CS451 Deformation

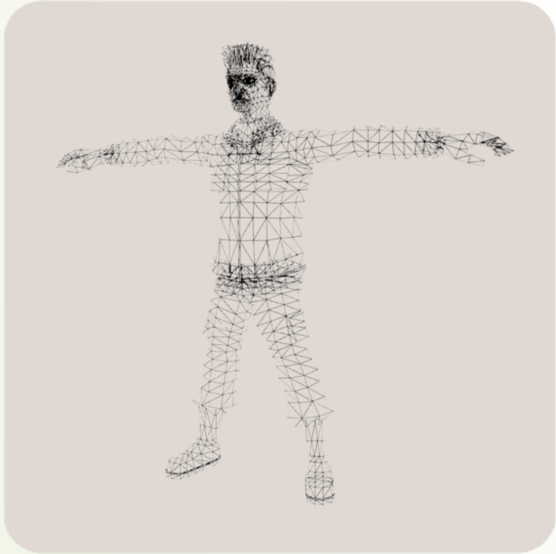
skeleton-subspace deformation

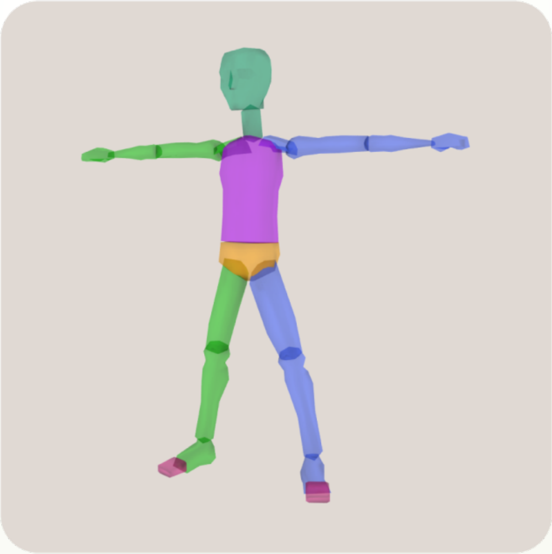
Jyh-Ming Lien
Department of Computer SCience
George Mason University

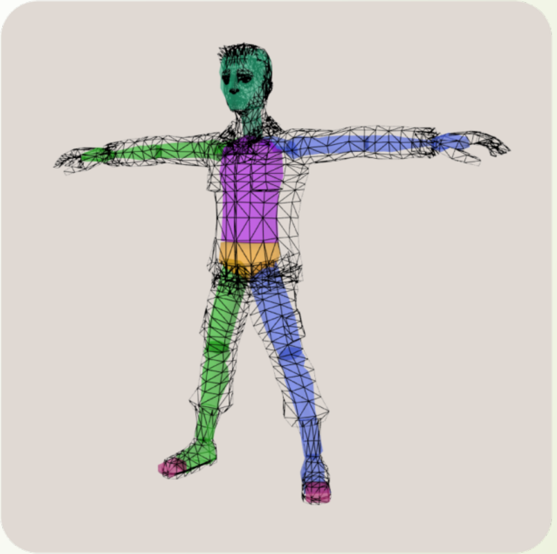
Based on Nicholas Park's lecture note

Skeleton-subspace deformation

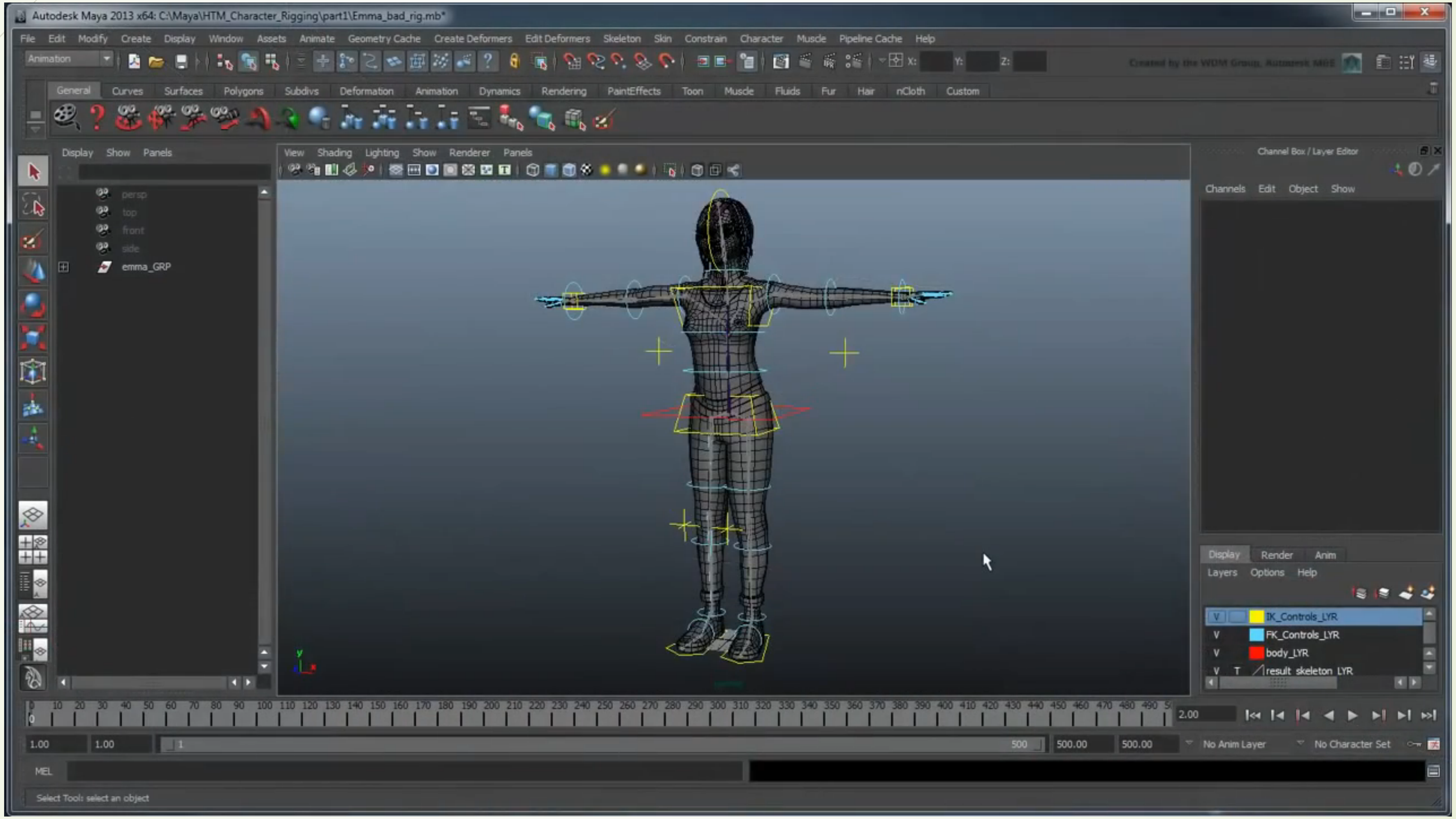
- A popular method for animating a character is to use a *skeleton*, which is composed of bones. The skeleton is embedded into the polygon mesh. When the skeleton is animated, the vertices of the polygon mesh will be accordingly animated





