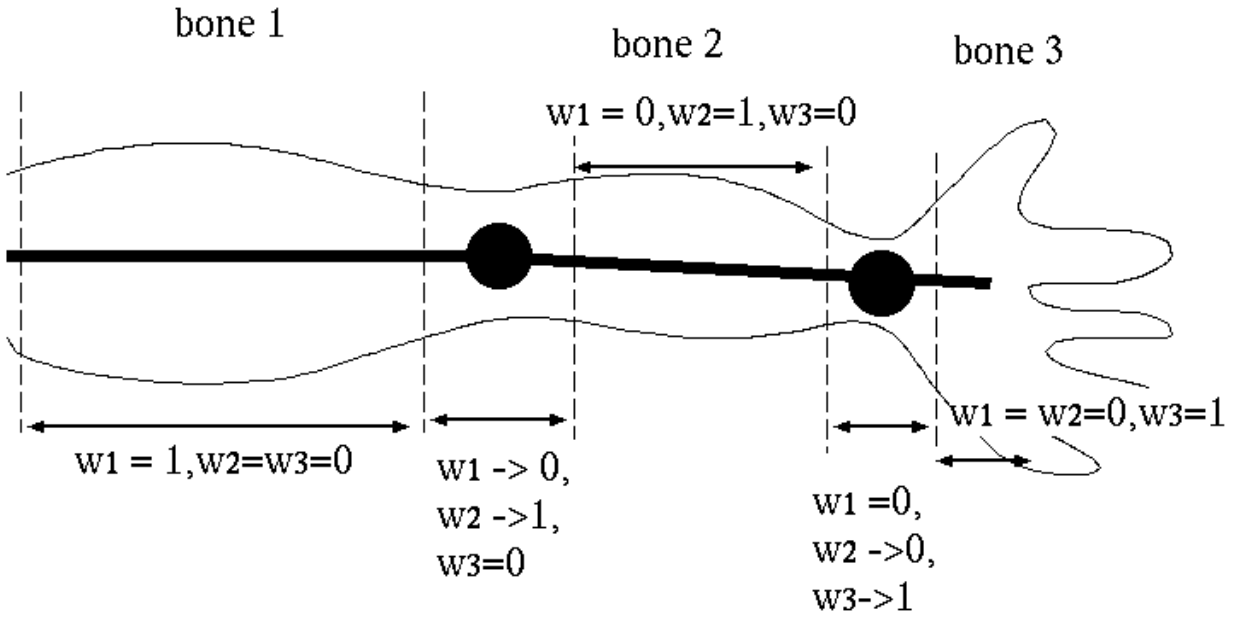

- Assuming it is a point of the upper arm, the position is (3,1)
- assuming it is a point of the forearm, it is (5,1)
- Assuming the weight is 0.8 for the upper-arm and 0.2 for the forearm, the position is

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

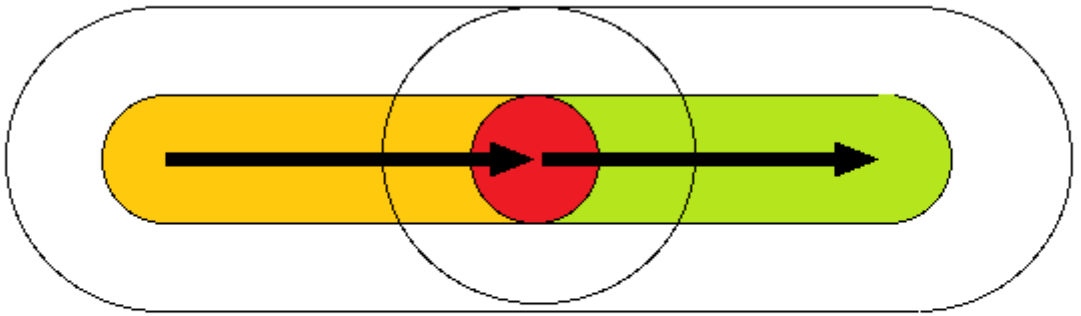
- If vertex v is in the middle of bone i, then $w_i = 1$ and for the rest $w_{j \neq i} = 0$
- If the vertex is near the border of bone i and i+1, *wi* will gradually decrease to 0 and *wi+1* will gradually increase to 1
- If the vertex is affected by more than three bones, the weight can be determined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

