
Computer Graphics

10 - Character Animation

Yoonsang Lee
Hanyang University

Spring 2026

Outline

- Skeletal Animation
- Forward Kinematics
- Creating Character Animation
 - Keyframe Animation
 - Motion Capture
 - BVH file format
 - Data-Driven Animation
 - Physics-Based Animation
- Beyond Today's Character Animation

Computer-Generated Animation

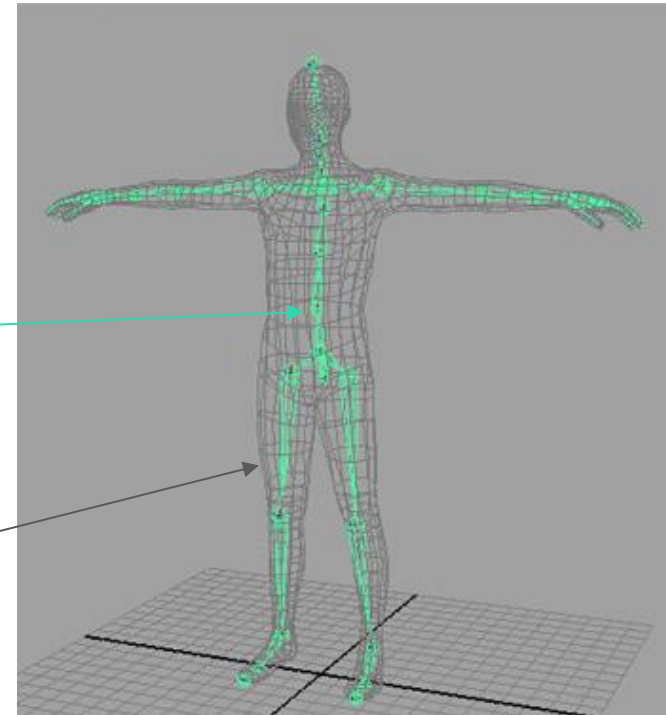
- Generating images by rendering animated 3D objects.
 - Modeling 3D objects or characters and giving them motion.
 - Although digitized hand-drawn 2D animation is still used, much of modern animation is created in 3D using this approach.



Encanto, 2021

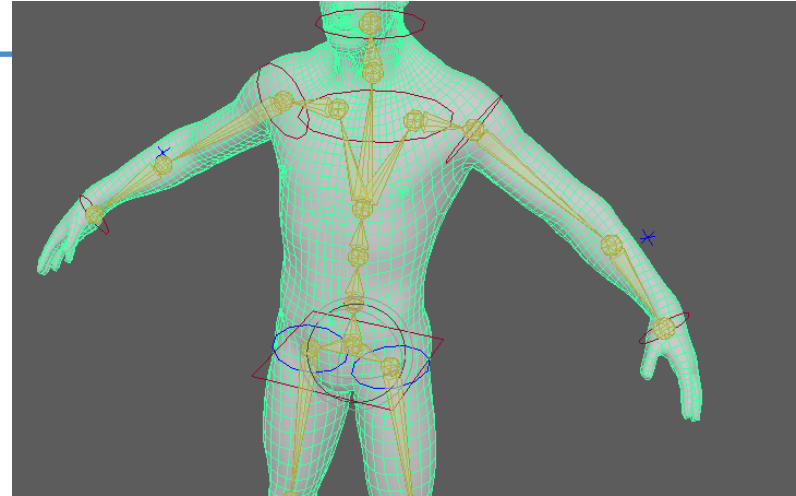
Computer-Generated Character Animation

- Most modern movies and games use *skeletal animation*.
- A skeletal animation system typically consists of:
 - **Skeleton:** bone/joint hierarchy
 - **Motion:** joint movement data
 - **Mesh / Skin:** surface representation for rendering



Visualizing Skeletal Animation

- Deformable mesh + skinning
 - Details of skinning will be omitted in this course.
 - If you are interested, many tutorials and references are available online.
- Simple rigid link visualization
- Today's focus: skeleton hierarchy and motion

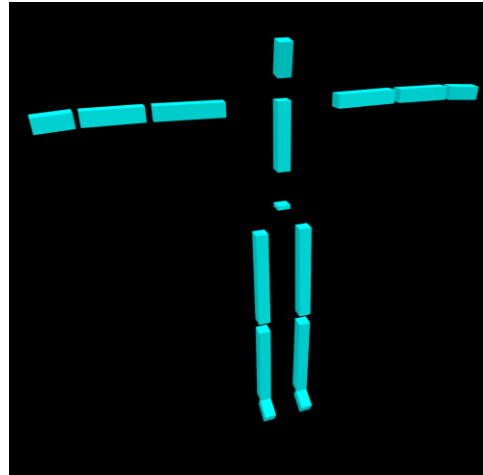


Skeletal Animation

Skeletal Animation

- Skeletal animation consists of two parts:

- Skeleton: static data

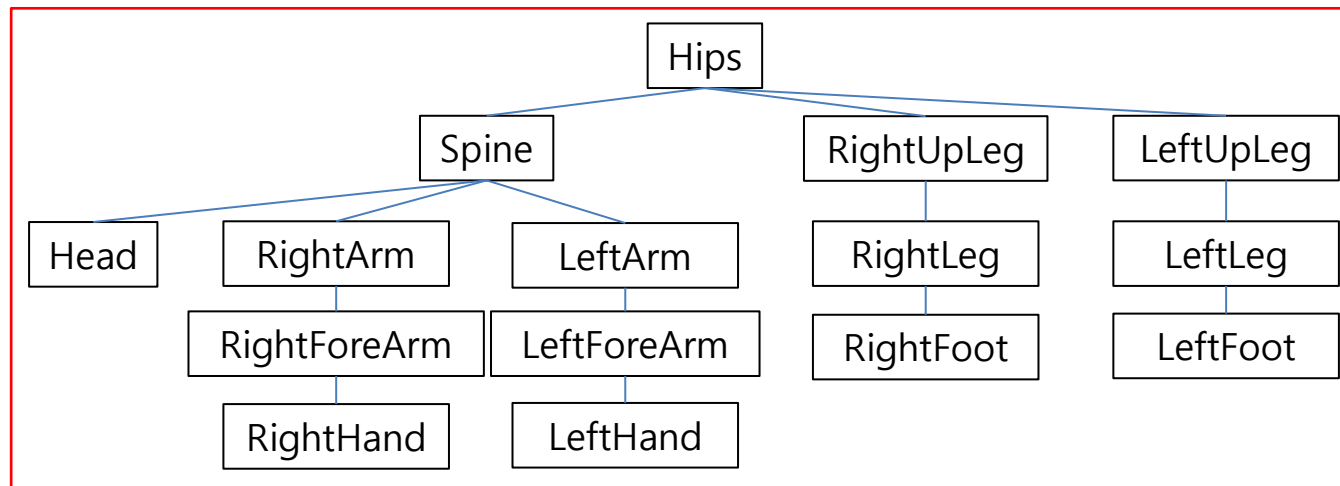
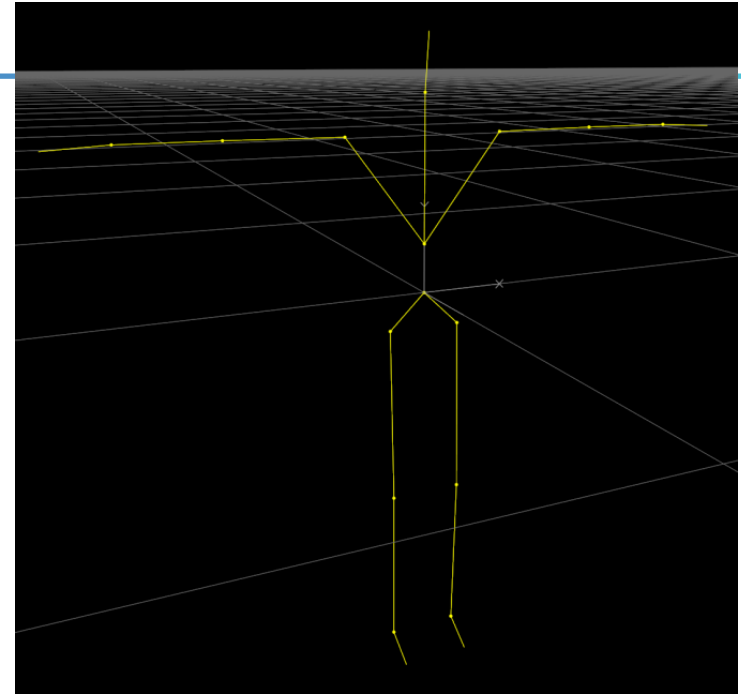


- Motion: time-varying data



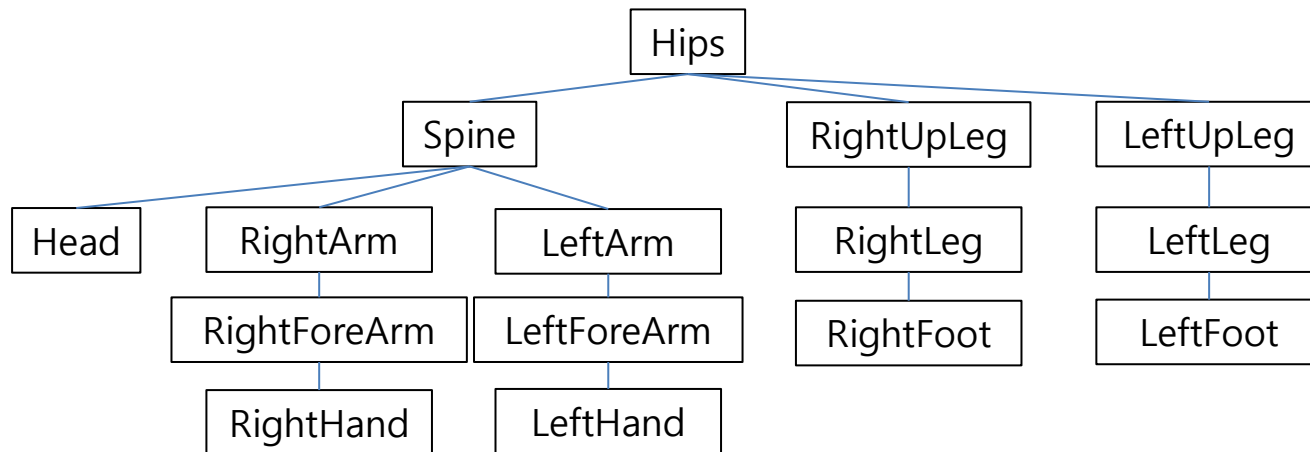
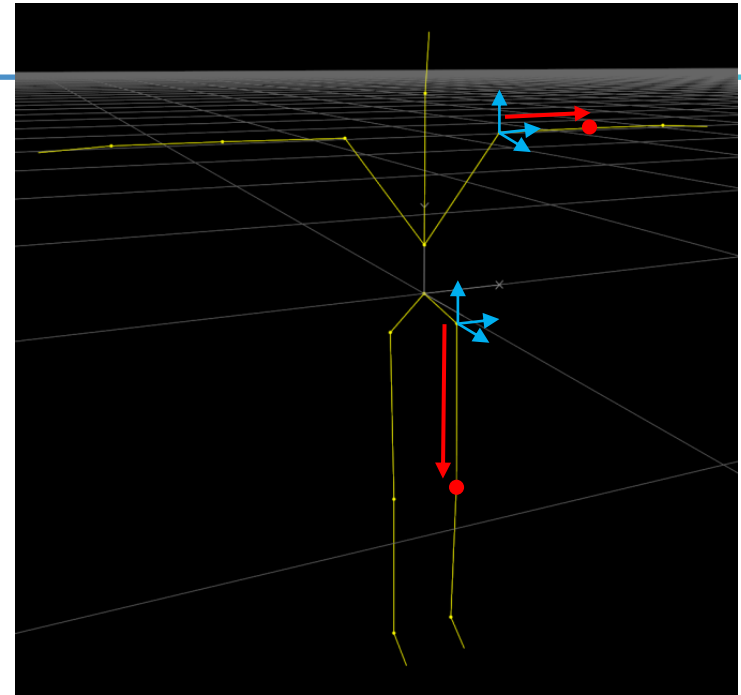
Skeleton Part

- Skeleton: static data
 - Joint hierarchy
 - Joint offset from its parent joint
 - usually translation
 - expressed in the parent frame

