
Computer Graphics

10 - Character Animation

Yoonsang Lee
Hanyang University

Spring 2025

Outline

- Computer-Generated Character Animation
- Skeletal Animation
- Forward Kinematics
- Creating Character Animation
- BVH File Format

Computer-Generated Animation

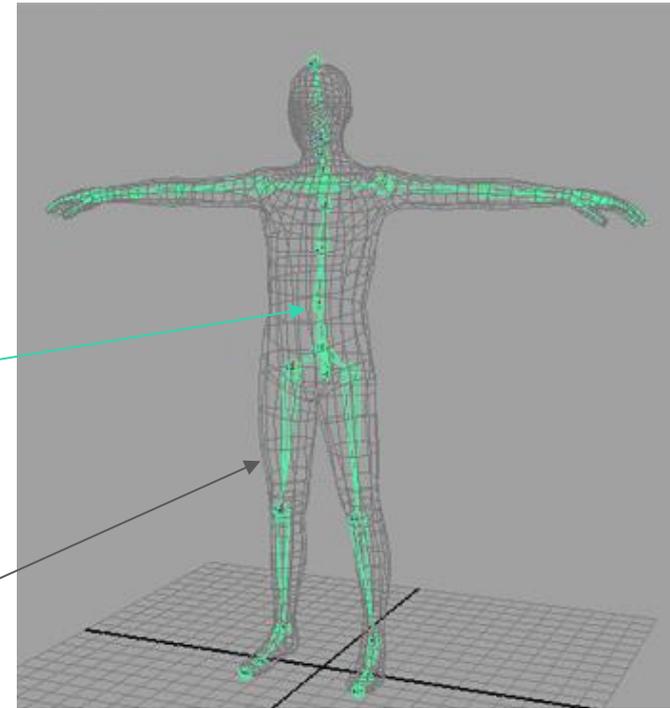
- Generating images by "rendering" 3D objects.
 - Modeling 3D objects or characters and giving them "motion".
 - Although digitized hand-drawn 2D animation is still used, much of animation today is created in 3D using this method.



Encanto, 2021

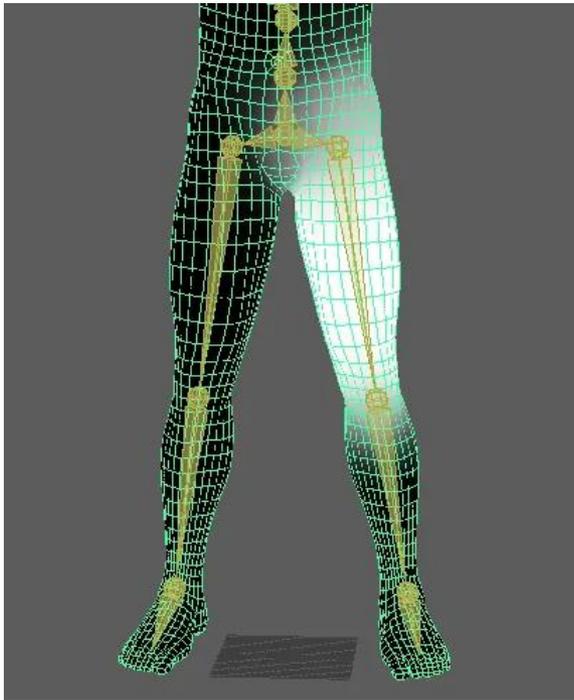
Computer-Generated Character Animation

- For character animation,
- *Skeletal animation* is the most popular, which consists of:
 - *Skeleton*: Bone or joint hierarchy and related data
 - *Motion*: Joint movement data
 - *Skin (or mesh)*: Surface representation for rendering the character

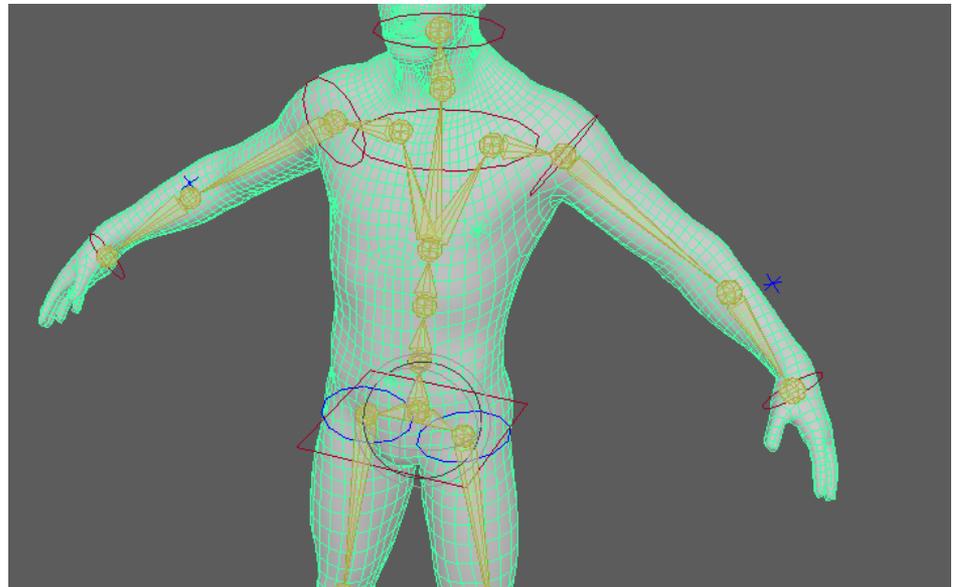


Visualizing Character in Skeletal Animation

- *Skin* can be deformable mesh(es)
 - *Skinning*: "Embedding" a skeleton into skin mesh (Details will be skipped).



A human thigh with simple **skin weights** painted onto it

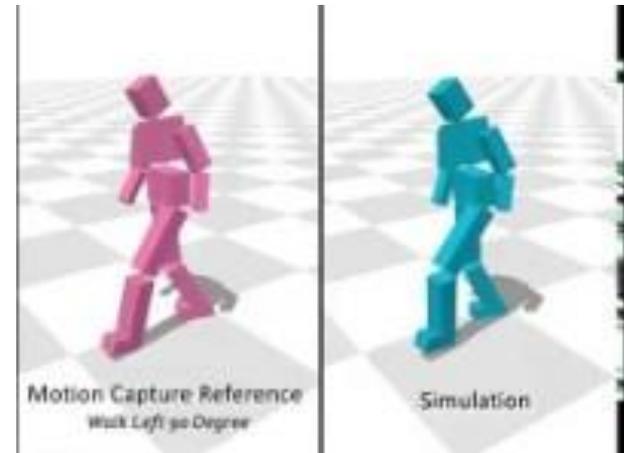


Visualizing Character in Skeletal Animation

- Characters can also be visualized in a simpler way using individual rigid meshes.



<https://youtu.be/PEhyWI8LGBY>



https://youtu.be/hpeqxc_1vwo

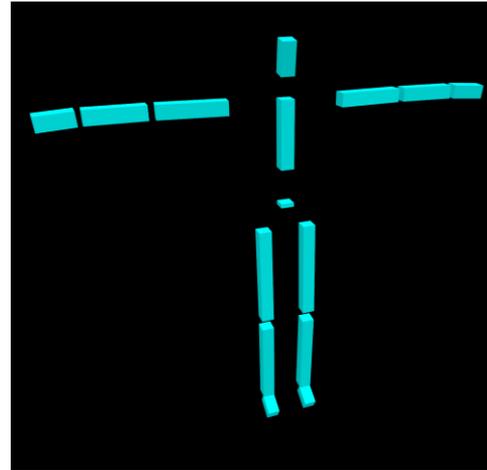
- Today's lecture focuses on "hierarchical structure" and "motion" in skeletal animation, not visualizing a character.

Skeletal Animation

Skeletal Animation

- Skeletal animation consists of two parts:

- "*Skeleton*": static data

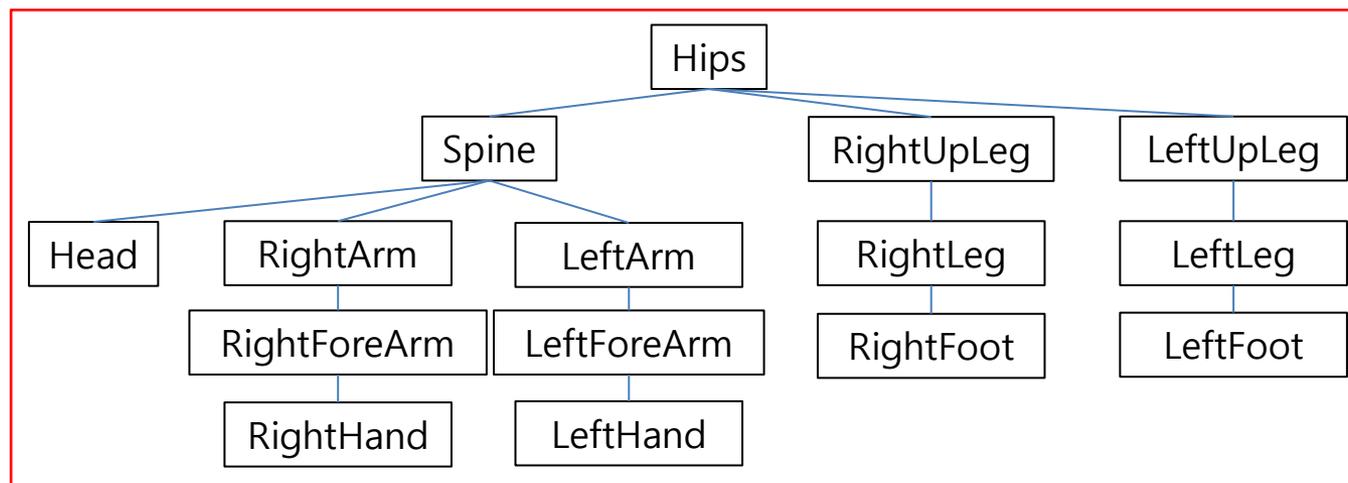
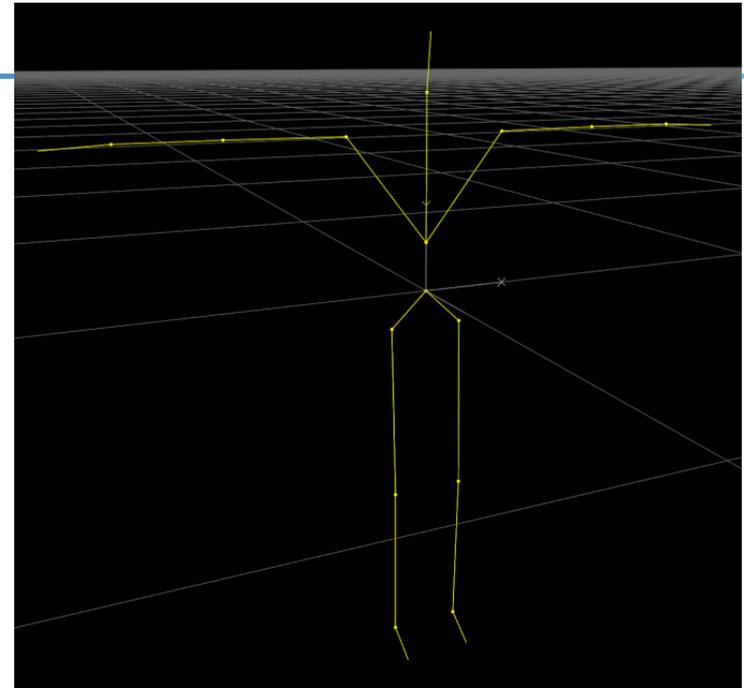