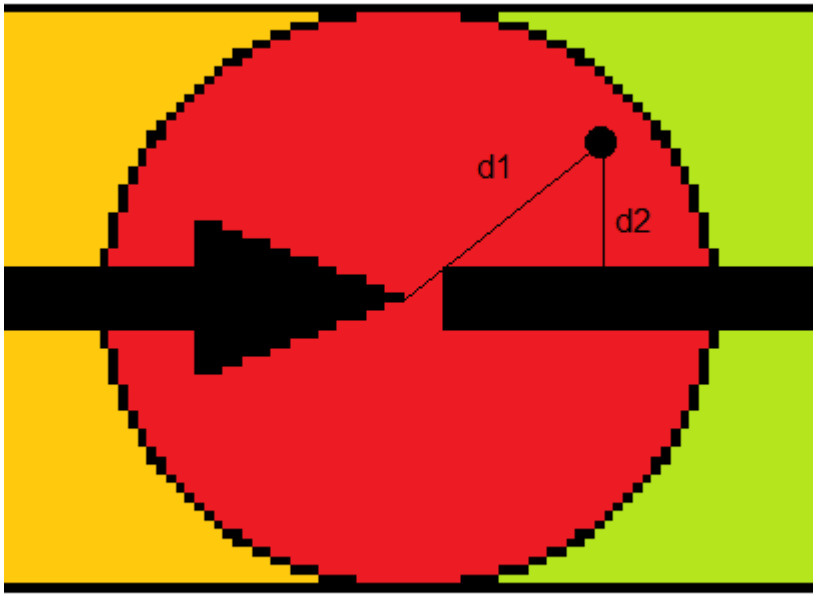


# How to decide the weights?

## 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.

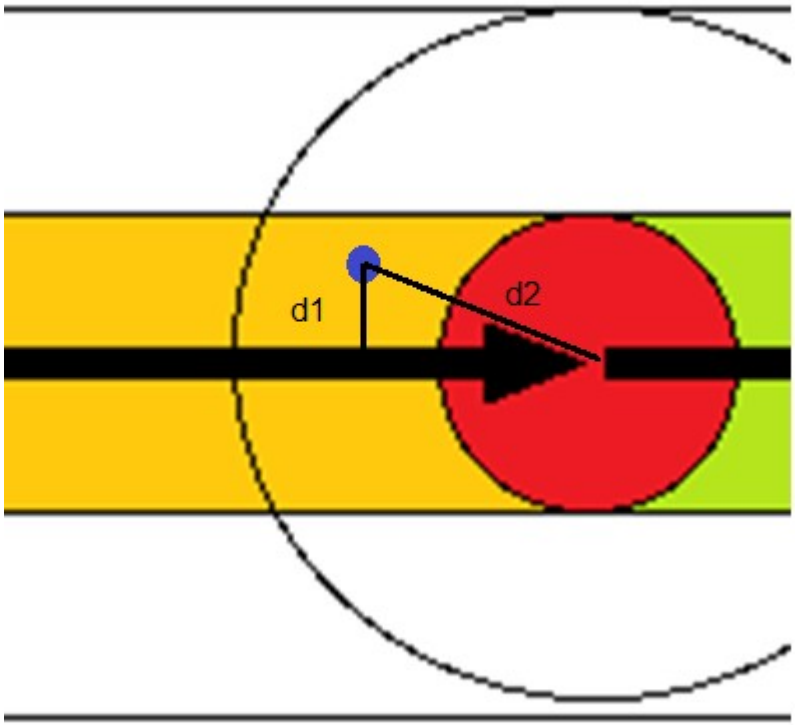
$$w_1 = \sqrt{\frac{\frac{1}{d_1}}{\frac{1}{d_1} + f(d_2)\frac{1}{d_2}}}, w_2 = \sqrt{\frac{f(d_2)\frac{1}{d_2}}{\frac{1}{d_1} + f(d_2)\frac{1}{d_2}}}$$