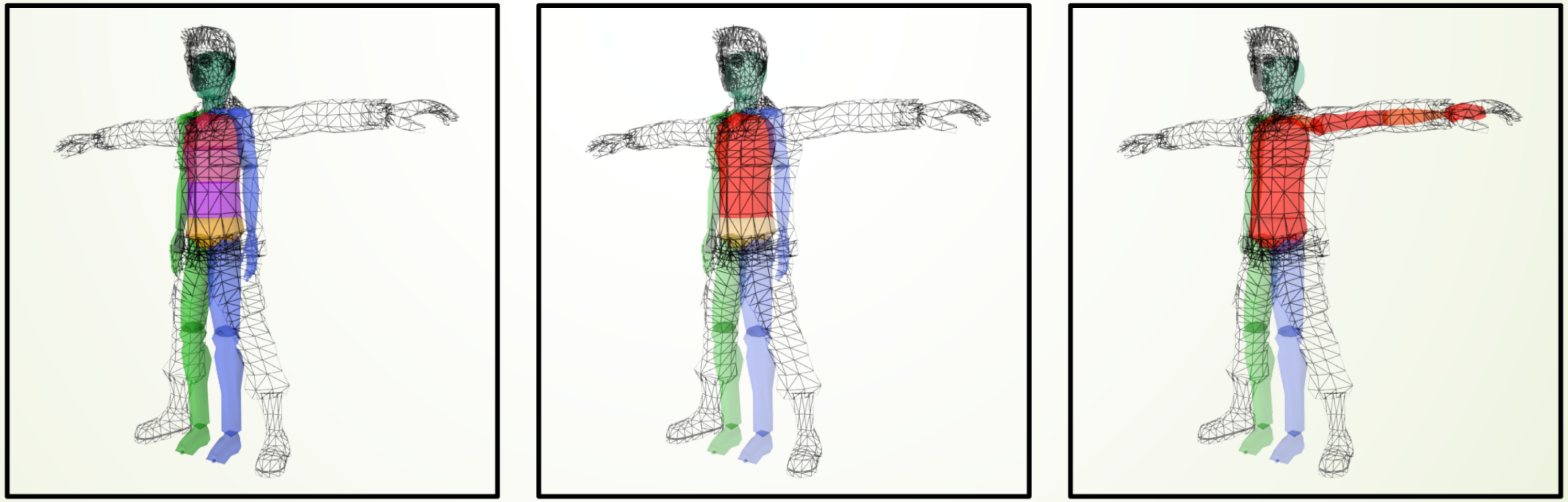


How does SSD Look Like



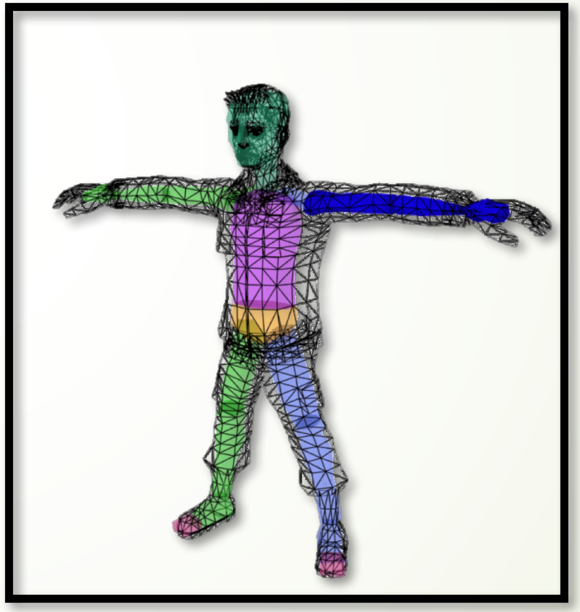
Skeleton

- Skeleton editing and embedding
  - The skeleton template such as 3ds Max biped is positioned in the *default pose*.
  - The bones are edited
  - The skeleton is made to fit to the polygon mesh

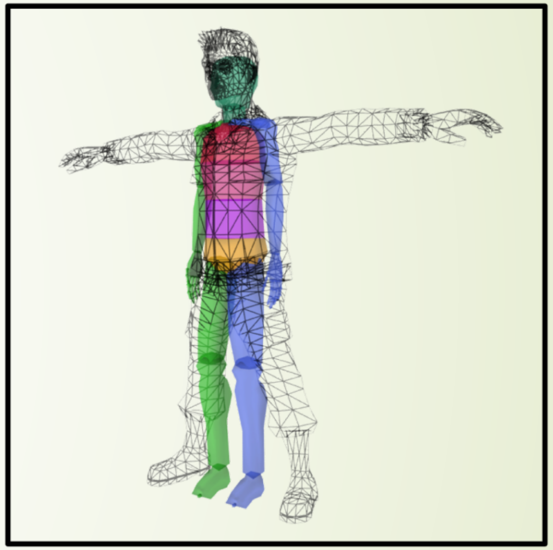


Important Terminology

- Zero pose or default pose
- Binding pose
- Bone/World space
- Skinning
- Blending weight