- "*Motion*": time-varying data



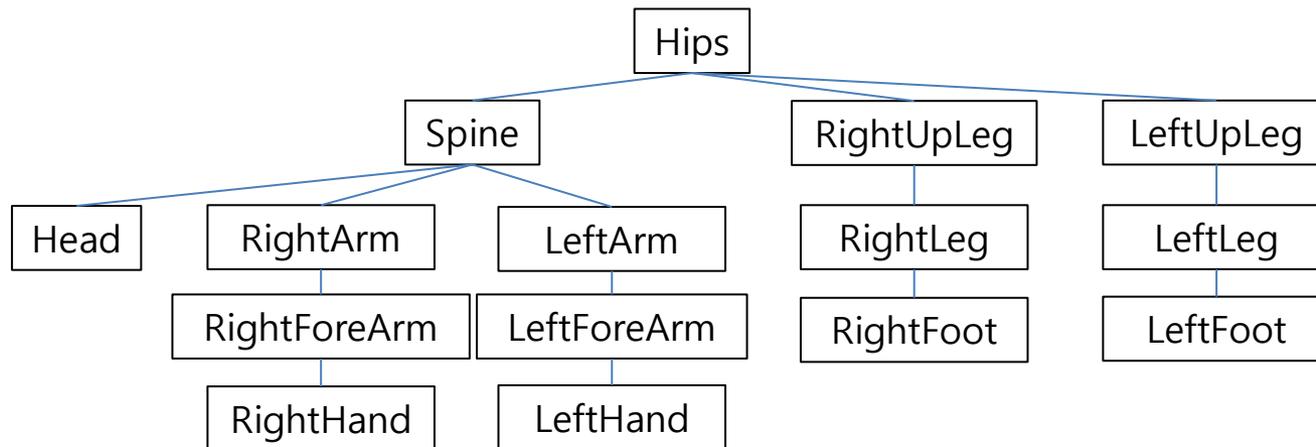
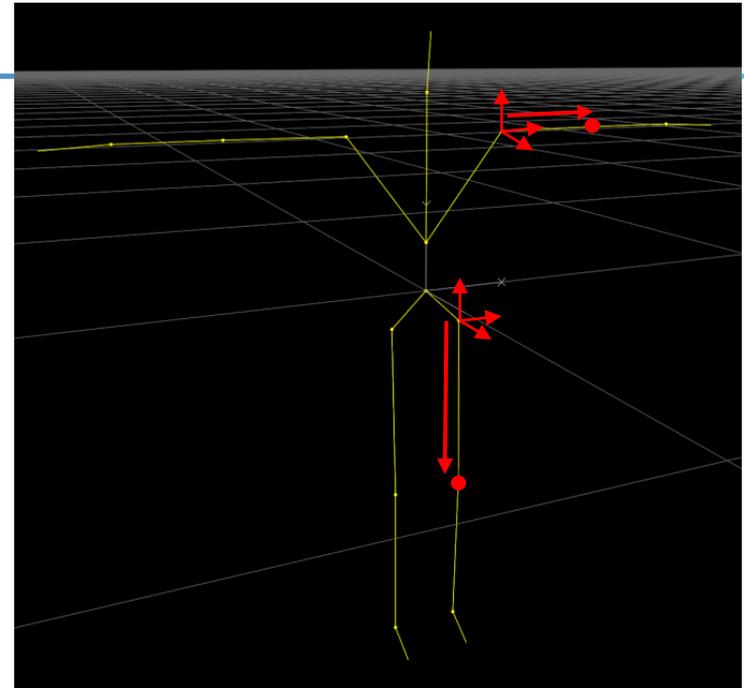
"Skeleton" Part

- "*Skeleton*": static data
 - joint hierarchy
 - joint offset from its parent joint
 - usually translation
 - w.r.t. parent frame

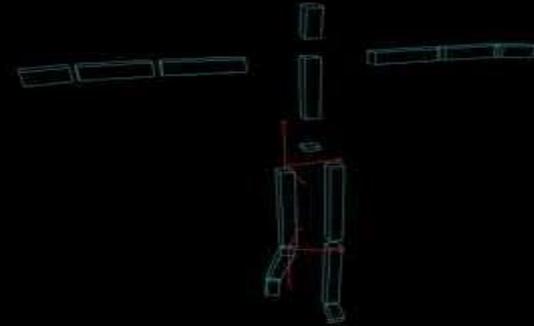
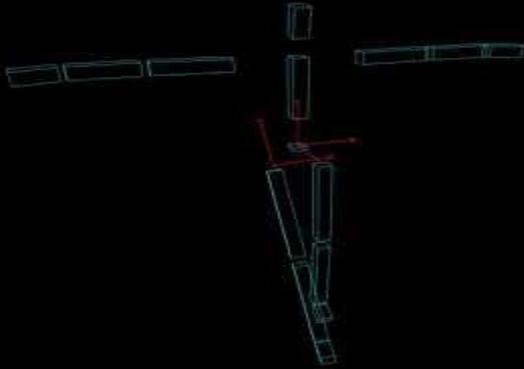


"Skeleton" Part

- "*Skeleton*": static data
 - joint hierarchy
 - joint offset from its parent joint
 - usually translation
 - w.r.t. parent frame



Recall: Hierarchical Model - Human Figure



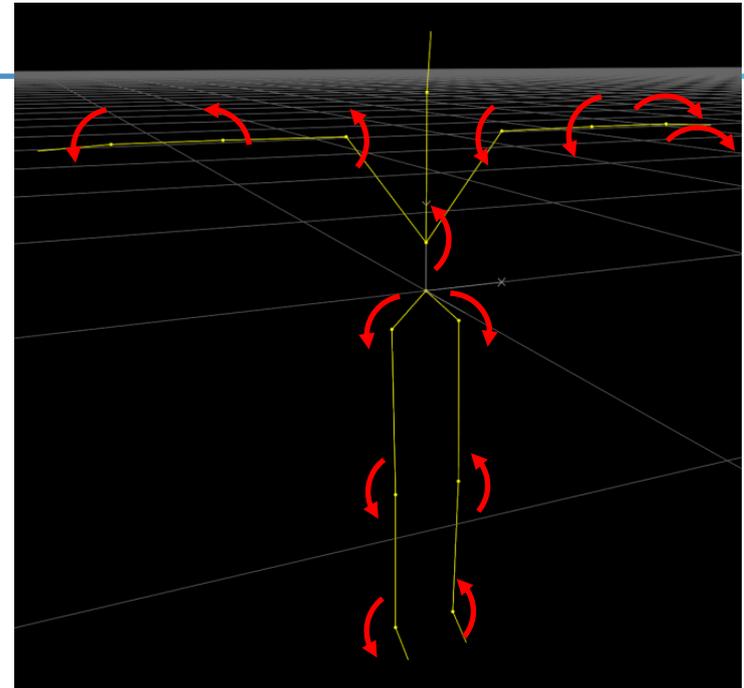
<https://youtu.be/Q7lhvMkCSCg>

<https://youtu.be/Q5R8WGUwpFU>

- Each part's movement is described w.r.t. its parent's frame.
 - → Each part has its own transformation w.r.t. parent's frame.
- Actually, this transformation is twofold:
 - **Static** transformation: **joint offset**
 - **Time-varying** transformation: "**motion**"

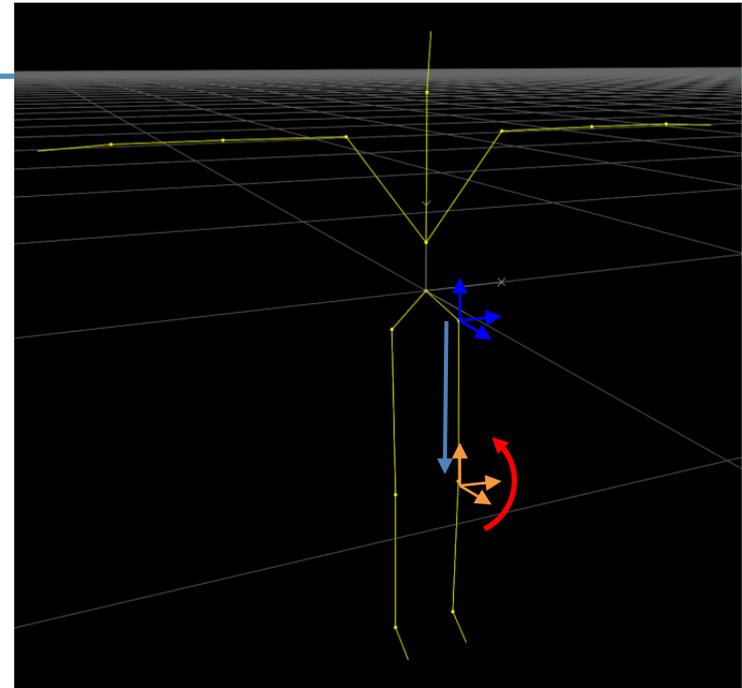
"Motion" Part

- "*Motion*": time-varying data
 - movement of internal joints
 - usually rotation
 - w.r.t. default frame of each joint
 - the frame after applying joint offset to the parent frame
 - (translational and rotational) movement of skeletal root
 - usually the pelvis part
 - w.r.t. world frame



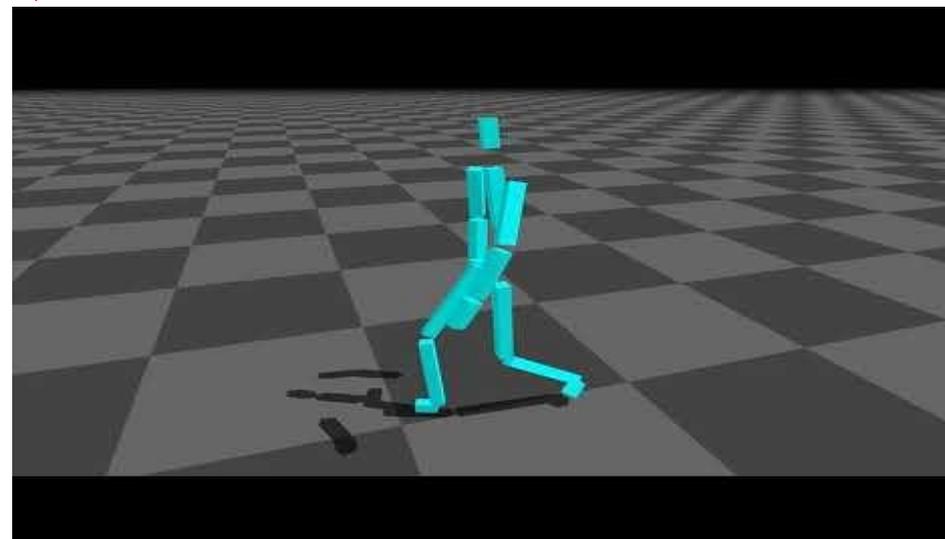
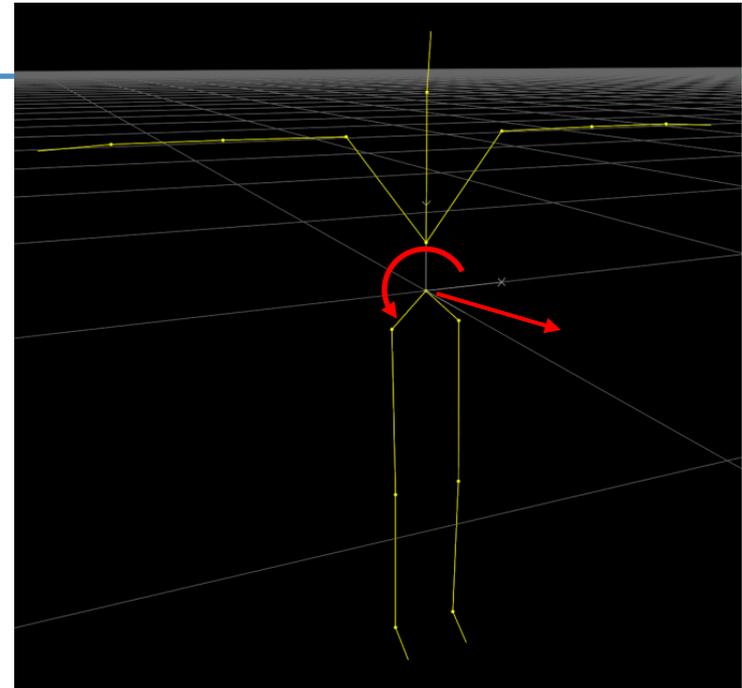
"Motion" Part

- "*Motion*": time-varying data
 - movement of internal joints
 - usually rotation
 - w.r.t. default frame of each joint
 - the frame after applying joint offset to the parent frame
 - (translational and rotational) movement of skeletal root
 - usually the pelvis part
 - w.r.t. world frame



