
Computer Graphics

10 - Lab - Character Animation

Yoonsang Lee
Hanyang University

Spring 2025

Outline

- Example: Joint & Link Transformations
- Brief Introduction to Recent Publications from CGR Lab.

Example: Joint & Link Transformations

Recall: Forward Kinematics Map

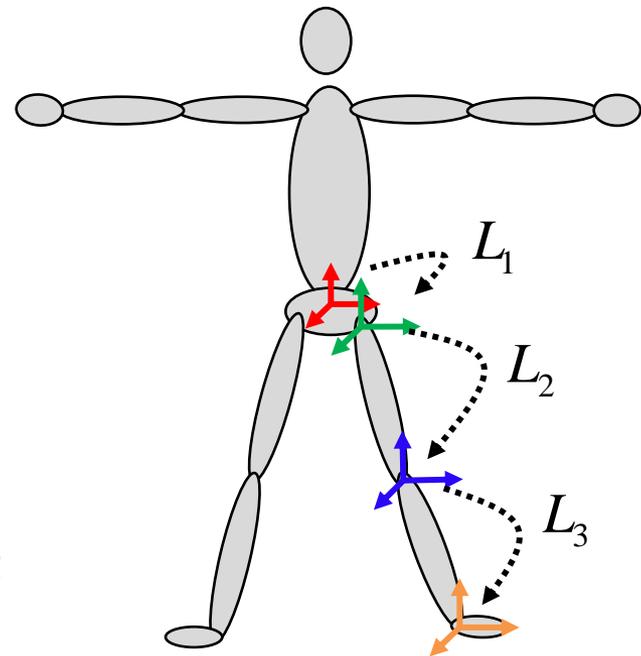
- A *forward kinematics map* T is an alternating multiple of ...
- **Joint transformations** (time-varying)
 - : Joint movement ("*motion*")
- **Link transformations** (static)
 - : Joint offset ("*skeleton*")

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

translation to 1st joint

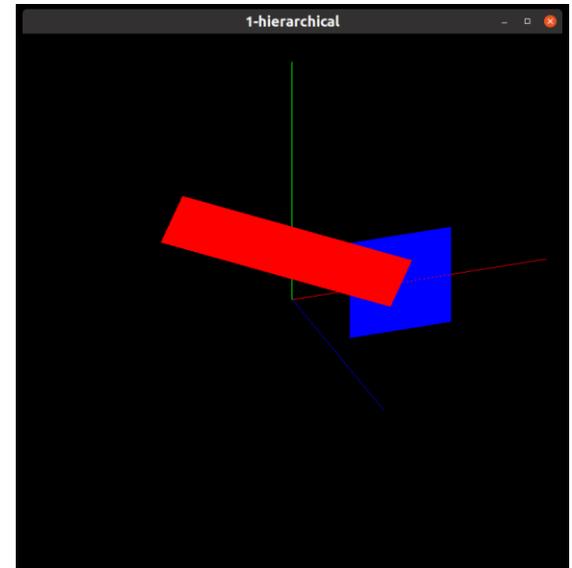
rotation of 1st joint

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



[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)))
```

Brief Introduction to Recent Publications from CGR Lab.

Introduction to CGR Lab.

- Computer Graphics & Robotics Lab.
 - (& - set intersection)

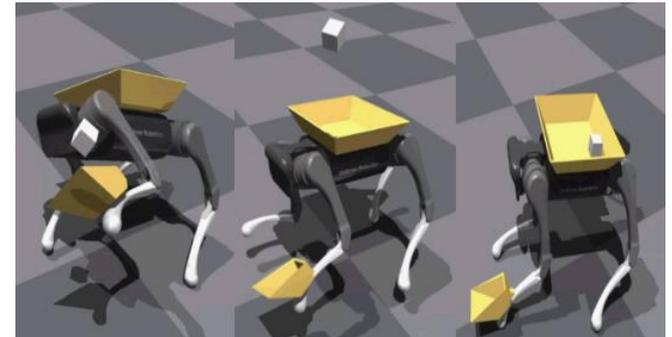
- Our research group aims to
 - understand **various aspects** (movements, underlying mechanisms, etc.) of
 - **natural** or **artificial creatures** (humans, animals, robots, etc.) and
 - realize them in **from virtual** environment **to real** world.

Main Research Direction

- "How to move?"
 - for articulated characters
 - robots



- “Learning How to Move”



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

SIGGRAPH 2025
Vancouver+ 10-14 August



HANYANG UNIVERSITY



Minsu Kim, Eunho Jung, Yoonsang Lee. “PhysicsFC: Learning User-Controlled Skills for a Physics-Based Football Player Controller.” **Accepted to ACM Transactions on Graphics (SIGGRAPH 2025)**, Yoonsang Lee, 2025

Utilizing Motion Matching with Deep Reinforcement Learning for Target Location Tasks

For target location tasks with moving obstacles, we propose two novel components: *hit reward* and *obstacle curriculum*.



(In this tasks, the policy receives supplementary sensor input.)

Jeongmin Lee, Taesoo Kwon, Hyunju Shin, Yoonsang Lee. "Utilizing Motion Matching with Deep Reinforcement Learning for Target Location Tasks." **Eurographics 2024** Short Papers, April 2024

Adaptive Tracking of a Single-Rigid-Body Character in Various Environments

Push boxes



10kg

(Edited) Walk



30kg

Taesoo Kwon, Taehong Gu, Jaewon Ahn, Yoonsang Lee. "Adaptive Tracking of a Single-Rigid-Body Character in Various Environments." SA '23: **SIGGRAPH Asia 2023** Conference Papers, Article No.: 118, December 2023

Motion Matching



Jeongmin Lee, Taesoo Kwon, Yoonsang Lee. "Interactive Character Path-Following Using Long-Horizon Motion Matching With Revised Future Queries." IEEE Access, January 2023

Hanyang University CSE4020, Yoonsang Lee, 2025

Time for Assignment

- Project 3
 - Due: 23:59, Jun 15, 2023 (NO SCORE for late submissions!)
 - Will be uploaded soon (Today).
- Let's start today's assignment.
- TA will guide you.