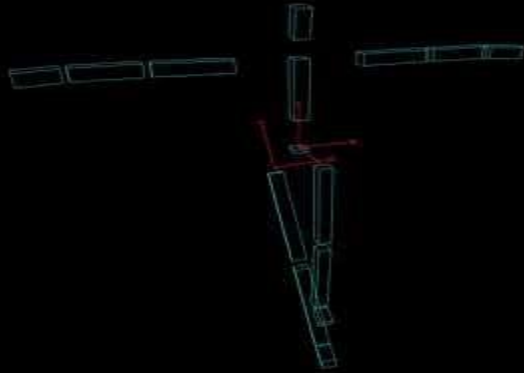


Skeleton Part

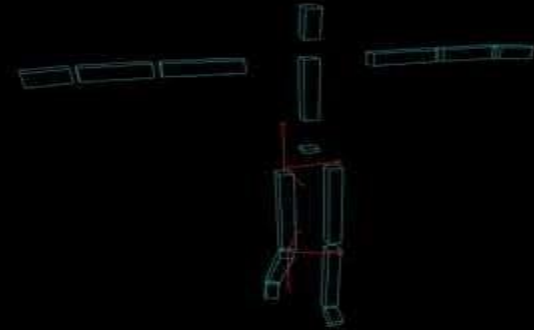
- Skeleton: static data
 - Joint hierarchy
 - Joint offset from its parent joint
 - usually translation
 - expressed in the parent frame



Recall: Hierarchical Model - Human Figure



<https://youtu.be/Q7lhvMkCSCg>

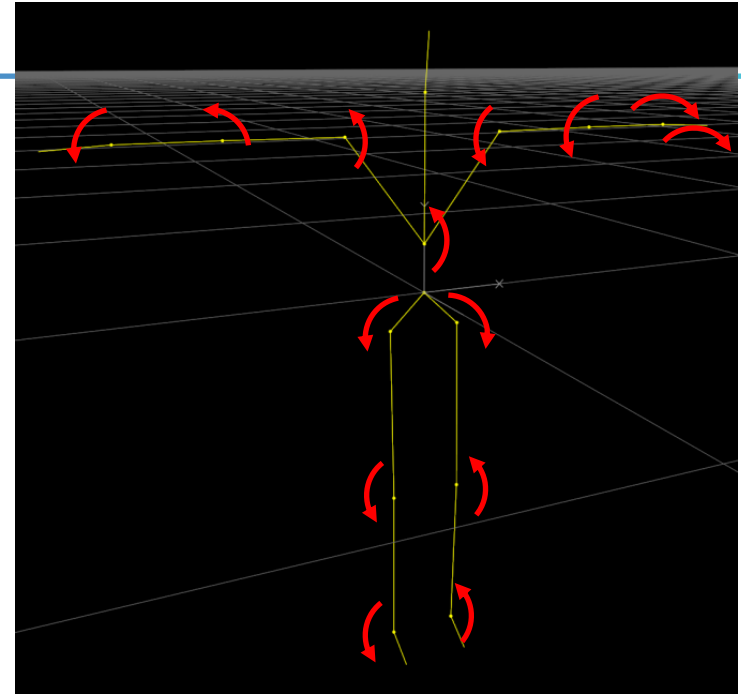


<https://youtu.be/Q5R8WGUwpFU>

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

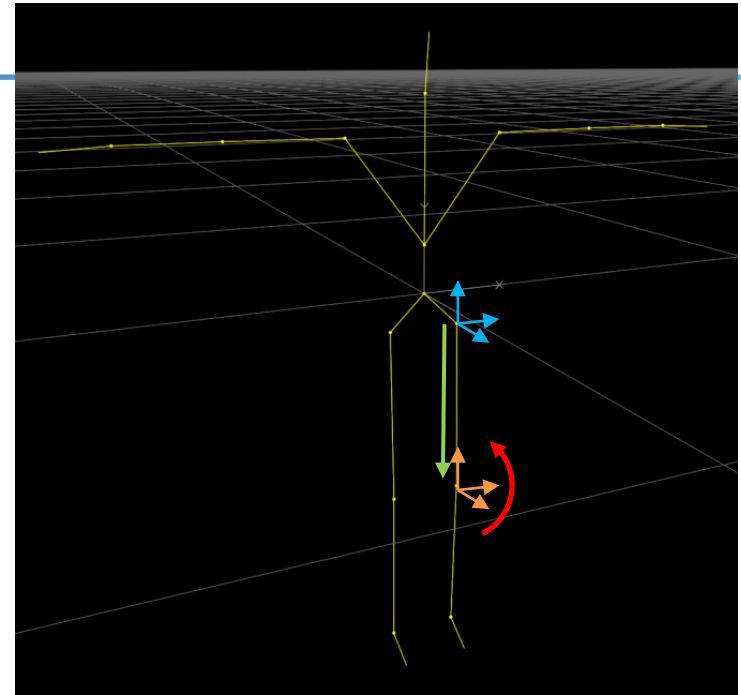
Motion Part

- Motion: time-varying data
 - Movement of internal joints
 - usually rotation
 - expressed in each joint's *default (rest) frame*
 - the frame obtained after applying joint offset to the parent frame
 - Movement of skeletal root
 - translation and rotation
 - expressed in the world frame



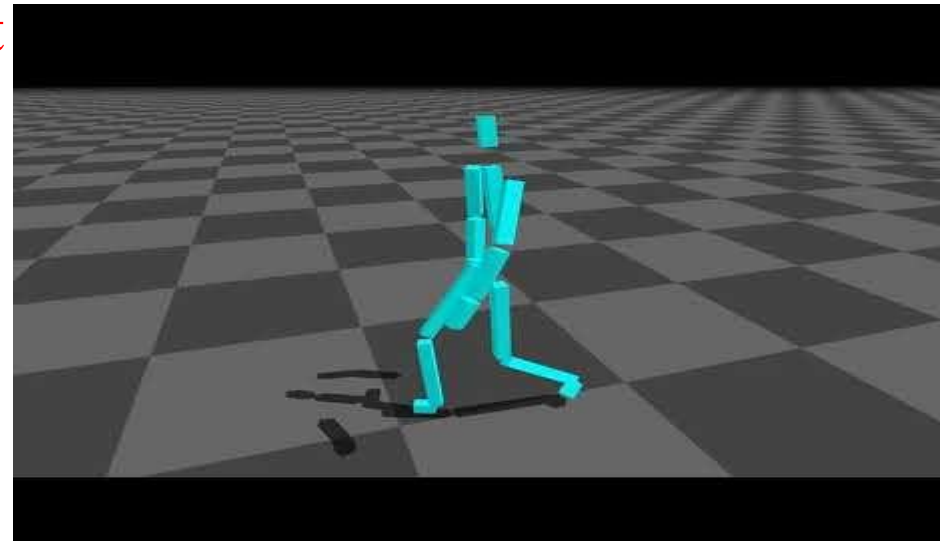
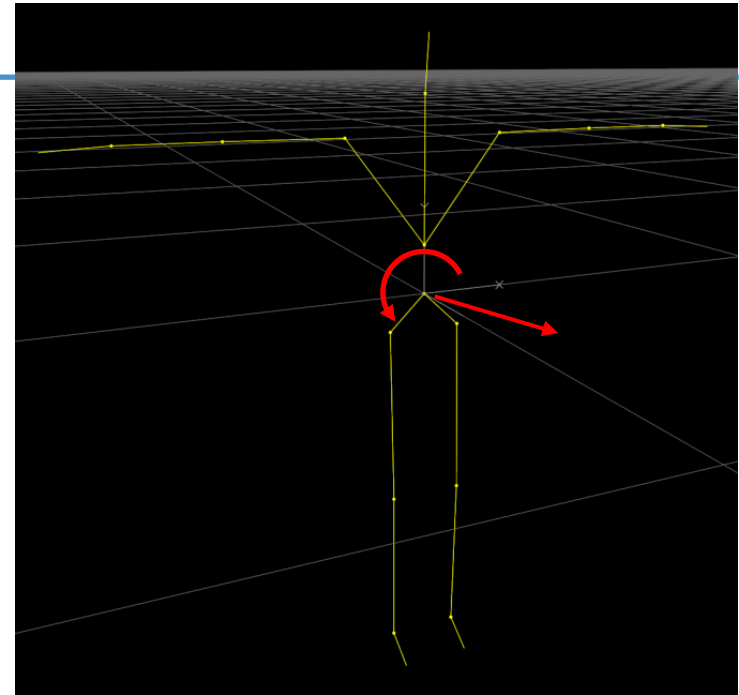
Motion Part

- Motion: time-varying data
 - Movement of internal joints
 - usually rotation
 - expressed in each joint's *default (rest) frame*
 - the frame obtained after applying *joint offset* to the *parent frame*
 - Movement of skeletal root
 - translation and rotation
 - expressed in the world frame



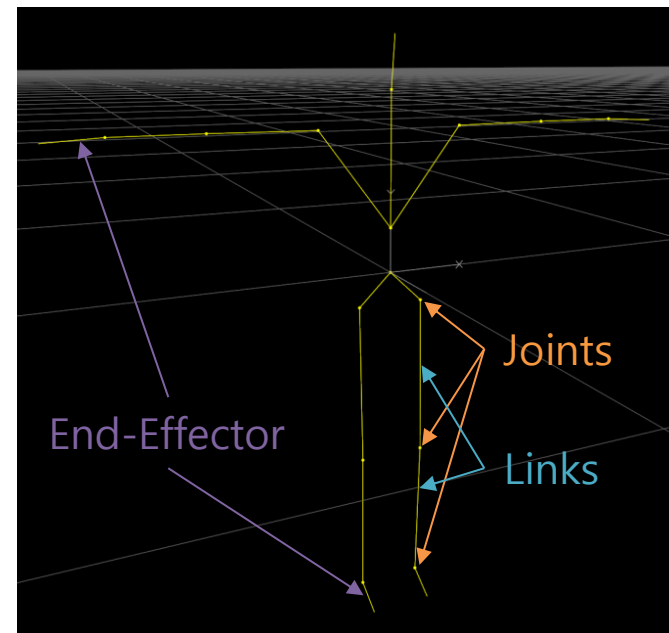
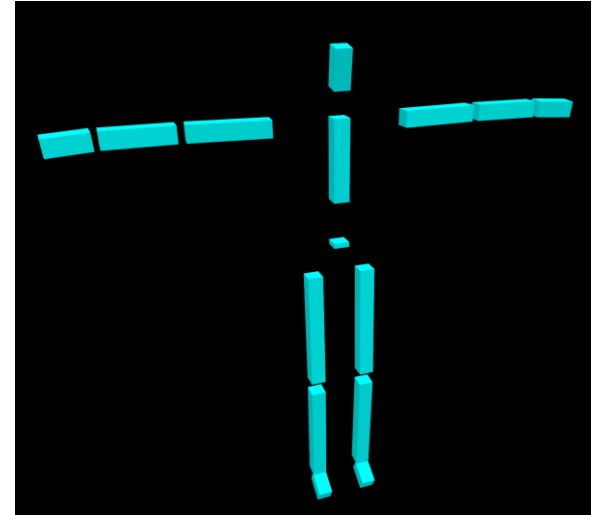
Motion Part

- Motion: time-varying data
 - Movement of internal joints
 - usually rotation
 - expressed in each joint's *default (rest) frame*
 - the frame obtained after applying joint offset to the parent frame
 - Movement of skeletal root
 - translation and rotation
 - expressed in the world frame



Terminologies

- ***Posture (pose)***: motion data at a single frame
- ***Rest pose***: a pose where all joint motions are "zero"
- ***Joint*** - a connection between two objects which allows relative movement
- ***Link*** - a rigid object between joints
- ***End effector*** - a free end of a kinematic chain (e.g., hand or foot)

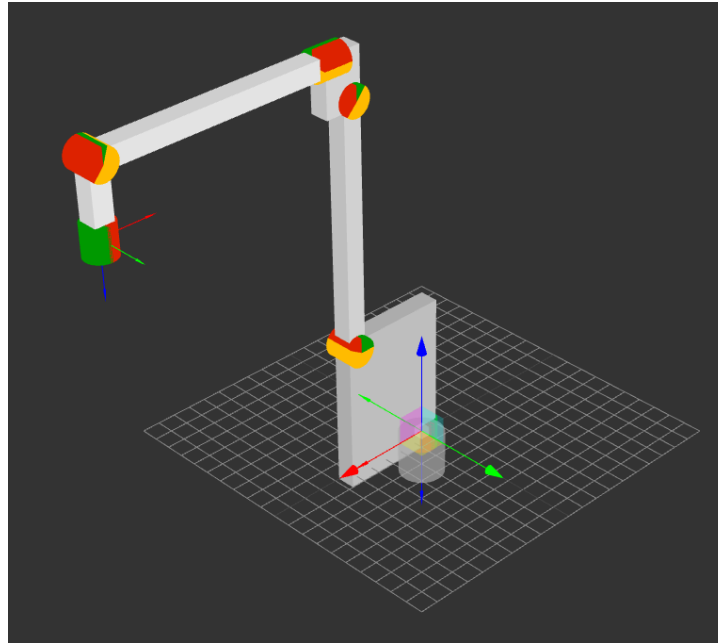


Forward Kinematics

Kinematics

- *Kinematics* is the study of motion 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 commonly used for 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**
- Forward kinematics is a fundamental concept in skeletal animation.