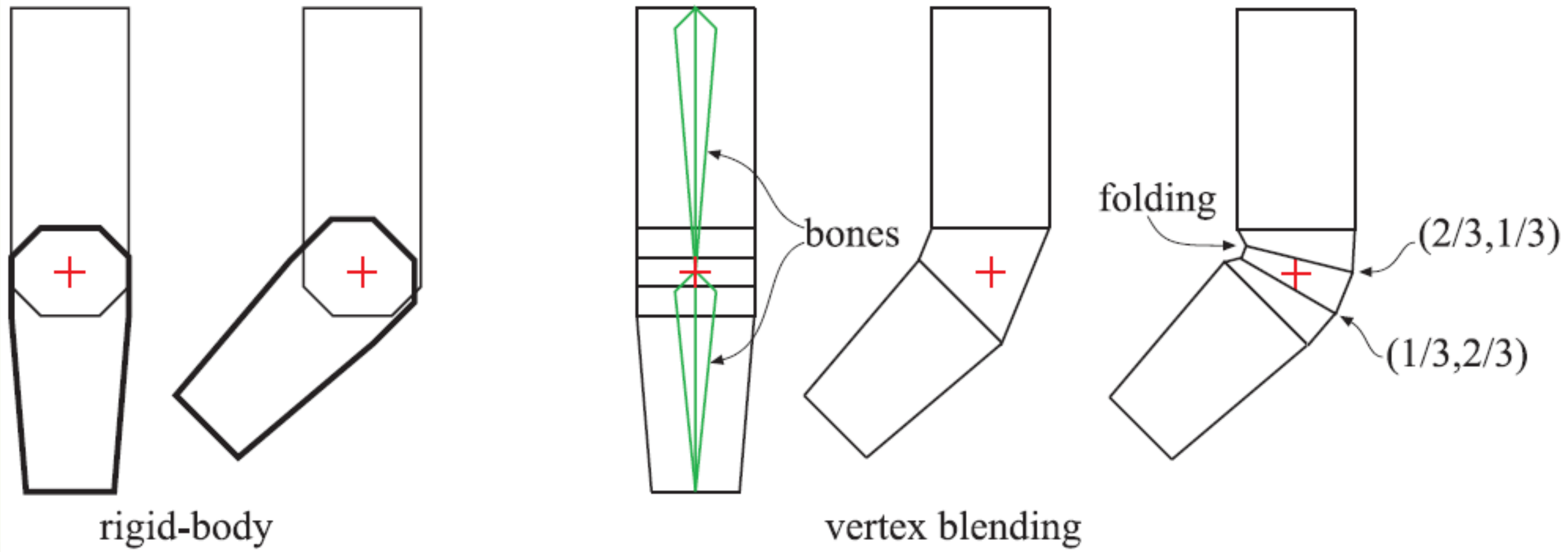
Binding pose



default pose

Skinning

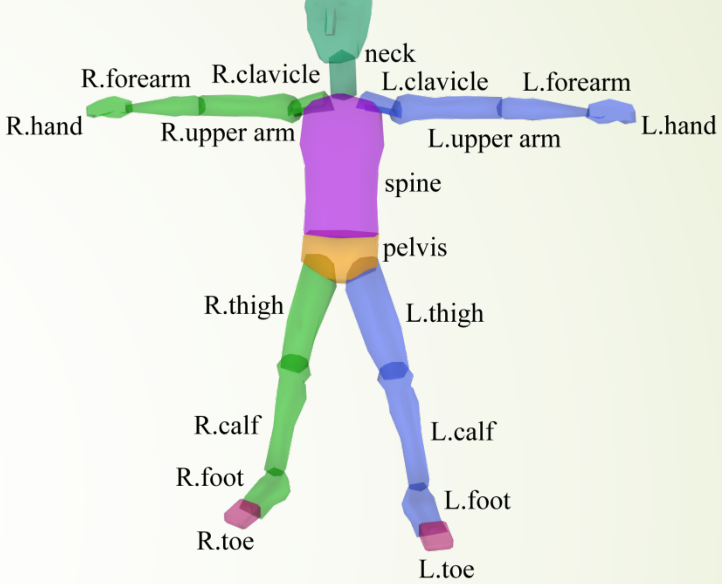
- Skinning is the process of attaching a skin (mesh) to an underlying articulated skeleton
  - skeletal subspace deformation (SSD)
    - A.K.A smooth skinning algorithm, blended skinning, multi-matrix skinning, linear blend skinning,, and sometimes just skinning

