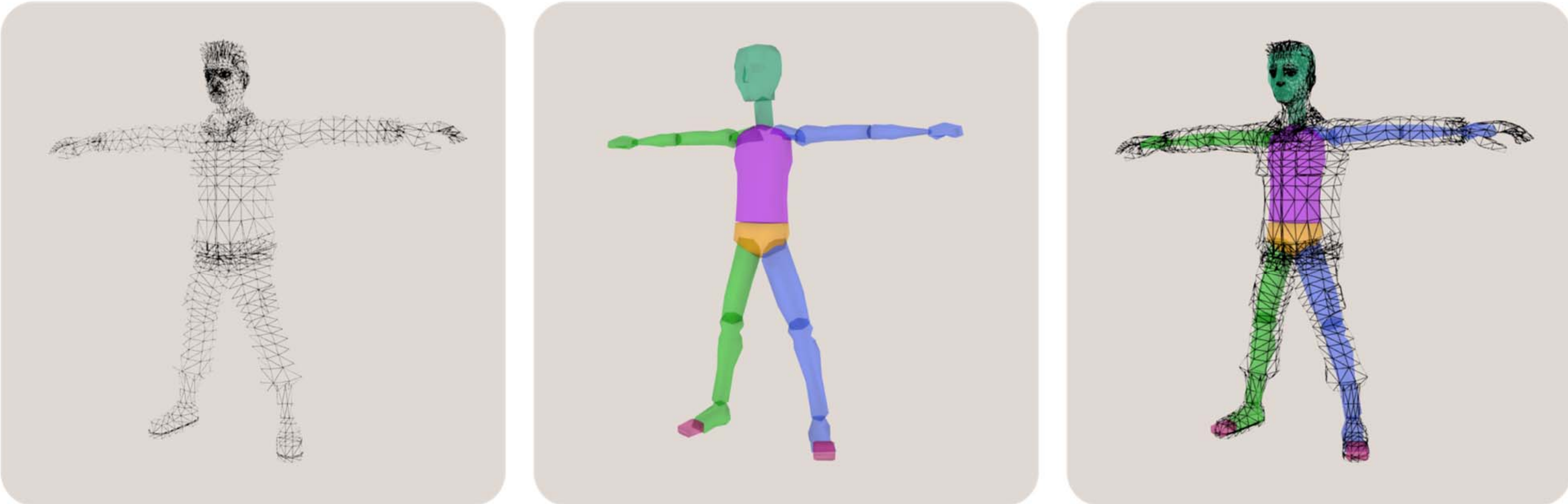
Chapter XIII
Character Animation

Skeleton

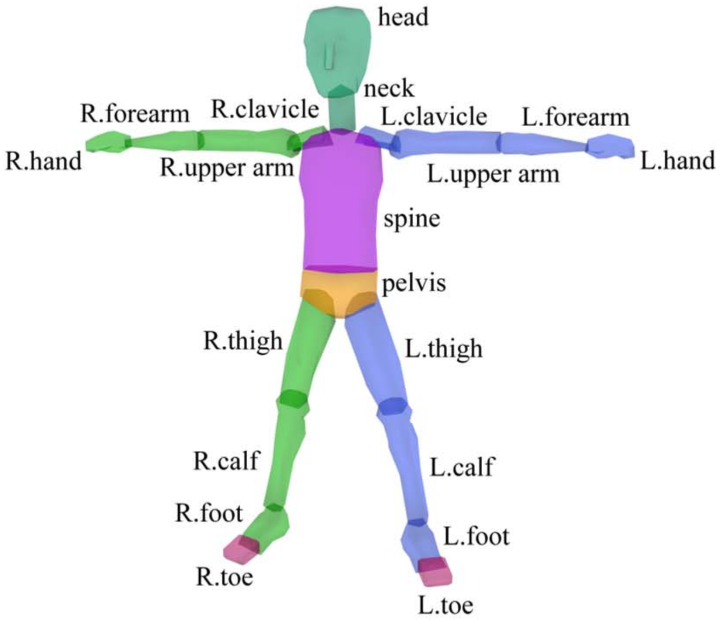
- A popular method for animating a character is to use a *skeleton*, which is composed of bones. The skeleton is embedded into the polygon mesh. When the skeleton is animated, the vertices of the polygon mesh will be accordingly animated.

