
Computer Graphics

10 - Lab - Character Animation

Yoonsang Lee
Hanyang University

Spring 2026

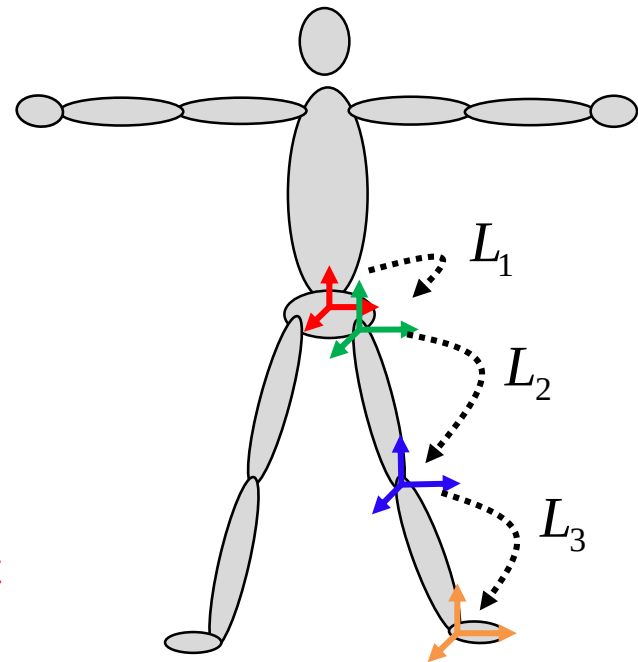
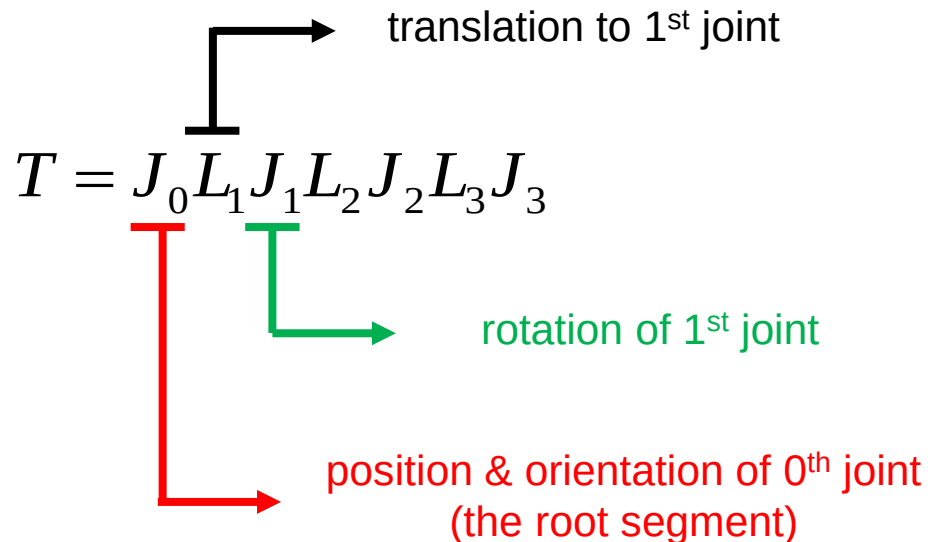
Outline

- Example: Joint & Link Transformations

Example: Joint & Link Transformations

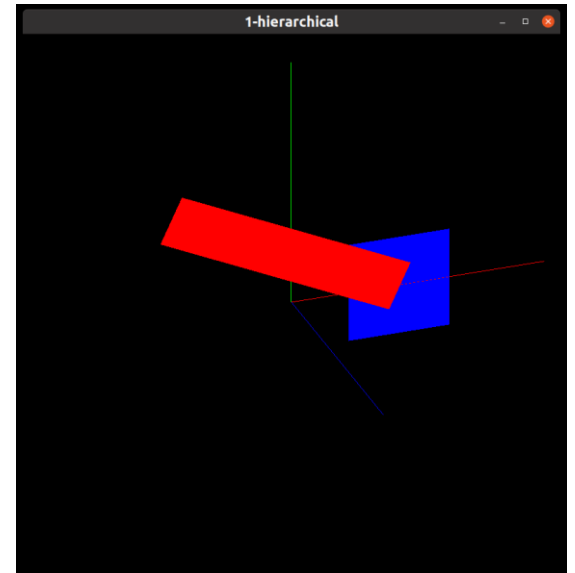
Recall: Forward Kinematics Map

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



[Code] 1-joint-link-transform

- Let's modify "7-Lab-Hierarchical-Mesh/1-hierarchical.py" to use **joint & link transformations** instead of a local transformation, while keeping the same functionality.



[Code] 1-joint-link-transform

1-hierarchical

```
class Node:
    def __init__(self, parent,
shape_transform, color):
        ...
        # transform
        self.transform = glm.mat4()
        self.global_transform =
glm.mat4()
        ...

    def set_transform(self, transform):
        self.transform = transform

    def
update_tree_global_transform(self):
        if self.parent is not None:
            self.global_transform =
self.parent.get_global_transform() *
self.transform
        else:
            self.global_transform =
self.transform
        ...
```

1-joint-link-transform

```
class Node:
    def __init__(self, parent,
link_transform_from_parent, shape_transform,
color):
        ...
        # transform
        self.link_transform_from_parent =
link_transform_from_parent
        self.joint_transform = glm.mat4()
        self.global_transform = glm.mat4()
        ...

    def set_joint_transform(self,
joint_transform):
        self.joint_transform = joint_transform

    def update_tree_global_transform(self):
        if self.parent is not None:
            self.global_transform =
self.parent.get_global_transform() *
self.link_transform_from_parent *
self.joint_transform
        else:
            self.global_transform =
self.link_transform_from_parent *
self.joint_transform
        ...
```

[Code] 1-joint-link-transform

1-hierarchical

```
# Node(parent, shape_transform, color)
base = Node(None, glm.scale((.2,.2,0.)), glm.vec3(0,0,1))
arm = Node(base, glm.translate((.5,0,.01)) * glm.scale((.5,.1,0.)),
glm.vec3(1,0,0))

while not glfwWindowShouldClose(window):
    ...
    # set local transformations of each node
    base.set_transform(glm.translate((glm.sin(t),0,0)))
    arm.set_transform(glm.translate((.2, 0, 0)) * glm.rotate(t, (0,0,1)))
```

1-joint-link-transform

```
# Node(parent, link_transform_from_parent, shape_transform, color)
base = Node(None, glm.mat4(), glm.scale((.2,.2,0.)), glm.vec3(0,0,1))
arm = Node(base, glm.translate(glm.vec3(.2,0,0)), glm.translate((.5,0,.01)) *
glm.scale((.5,.1,0.)), glm.vec3(1,0,0))

while not glfwWindowShouldClose(window):
    ...
    # set local transformations of each node
    base.set_joint_transform(glm.translate((glm.sin(t),0,0)))
    arm.set_joint_transform(glm.rotate(t, (0,0,1)))
```

Time for Lab Check

- Project 3
 - Due: 23:59, Jun 7, 2026 (NO SCORE for late submissions!)
 - Will be uploaded soon (today or tomorrow).
- Let's start today's lab check.
- TAs will guide you.