[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 (FK)

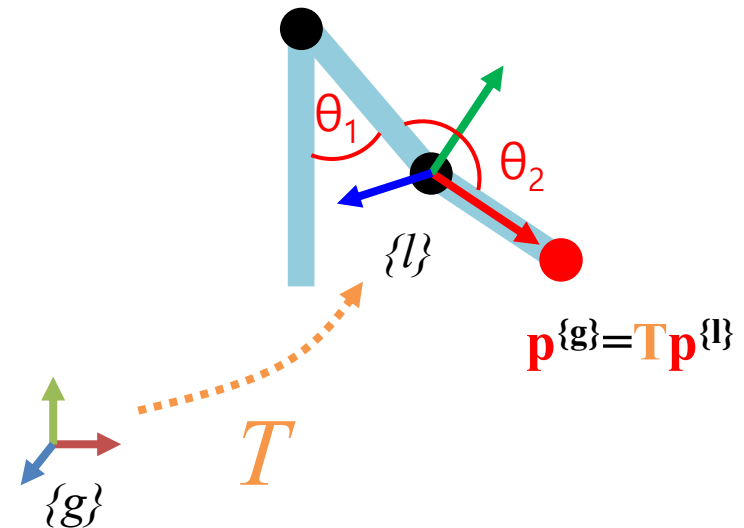
- Forward kinematics computes transformations T :
 - defining a joint frame w.r.t. the world frame,
 - and transforming points defined relative to each joint frame into the world frame.

- T is determined by:

- joint hierarchy
- joint offsets
- joint angles

- Using T , we can compute:

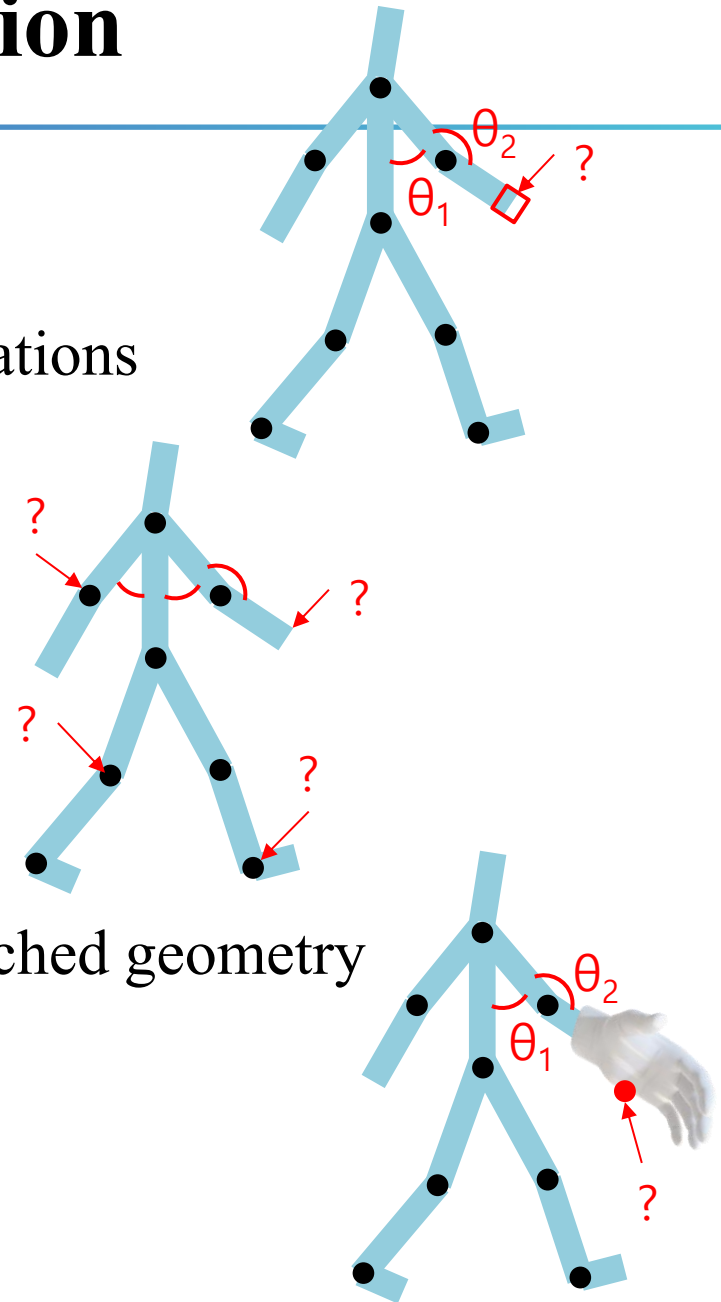
- global positions & orientations of each joint



- skeleton + joint angles $\rightarrow$ global positions & orientations of all joints

FK for Skeletal Animation

- FK is often used to compute:
 - end-effector positions & orientations
 - joint positions & orientations
 - positions & orientations of attached geometry



Forward Kinematics Map

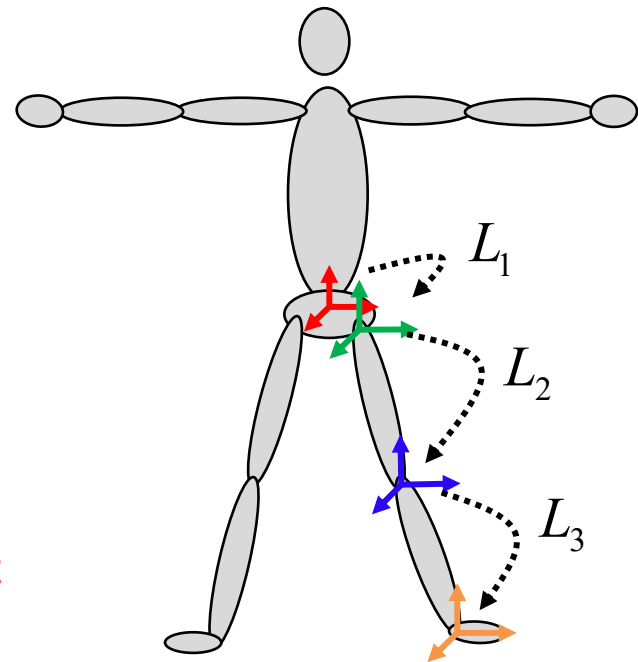
- A *forward kinematics map* T is an alternating product of ...
- **Joint transformations** (time-varying)
 - : Joint movement (*motion*)
- **Link transformations** (static)
 - : Joint offset (*skeleton*)

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

translation to 1st joint

rotation of 1st joint

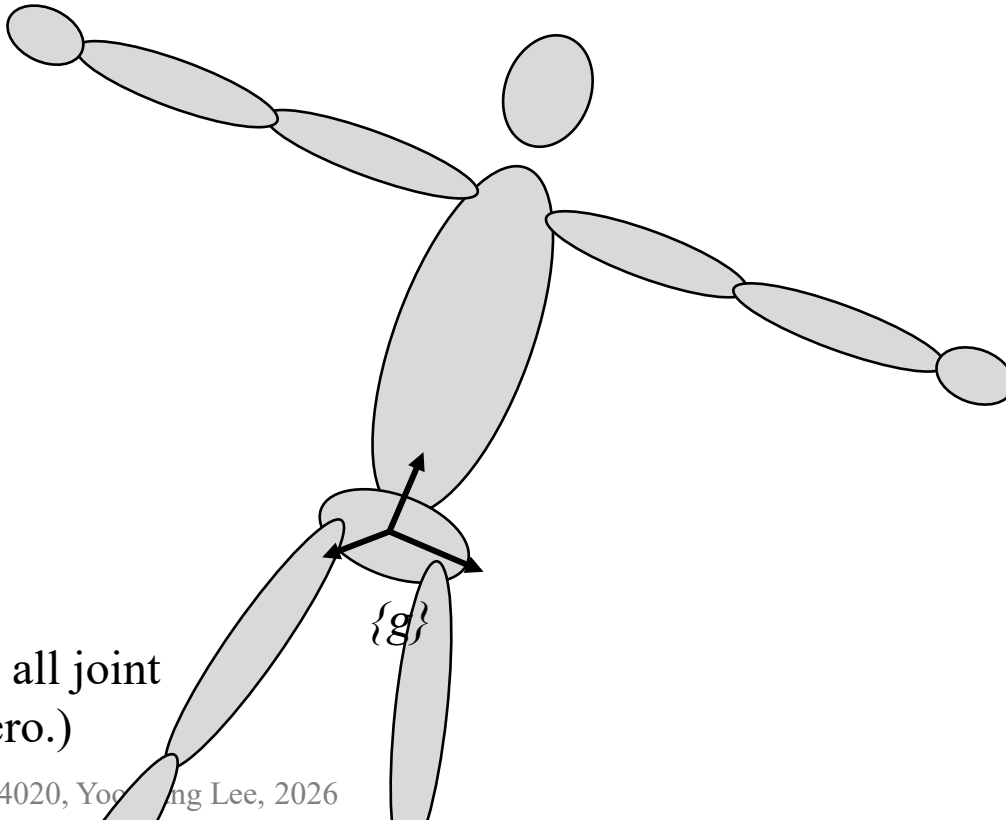
position & orientation of 0th joint
(the root segment)



Forward Kinematics Map

- Initialize accumulated transform
- Initially, the current frame is the world frame.