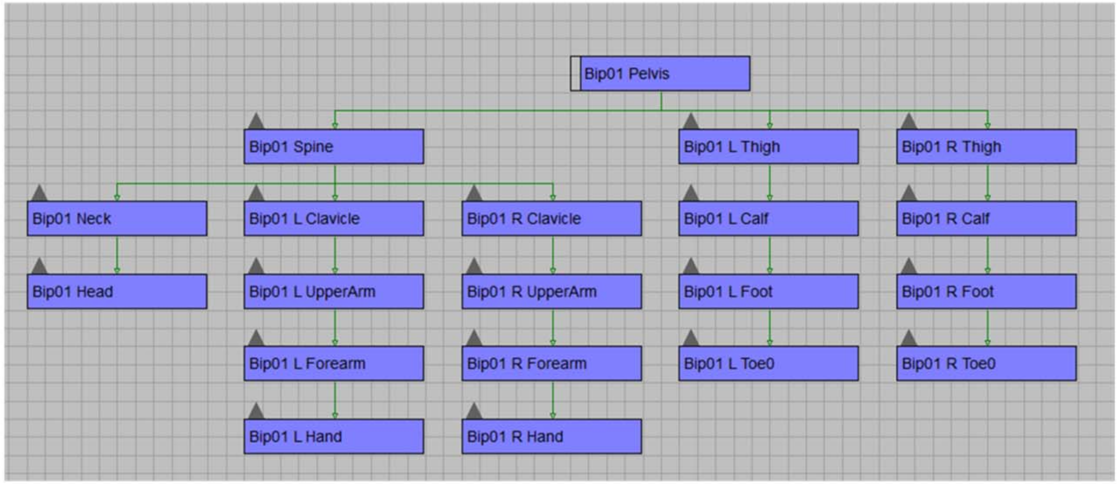


(a) (b) (c)

- The initial pose of the character is also called default pose, rest pose, dress pose, bind pose, etc.

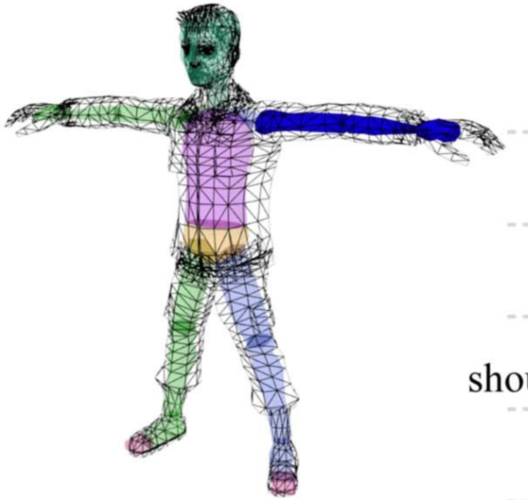
Skeleton (cont'd)

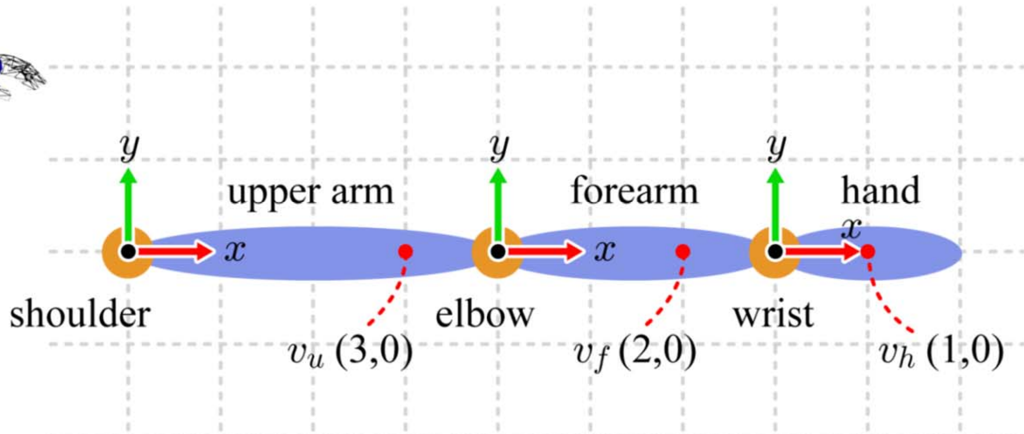




Space Change between Bones

- The bones are connected at joints, which allow the rigid skeleton to animate in an articulated fashion. See the example, which consists of three bones (upper arm, forearm, and hand) and three joints (shoulder, elbow, and wrist).

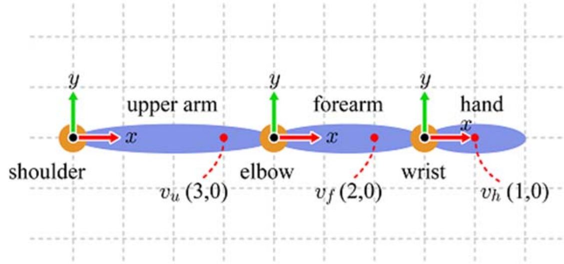




- When the forearm moves, for example, v_f has to move accordingly. It is simply achieved if v_f is defined in the forearm's object space.
- Initially, all vertices of the default pose are defined in the object space of the character. To avoid confusion with the object space of a bone, let's simplay call the character's object space *character space*.
- Every character-space vertex of the default pose needs to be transformed into the object space of the bone (which we call *bone space* henceforth) it belongs to. 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)

- For now, consider the opposite direction, i.e., from the bone space to the character space. If we can compute such a bone-to-character transform, its inverse can be used to convert a character-space vertex into the bone space.
- Consider the *to-parent transform* of the forearm, which transforms a forearm vertex to the space of its parent.



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

- Now consider the to-parent matrix of the hand.

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

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

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

- Given a vertex at a bone, we can concatenate the to-parent matrices so as to transform the vertex into the bone space of any ancestor in the skeleton.