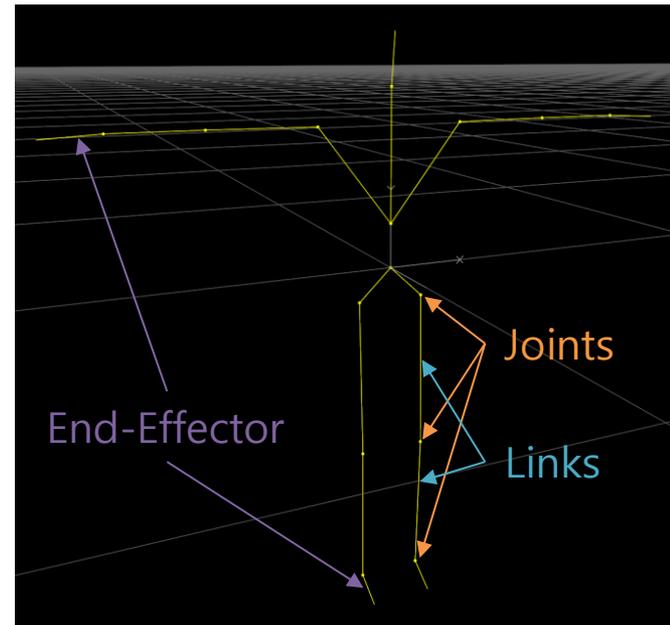
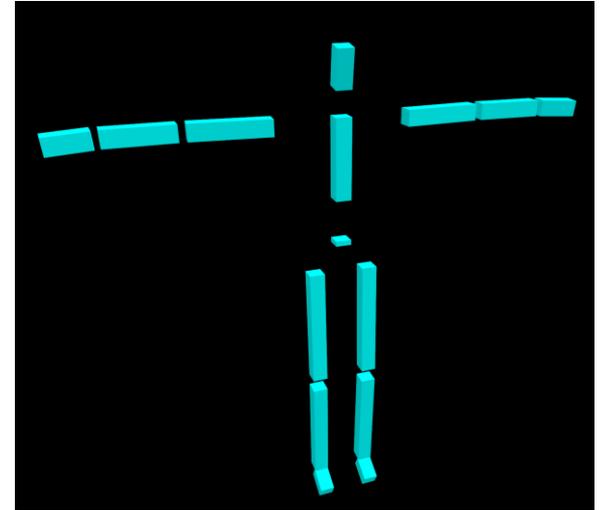
"Motion" Part

- "*Motion*": time-varying data
 - movement of internal joints
 - usually rotation
 - w.r.t. default frame of each joint
 - the frame after applying joint offset to the parent frame
 - (translational and rotational) movement of skeletal root
 - usually the pelvis part
 - w.r.t. world frame



Terminologies

- **Posture (pose):** "motion" at a single frame
- **Rest pose:** a pose where all joint movements are "zero"
- **Joint** - a connection between two objects which allows some movement
- **Link** - a rigid object between joints
- **End effector** - a free end of a kinematic chain



Forward Kinematics

Kinematics

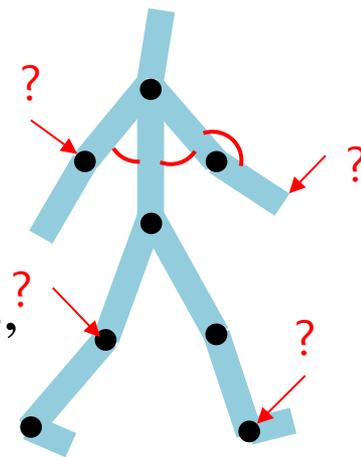
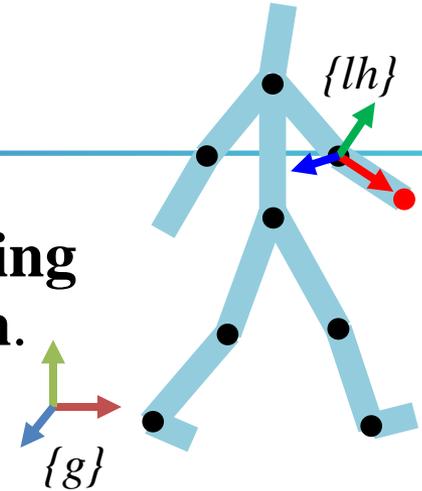
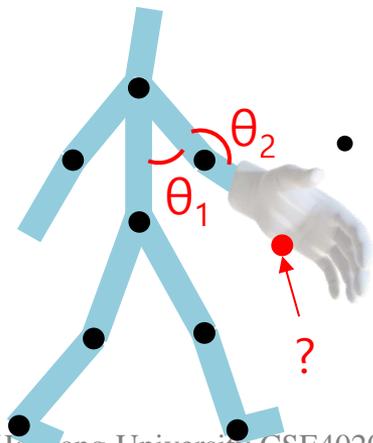
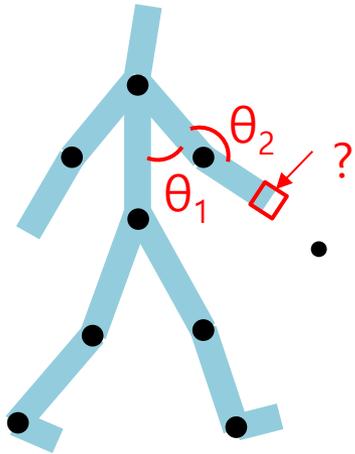
- *Kinematics* is the study of motion of objects without considering mass or forces.
 - By contrast, *Dynamics* is the study of the relationship between motion and its causes, specifically, forces and mass.
- In computer graphics, kinematics is about skeletal animation.
 - *Forward kinematics*: Given **joint angles**, compute **the position & orientation of end-effector**
 - *Inverse kinematics*: Given **the position & orientation of end-effector**, compute **joint angles**
- Understanding forward kinematics helps in understanding skeletal animation.

Forward Kinematics (FK)

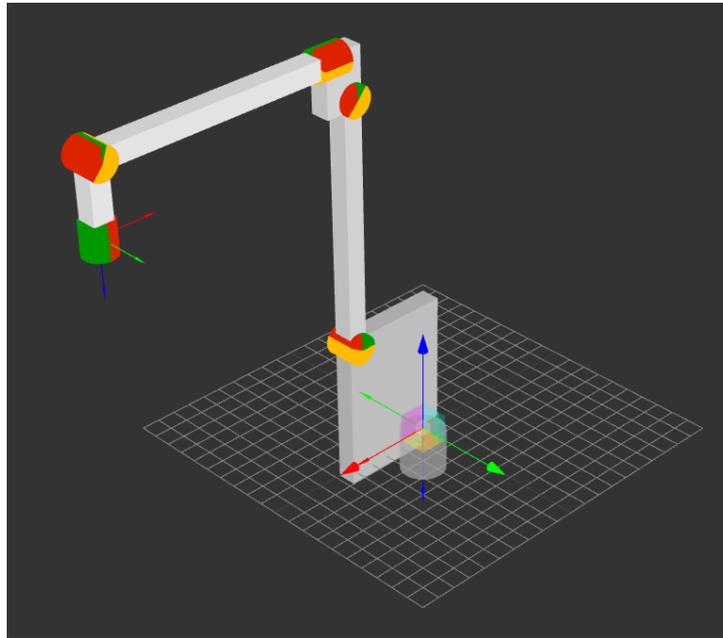
- *Forward kinematics map T is a mapping from local position to global position.*
 - T is calculated using joint angles.

- *Forward kinematics: (Calculate T from) given joint angles, compute the position & orientation of end-effector (using this T).*

- In fact, it's not limited only to end effectors,
- and can be applied to any point "attached" to any link.



[Demo] Forward / Inverse Kinematics

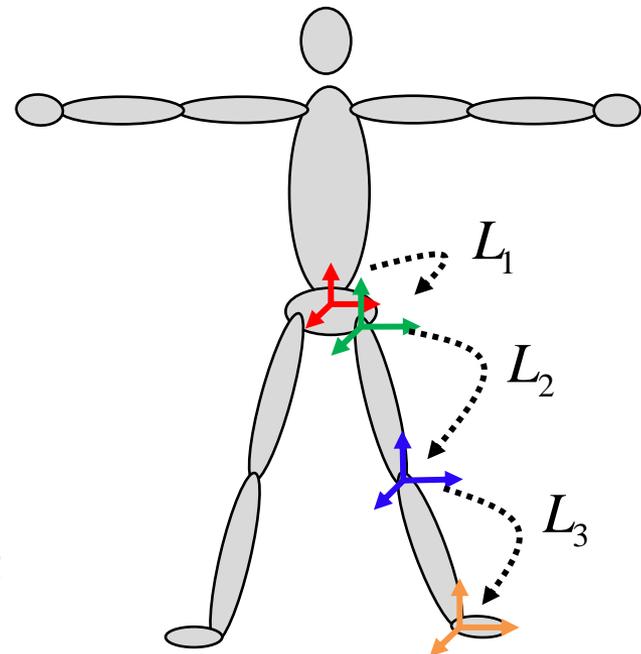
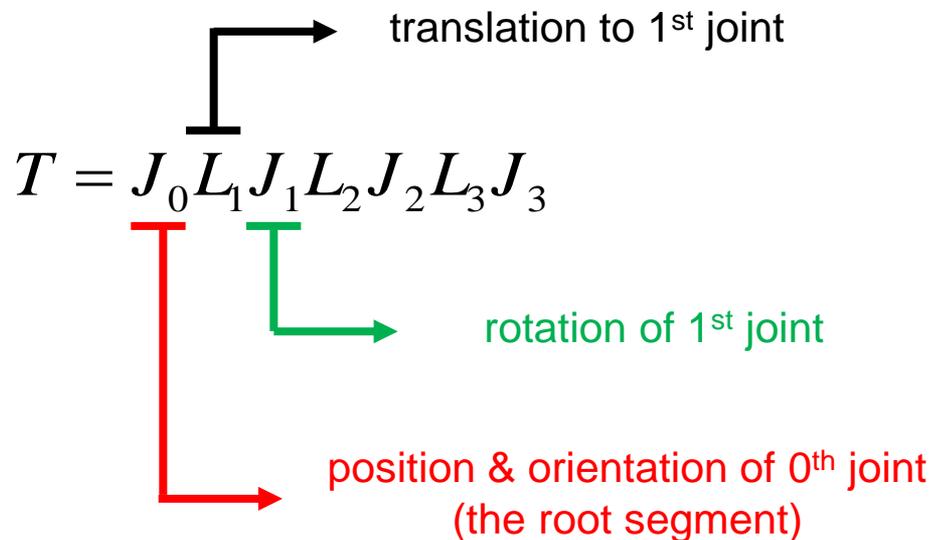


<http://robot.glumb.de/>

- Forward kinematics : Open “angles” menu and change values
- Inverse kinematics : Move the end-effector position by mouse dragging

Forward Kinematics Map

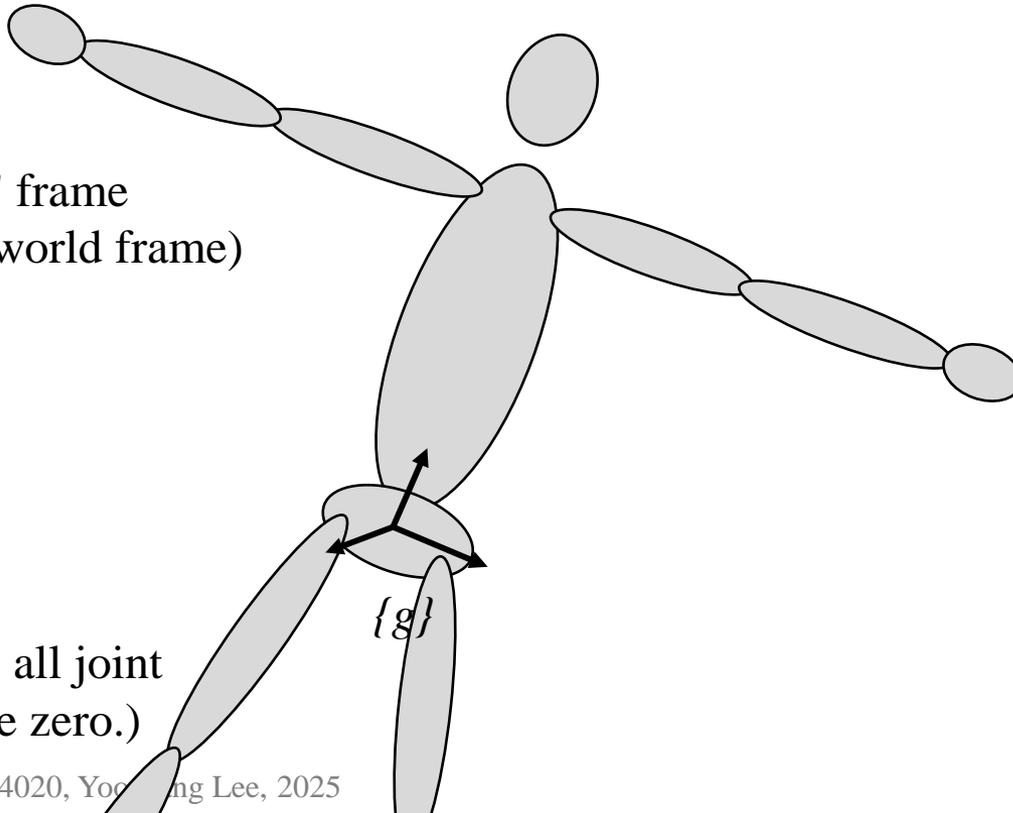
- A *forward kinematics map* T is an alternating multiple of ...
- **Joint transformations** (time-varying)
 - : Joint movement ("*motion*")
- **Link transformations** (static)
 - : Joint offset ("*skeleton*")