$f(x)$ : fall-off function

$f(x) = 1$ if x $< r_1$

$f(x) = 0$ if x $> r_2$

$$f(x) = 1 - \frac{x - r_1}{r_2 - r_1}$$

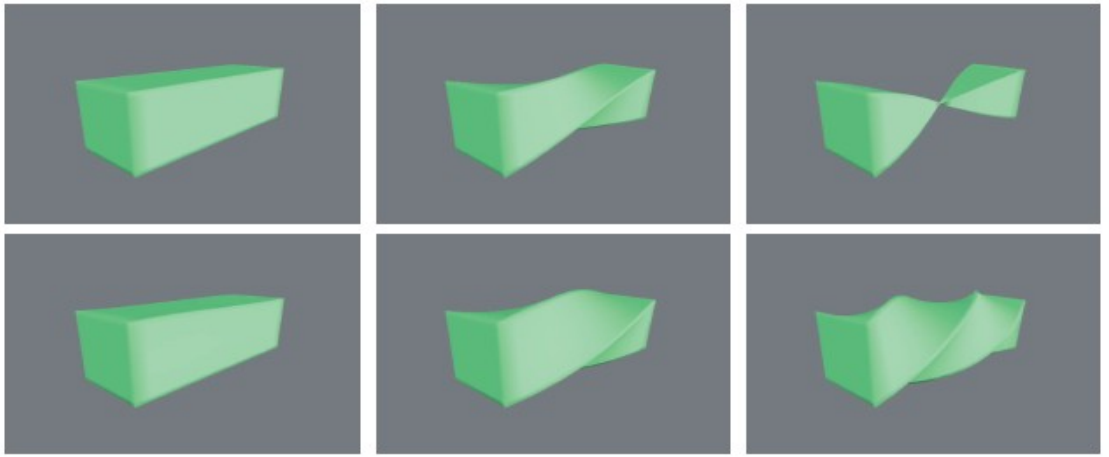


# Deciding the weights manually

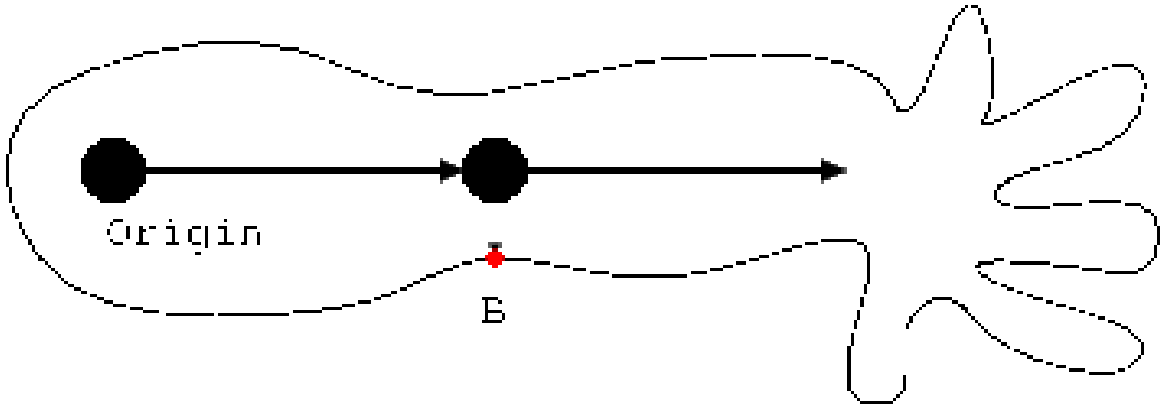
http://www.youtube.com/watch?v=z0QWBdk8MCA&feature=related