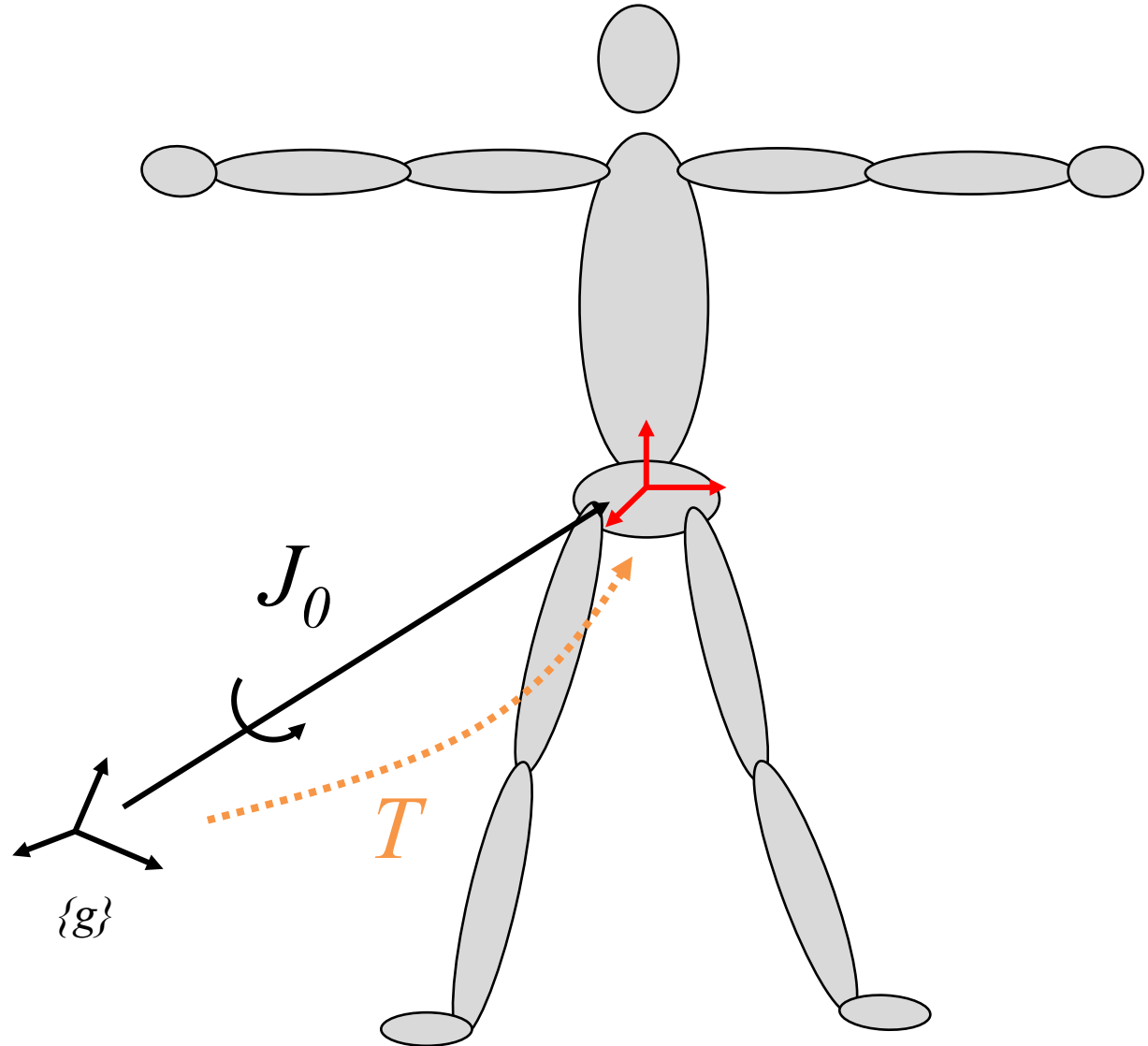
$$T = I$$



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

Forward Kinematics Map

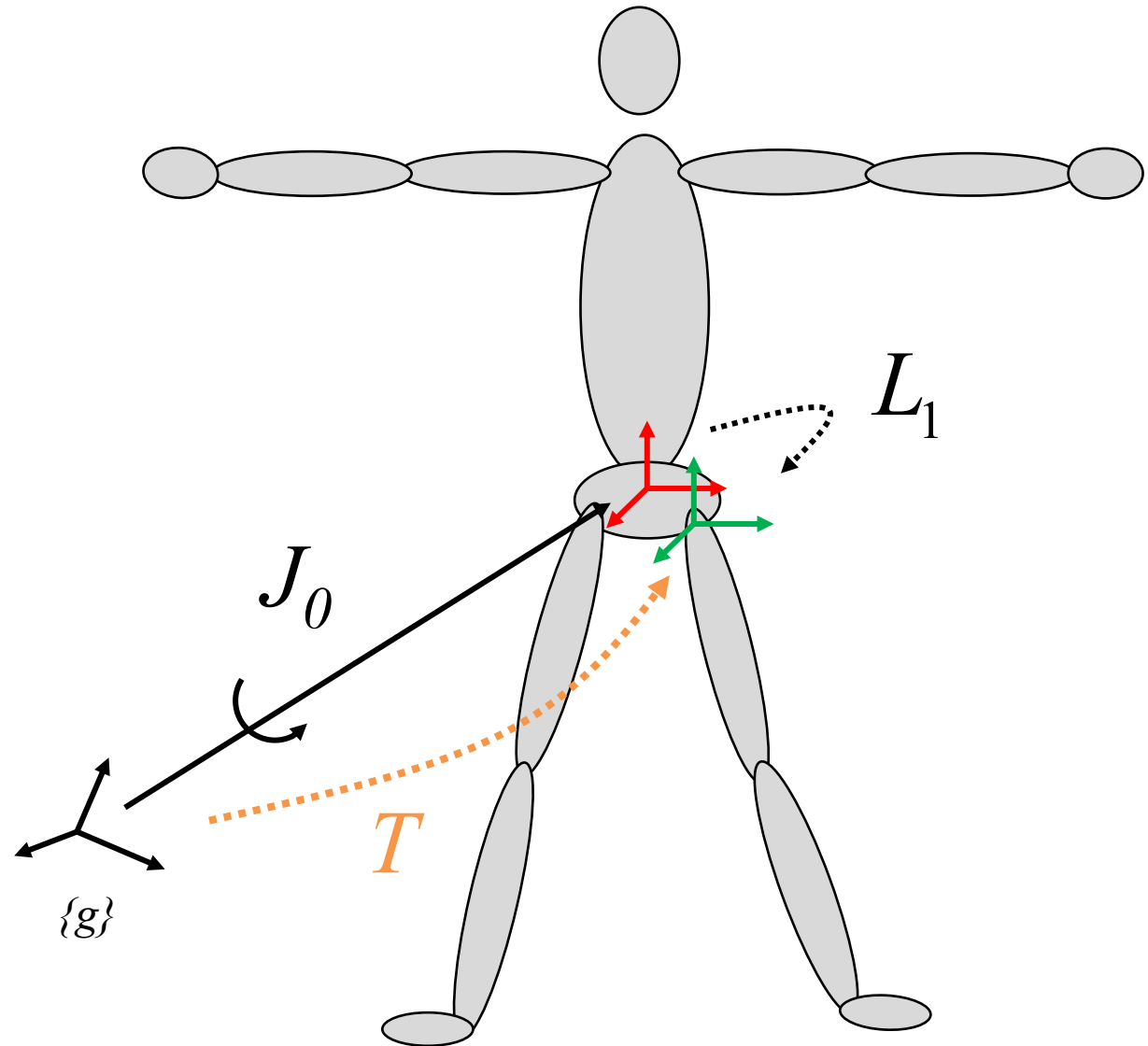
$$T = J_0$$



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

Forward Kinematics Map

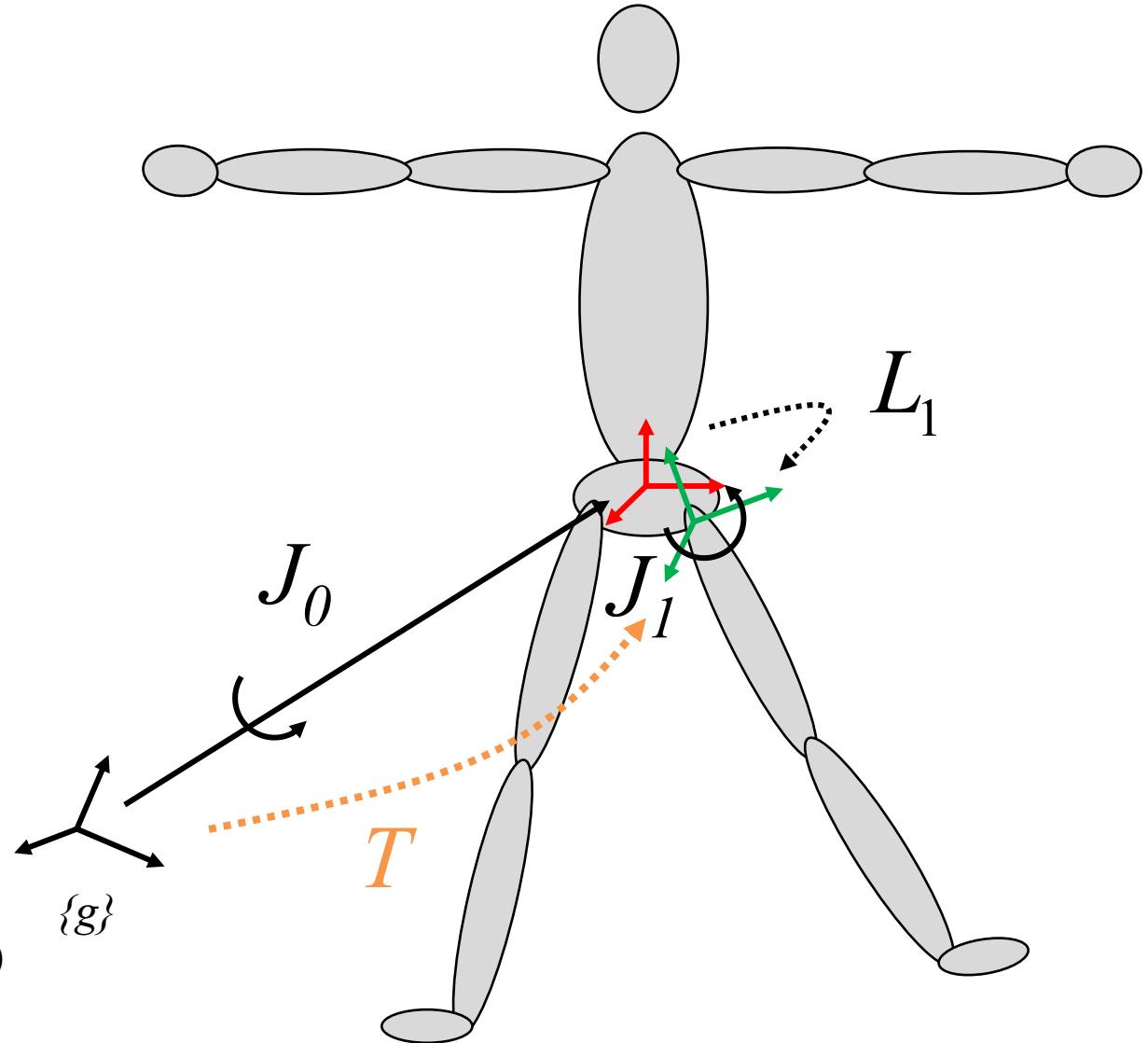
$$T = J_0 L_1$$



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

Forward Kinematics Map

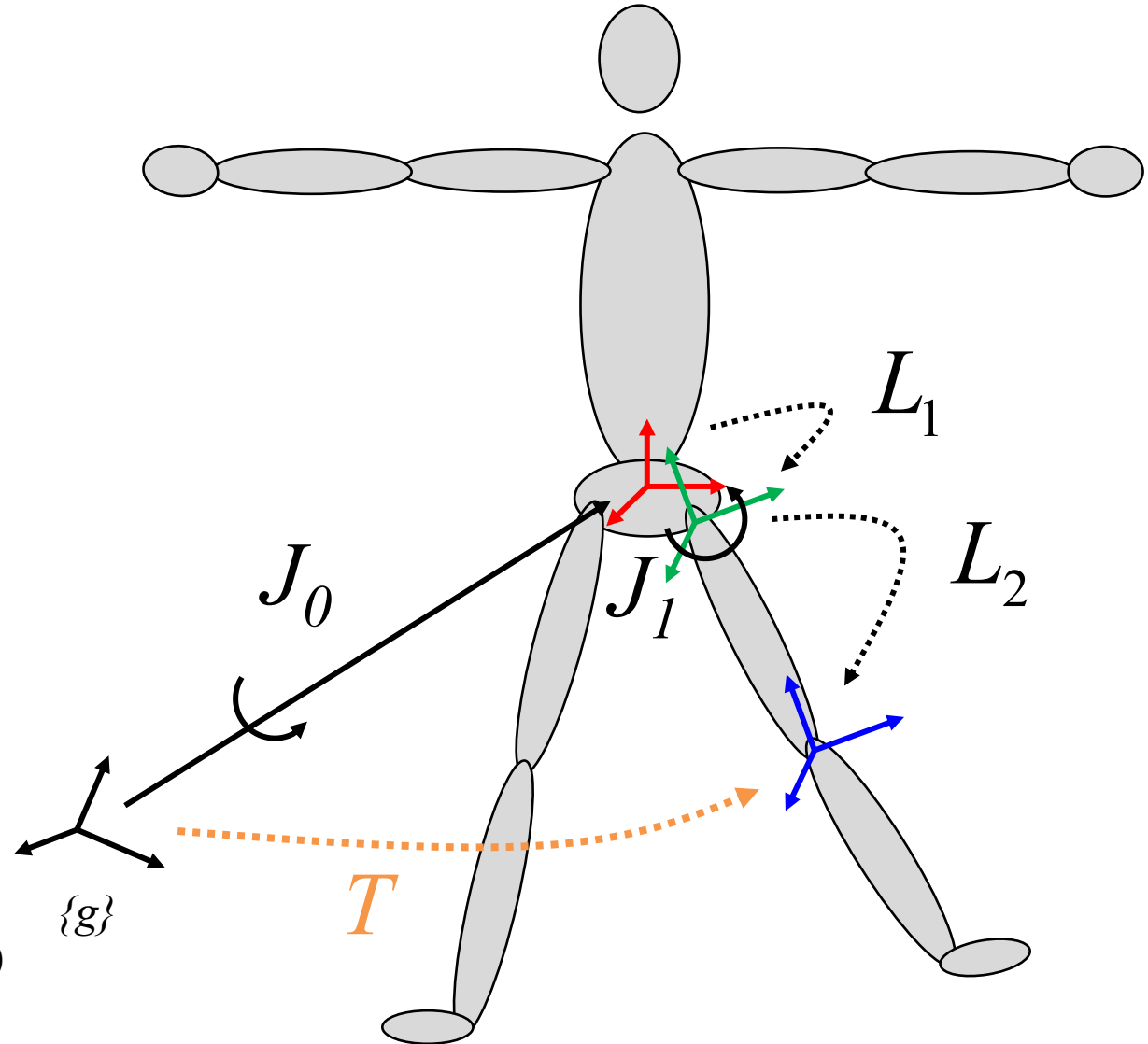
$$T = J_0 L_1 J_1$$



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

Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2$$

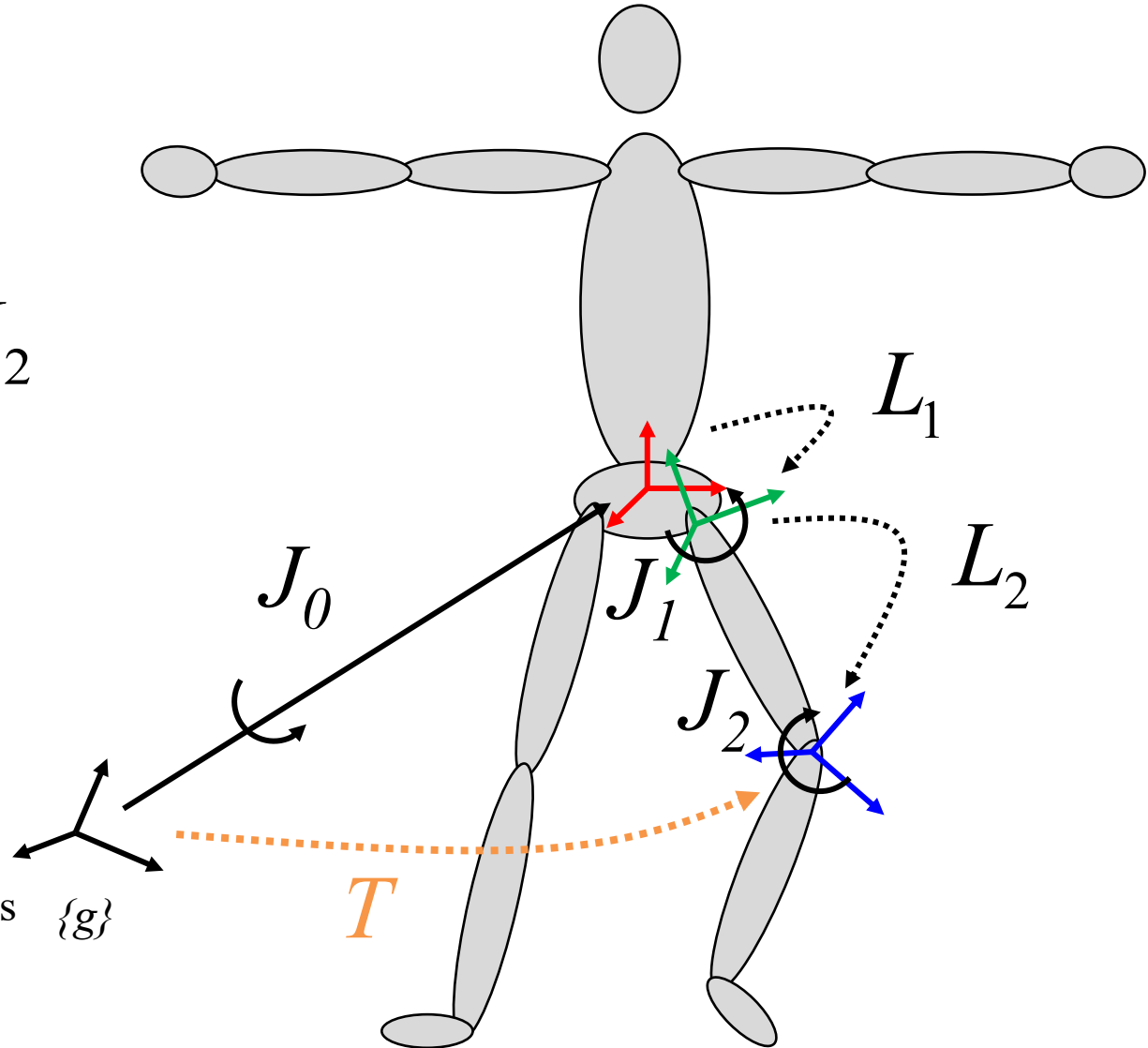


(In this figure, all joint motions except J_0 , J_1 are zero.)

Forward Kinematics Map

$$T = J_0 L_1 J_1 L_2 J_2$$

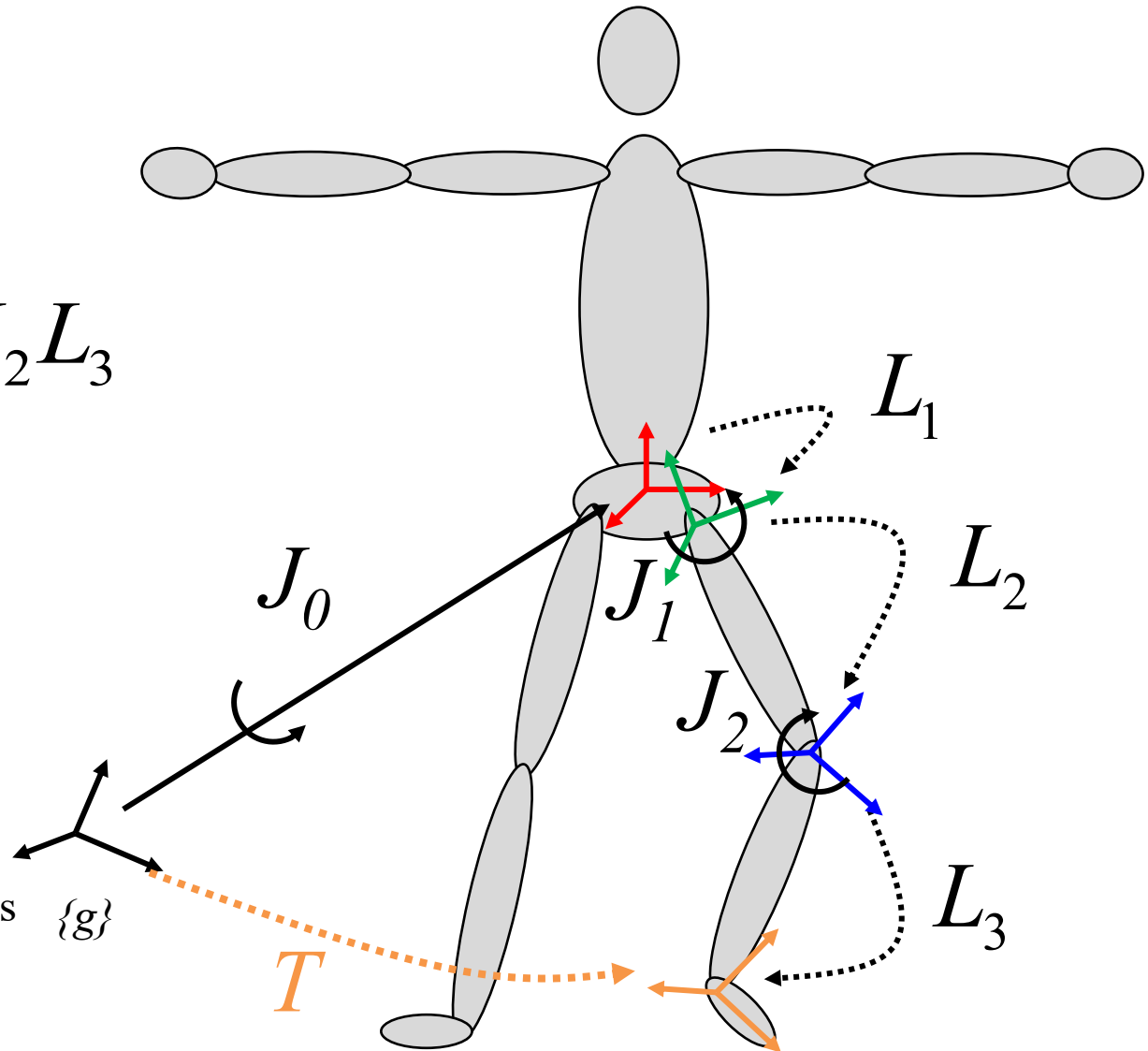
(In this figure, all joint motions except J_0, J_1, J_2 are zero.)



Forward Kinematics Map

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

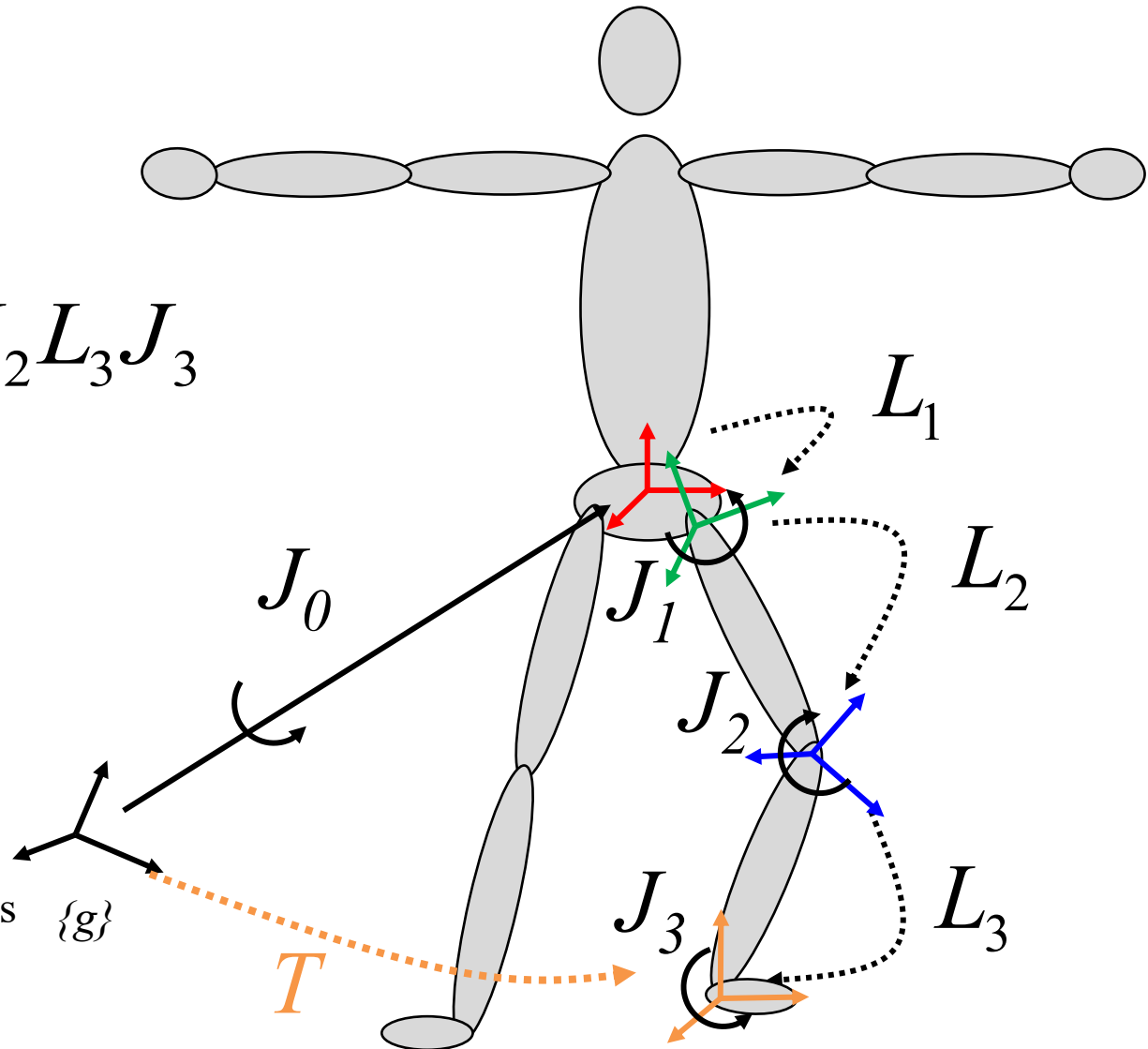
(In this figure, all joint motions except J_0, J_1, J_2 are zero.)