Forward Kinematics Map

$$T = \mathbf{I}$$

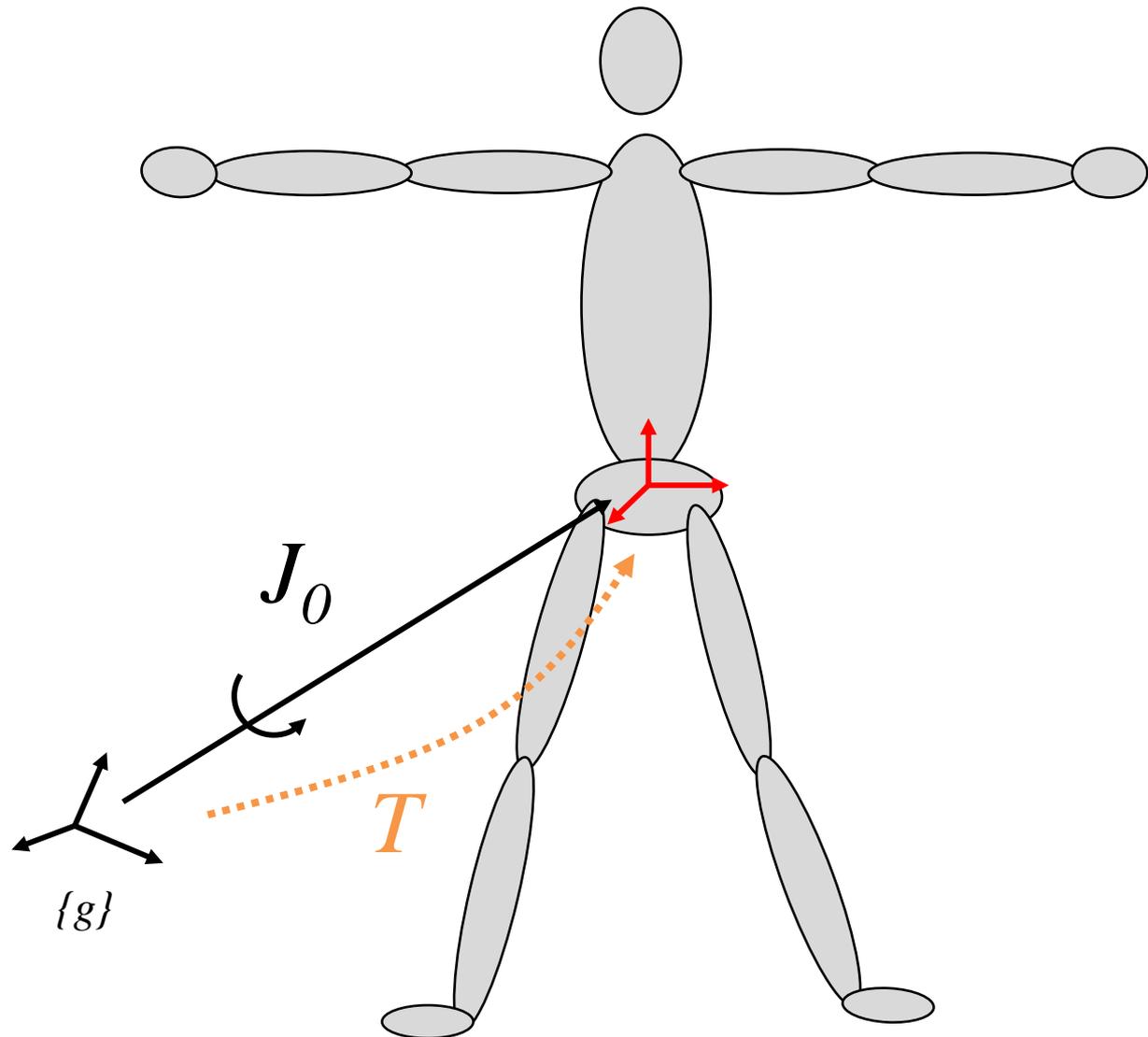
current "local" frame
(expressed in world frame)



(In this figure, all joint
movements are zero.)

Forward Kinematics Map

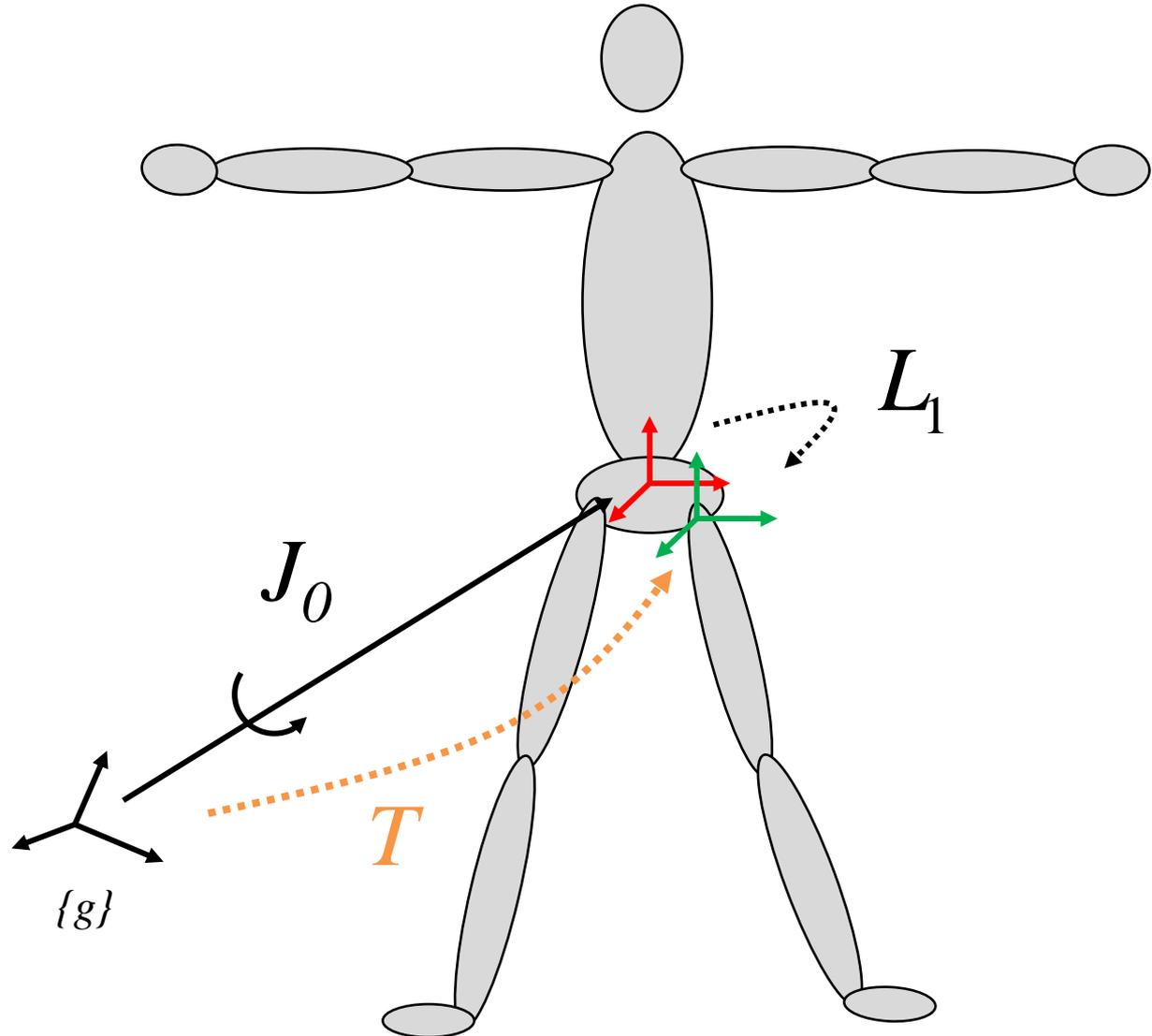
$$T = J_0$$



(In this figure, all joint movements except J_0 are zero.)

Forward Kinematics Map

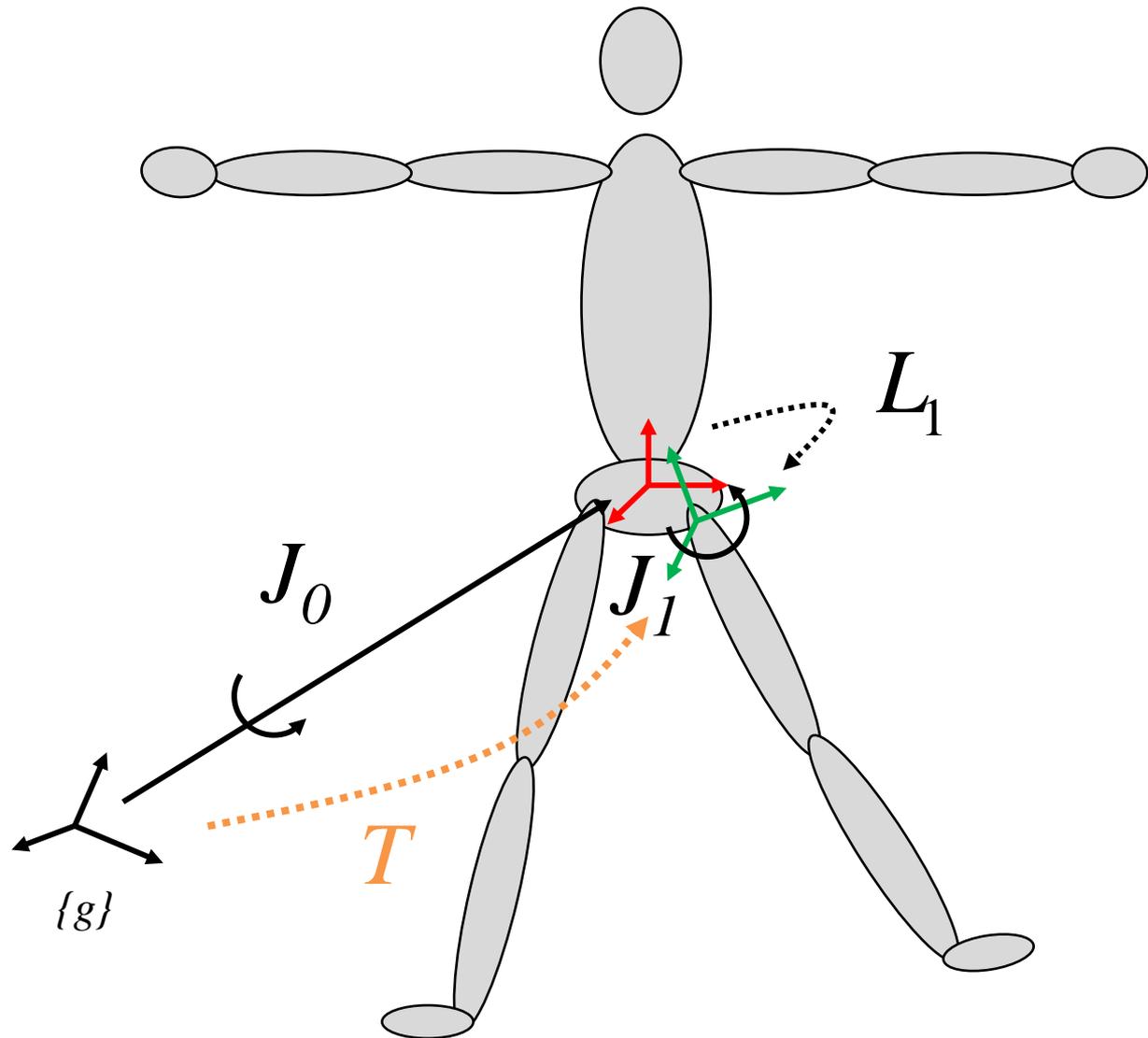
$$T = J_0 L_1$$



(In this figure, all joint movements except J_0 are zero.)