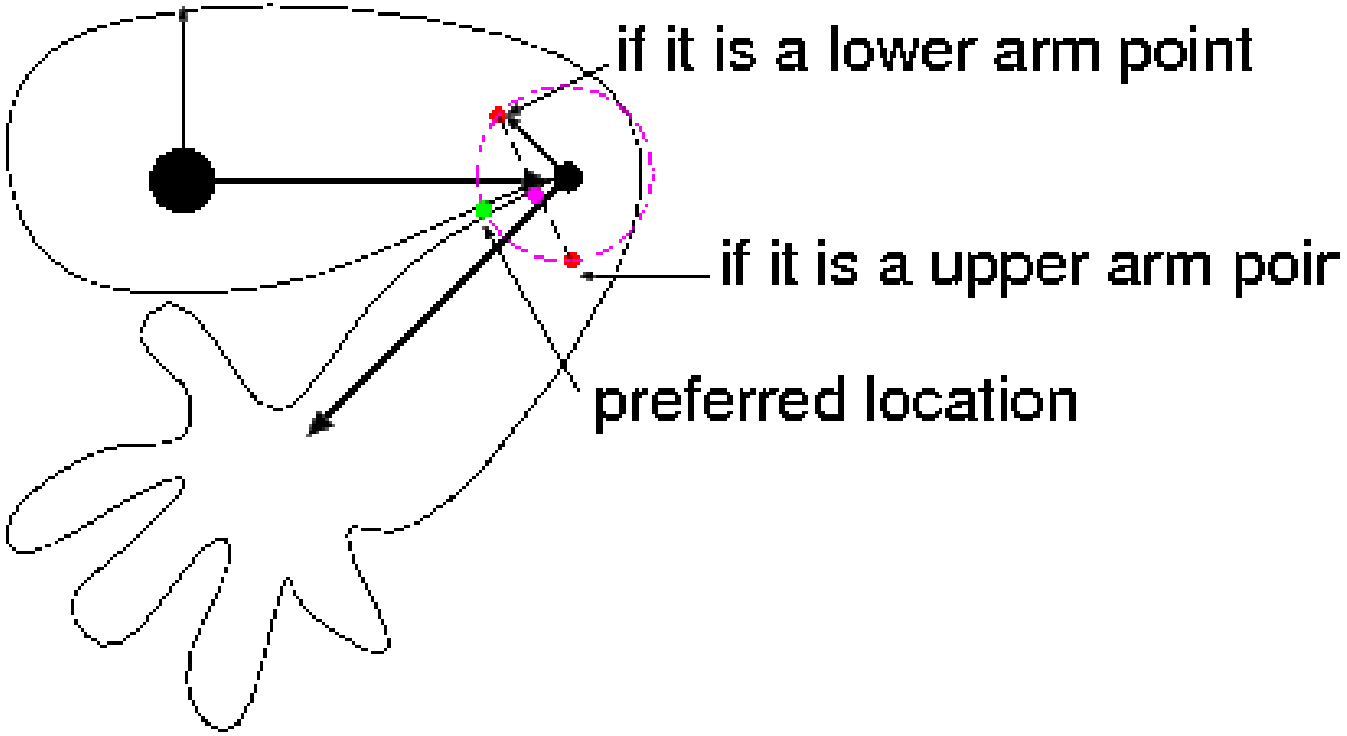
# Problems with Linear Blending

- 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

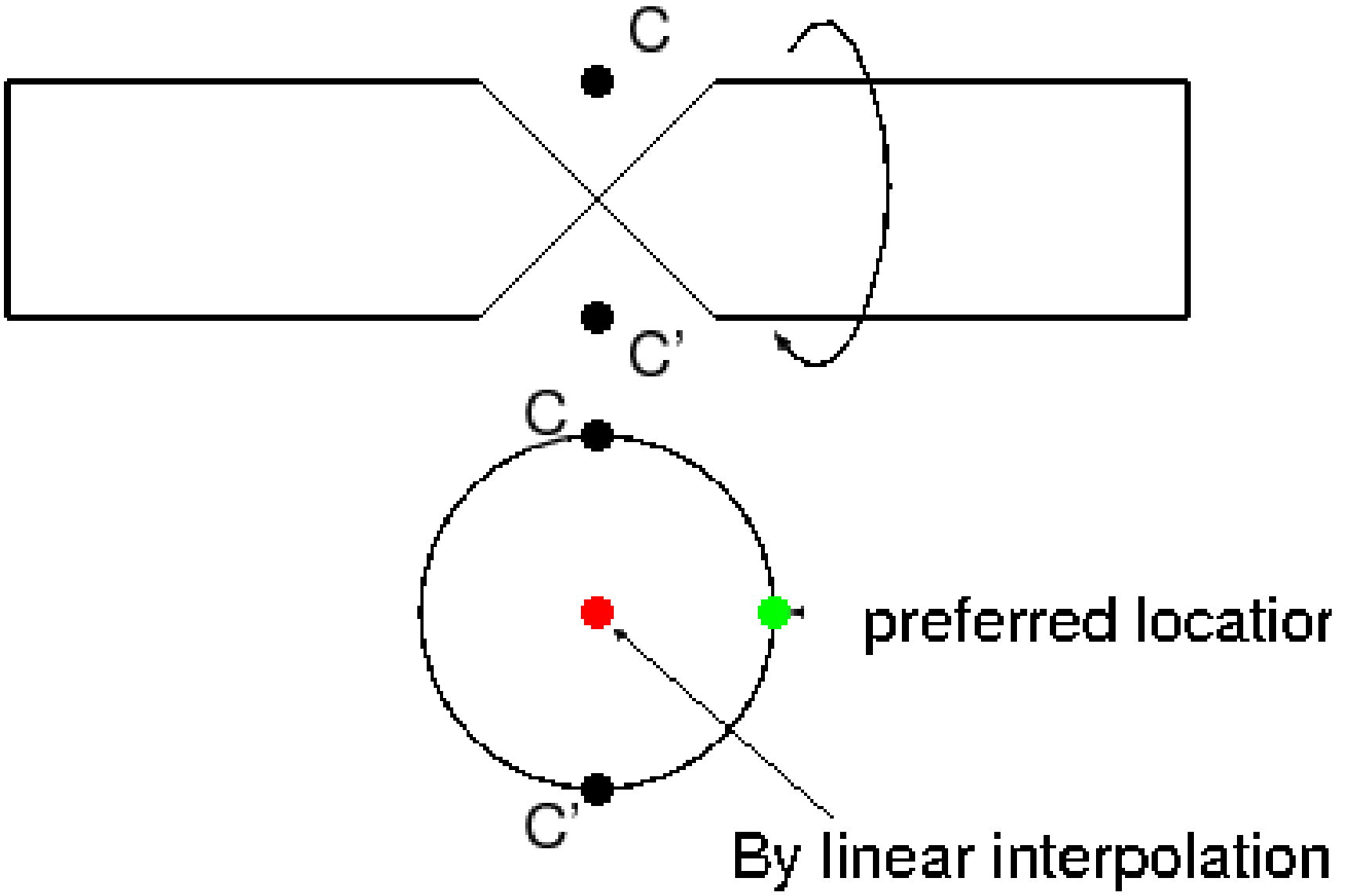