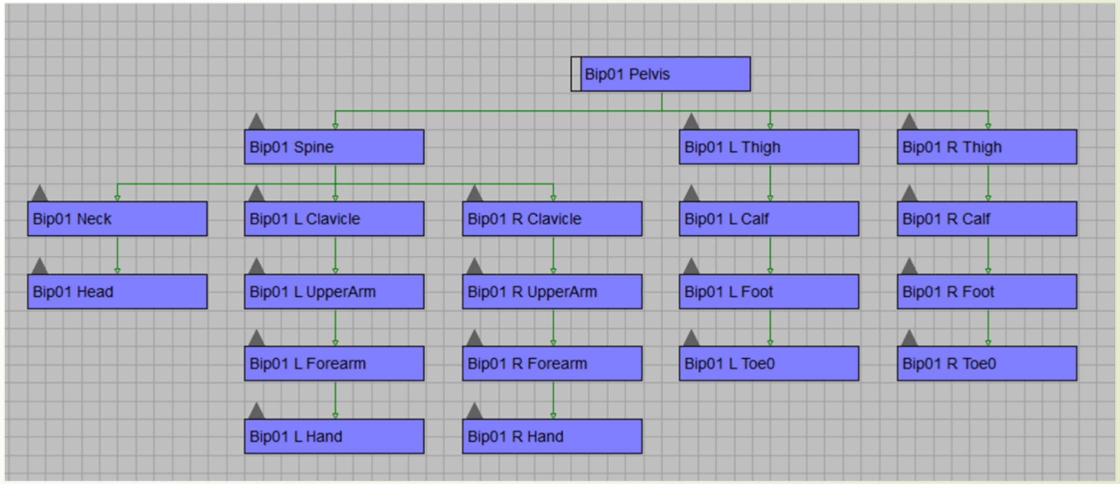


Skeleton Hierarchy

Hierarchy saves your live

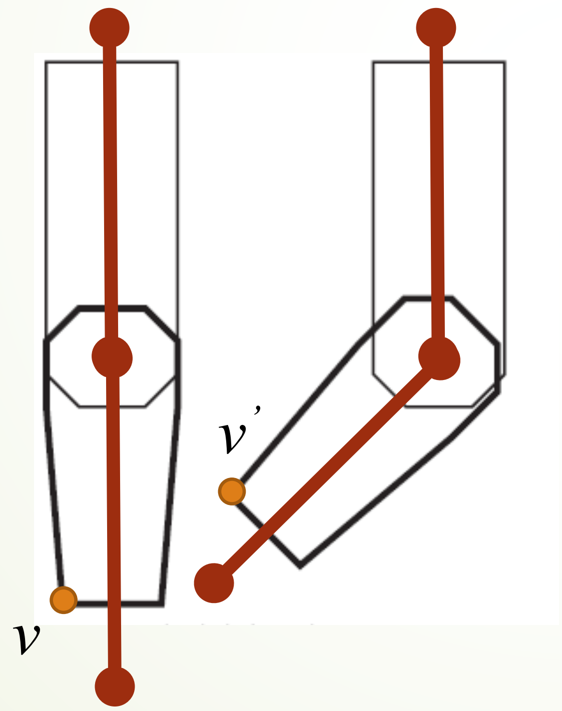
- The bones are connected at joints, which allow the rigid skeleton to animate in an articulated fashion

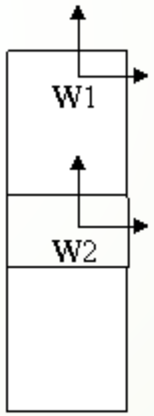




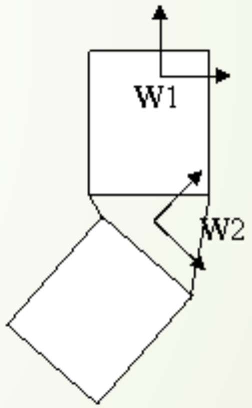
Rigid Skinning

- Rendering characters as rigid components, e.g. as a robot
  - $v' = \mathbf{W}v$, where $\mathbf{W}$ is world transform of the lower bone



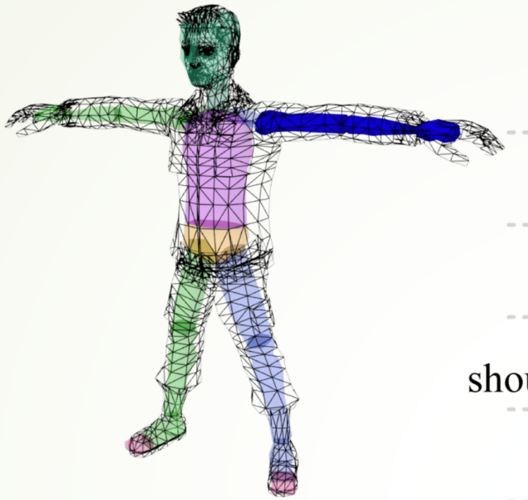


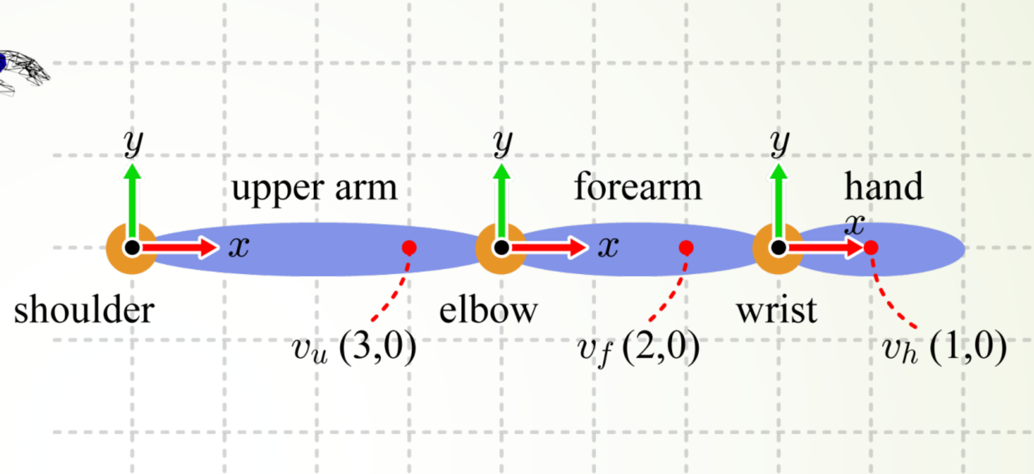
An unbent knee with skin attached to joints 1 and 2



Every vertex is attached to exactly one joint, so as the knee bends, we get some distortion

Space Change between Bones





- When the forearm moves, for example, v_f has to move accordingly. It is simply achieved if v_f is defined in the forearm's object space.
- Every world-space vertex of the default pose needs to be transformed into the object space of the bone (which we call *bone space* henceforth). For example, v_f will be transformed into the forearm's bone space so as to have the coordinates (2,0).