Forward Kinematics Map

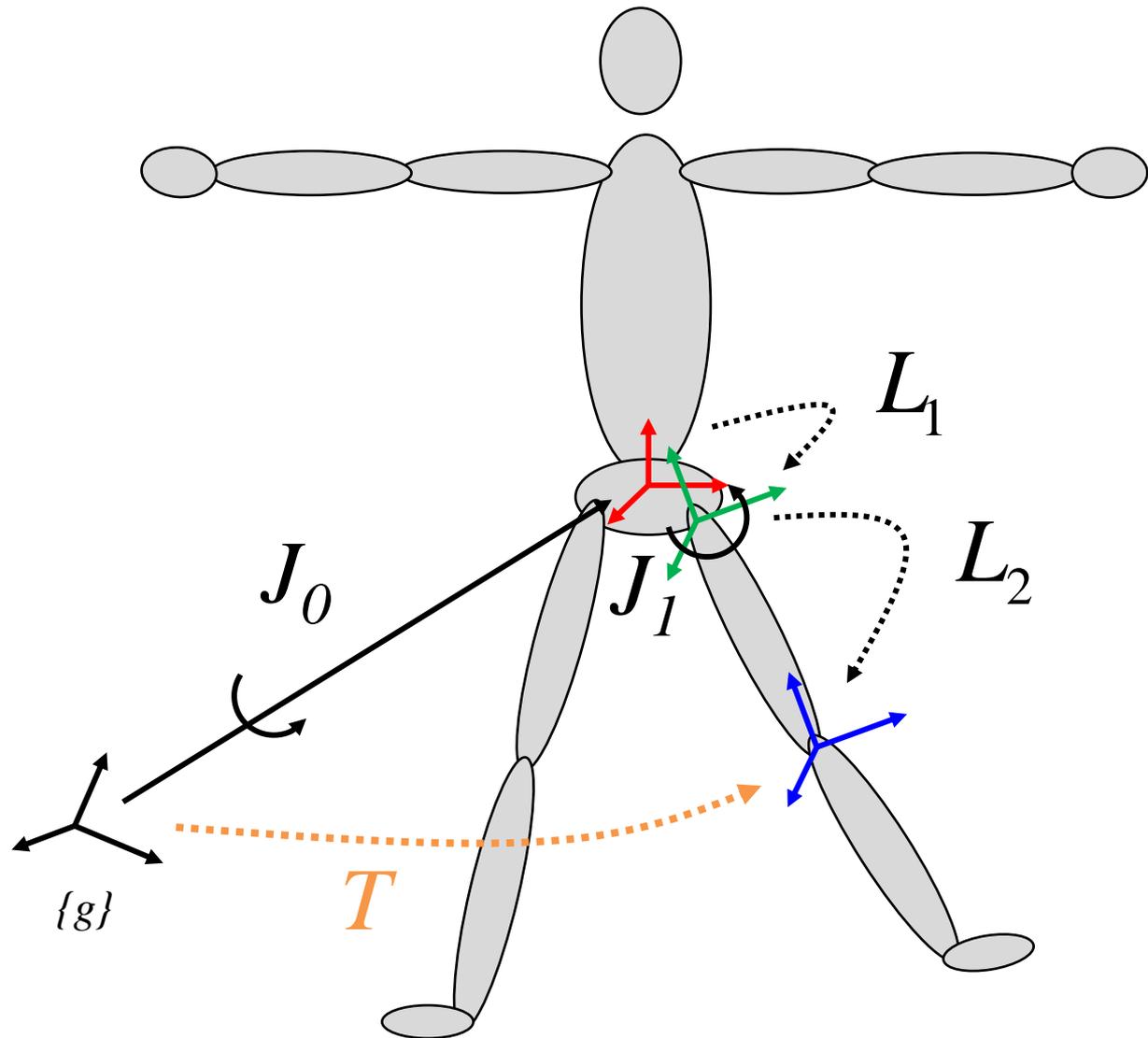
$$T = J_0 L_1 J_1$$



(In this figure, all joint movements except J_0 , J_1 are zero.)

Forward Kinematics Map

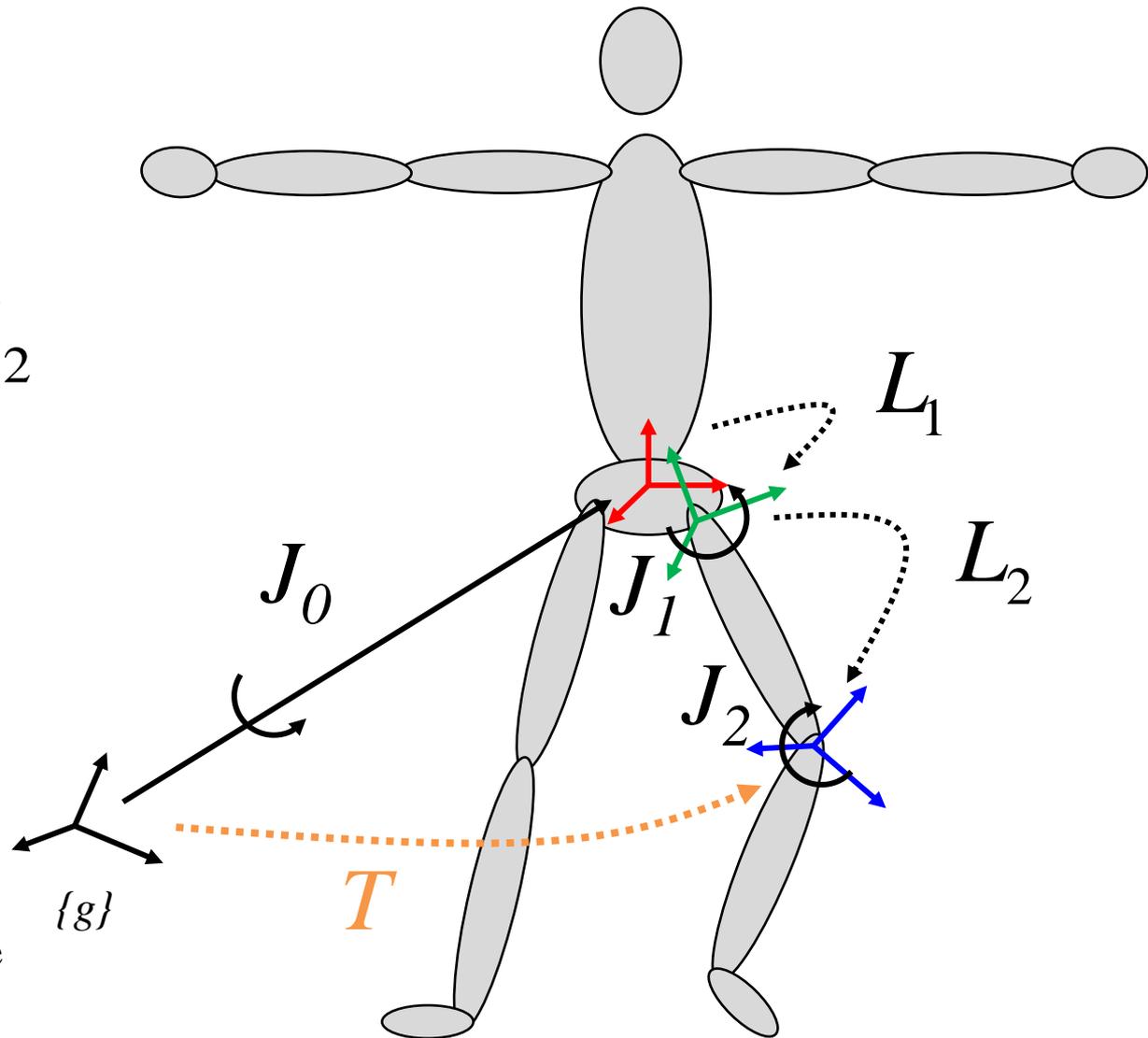
$$T = J_0 L_1 J_1 L_2$$



(In this figure, all joint movements except J_0 , J_1 are zero.)

Forward Kinematics Map

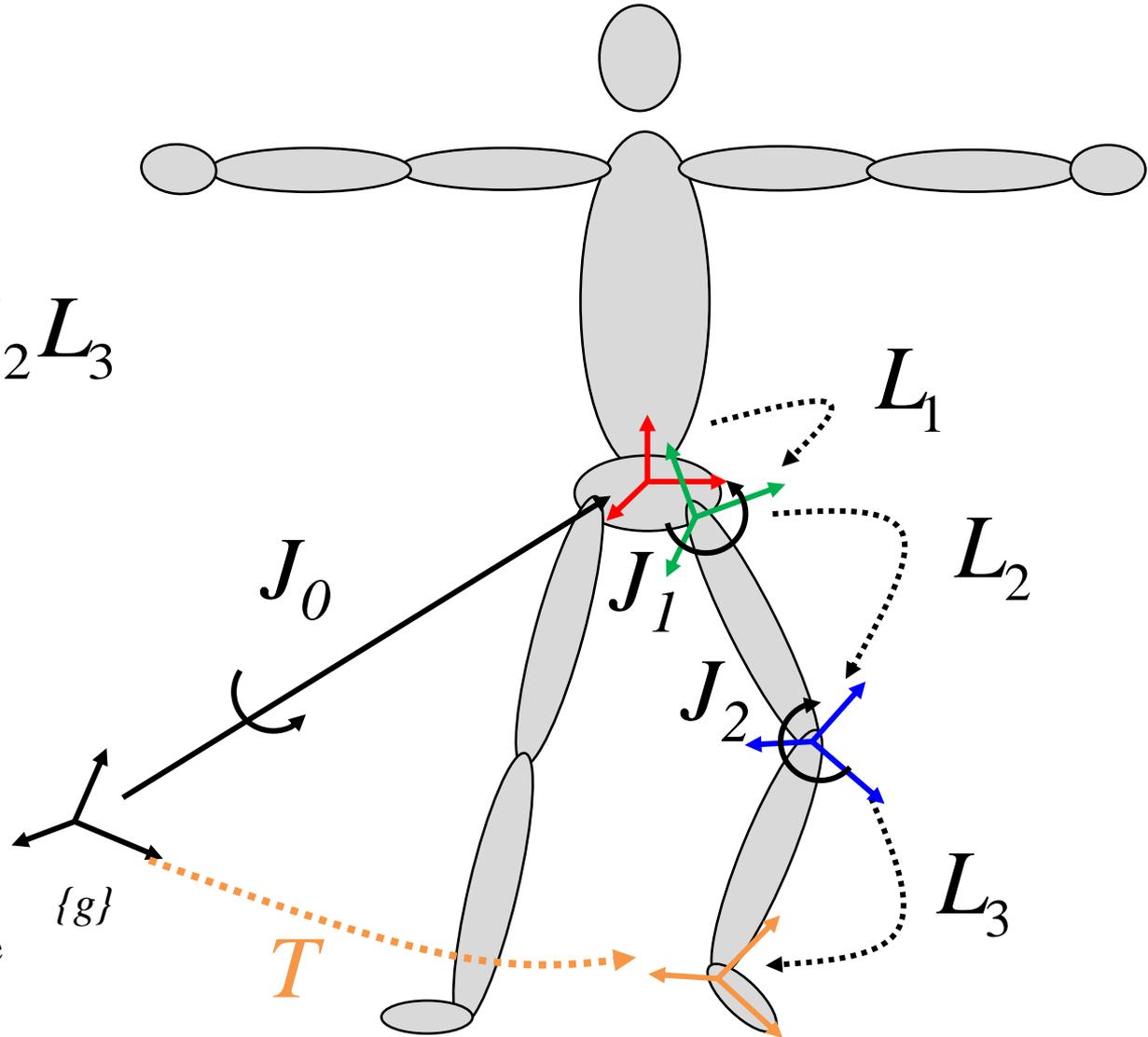
$$T = J_0 L_1 J_1 L_2 J_2$$



(In this figure, all joint movements except J_0 , J_1 , J_2 are zero.)