# Why does it happen?



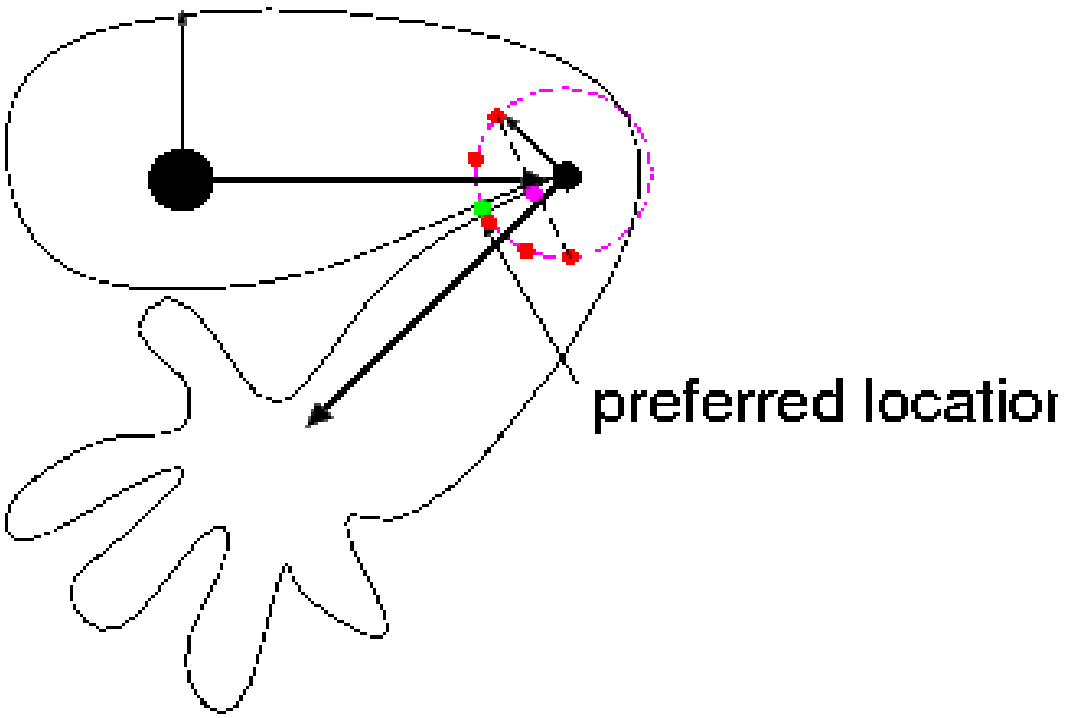



# Why does it happen?



# One Simple Solution

- Find the closest joint from the vertex
- Use this joint position as the rotation center for both body parts
- Interpolate along the arc (can use quaternion interpolation, SLERP, see notes of lecture 1)
- Works well for 2 bones