Character Space to Bone Space

- The to-parent matrix is denoted by $M_{i,p}$.
- Let us denote the transform from the bone space to the character space by $M_{i,d}$, where d stands for default pose.
- Assume that $M_{1,d}=I$.
- The spine's transform to the character space:

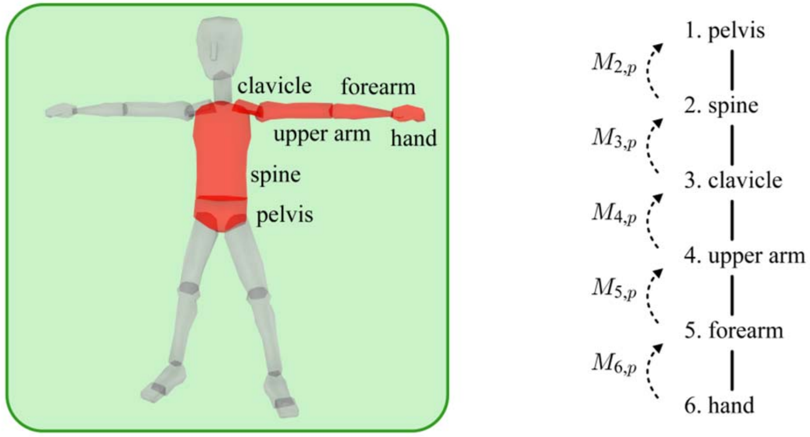
$$M_{2,d} = M_{1,d}M_{2,p}$$

- The clavicle's transform to the character space:

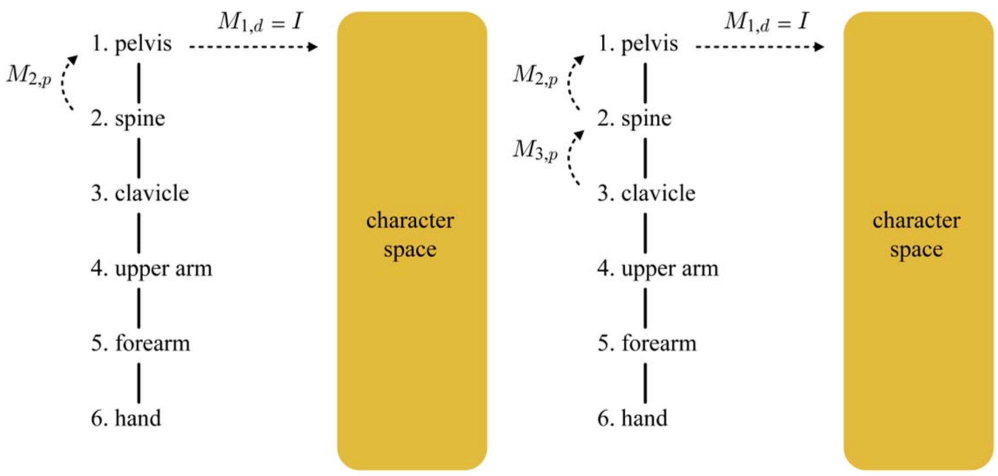
$$\begin{aligned} M_{3,d} &= M_{1,d}M_{2,p}M_{3,p} \\ &= M_{2,d}M_{3,p} \end{aligned}$$

- Let's generalize

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



(a) To-parent transforms



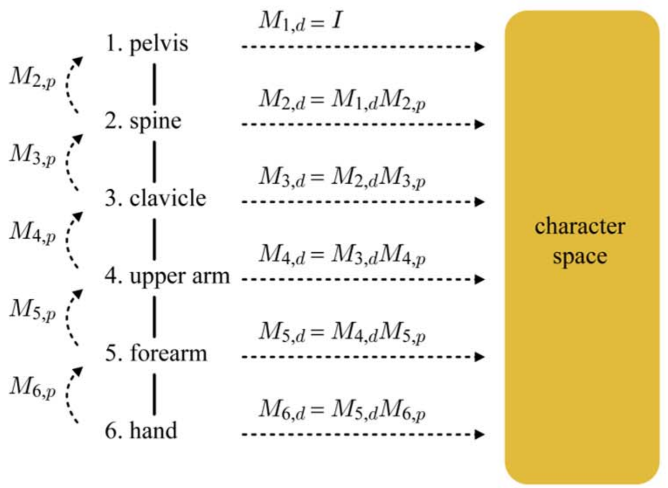
(b) From spine to the character space (c) From clavicle to the character space

Character Space to Bone Space (cont'd)

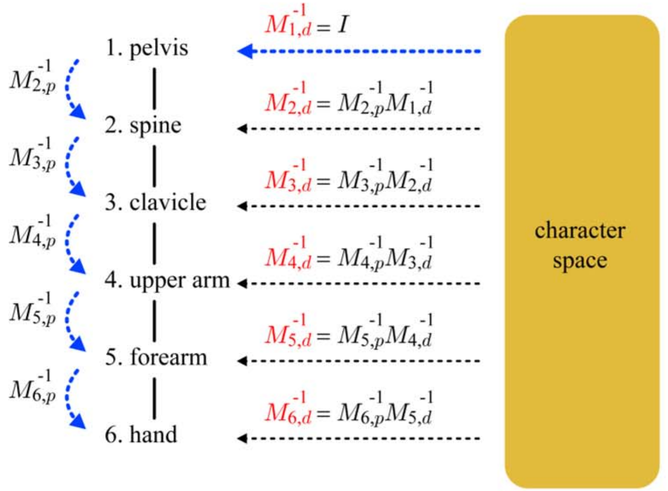
- So far, we have considered the transform from the bone space to the character space. However, what is needed in an articulated-body animation is its inverse.

