Computer Graphics

10 - Lab - Character Animation

Yoonsang Lee
Hanyang University

Spring 2023

Outline

- Example: Joint & Link Transformations
- Brief Intro to Motion Graph and Motion Matching

Example: Joint & Link Transformations

Recall: Forward Kinematics Map

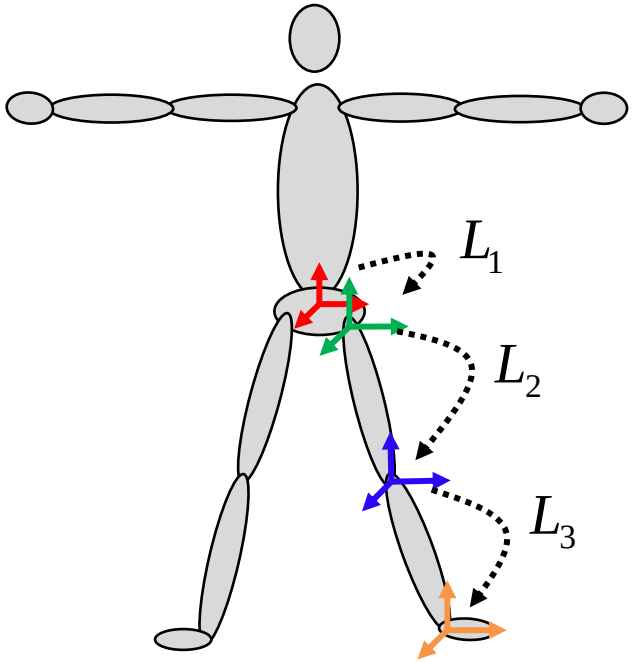
- A *forward kinematics map T* is an alternating multiple of ...
- **Joint transformations** (time-varying)
  - : Joint movement (*"motion"*)
- **Link transformations** (static)
  - : Joint offset (*"skeleton"*)

translation to 1st joint

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

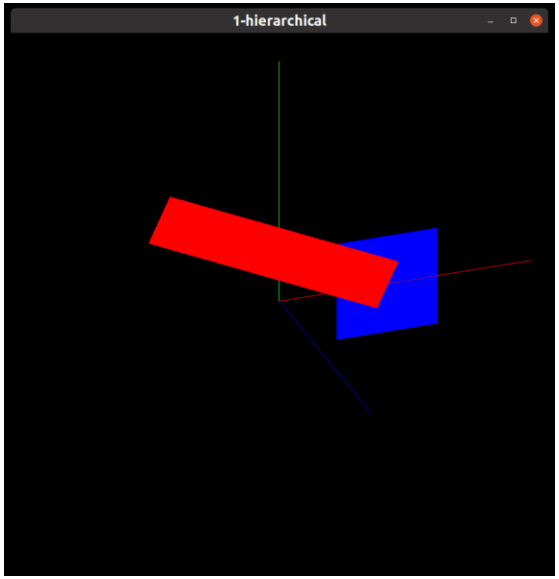
rotation of 1st joint

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)))
```

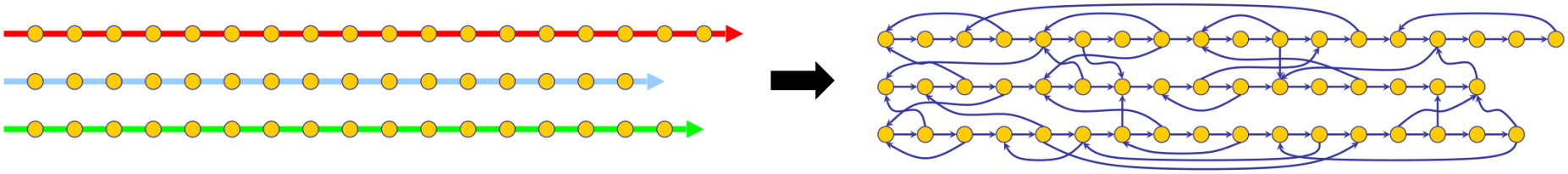
1-joint-link-transform

```
    # 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 Intro to Motion Graph and Motion Matching

Motion Graph [Lee et al. 2002] [Kovar et al. 2002] [Arikan&Forsyth 2002]



- Consideration for creating transitions:
  - Contact states, pose similarity, avoiding dead-ends

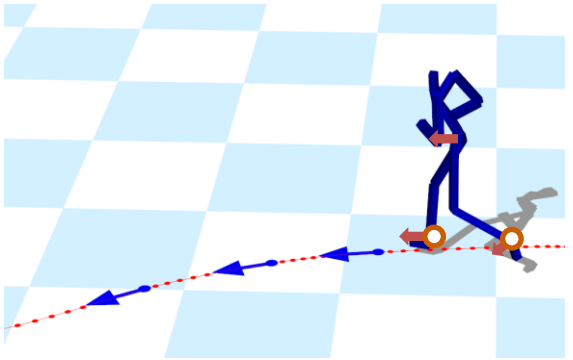
- Once a motion graph is constructed, you can find a series of transitions passing through...
  - Specified poses
  - Specified locations / continuous path
  - Specified poses and times
  - ...
- by using graph search algorithms (such as Dijkstra, A*, ...) or dynamic programming.

Motion Graph

[Lee et al. 2002] http://graphics.cs.cmu.edu/projects/Avatar/

Motion Matching [Büttner and Clavet 2015]

- *Motion DB* stores the pose for each frame of motion data.
- *Feature DB* stores extracted "features" for each frame of motion data.
  - Feature: (current state, future information)



- Matching (performed periodically):
  - Query q: (current character state, future information created by user input)
  - Search for the frame j^* that corresponds to the feature closest to the query q.
- , then motions are played sequentially from the j^* frame in the *motion DB*.

- Widely used in AAA games.

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*

Time for Assignment

- Project 3
  - Due: 23:59, Jun 4, 2023 (NO SCORE for late submissions!)

- Let's start today's assignment.
- TA will guide you.