Space Change between Bones (cont'd)

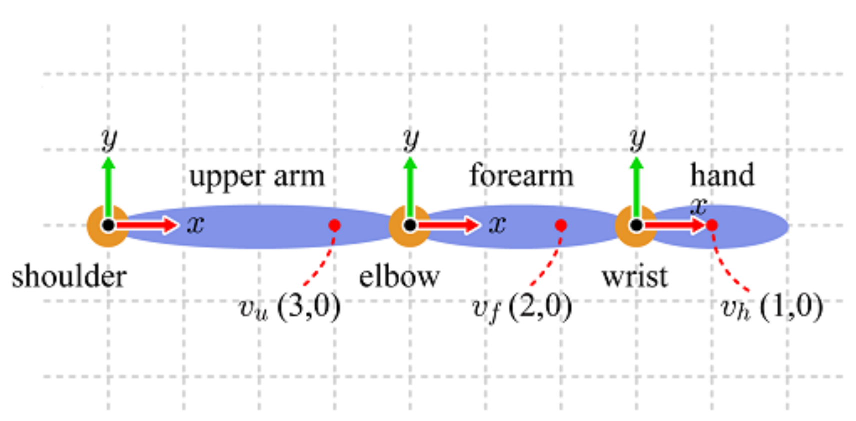
- Let us consider the opposite direction first
  - i.e., from the bone space to the world space
  - Given a bone-to-world transform, its inverse can convert a world-space vertex into the bone space

- Ex1: to-parent transform of the forearm, which transforms a forearm vertex to the space of its parent

$$M_{f,p}v_f = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \\ 1 \end{pmatrix}$$

Note this is in 3x3 homogenous coordinates

- Ex2: to-parent matrix of the hand.



$$v'_h = M_{h,p}v_h = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$M_{f,p}v'_h = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 1 \end{pmatrix}$$

$$M_{f,p}v'_h = M_{f,p}M_{h,p}v_h = \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 0 \\ 1 \end{pmatrix}$$

Bone Space to World Space

- The root node (pelvis) is associated with a transform used to position and orient it in the world space for the default pose. Let us denote the world matrix by $M_{1,d}$.

- The spine's world transform

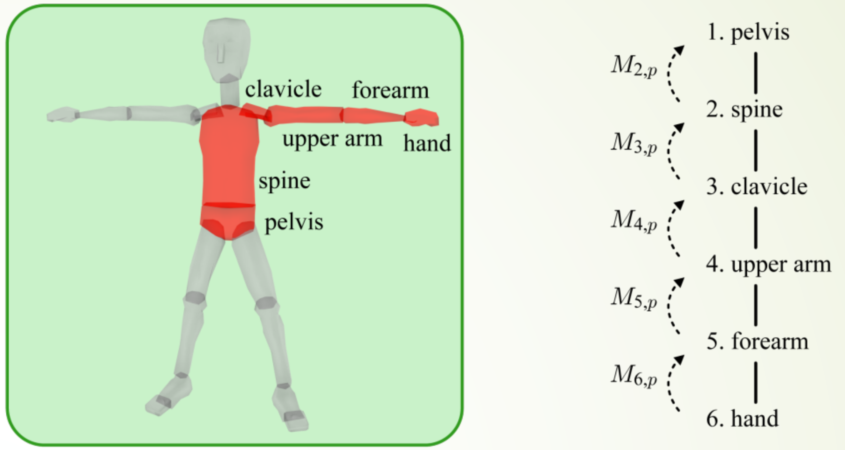
$$M_{2,d} = M_{1,d}M_{2,p}$$

- The clavicle's world transform

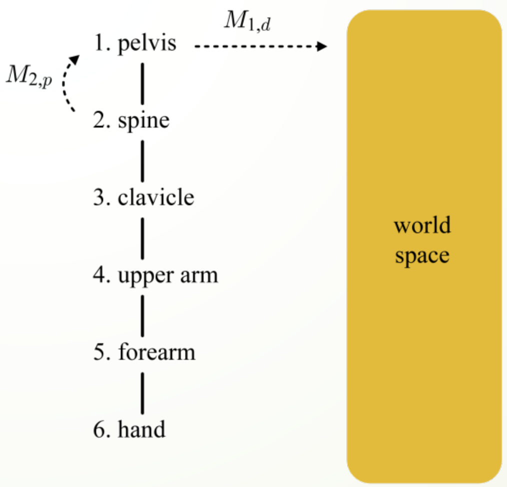
$$M_{3,d} = M_{1,d}M_{2,p}M_{3,p}$$
$$= M_{2,d}M_{3,p}$$

- Let's generalize

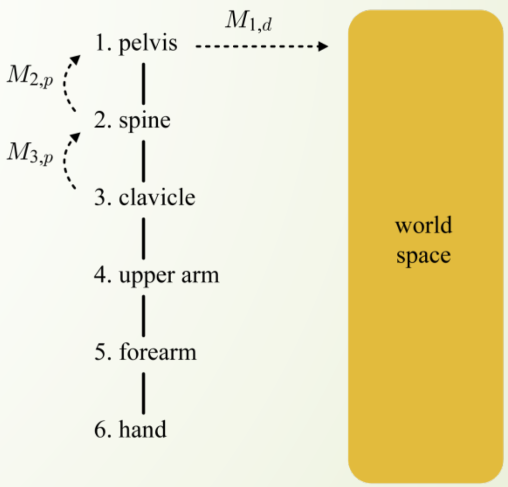
$$M_{i,d} = M_{i-1,d}M_{i,p}$$



(a) To-parent transforms



(b) From spine to the world



(c) From clavicle to the world

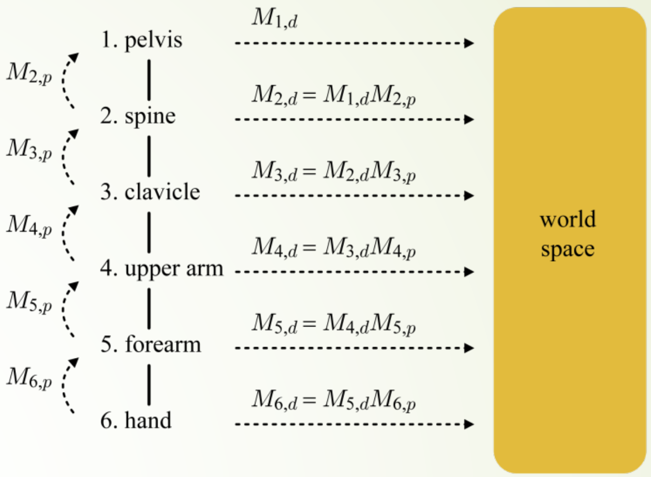
World Space to Bone Space

- So far, we have considered the world transform from the bone space to the world space. However, what is needed in an articulated-body animation is its inverse.