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

- Once the default pose is fixed, such inverse 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 traversed top down.



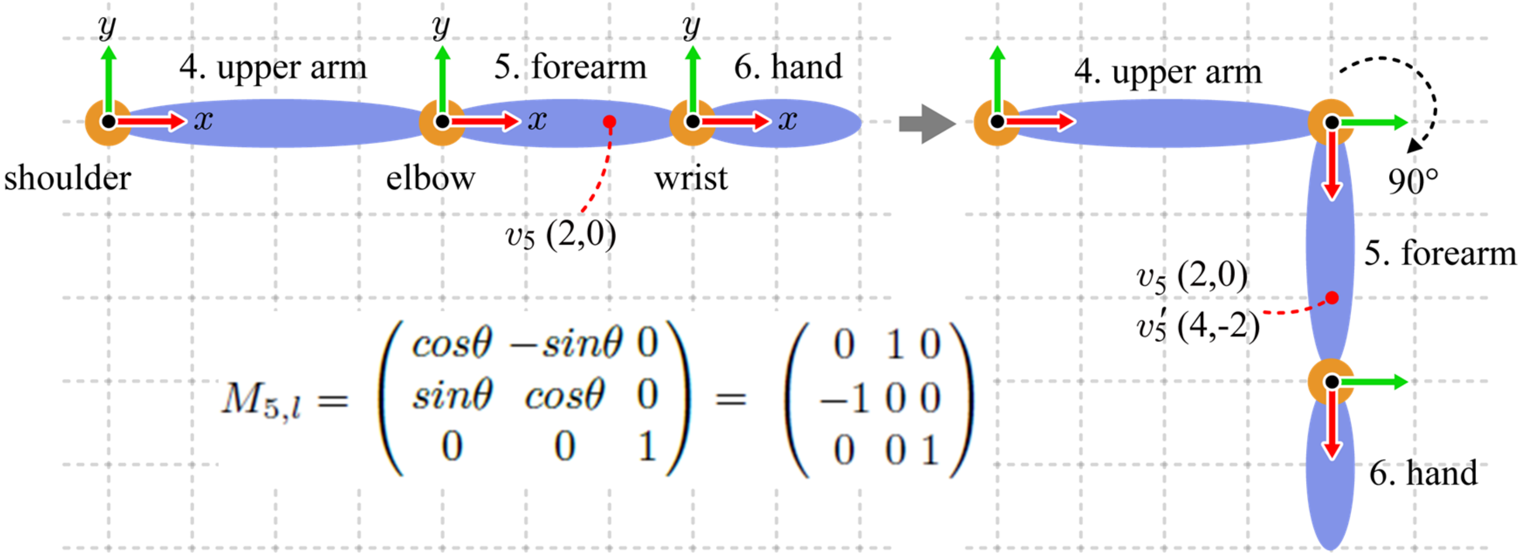
(d) All bones' transforms to the character space



(e) Inverse transforms

Forward Kinematics

- We have identified the transform that converts a character-space vertex “in the default 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.