Forward Kinematics Map

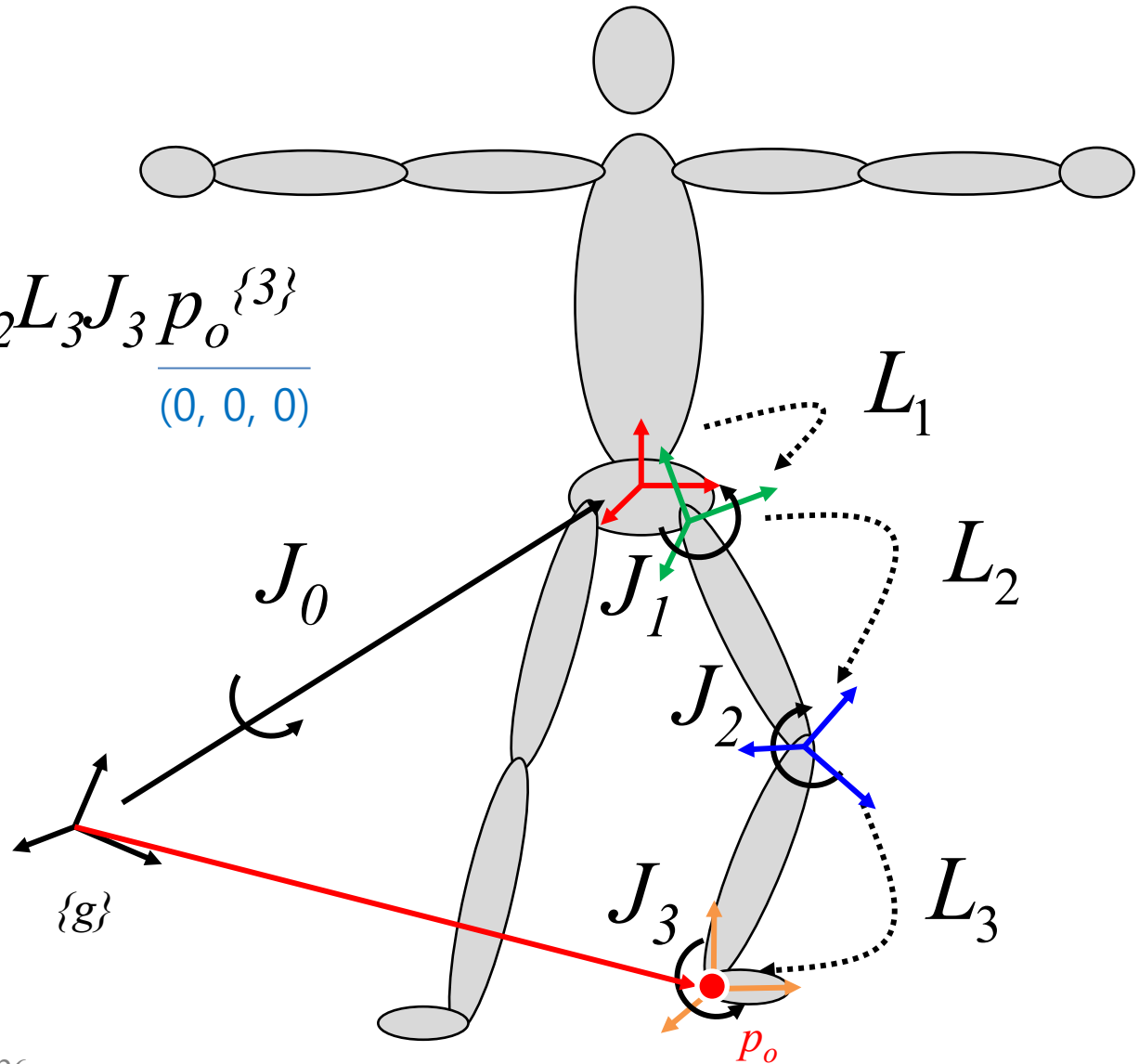
$$T = J_0 L_1 J_1 L_2 J_2 L_3 J_3$$

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



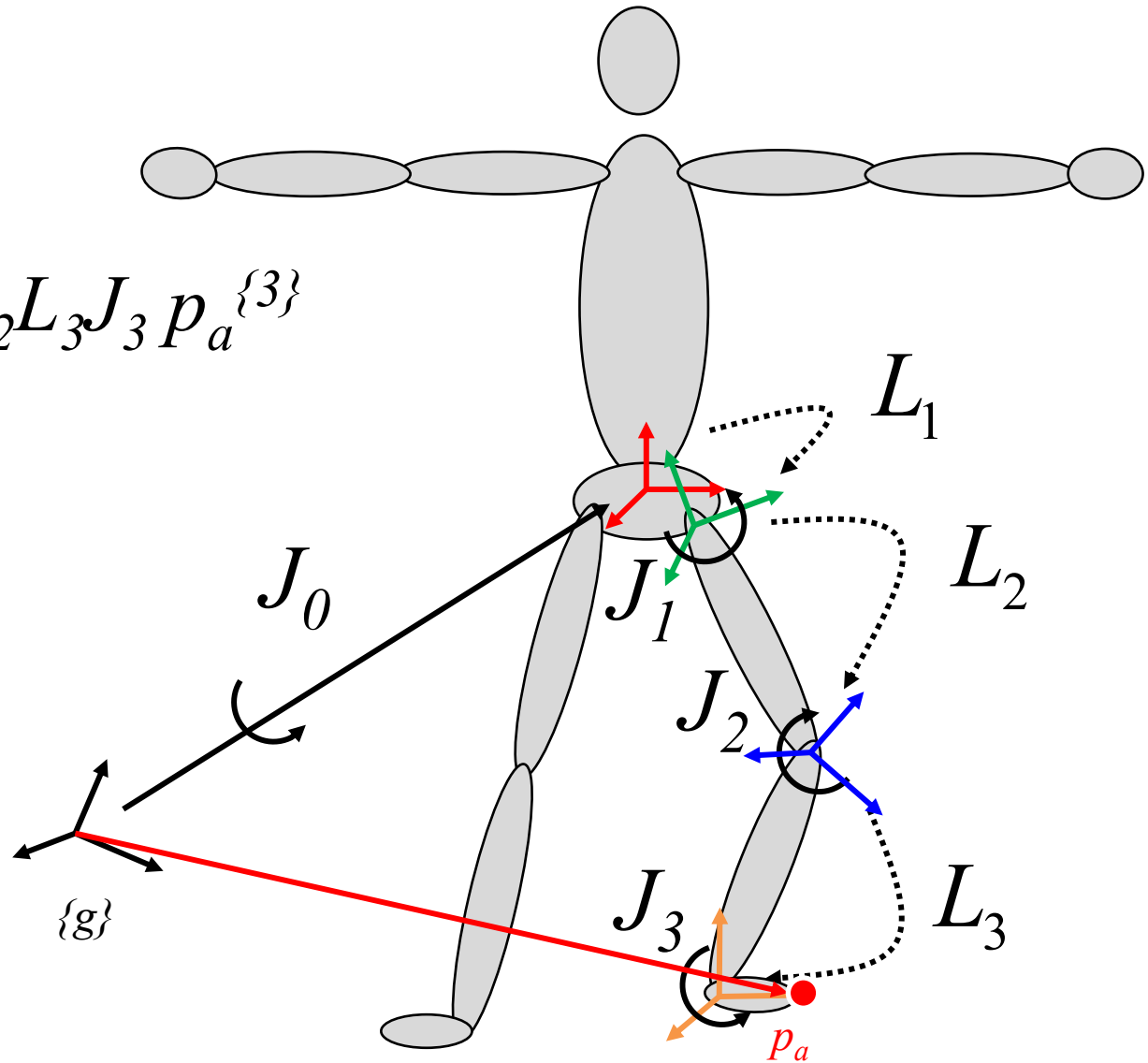
Forward Kinematics Map

$$p_o^{\{g\}} = J_0 \underbrace{L_1 J_1 L_2 J_2 L_3 J_3}_{(0, 0, 0)} p_o^{\{3\}}$$



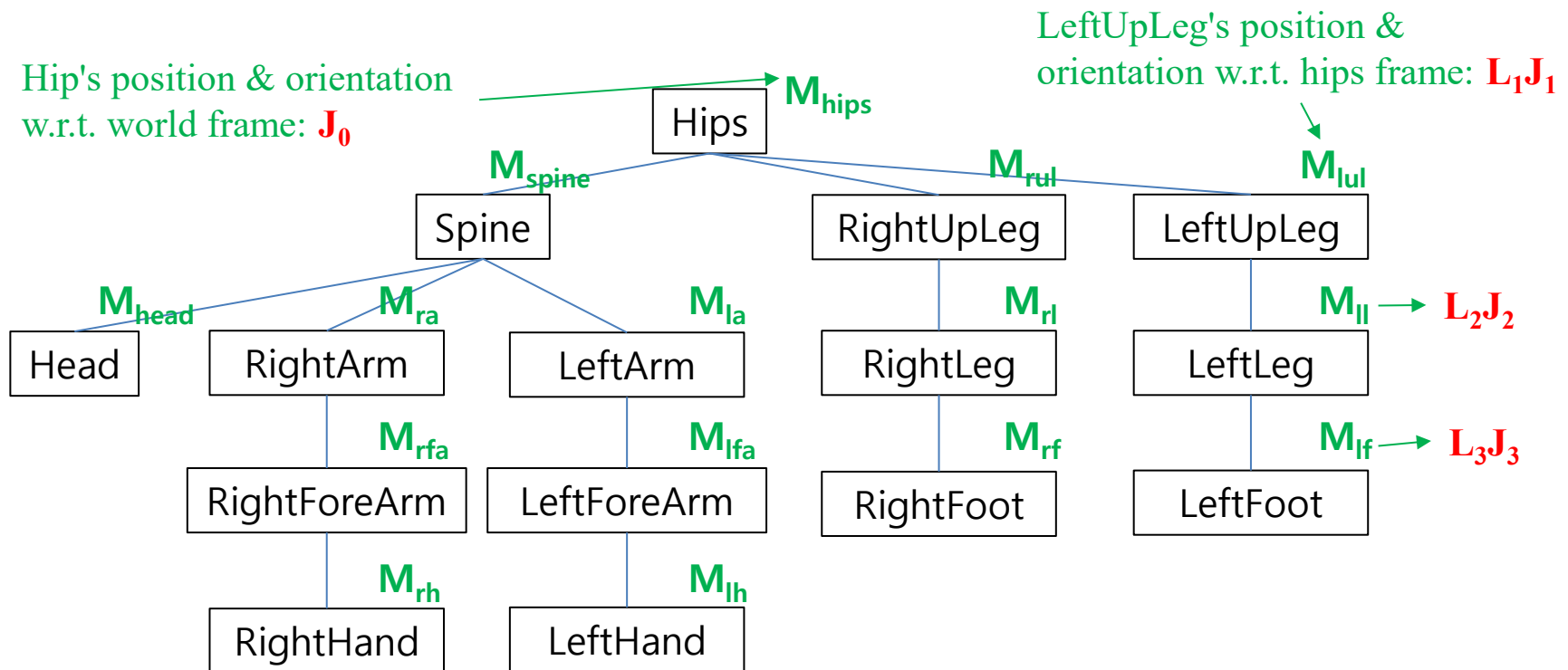
Forward Kinematics Map

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



Recall: Local Transformation

- Each node has its **own transformation** relative to its parent.
- **Local transformation**



Skeletal Animation

- Skeletal animation is defined by:
- Joint hierarchy
- Static link transformations L_i
 - joint offsets
 - $\rightarrow$ Skeleton
- Time-varying joint transformations J_i
 - joint motions / rotations
 - $\rightarrow$ 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. **2024123456: 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. **2024123456: 4.0**
- Note that your quiz answer must be submitted **in the above format** to be counted for attendance.