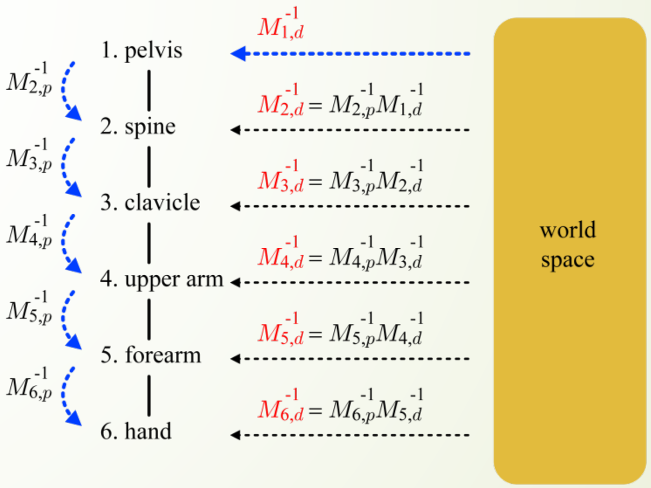
$$M_{i,d} = M_{i-1,d}M_{i,p}$$

$$M_{i,d}^{-1} = M_{i,p}^{-1}M_{i-1,d}^{-1}$$

- Once the default pose is fixed, the inverse world transforms can be computed for all bones
  - In the default pose, $M_{i,p}^{-1}$ can be immediately obtained
  - Computing $M_{i,d}^{-1}$ requires $M_{i-1,d}^{-1}$ to be computed in advance, and therefore the skeleton hierarchy is travdersed top down



(d) World transforms for all bones

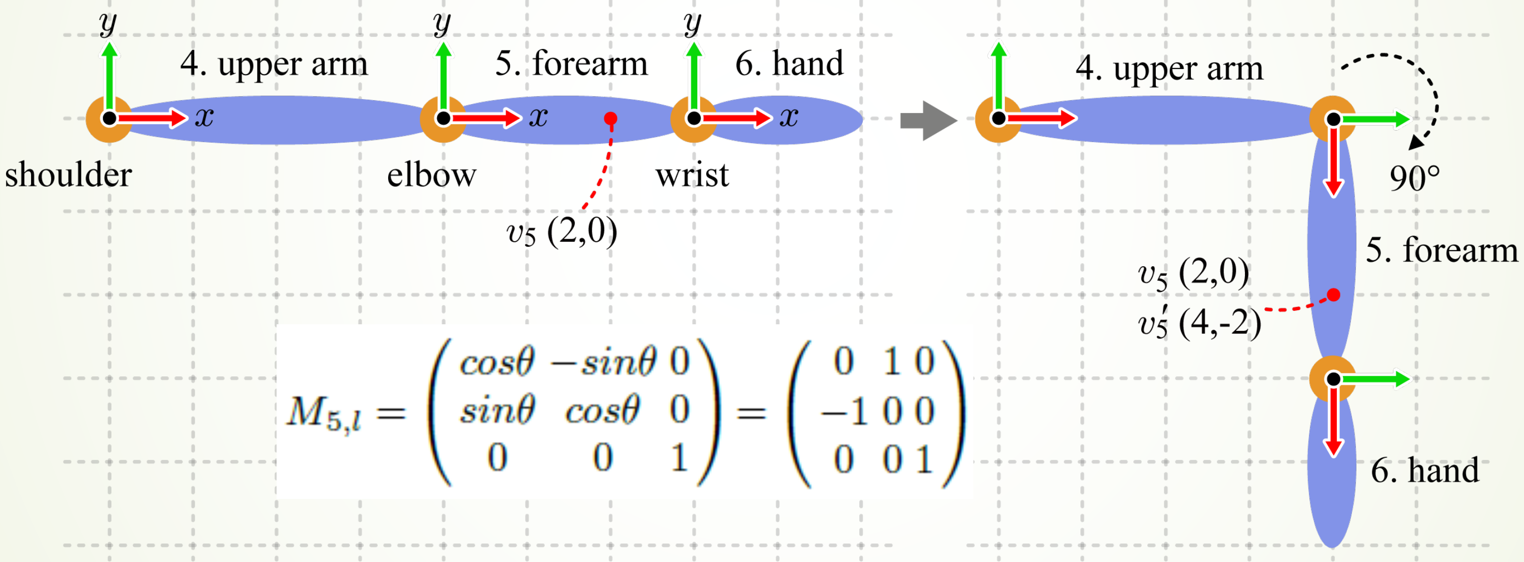


(e) Inverse world transforms for all bones

Forward Kinematics

- The inverse of the world transform converts a world-space vertex “in the binding pose” to the *i*-th bone's space

- Now the *i*-th bone is animated. Then, the vertices belonging to the bone are accordingly animated. The animation is often called local transform