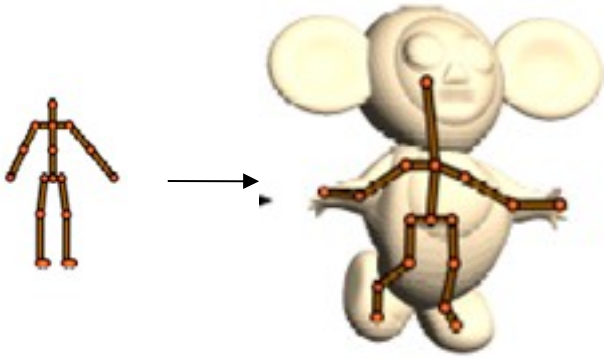


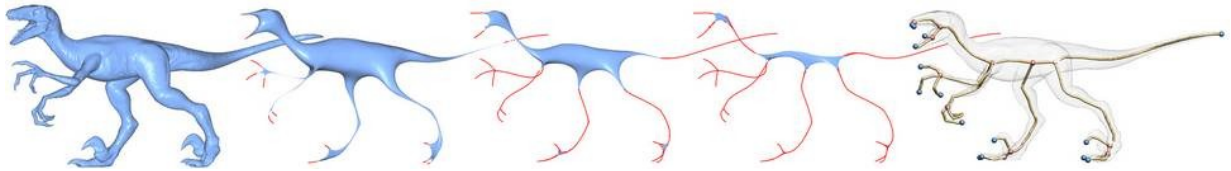
# Some advanced stuff

- Automatic computation of the weights from examples

examples

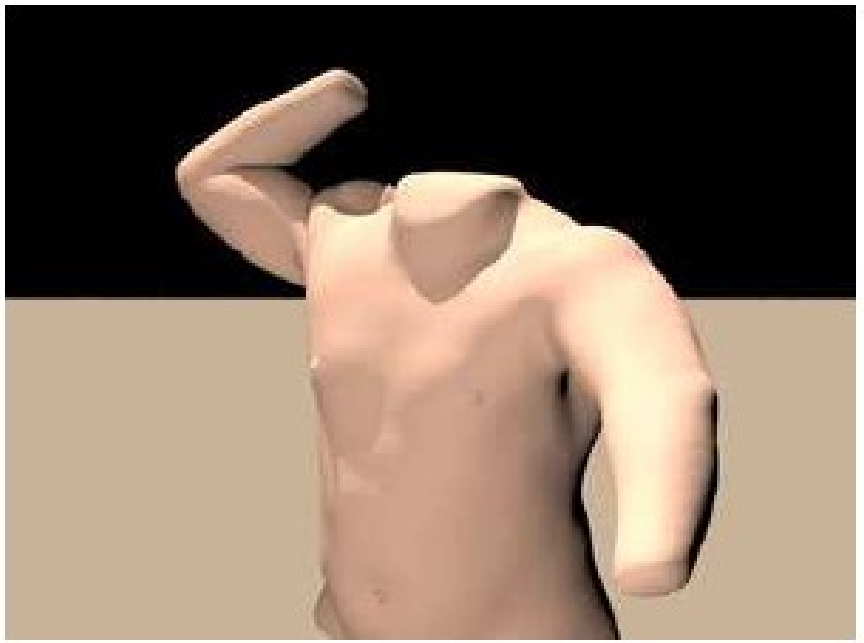
- Automatic skeleton fitting / extraction





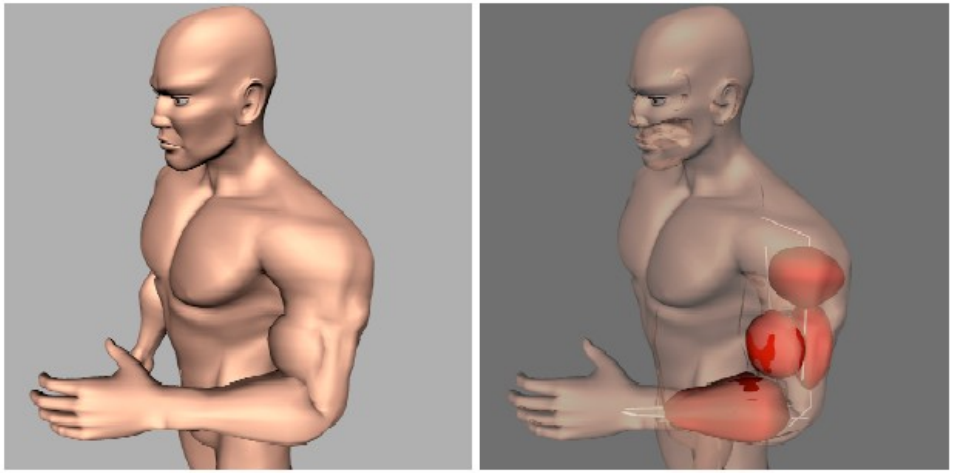
# Anatomical models

- 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.nlm.nih.gov/research/visible/visible_human.html

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

# Summary

- Skinning
  - Linear Blending
  - Example-based weight estimation
  - Anatomical models

# Readings

Skinning

- **A Comparison of Linear Skinning Techniques for Character Animation Afrigraph 2007**
- **Multi-Weight Enveloping: Least-Squares Approximation Techniques for Skin Animation**
  - Wang and Phillips, SCA 02
  - http://portal.acm.org/citation.cfm?id=545283

- Guessing the weights from examples
  - Alex Mohr Michael Gleicher **Building Effcient, Accurate Character Skins from Examples.** SIGGRAPH 2003
- **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