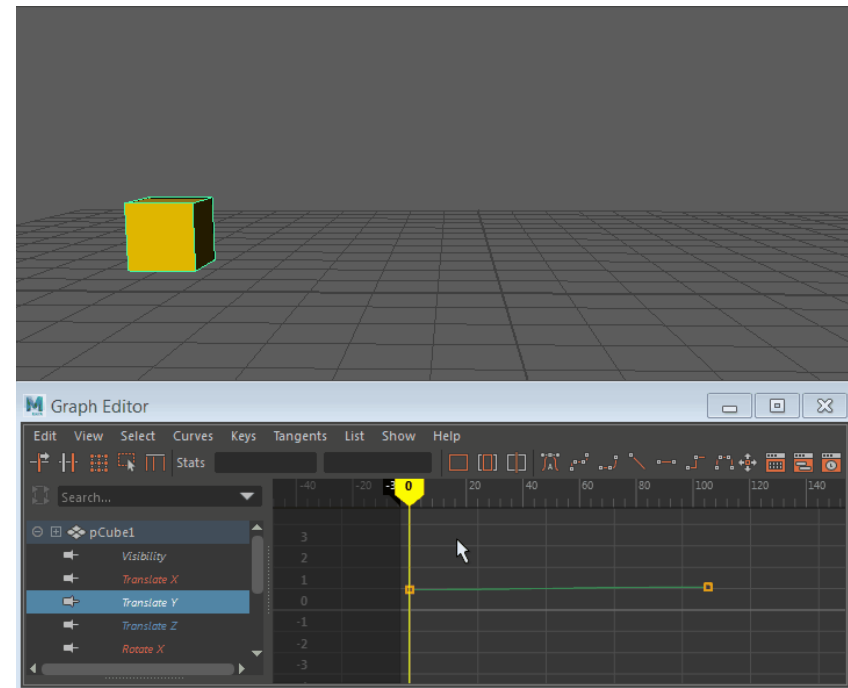
Creating Character Animation

Methods for Creating Character Animation

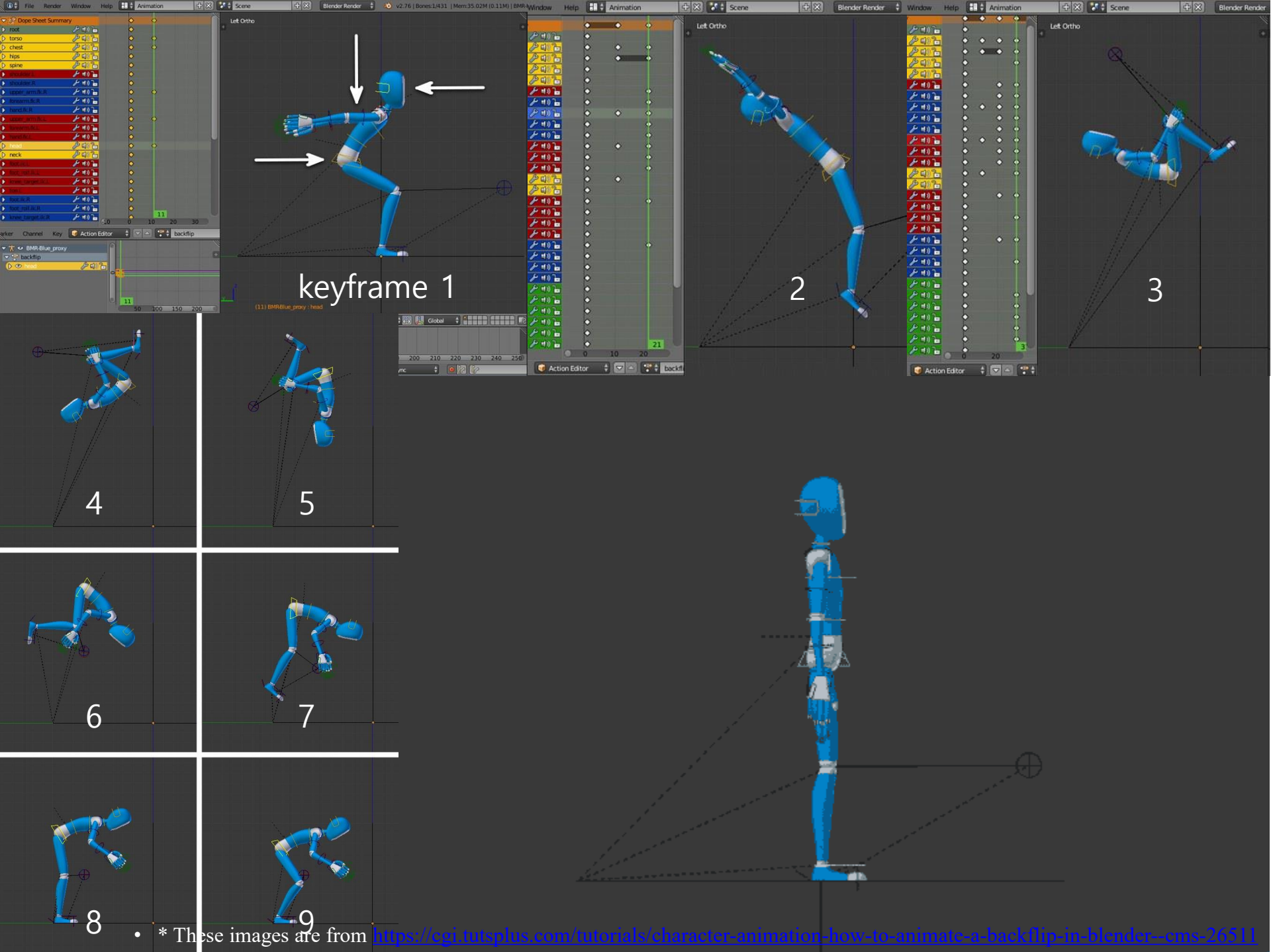
- How are skeletons and motions created in practice?
- Manual animation
 - Keyframe Animation
- Captured animation
 - Motion Capture
- Data-Driven Animation
- Physics-Based Animation

Keyframe Animation

- Animators specify important poses at *keyframes*.
- Motion between keyframes is generated by interpolation.
 - Interpolation automates the traditional “inbetweening” process.
- Skeletons are typically rigged manually.
- Motion is controlled via time-varying parameter curves with keyframes.



* This image is from <https://help.autodesk.com/view/MAYAUL/2022/ENU/?guid=GUID-E9E69ABB3-E94B-4D67-9935-40F77FD2E9E0>



Keyframe Animation

- A traditional method for computer animation,
 - rooted in hand-drawn animation.
- Highly controllable and artist-friendly; therefore still widely used in games and films.
- Difficult to create realistic and physically plausible motions.
 - Output quality largely depends on the animator's skill.

Motion Capture

- Use measured real-world motion to create realistic animation.
- Motion capture systems record:
 - marker trajectories on the body
 - body/joint positions and orientations
- Through post-processing:
 - a skeleton is reconstructed
 - motion parameters are extracted