Forward Kinematics Map

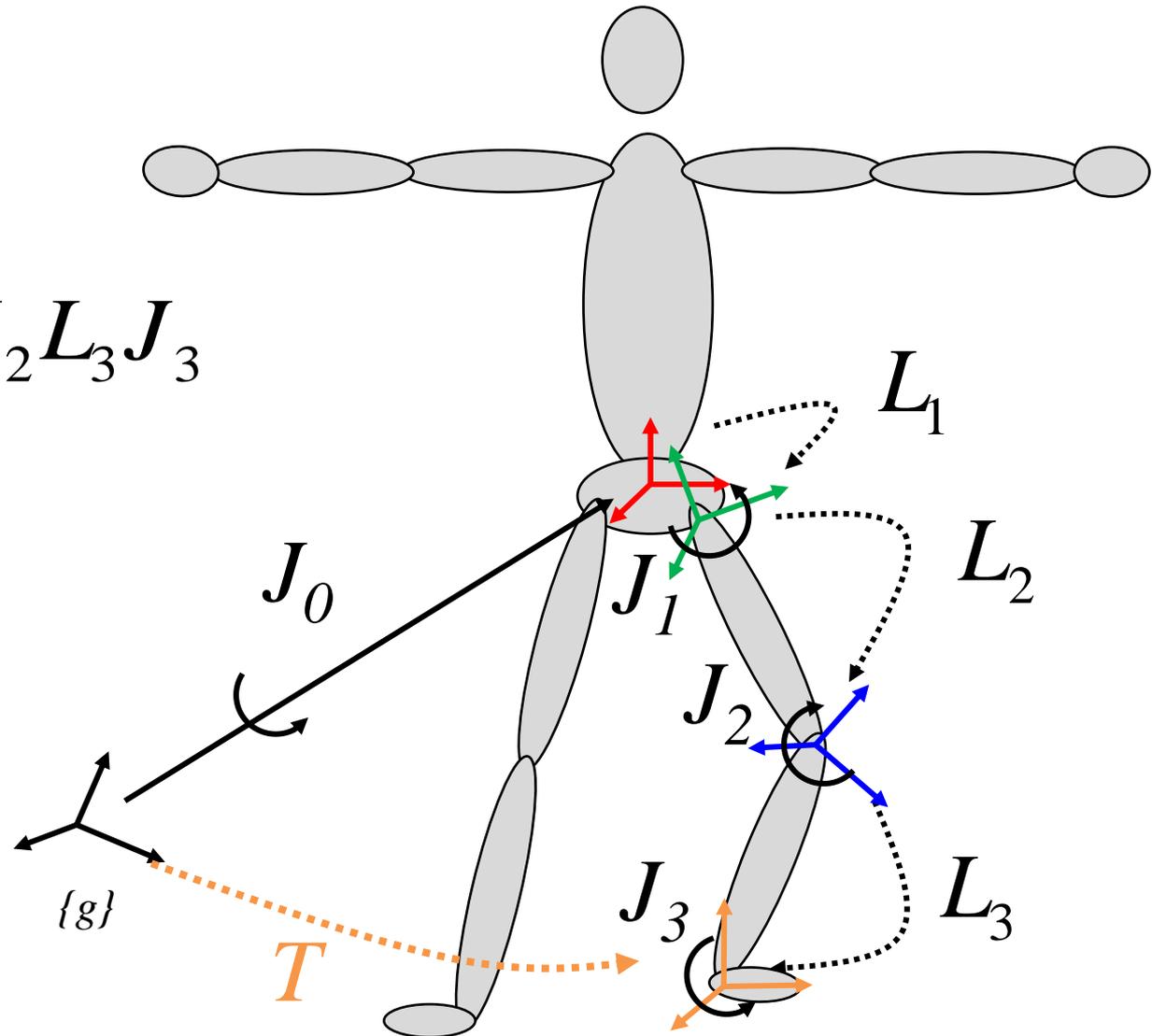
$$T = J_0 L_1 J_1 L_2 J_2 L_3$$



(In this figure, all joint movements except J_0 , J_1 , J_2 are zero.)

Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2 J_2 L_3 J_3$$

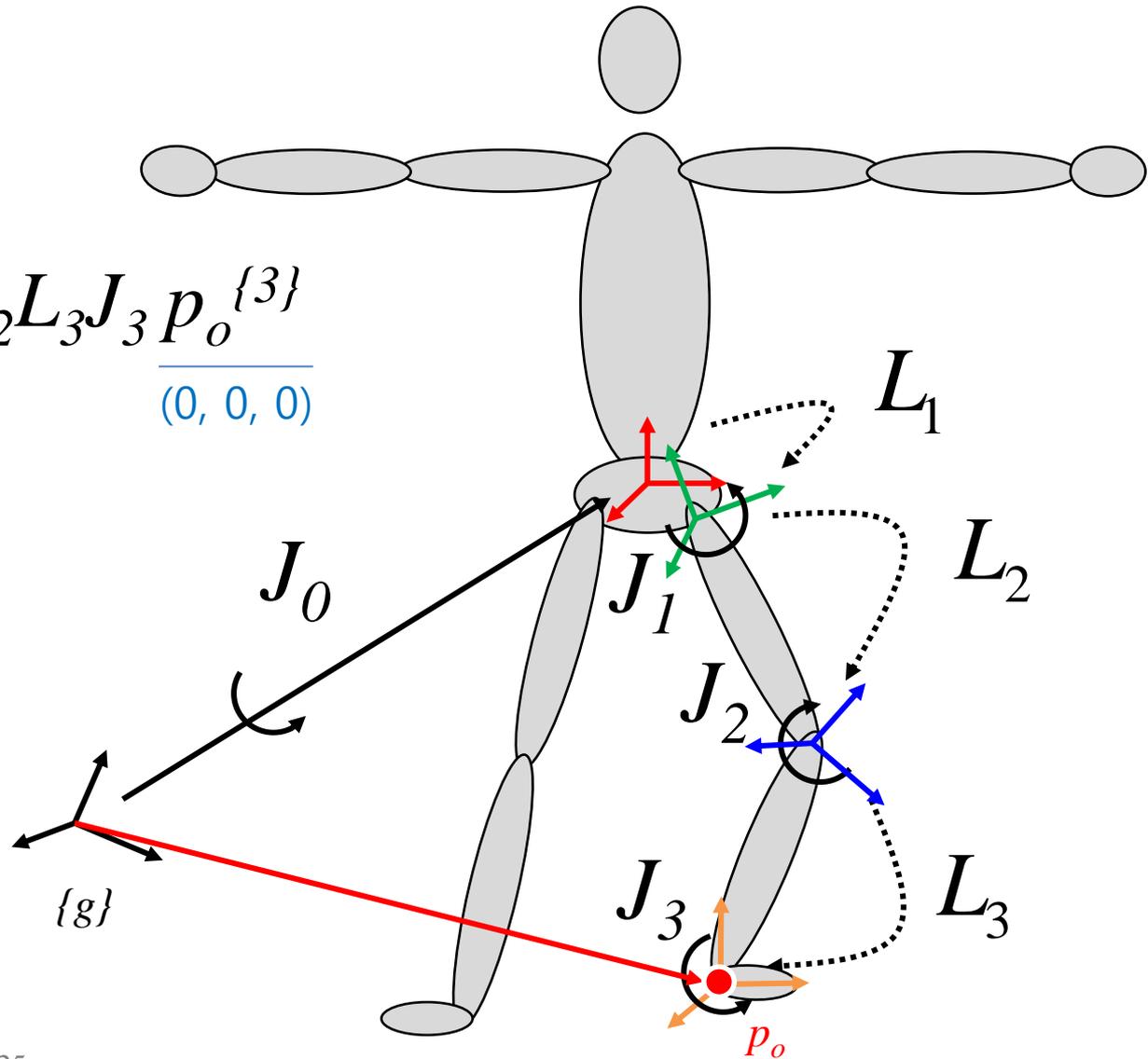


(In this figure, all joint movements except J_0, J_1, J_2, J_3 are zero.)

Forward Kinematics Map

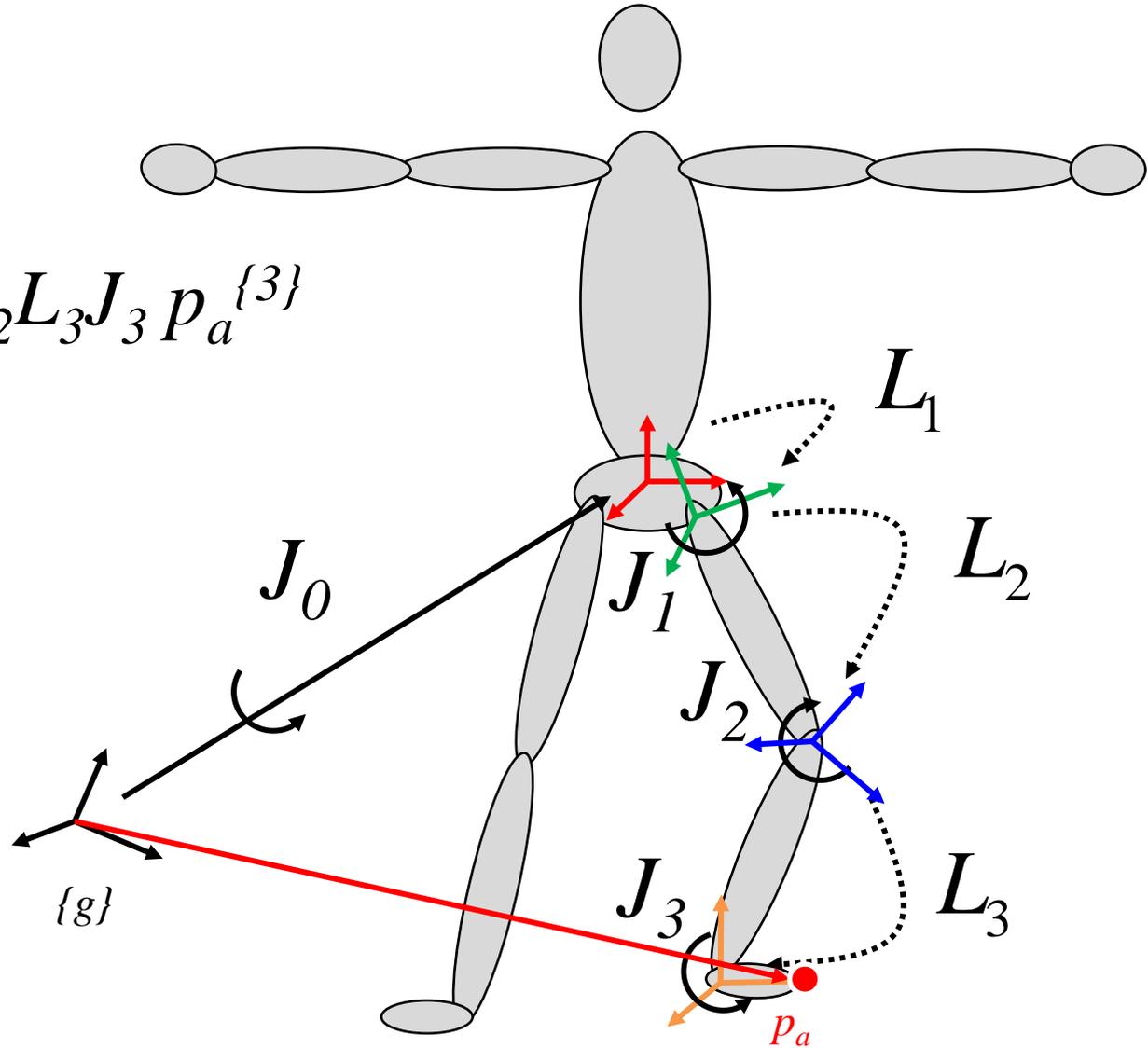
$$p_o^{\{g\}} = J_0 L_1 J_1 L_2 J_2 L_3 J_3 p_o^{\{3\}}$$