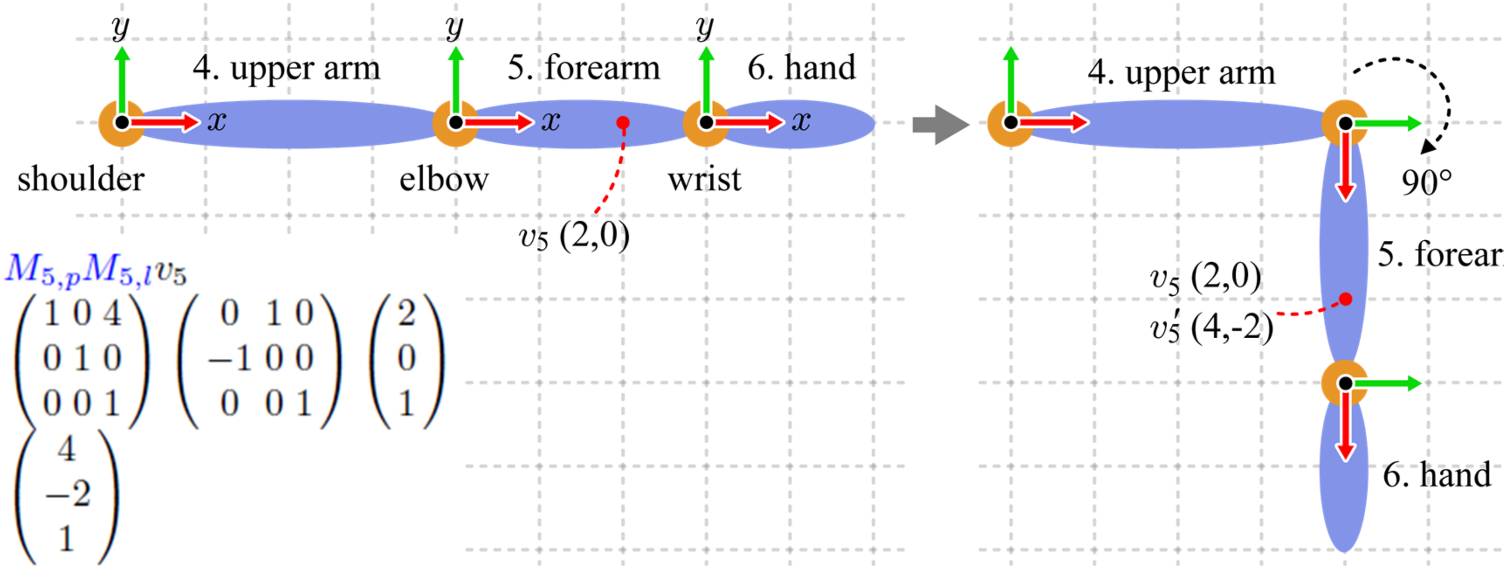


$$M_{5,l} = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- For rendering, the animated vertices should be transformed back to the character space. (Then, they will be transformed to the world space, camera space and so forth, along the pipeline.)
- Let us compute the matrix for animating v_5 and transforming it to the character space. We name the matrix $M_{5,g}$.

Forward Kinematics (cont'd)

- As the first step for computing $M_{5,a}$, let us find its coordinates in the upper arm's space.



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

- The upper arm can also be animated. Let $M_{4,a}$ denote the matrix that animates a vertex of the upper arm and converts it into the character space. Then, $M_{5,a}$ is defined as follows.

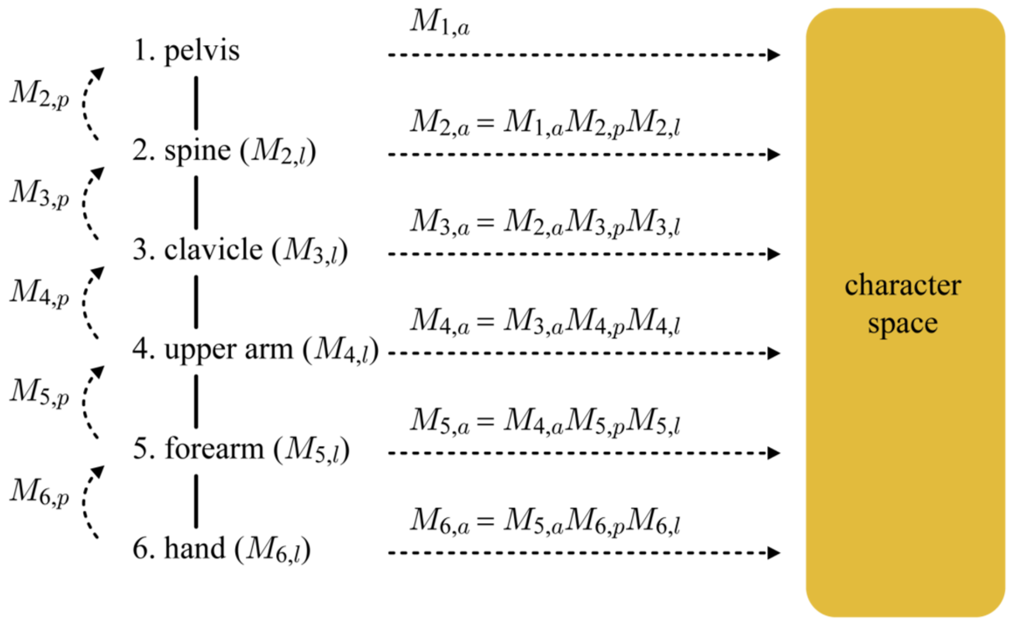
$$M_{5,a} = M_{4,a}M_{5,p}M_{5,l}$$

- Let's generalize.

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

Forward Kinematics (cont'd)

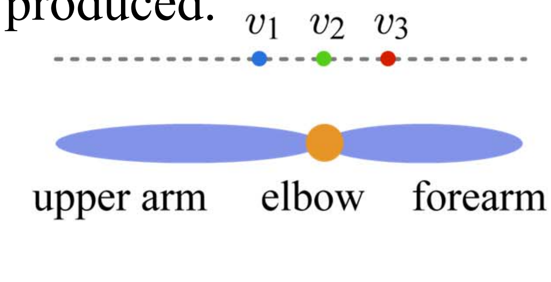
- When the artist defines the animated pose of the i-th bone, $M_{i,l}$ is obtained.
- $M_{i,p}$ was obtained from the default pose.
- So, computing $M_{i,a}$ simply requires $M_{i\text{-}1,a}$ to be computed in advance.
- $M_{1,a}$ representing the pose of the animated pelvis is defined by the artist.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



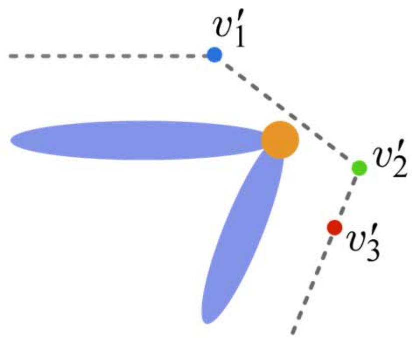
- When v_d and v_a denote the character-space vertices "in the default pose" and "in the animated pose," respectively, we have the relation: $v_a = M_{i,a}M_{i,d}^{-1}v_d$

Skinning

- When v_2 belongs to the forearm, $M_{f,a}M_{f,d}^{-1}$ produces v_2 ´. Similarly, v_1 ´and v_3 ´ can be produced.