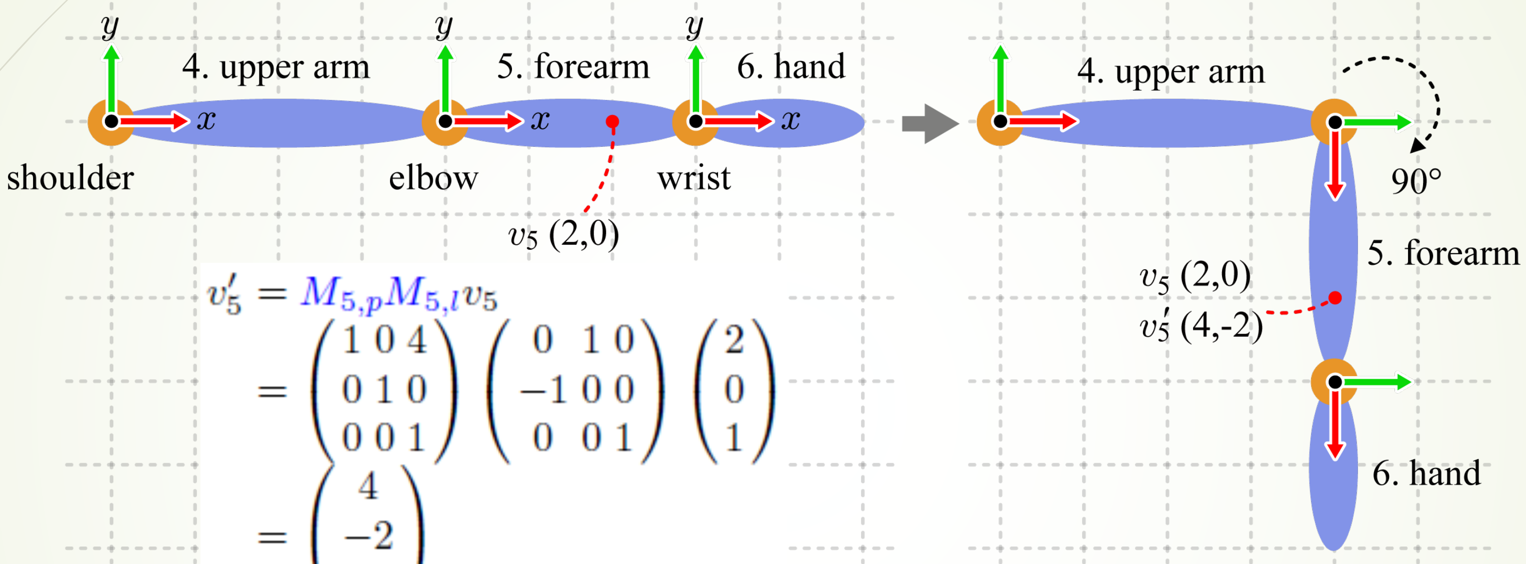


$$M_{5,l} = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- For rendering, the animated vertices should be transformed back to the world space. (Then, they will be transformed to the camera space and so forth, along the pipeline.) Let us call the transform matrix $M_{5,w}$.

Forward Kinematics (cont'd)

- As the first step for computing the world-space position of "animated v_5," let us find its coordinates in the upper arm's space.



$$
\begin{aligned}
v_5' &= M_{5,p}M_{5,l}v_5 \\
&= \begin{pmatrix} 1 & 0 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}
\end{aligned}
$$

- The upper arm can also be animated. Let $M_{4,w}$ denote the matrix that transforms the animated vertices of the upper arm into the world space. Then, $M_{5,w}$ for transforming "animated v_5" into the world space is defined.

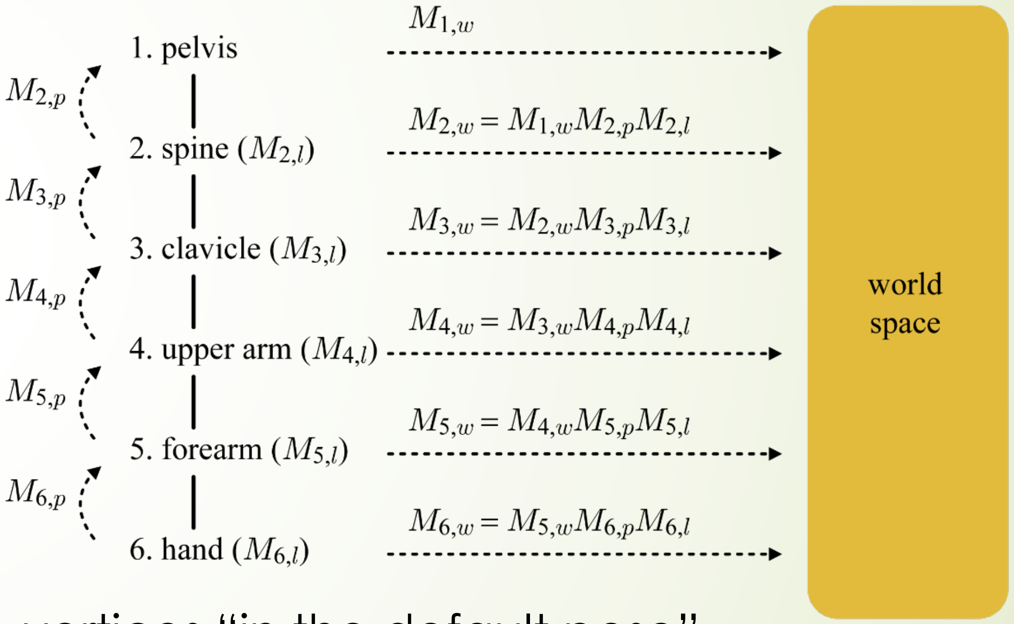
$$M_{5,w} = M_{4,w}M_{5,p}M_{5,l}$$

- Let's generalize.

$$M_{i,w} = M_{i-1,w}M_{i,p}M_{i,l}$$

Forward Kinematics (cont'd)

- When the artist defines the animated pose of the *i*-th bone, $M_{i,l}$ is obtained
- $M_{i,p}$ was obtained from the default pose.
- So, computing $M_{i,w}$ simply requires $M_{i-1,w}$ to be computed in advance.
- $M_{1,w}$ representing the pose of the animated pelvis is defined by the artist.
- We can compute the world transforms of all bones "in the animated pose" also in the top-down fashion.