$\underbrace{\hspace{10em}}_{(0, 0, 0)}$



Forward Kinematics Map

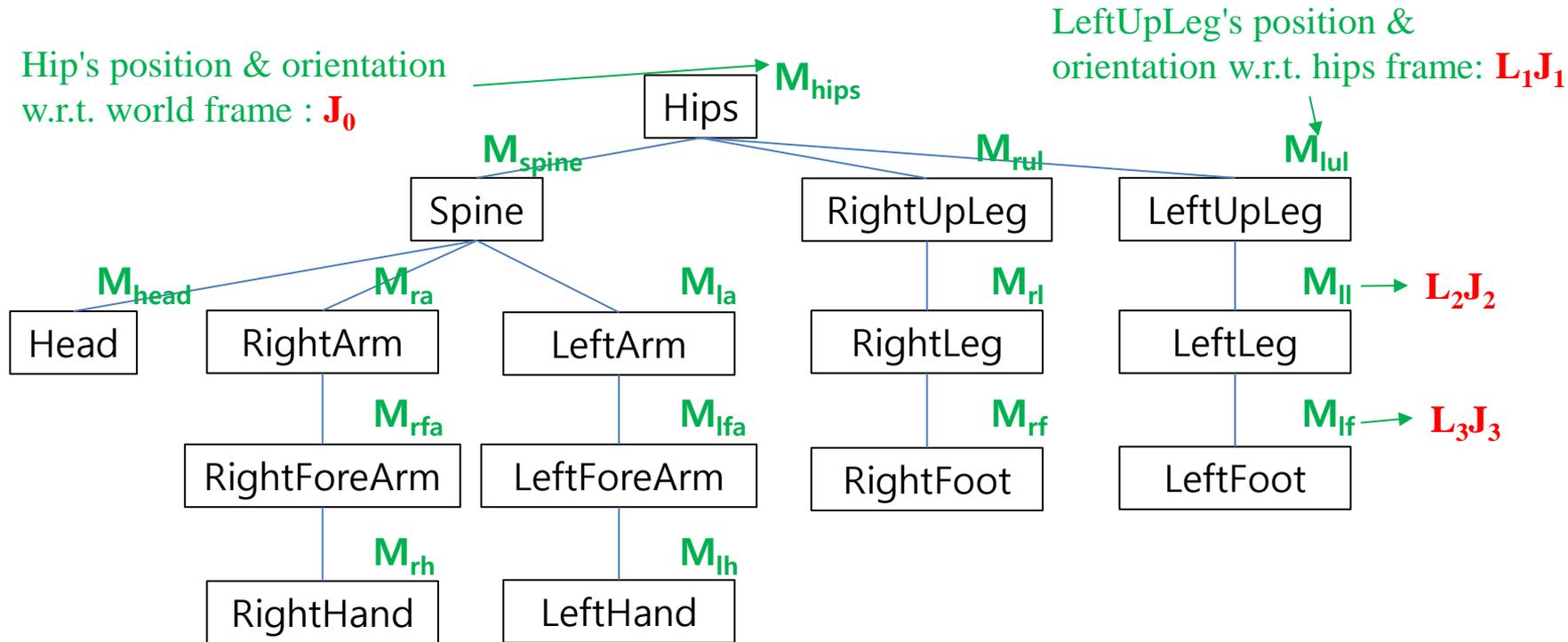
$$p_a^{\{g\}} = J_0 L_1 J_1 L_2 J_2 L_3 J_3 p_a^{\{3\}}$$



Recall: Rendering Hierarchical Models

- Each node has its own transformation w.r.t. parent node's frame.

→ **Local transformation**



Forward Kinematics Map

- This is how a skeletal animation is defined by
 - Joint hierarchy
 - Link transformations L_i
 - \rightarrow "*Skeleton*"
- Joint transformations J_i
- \rightarrow "*Motion*"

Quiz 1

- Go to <https://www.slido.com/>
- Join #cg-ys
- Click "Polls"

- Submit your answer in the following format:
 - **Student ID: Your answer**
 - e.g. **2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to be counted for attendance.

Quiz 2

- Go to <https://www.slido.com/>
- Join #cg-ys
- Click "Polls"

- Submit your answer in the following format:
 - **Student ID: Your answer**
 - e.g. **2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to be counted for attendance.

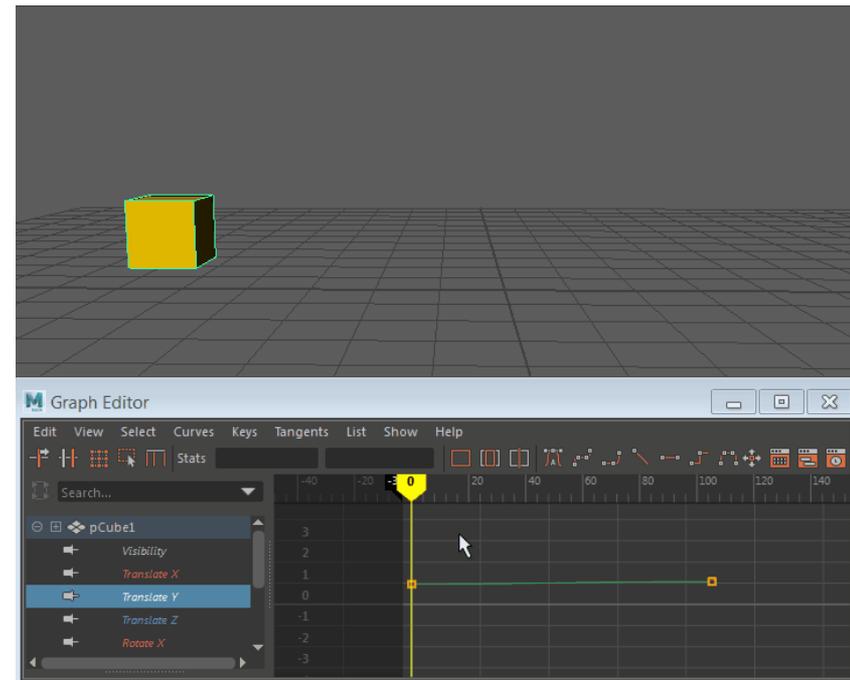
Creating Character Animation

Methods for Creating Character Animation

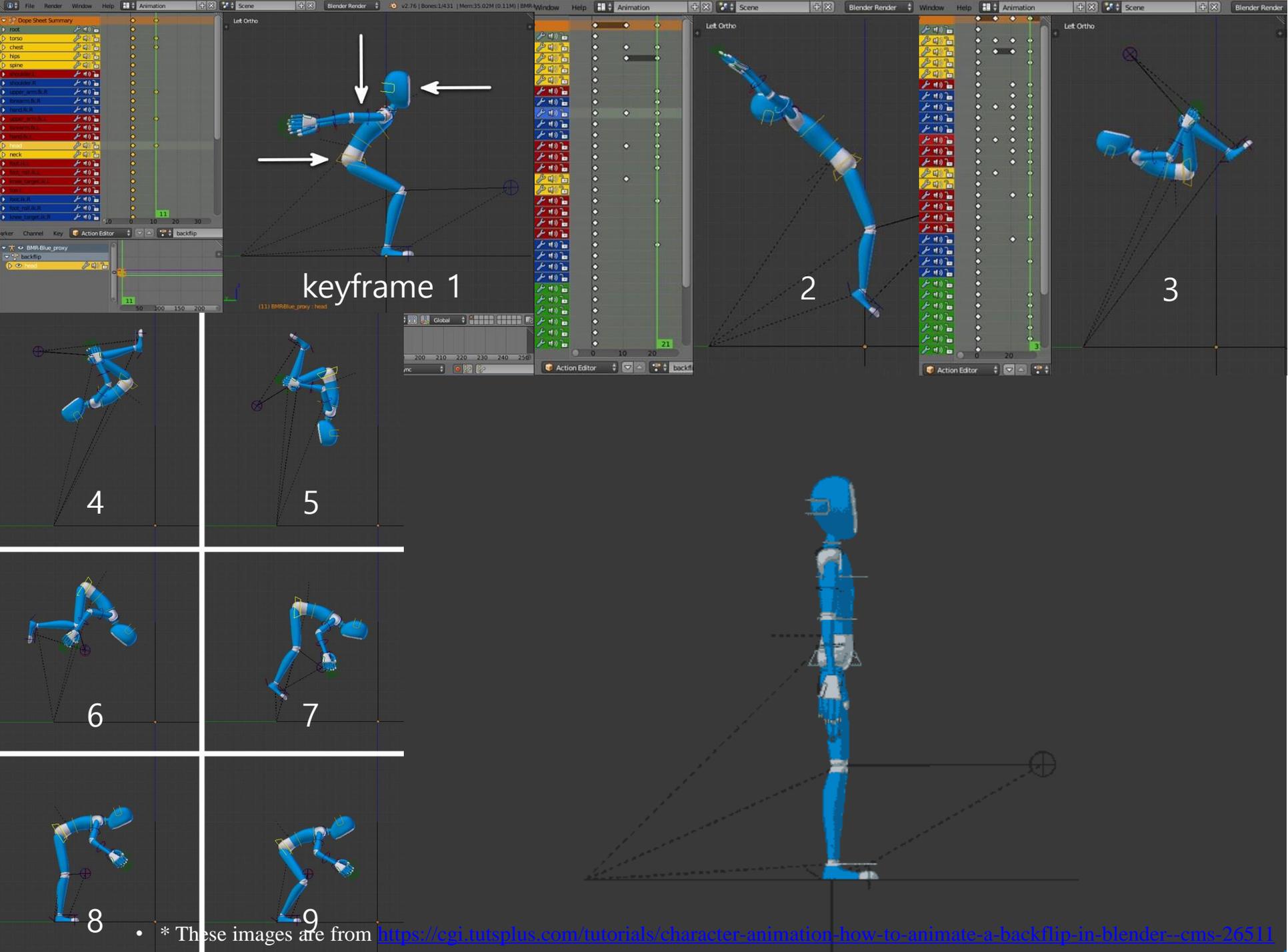
- How to create and set "*skeleton*" and "*motion*"?
- Keyframe Animation
- Motion Capture
- Data-Driven Animation
- Physics-Based Animation

Keyframe Animation

- Animators specifies important events at *key frames*.
- Computer software generates inbetween frames using interpolation.
 - a task that was traditionally done by human assistants in 2D animation.
- → Replace labor-intensive "inbetweening" processes with computing
- Usually, "*skeleton*" is manually created.
- "*Motion*" is controlled by setting keyframes and manipulating various parameter curves.



* This image is from <https://help.autodesk.com/view/MAYAUL/2022/ENU/?guid=GUID-9E69ABB3-E94B-4D67-9935-40F77FD2E9E0>



keyframe 1

2

3

4

5

6

7

8

9

* These images are from <https://cgi.tutspplus.com/tutorials/character-animation-how-to-animate-a-backflip-in-blender--cms-26511>

Keyframe Animation

- A traditional method to produce computer animation,
 - which has roots in hand-drawn traditional animation.
- Difficult to create “realistic” and “physically plausible” motions.
 - The quality of the output largely depends on the skill of the individual artist.
- Still used a lot.

Motion Capture

- Idea: Use “real” human motion to create realistic animation.
- Motion capture system “captures” movement of people or objects by measuring
 - position of each marker on the skin
 - position and orientation of each body part (or joint)
- "*Motion*" and "*skeleton*" are extracted through post-processing.

Motion Capture



<https://youtu.be/YzS73UCOk20>

Bvh Motion Capture Data Example

HIERARCHY → *"Skeleton"* ↓

ROOT Hips

```
{
  OFFSET 0.0 0.0 0.0
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTATION
  JOINT Spine
  {
    OFFSET 0.0 0.118905 0.0
    CHANNELS 3 ZROTATION XROTATION YROTATION
    JOINT Head
    {
      OFFSET 0.0 0.3439156 0.00139004
      CHANNELS 3 ZROTATION XROTATION YROTATION
      End Site
      {
        OFFSET -0.0029 0.127742 0.0251395
      }
    }
    JOINT RightArm
    {
      OFFSET -0.192727 0.25291768 0.00229786
      CHANNELS 3 ZROTATION XROTATION YROTATION
      JOINT RightForeArm
      {
        ...
      }
    }
  }
}
```

MOTION → *"Motion"* ↓

Frames: 199

Frame Time: 0.033333

```
1.95769 0.989769479321 0.039193 -4.11275998891 -0.490682977769 -91.3519974695 0.45458697547 ...
1.95769 0.989769479321 0.0392908 -4.11760985011 -0.48626597981 -91.3734989051 0.513819016282 ...
1.95769 0.989769479321 0.039424 -4.12004011679 -0.488125979059 -91.387002189 0.592700017233 ...
1.95771 0.989769479321 0.0395518 -4.0961698863 -0.500940000911 -91.3840993586 0.61126399115 ...
1.95779 0.989759479321 0.0396999 -4.05799980101 -0.510696019006 -91.3839969058 0.58299101005 ...
1.9579 0.989719479321 0.0398625 -4.0423300664 -0.503295989288 -91.3842018115 0.57718001317 ...
...
```