(a)

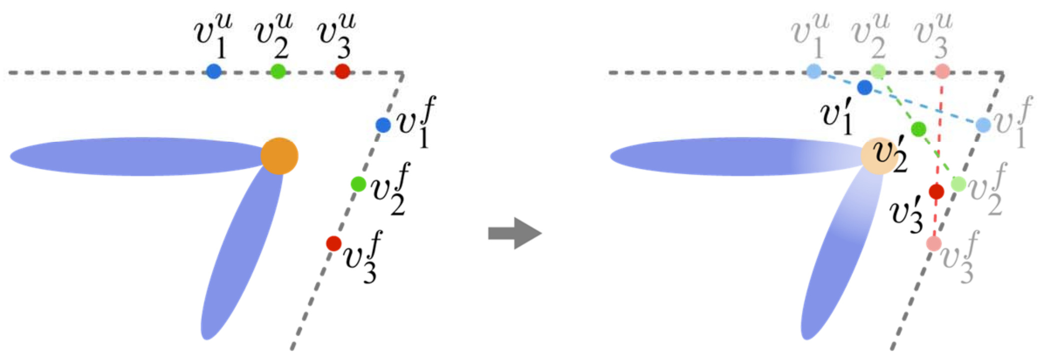


(b)

- In skinning, v_2 is transformed not only by $M_{f,a}M_{f,d}^{-1}$ but also by $M_{u,a}M_{u,d}^{-1}$. 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

(c)

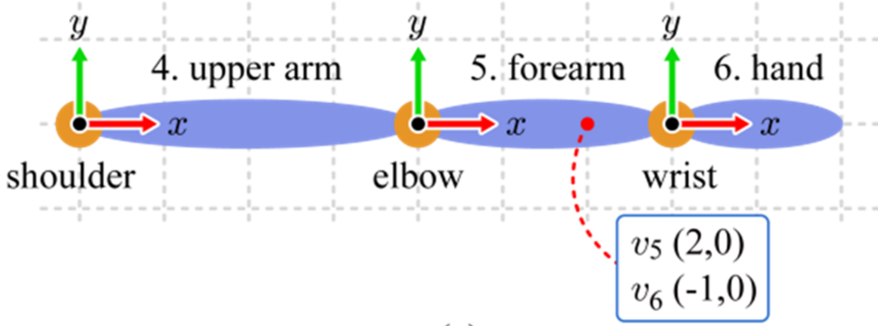


(d)

Skinning (cont'd)

- Suppose that w_5 and w_6 are equal.

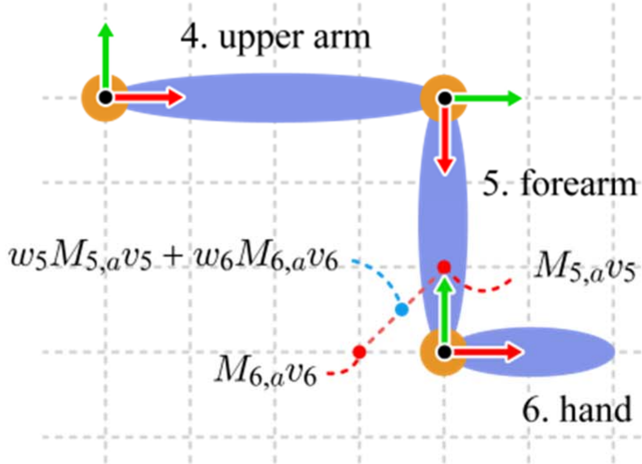
$$v_5 = M_{5,d}^{-1} v_d$$
$$v_6 = M_{6,d}^{-1} v_d$$



(a)

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

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



(b)

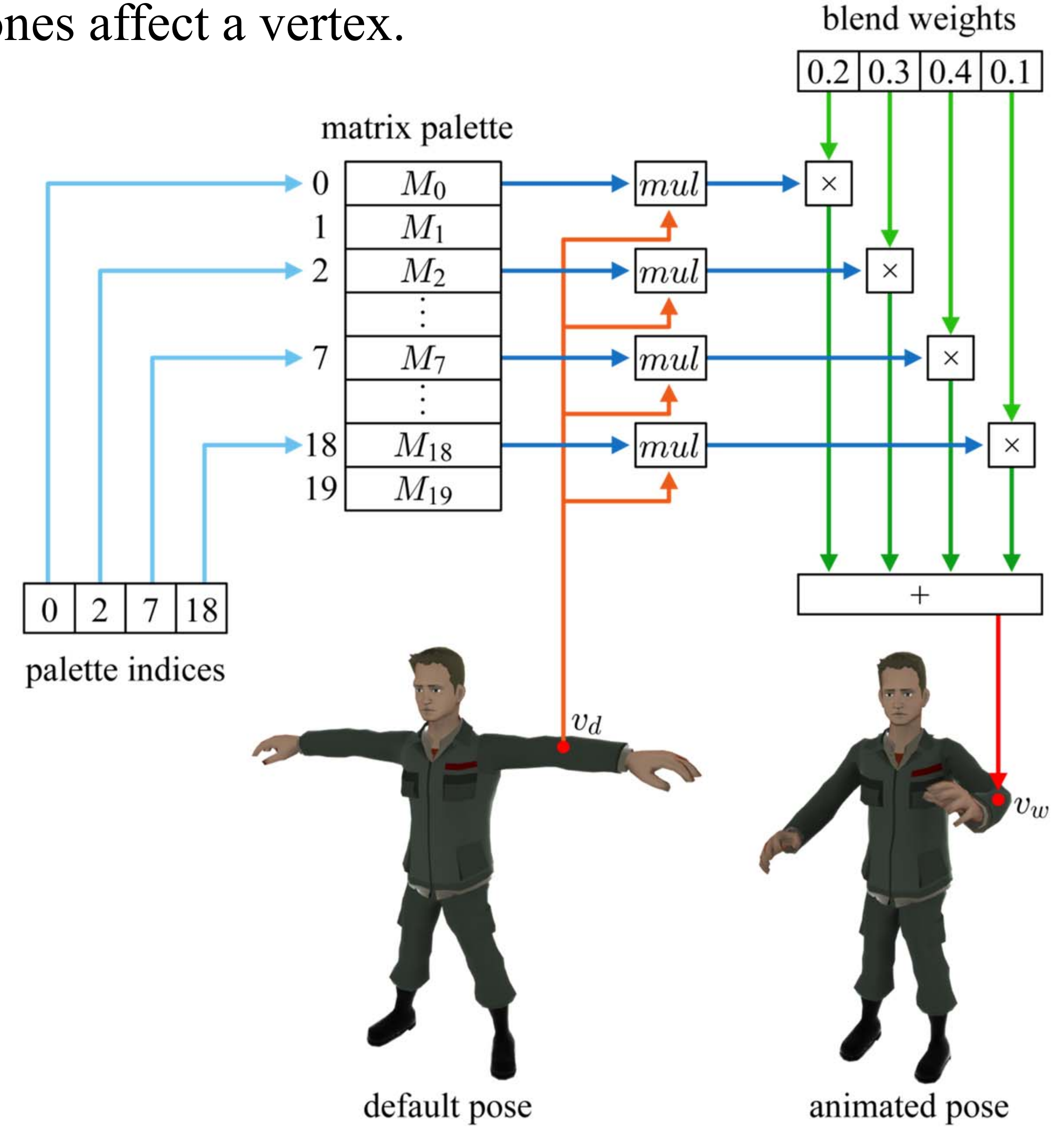
$$v_a = \sum_{i=1}^{m} w_i M_{i,a} M_{i,d}^{-1} v_d$$

$$v_a = \sum_{i=1}^{m} w_i M_i v_d$$

Skinning (cont'd)

- In general, at most 4 bones affect a vertex.

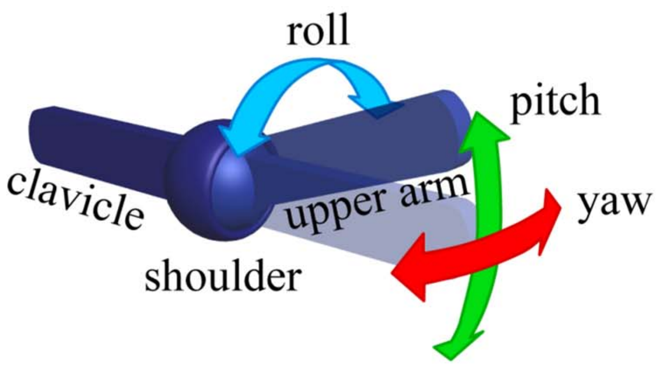
$$v_a = \sum_{i=1}^{m} w_i M_i v_d$$



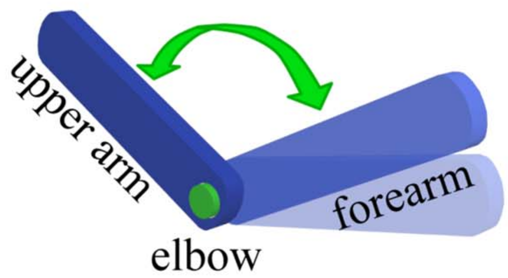
Inverse Kinematics

- In robotics, an end effector refers to the device located at the end of a robotic arm.
- Forward kinematics computes the pose of the end effector as a function of the joint angles of the robotic arm.
- The reverse process is called inverse kinematics (IK). Given the desired pose of the end effector along with the initial pose of the robotic arm, the joint angles of the final pose are calculated.

- The number of independent variables defining the state of an object is called the degrees of freedom (DOF).

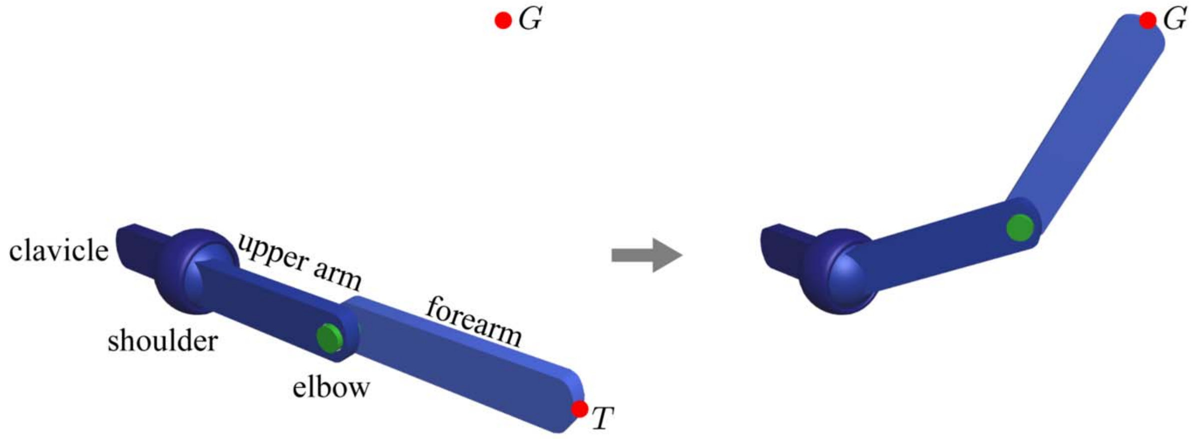


(a)



(b)

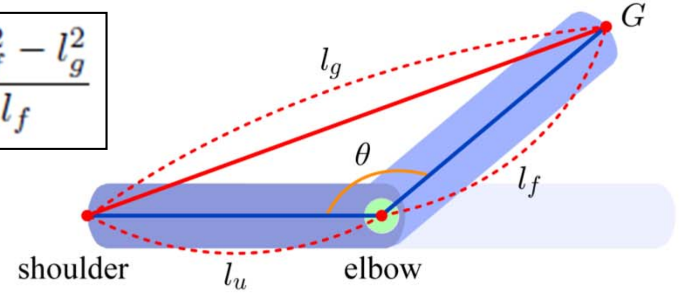
Inverse Kinematics - Analytic Solution



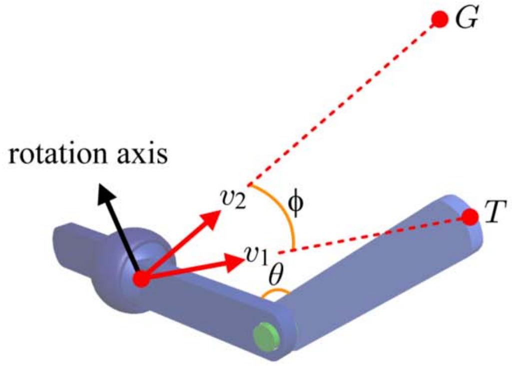
(a)

$$\boxed{l_g^2 = l_u^2 + l_f^2 - 2l_u l_f cos\theta}$$

$$\boxed{\theta = acos\frac{l_u^2 + l_f^2 - l_g^2}{2l_u l_f}}$$

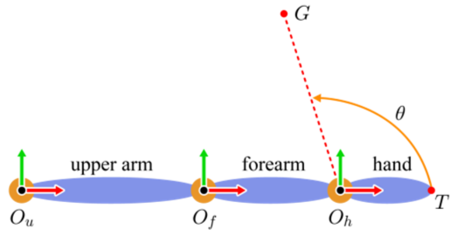


(b)

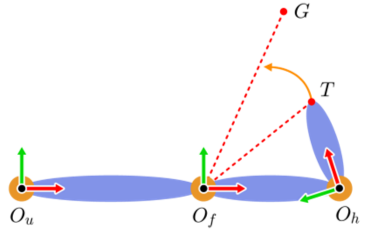


(c)

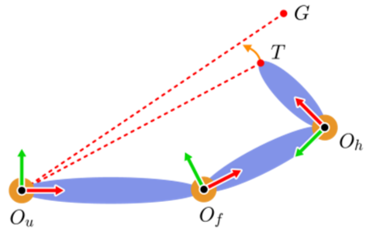
Inverse Kinematics - Cyclic Coordinate Descent



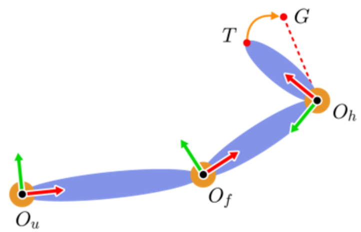
(a) Initial pose



(b) The hand has been rotated, and now it is the forearm's turn.



(c) The forearm has been rotated, and now it is the upper arm's turn.



(d) The upper arm has been rotated, and the next iteration starts.

Inverse Kinematics - Example