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 ...  
...
```

Consists of two parts:

Hierarchy section

Describes *Skeleton*: static data

Motion section

Describes *Motion*: time-varying data

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 "**

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

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

in this example,

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

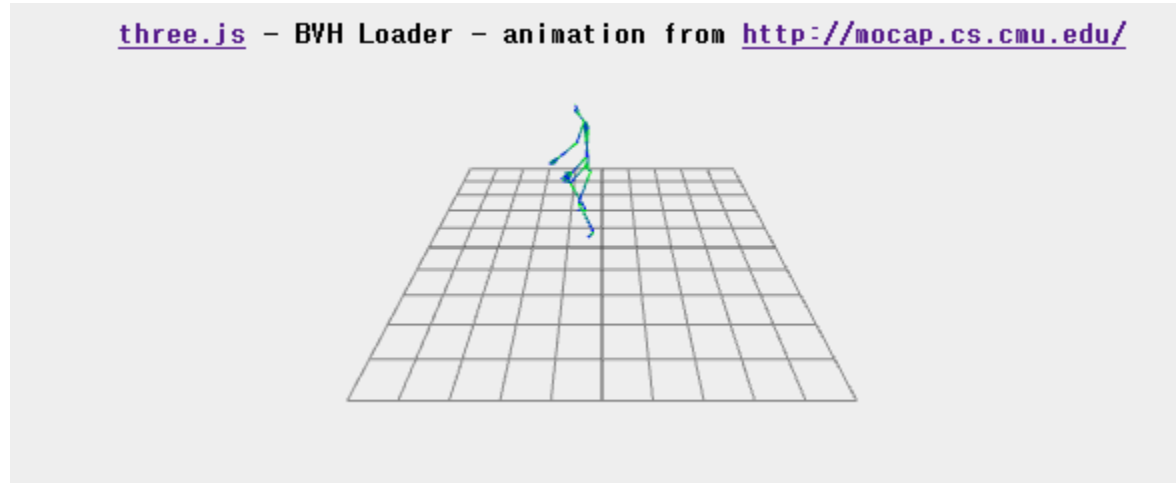
Frame Time: 0.033333

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. **2024123456: 4.0**
- Note that your quiz answer must be submitted **in the above format** to be counted for attendance.

Motion Capture

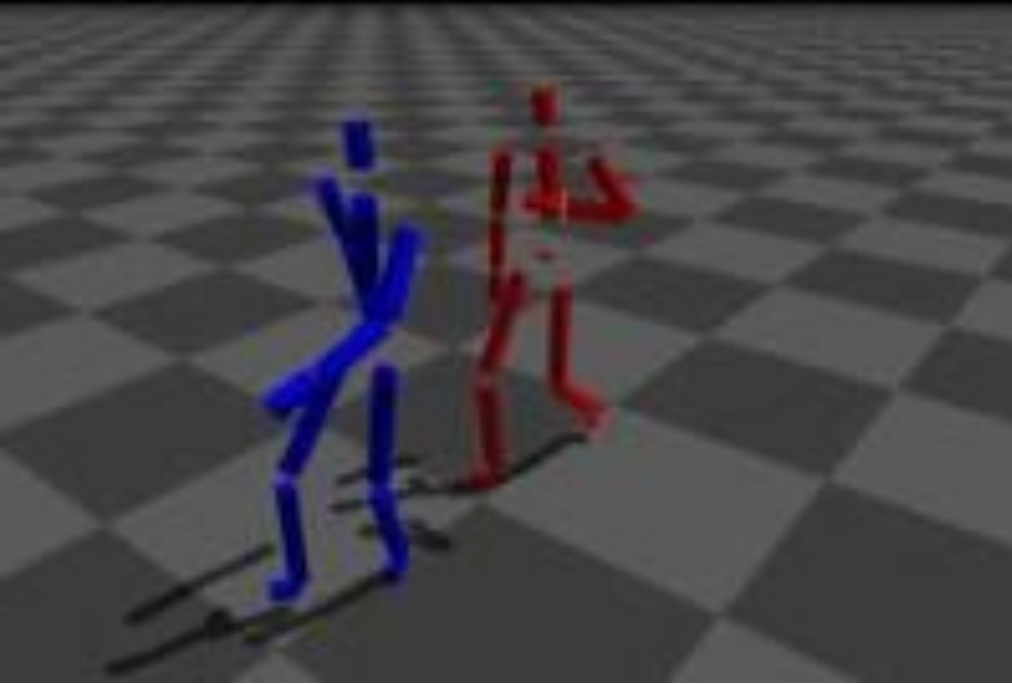
- Widely used in movies and AAA games.
- Produces highly realistic motion.
- Expensive:
 - capture equipment
 - studio setup and operation
 - post-processing / cleanup
- Limitation:
 - captured motion is often specialized to the capture environment and task.

Data-Driven Animation

- Goal: reuse and generalize motion data to generate flexible motion behaviors
- Motion data is:
 - edited
 - blended
 - restructured
 - learned & synthesized
- Typical approaches:
 - motion editing / blending
 - 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: Generate motion using physics simulation.
 - Physical reality plays a key role in high-quality animation.
 - Physics simulation enforces physical constraints.
- Skeleton / body model: Usually defined using articulated rigid-body models
 - e.g., URDF, MuJoCo XML, Isaac Gym assets, ...
- Motion: Emerges from simulated dynamics and control inputs
- Controller / Policy
 - Computes control actions (torques, target pose, etc) at each timestep
 - Maintains balance while performing desired behaviors
- Modern trend
 - Deep reinforcement learning (DRL) and imitation learning are widely used.
 - Physics simulation is widely used in robotics and embodied AI research.

DeepMimic [Peng et al. 2018]

<https://youtu.be/vppFvq2quQ0>



PhysicsFC [Kim et al. 2025]

<https://youtu.be/dSfW8EvV5CA>



Beyond Today's Character Animation

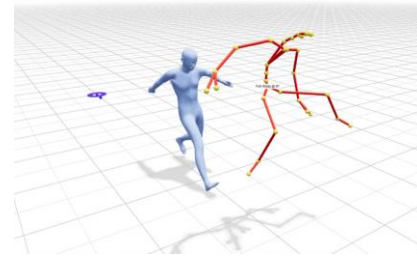
Modern Character Animation

- Animation is still an open problem.
- Key challenges:
 - controllability
 - environment adaptation
 - morphology generalization
 - expressive motion generation
 - physically grounded behaviors
 - intuitive user interaction
 - robust long-horizon control
 - ...
- Modern systems still rely on skeletons, transformations, and motion representations.

Current Research Directions

- Generative motion models

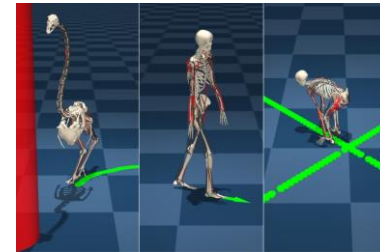
- diffusion models
- text-to-motion



Kimodo (2026)

- Physics-aware animation

- combining generative models with simulation
- balance and agile skill control



FreeMusco (2025)

- Embodied AI

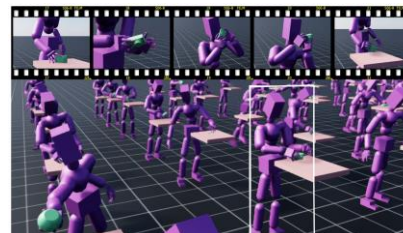
- robot learning
- sim-to-real transfer



HUSKY (2026)

- Human-scene interaction

- terrain adaptation
- object interaction



MaskedManipulator (2025)

Recent Work from Our Group

- Controllable
 - **PhysicsFC** (skillful interaction control)
 - **TouchWalker** (mapping user intent to motion)
- Adaptive
 - **FreeMusco** (motion-free, morphology-driven generalization)
- Expressive
 - **B2F** (from body to facial expression)

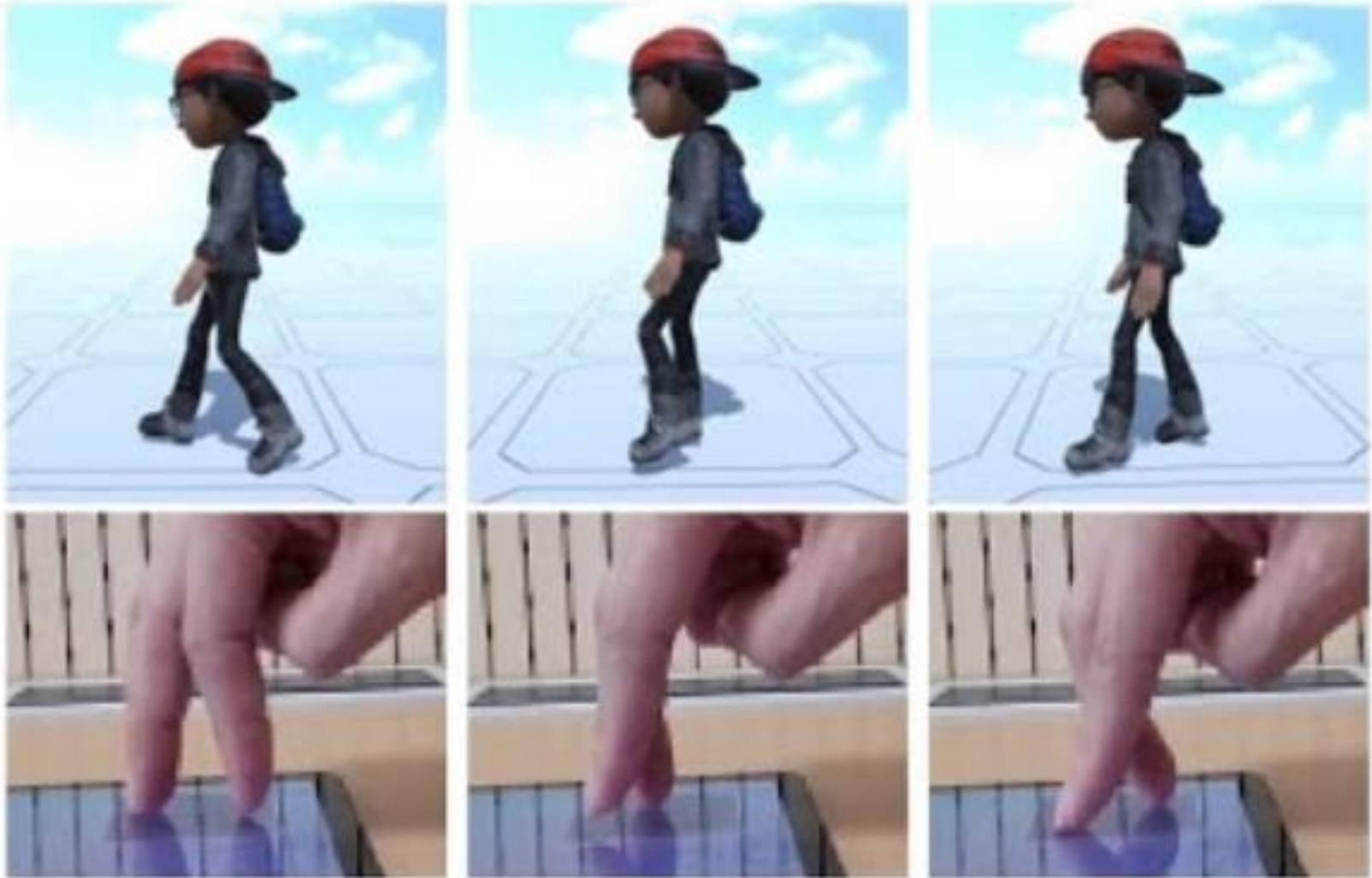
PhysicsFC: Learning User-Controlled Skills for a Physics-Based Football Player Controller



Minsu Kim, Eunho Jung, Yoonsang Lee. “PhysicsFC: Learning User-Controlled Skills for a Physics-Based Football Player Controller.” **SIGGRAPH 2025**

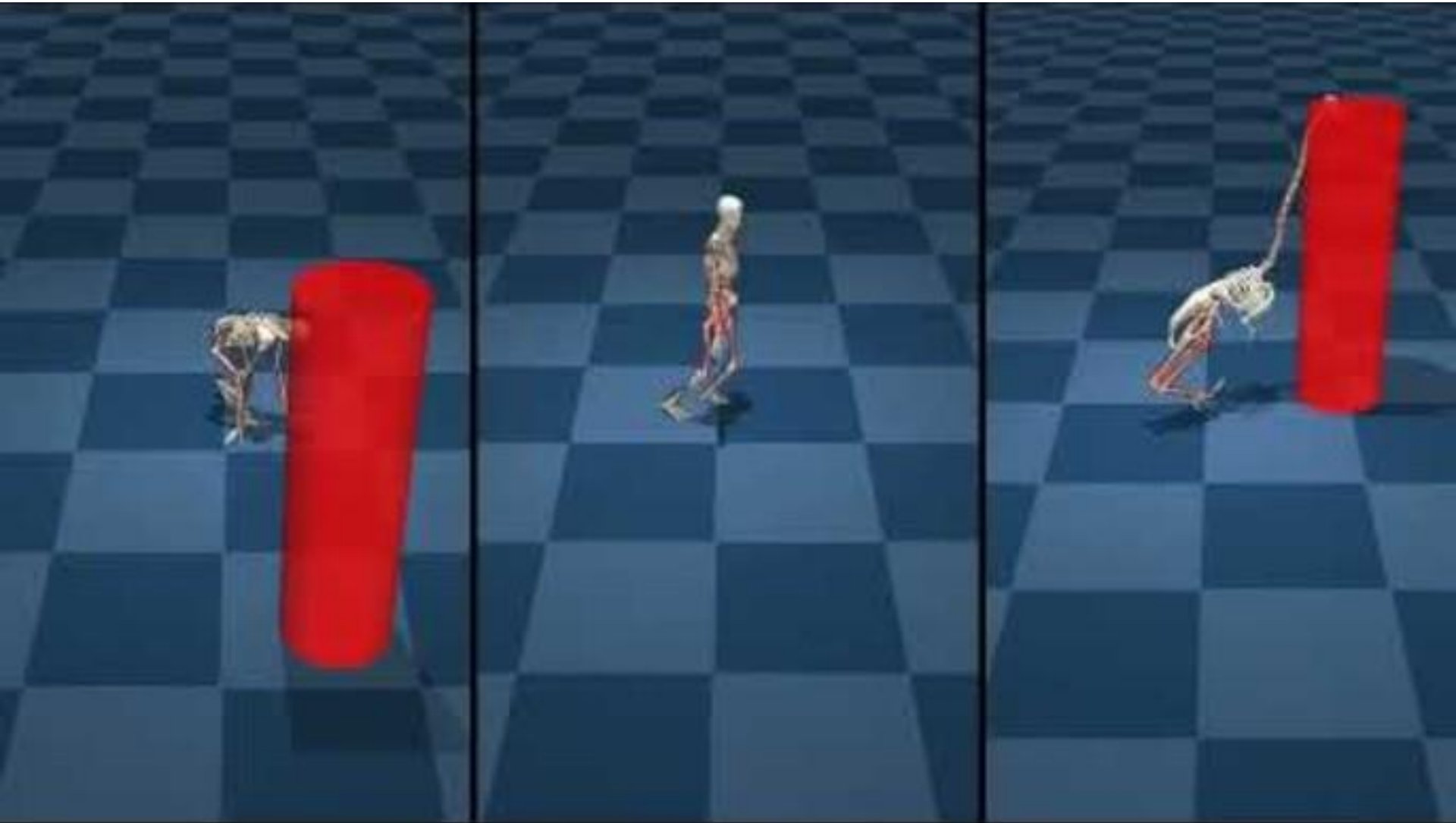
Hanyang University CSE4020, Yoonsang Lee, 2026

TouchWalker: Real-Time Avatar Locomotion from Touchscreen Finger Walking



Geuntae Park, Jiwon Yi, Taehyun Rhee, Kwanguk Kim, Yoonsang Lee. "TouchWalker: Real-Time Avatar Locomotion from Touchscreen Finger Walking." **ISMAR 2025**

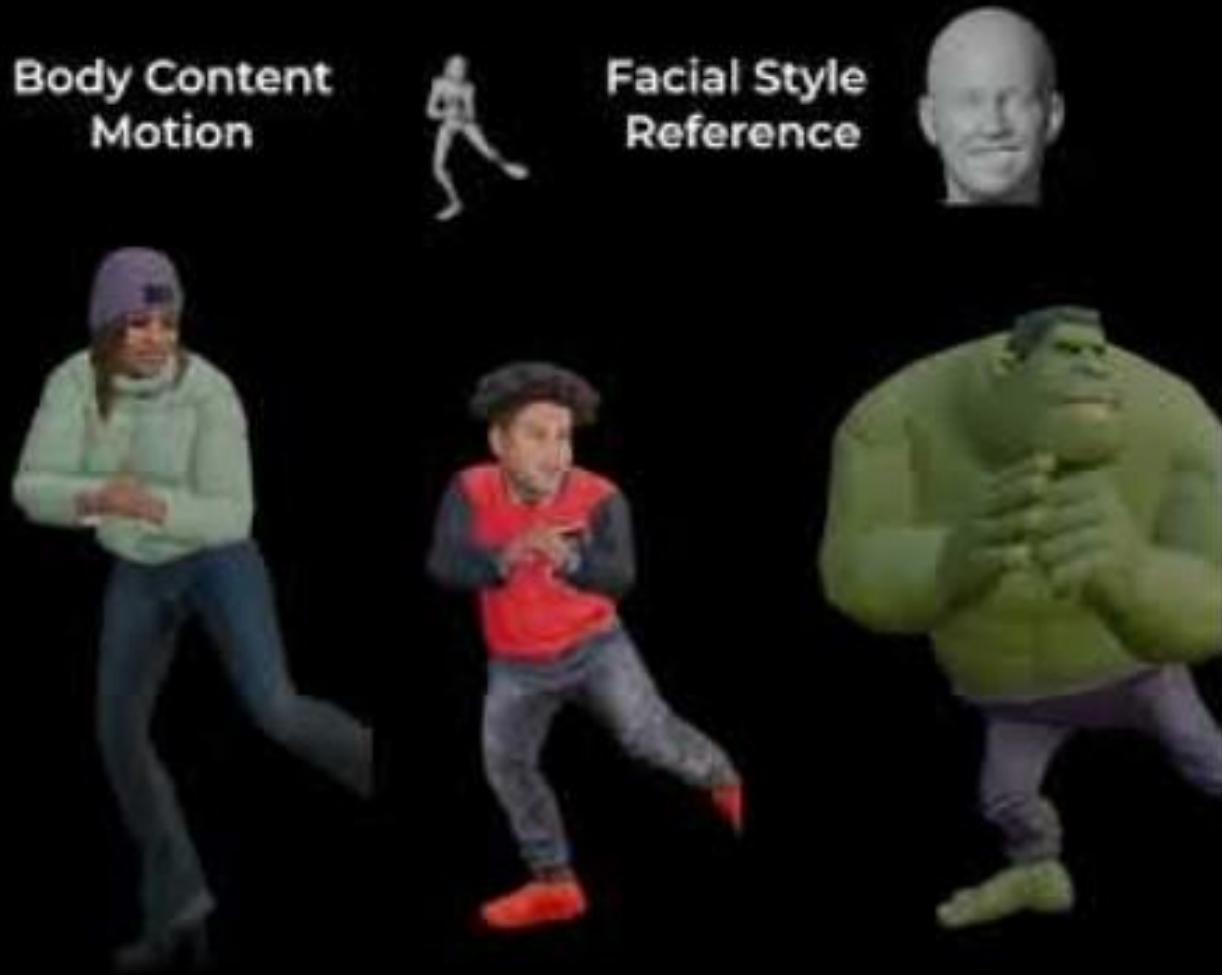
FreeMusco: Motion-Free Learning of Latent Control for Morphology-Adaptive Locomotion in Musculoskeletal Characters



Minkwan Kim, Yoonsang Lee. “FreeMusco: Motion-Free Learning of Latent Control for Morphology-Adaptive Locomotion in Musculoskeletal Characters.” **SIGGRAPH Asia 2025**

Hanyang University CSE4020, Yoonsang Lee, 2026

B2F: End-to-End Body-to-Face Motion Generation with Style Reference



Bokyung Jang, Eunho Jung, Yoonsang Lee. “B2F: End-to-End Body-to-Face Motion Generation with Style Reference.” **Pacific Graphics 2025**

CGR Lab.

- Computer Graphics & Robotics Lab.
 - (& - set intersection)
 - <https://cgrhyu.github.io/>
- Our research group aims to
 - understand **various aspects** (movements, underlying mechanisms, etc.) of
 - **natural** or **artificial creatures** (humans, animals, robots, etc.) and
 - realize them in **from virtual** environment **to real** world.

Lab Session

- Now, let's start the lab today.