Motion Capture

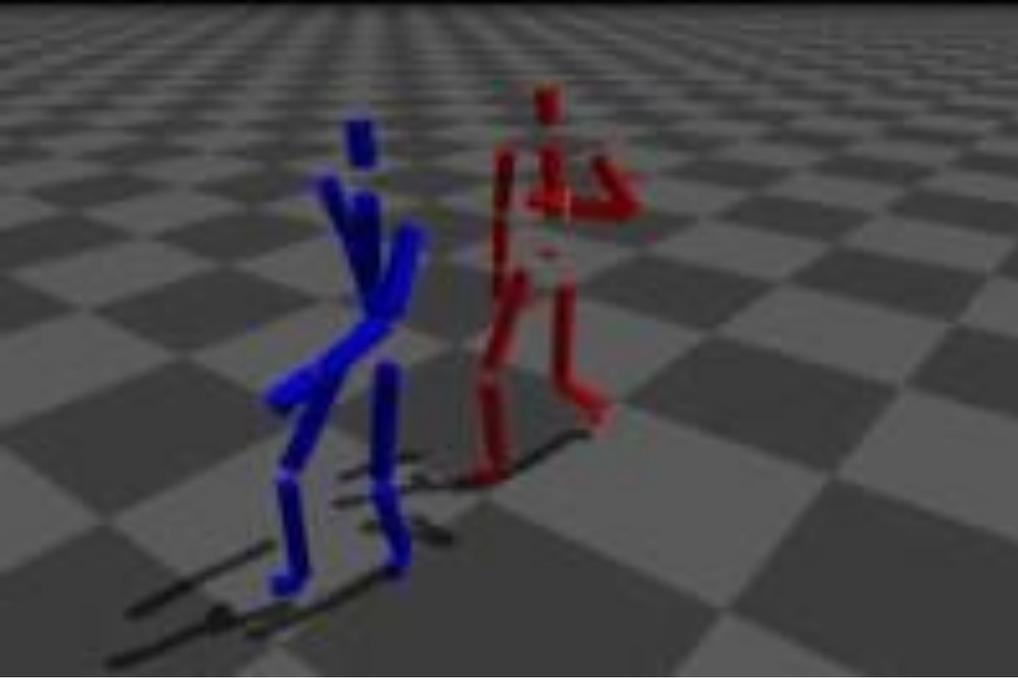
- Now widely used in movies & games (mainly by major companies).
- Very expansive
 - Expensive devices
 - High operating cost
- Limitation: Motion captured data is very realistic **only** in the same virtual environment as capture environment.

Data-Driven Animation

- Idea: Maximize the reuse of / add flexibility to captured motion data by editing, restructuring, or learning.
- "*Skeleton*" from the motion dataset is used.
- "*Motion*" is edited / reconstructed / synthesized based on the motion dataset.
- Motion editing techniques
 - time warping, motion blending, ...
- Data-driven techniques
 - motion graphs, motion matching, ...
- Deep learning-based motion synthesis

Motion Warping example

<https://youtu.be/VH4QuV2mFcg>



Deep Learning-Based Motion Synthesis - PFNN (2017)

<https://youtu.be/UI0Gilv5wvY>



Physics-Based Animation (Simulation-Based)

- Idea: Use physics simulation to generate motion.
 - Physical reality plays a key role in creating high-quality motion.
 - Physics simulation generates a motion that is always physically plausible.
- "*Skeleton*" is often loaded from a model file description such as .urdf.
- "*Motion*" is computed by physics simulation.
- Requires a "controller".
 - Determines joint torques at each timestep to perform desired action while maintaining balance.
- Now deep reinforcement learning (DRL) is widely used to learn control policies.

DeepMimic [Peng et al. 2018]

<https://youtu.be/vppFvq2quQ0>



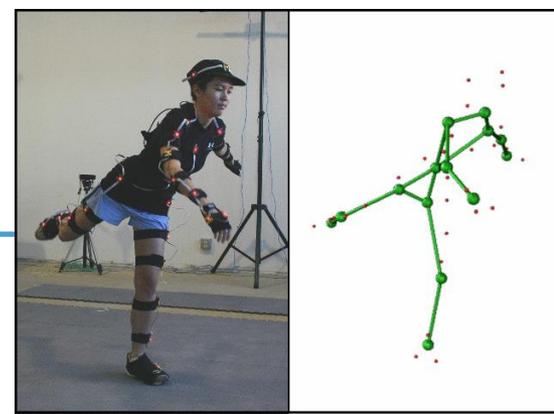
PhysicsFC [Kim et al. 2025]

<https://youtu.be/dSfW8EvV5CA>



BVH File Format

BVH File Format



- BVH (**B**io**V**ision **H**ierarchical data)
 - Developed by Biovision, a motion capture company
- Consists of two parts:
- **Hierarchy section**
 - Describes the "*Skeleton*": static data
- **Motion section**
 - Describes the "*Motion*": time-varying data
- Text file format (human-readable)

Biovision BVH

■ Hierarchy section

➤ "HIERARCHY"

• "ROOT"

- followed by the name of the root
- "{ " and " } " pair
- **"OFFSET"** : *link transformation L*
 - » X,Y and Z offset of the segment from its parent
- **"CHANNELS"** : *type of its joint transformation J*
 - » the number of channels
 - » the type of each channel

• "JOINT"

- identical to the root definition except for the number of channels
- **"OFFSET "** , **"CHANNELS"**

• "End Site"

- indicates that the current segment is an end effector (no children)
- **"OFFSET "**

```
ROOT Hips
{
  OFFSET 0.00 0.00 0.00
  CHANNELS 6 Iposition Yposition Zposition Zrotation Irotation Yrotation
  JOINT LeftHip
  {
    OFFSET 3.430000 0.000000 0.000000
    CHANNELS 3 Zrotation Irotation Yrotation
    JOINT LeftKnee
    {
      OFFSET 0.000000 -18.469999 0.000000
      CHANNELS 3 Zrotation Irotation Yrotation
      JOINT LeftAnkle
      {
        OFFSET 0.000000 -17.950001 0.000000
        End Site
        {
          OFFSET 0.000000 -3.119999 0.000000
        }
      }
    }
  }
  ...
}
```

in this example,

- 6 channels for the root (Tx Ty Tz Rz Rx Ry)
- 3 channels for every other object (Rz Rx Ry)

HIERARCHY

ROOT Hips

```
{  
  OFFSET 0.0 0.0 0.0  
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTA  
  JOINT chest  
  {  
    OFFSET 0.096536 3.475309 -0.289609  
    CHANNELS 3 Xrotation Zrotation Yrotation  
    JOINT neck  
    {  
      OFFSET -0.096536 13.901242 -2.027265  
      CHANNELS 3 Xrotation Zrotation Yrotation  
      JOINT head  
      {  
        OFFSET -0.166775 1.448045 0.482682  
        CHANNELS 3 Xrotation Zrotation Yrotation  
        JOINT leftEye  
        /
```

HIERARCHY

ROOT Hips

```
{ Root offset is generally zero
  OFFSET 0.0 0.0 0.0 J0 channels
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTA
  JOINT chest
  {
    OFFSET 0.096536 3.475309 -0.289609 L1
    CHANNELS 3 Xrotation Zrotation Yrotation J1 channels
    JOINT neck
    {
      OFFSET -0.096536 13.901242 -2.027265 L2
      CHANNELS 3 Xrotation Zrotation Yrotation J2 channels
      JOINT head
      {
        OFFSET -0.166775 1.448045 0.482682 L3
        CHANNELS 3 Xrotation Zrotation Yrotation
        JOINT leftEye J3 channels
        -
```

Biovision BVH

```
MOTION
Frames:      20
Frame Time:  0.033333
0.00 39.68  0.00  0.65  ...
...
```

■ Motion Section

➤ "MOTION"

- followed by a line indicating the number of frames
- **"Frames:"**
 - the number of frames
- **"Frame Time:"**
 - the sampling rate of the data
 - Ex) 0.033333 → 30 frames a second
- The rest of the file contains the actual motion data
 - The numbers appear in the order of the channel specifications as the skeleton hierarchy was parsed
- **Each line** has motion data for **a single frame**
- **Each number** in a line is **a value for a single channel**
- The unit of rotation channel values is **degree**

HIERARCHY

ROOT Hips

```
{
  OFFSET 0.0 0.0 0.0
  CHANNELS 6 XPOSITION YPOSITION ZPOSITION ZROTATION XROTATION YROTATION
    JOINT chest Column 1 Column 2 Column 3 Column 4 Column 5 Column 6
    {
      OFFSET 0.096536 3.475309 -0.289609
      CHANNELS 3 Xrotation Zrotation Yrotation
      JOINT neck Column 7 Column 8 Column 9
      {
        OFFSET -0.096536 13.901242 -2.027265
        CHANNELS 3 Xrotation Zrotation Yrotation
        JOINT head Column 10 Column 11 Column 12
        {
          OFFSET -0.166775 1.448045 0.482682
          CHANNELS 3 Xrotation Zrotation Yrotation
            Column 13 Column 14 Column 15
        }
      }
    }
  }
}
```

MOTION

Frames: 199

Frame Time: 0.033333

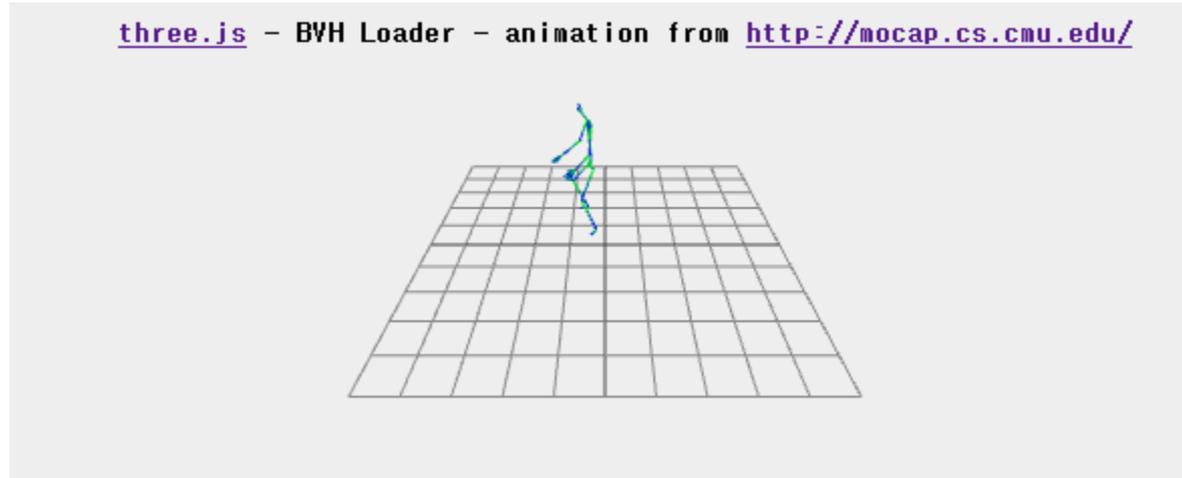
```
1.95769 0.989769479321 0.039193 -4.11275998891 -0.490682977769 -91.3519974695 0.45458697547 ...
1.95769 0.989769479321 0.0392908 -4.11760985011 -0.48626597981 -91.3734989051 0.513819016282 ...
1.95769 0.989769479321 0.039424 -4.12004011679 -0.488125979059 -91.387002189 0.592700017233 ...
1.95771 0.989769479321 0.0395518 -4.0961698863 -0.500940000911 -91.3840993586 0.61126399115 ...
1.95779 0.989759479321 0.0396999 -4.05799980101 -0.510696019006 -91.3839969058 0.58299101005 ...
1.9579 0.989719479321 0.0398625 -4.0423300664 -0.503295989288 -91.3842018115 0.57718001317 ...
```

...

■ Interpreting the data

- To calculate the position of a segment
 - Translation information
 - For any joint segment
 - » the translation information will simply be the **offset** as defined in the hierarchy section
 - For the root object
 - » The translation data will be the sum of the **offset data** and the **translation data** from the motion section
 - Rotation information
 - comes from the **motion section**
 - **The "CHANNELS" order is important:** If the order is "ZROTATION XROTATION YROTATION"
 - Apply transformation in order of rotation about z, rotation about x, rotation about y w.r.t. body frame
 - → ZXY Euler angles
 - Do not assume ZXY Euler angles. Other sequences may also be used.

[Demo] BVH Viewer



<http://motion.hahasoha.net/>

- Select other motions from the list.
- Download corresponding BVH files and open them in a text editor.

Quiz 3

- Go to <https://www.slido.com/>
- Join #cg-ys
- Click "Polls"

- Submit your answer in the following format:
 - **Student ID: Your answer**
 - e.g. **2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to be counted for attendance.

Lab Session

- Now, let's start the lab today.