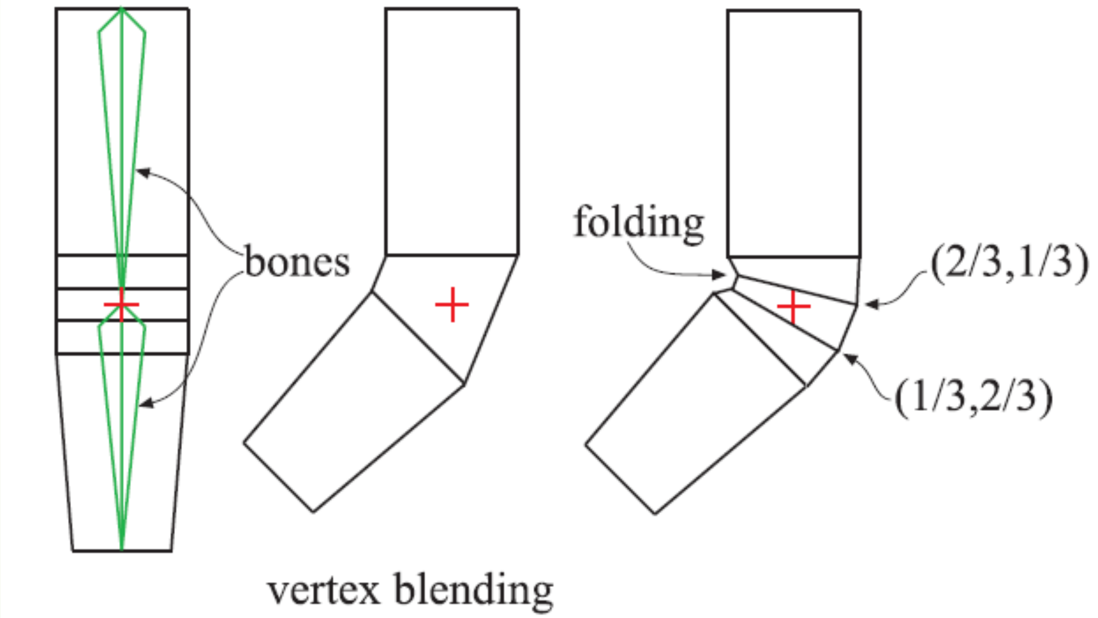


- When v_d and v_w denote the world-space vertices "in the default pose" and "in the animated pose," respectively, we have the following relation:

$$v_w = M_{i,w}M_{i,d}^{-1}v_d$$

Smooth Skinning

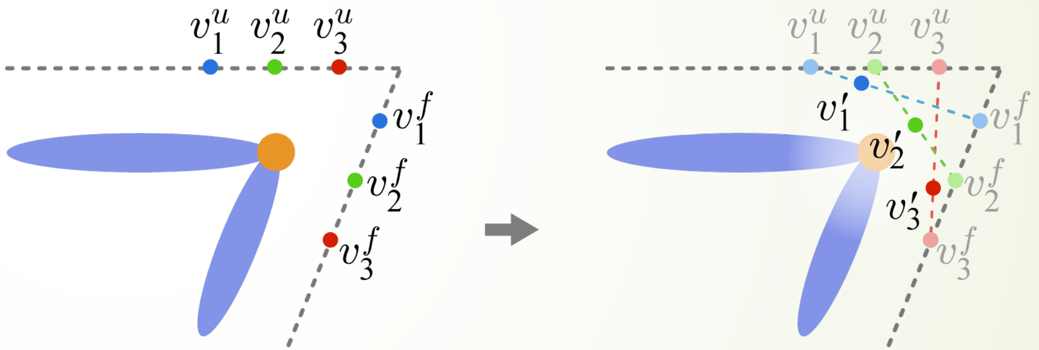
- Skinning is the process of attaching a skin (mesh) to an underlying articulated skeleton
  - skeletal subspace deformation (SSD)
    - A.K.A smooth skinning algorithm, blended skinning, multi-matrix skinning, linear blend skinning,, and sometimes just skinning



Smooth Skinning

- In smooth skinning, v_2 is transformed not only by $M_{f,w}M_{f,d}^{-1}$ but also by $M_{u,w}M_{u,d}^{-1}$. Then, the transformed vertices are interpolated using the predefined weights. The same applies to v_1 and v_3.

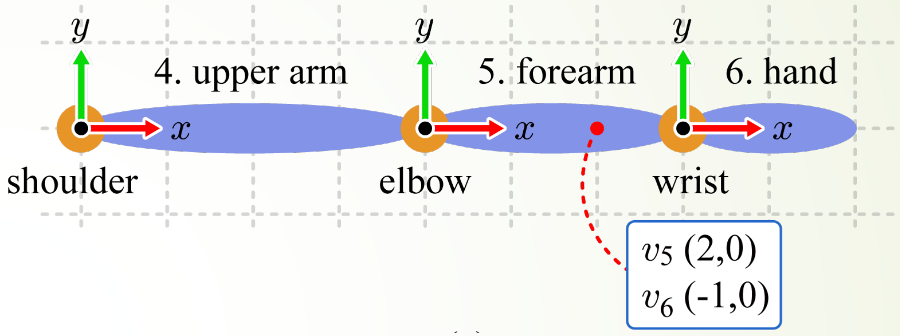
	upper arm	forearm
v_1	0.7	0.3
v_2	0.5	0.5
v_3	0.3	0.7



Smooth Skinning Algorithm

- Suppose that w_5 and w_6 are equal
  - They are the same point defined in different sub-spaces

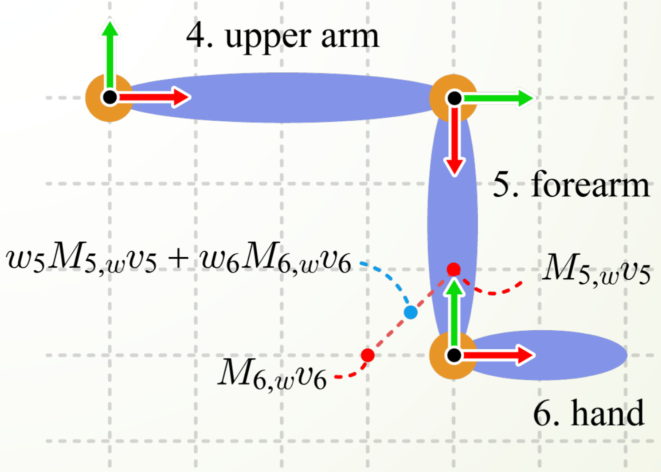
$$v_5 = M_{5,d}^{-1} v_d$$
$$v_6 = M_{6,d}^{-1} v_d$$



$$w_5 M_{5,w} v_5 + w_6 M_{6,w} v_6$$

$$w_5 M_{5,w} M_{5,d}^{-1} v_d + w_6 M_{6,w} M_{6,d}^{-1} v_d$$

$$\boxed{v_w = \sum_{i=1}^{n} w_i M_{i,w} M_{i,d}^{-1} v_d}$$



Smooth Skinning

- Given a mesh vertex v and a list of weights w_i for each bone, the position of v is defined as
  - $v' = \sum w_i \mathbf{W}_i \mathbf{B}_i^{-1} v$
- $\mathbf{B}_i$ is matrix at binding pose for bone i (a.k.a. binding matrix)
- $\mathbf{W}_i$ is world matrix for bone i at a given pose (specified by animator)
- How is w_i determined?!
  - Again, similar to FFD, $\sum w_i = 1$
- Why does most binding post look like this: