Data Driven Computer Animation

HKU COMP 3360

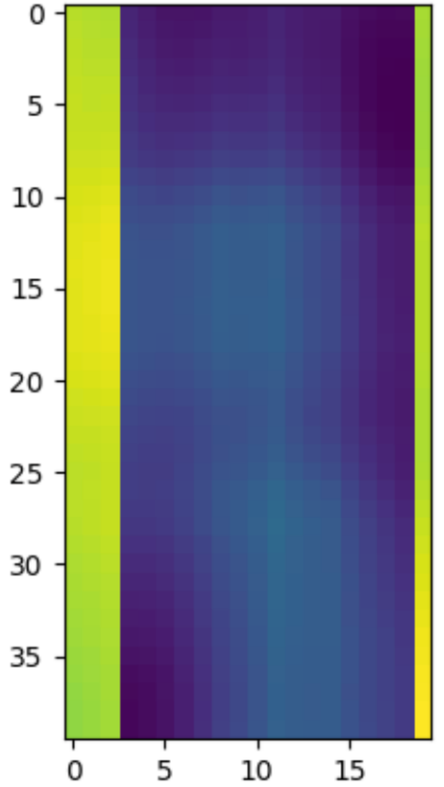
Tutorial 4 - Interactive Character Animation

Prof. Taku Komura
TA: Shi Mingyi (myshi@cs.hku.hk), Huancheng Lin (lamws@connect.hku.hk)

SECTION 2A, 2022

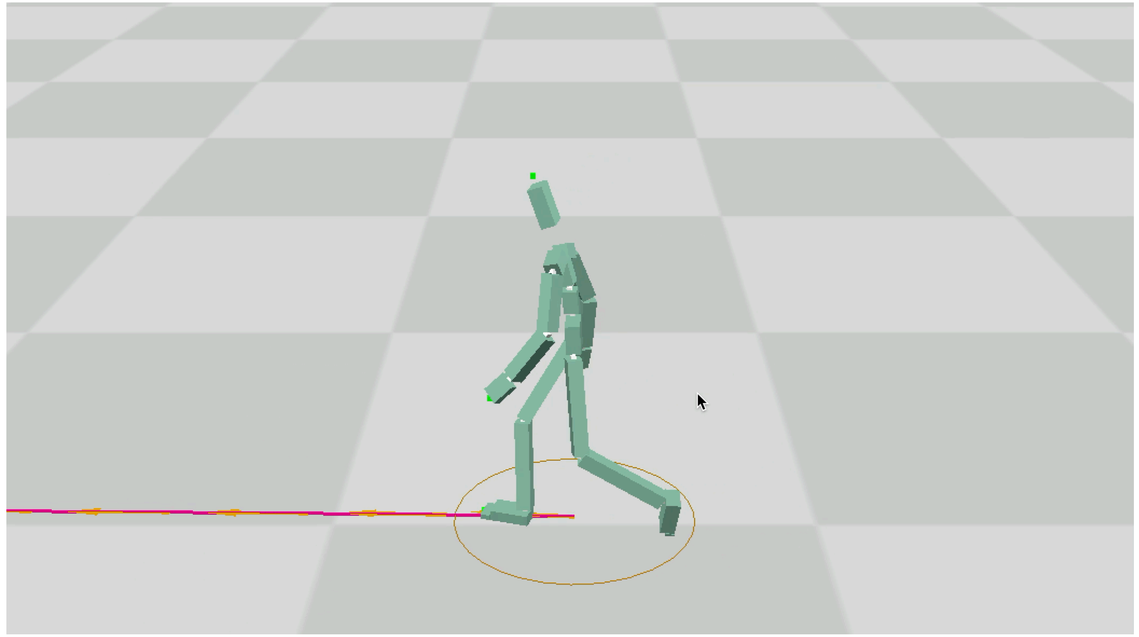
Some question on Part 1&2

1. Can I prepare more interpolation method?
   - Sure, but no bonus for this task (it's not hard)
2. The similarity matrix is different with the one in tutorial slides
   - -> ok, because the one in tutorial slide is a screenshot from course slides
   - -> I also got the same one



Agenda

1. The overview of character animation system
2. Why motion matching is better than status machine
3. The details of motion matching



Some slides content is from GDC talk by Daniel Holden
https://www.youtube.com/watch?v=o-QLSjSSyVk

The code environment is based on GAMES105

GAME OF THE YEAR

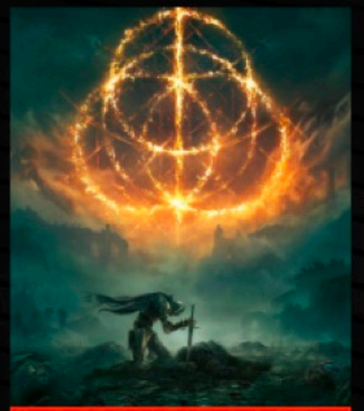
RECOGNIZING A GAME THAT DELIVERS THE ABSOLUTE BEST EXPERIENCE ACROSS ALL CREATIVE AND TECHNICAL FIELDS.

SHARE

SIGN IN TO VOTE

A PLAGUE TALE: REQUIEM

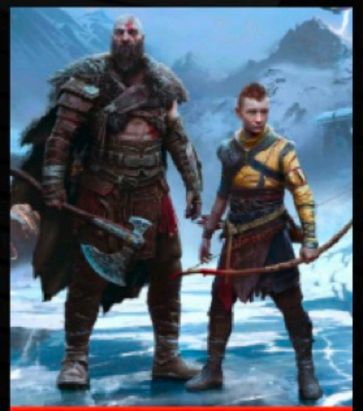
ASOBO STUDIO / FOCUS ENTERTAINMENT



SIGN IN TO VOTE

ELDEN RING

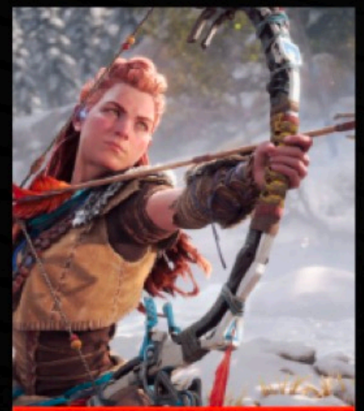
FROMSOFTWARE / BANDAI NAMCO



SIGN IN TO VOTE

GOD OF WAR RAGNARÖK

SONY SANTA MONICA / SIE



SIGN IN TO VOTE

HORIZON FORBIDDEN WEST

GUERRILLA GAMES / SIE

SIGN IN TO VOTE

STRAY

BLUETWELVE STUDIO / ANNAPURNA

SIGN IN TO VOTE

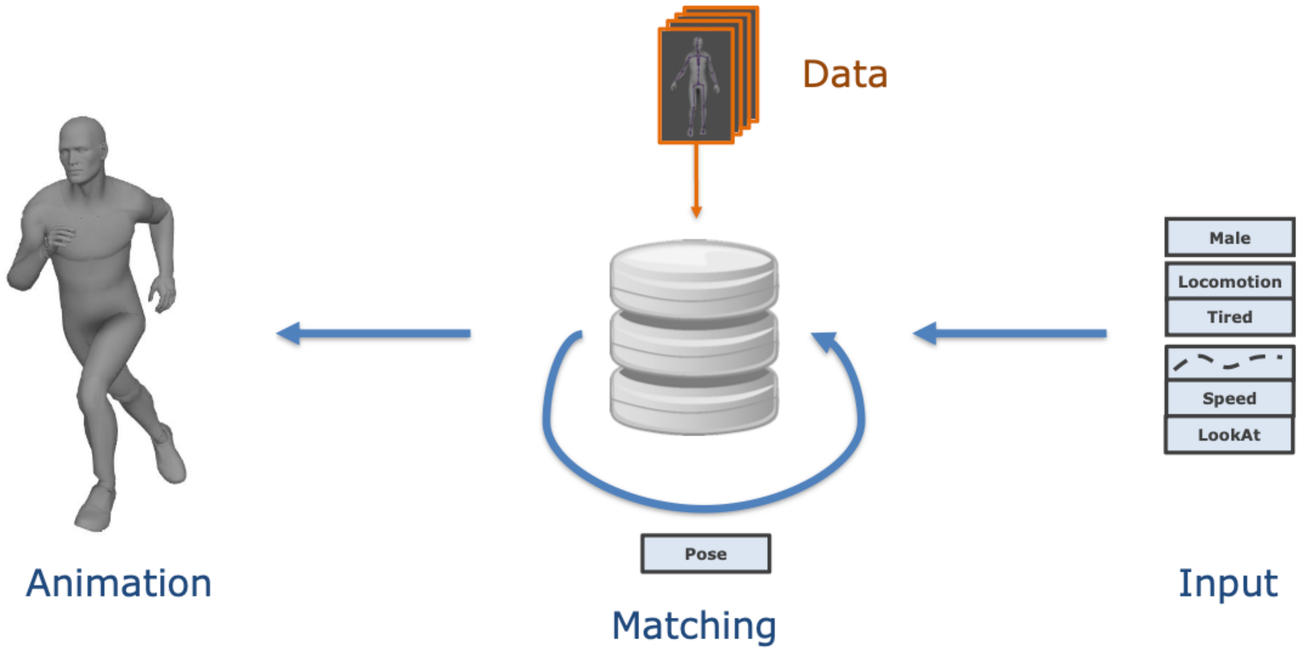
XENOBLADE CHRONICLES 3

MONOLITH SOFT / NINTENDO

Overall and Background of Interactive Character Animation

GDC talk by Daniel Holden
https://www.youtube.com/watch?v=o-QLSjSSyVk

Programming Details



Animation = Query(Dataset, variables)

Select animation where variables=xxxxx from dataset

Programming Details

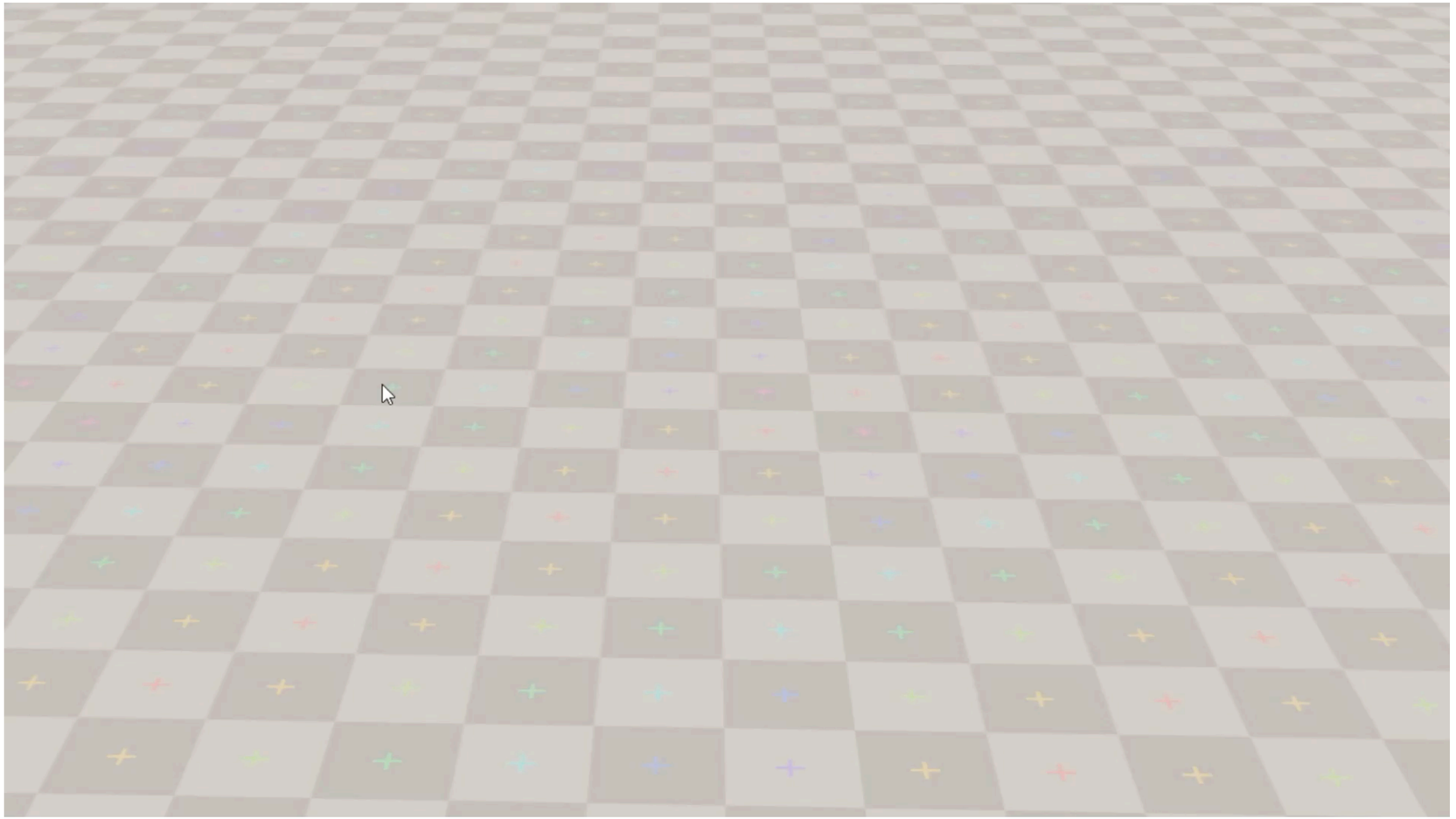
Prepare:

- A motion dataset
  - Oc-tree to accelerate the searching
- A set of feature variables
- A animation system to play the animation
  - Interactive, responsible, and smooth

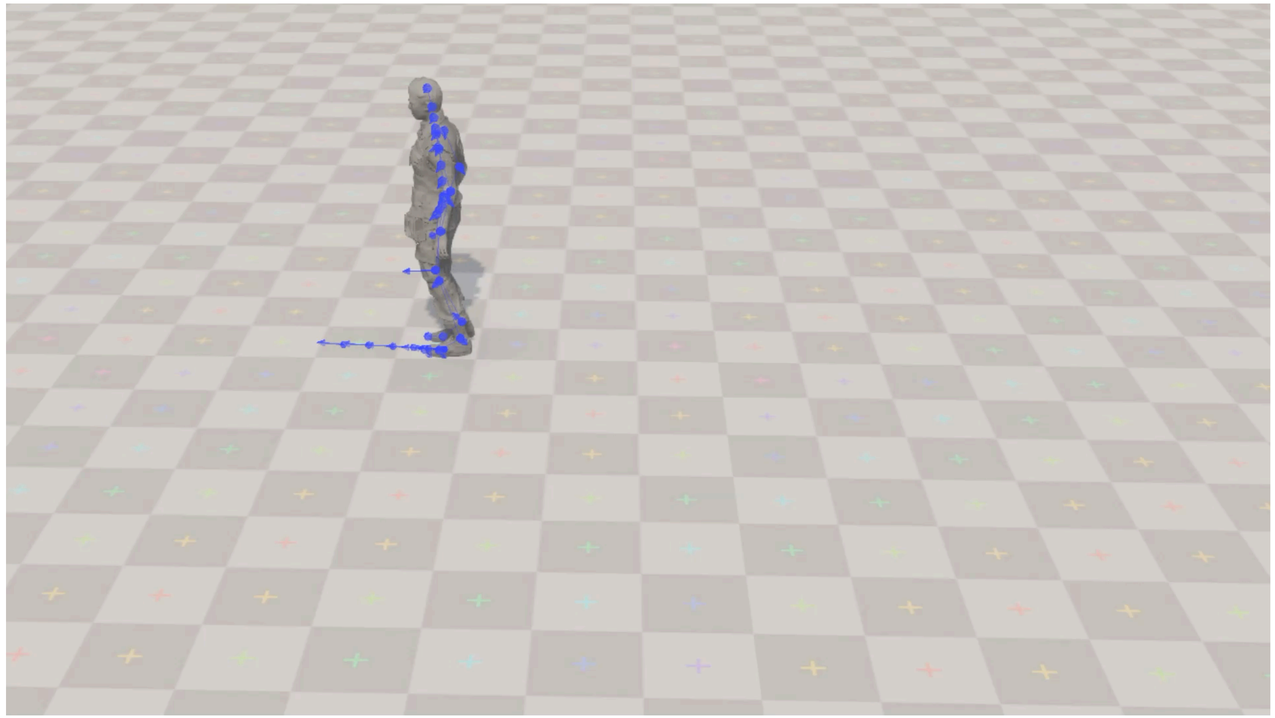
The dataset and animation are prepared in this assignments

you only need to care the variables design

Step1: Give control signals



Step2: Analyze the motion data



```
94          # Extract the data terms
95          pos, rot = forward_kinematics_with_channel(self.joint_parent, self.joint_channel, sel
96          rot = align_quat(rot, False)
97          vel = np.zeros_like(pos)
98          vel[1:] = (pos[1:] - pos[:-1])/self.dt
99          vel[0] = vel[-1]
100         avel = np.zeros_like(vel)
101         avel[1:] = quat_to_avel(rot, self.dt)
102         avel[0] = avel[-1]
```

We know:

1.Joint position

2.Joint velocity

3.Joint rotations

4.Joint angular velocity

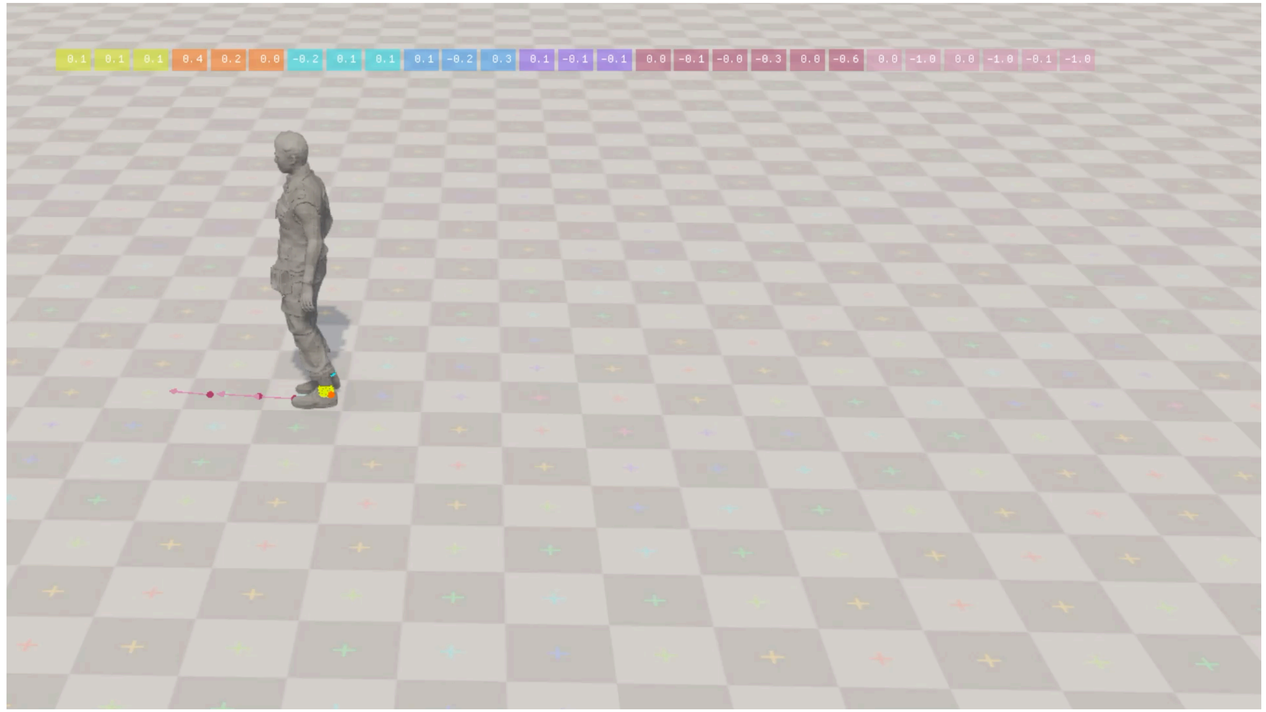
…

relative position

waving timing

…

Step3: Design the matching variables




```
How can we use these data to match the motion?
'''
feature_mapping = {
    'lFootPos': 3,
    'rFootPos': 3,
    'lFootVel': 3,
    'rFootVel': 3,
    'lFootRot': 3,
    'rFootRot': 3,
    'rKneeAVel': 3,
    'lKneeAVel': 3,
    'lHandPos': 3,
    'lHandVel': 3,
    'rHandPos': 3,
    'rHandVel': 3,
    'rKneeAVel': 3,
    'lKneeAVel': 3,
    'rHipPos': 3,
    'lHipPos': 3,
    'rHipVel': 3,
    'lHipVel': 3,
    'hipVel': 3,
```

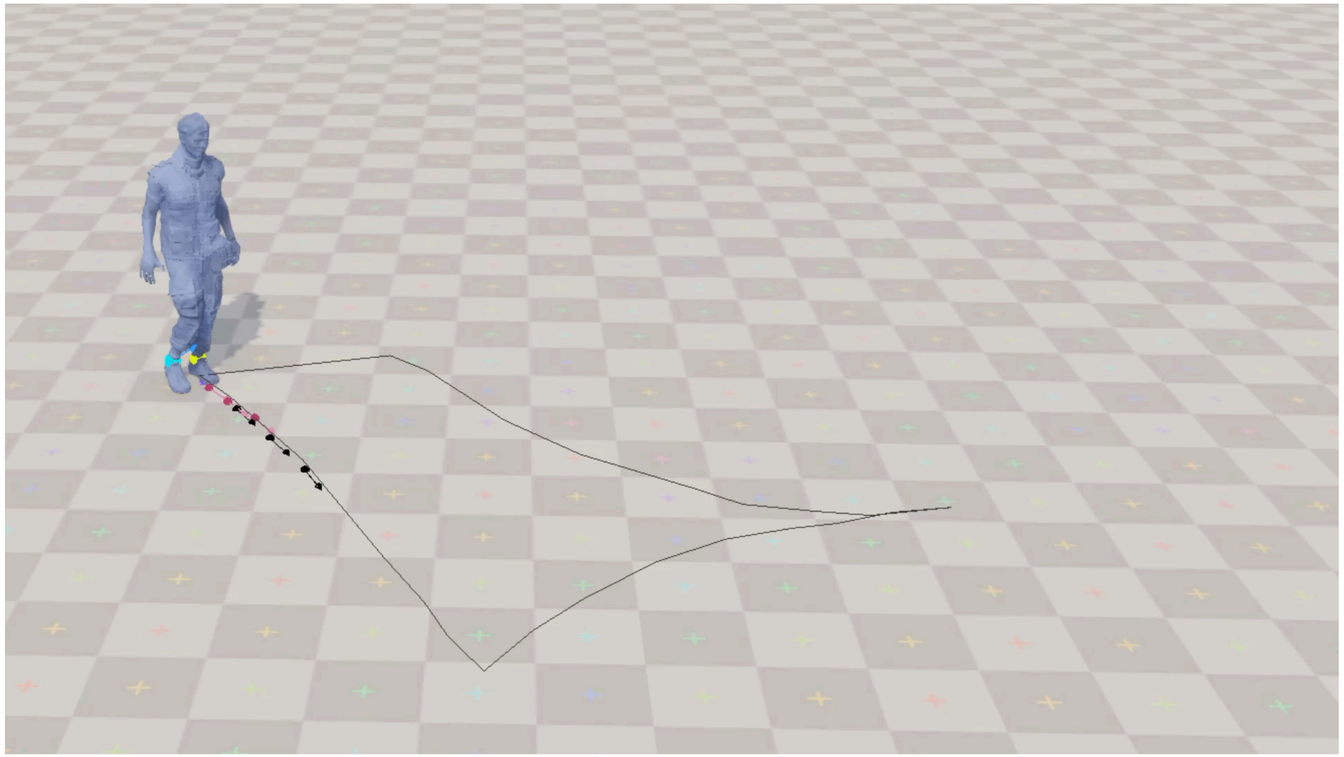
```
    for feature_name in self.feature_names:
        '''
        Extract the position:
            features.append(self.extract_offset(roo
        Extract the rotation:
            features.append(self.extract_rotation(r
        Extract the velocity:
            features.append(self.extract_vel(root_r
        Extract the future position:
        '''
        ##### Code Start #####

        if feature_name == 'lFootPos':
            features.append()
        elif feature_name == 'lFootVel':
            features.append()
        elif feature_name == 'trajectoryPos2D':
            features.append()

        ##### Code End #####
```

Not all the variables is needed
Find a balance between speed and accuracy

Step4: Matching

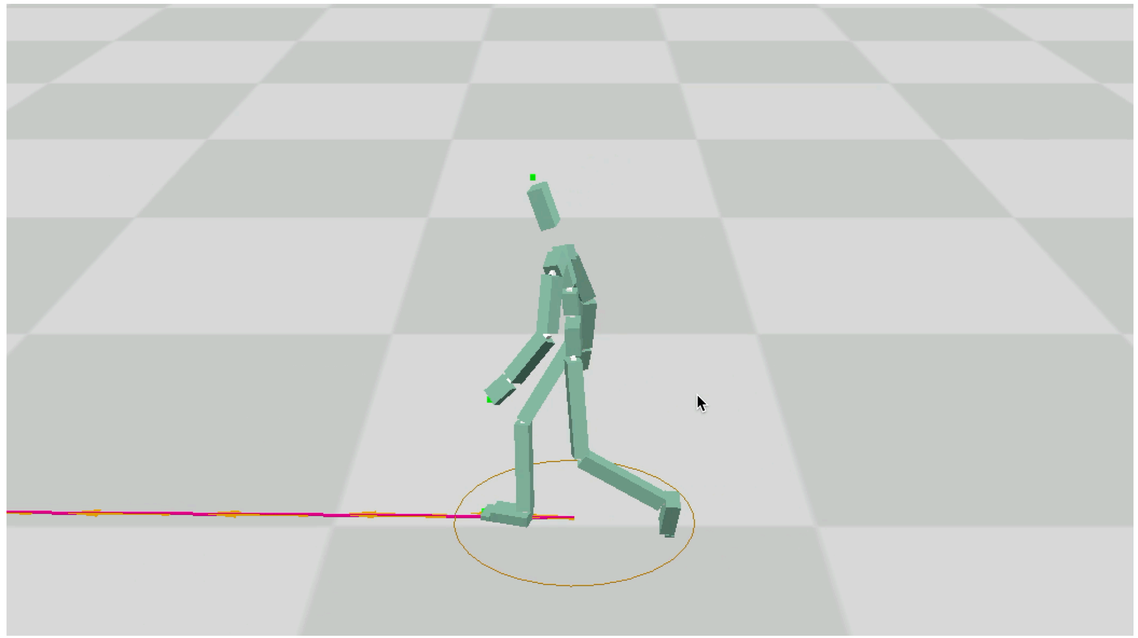


```
259             normalized_query = self.db.normalize_features(query_feature)
260
261             # Do the query
262             idx = self.db.query_tree.query(normalized_query.reshape(1,-1), k=1)[1][0]
```

Because we only use small dataset
We also store it with OC-tree
So the searching is fast

Desired Results

1. The character should follow the controller (10%)
2. Less variables, and better performance (total 15%, 22% - your_variable_num)



Overview

- part1_key_framing (30%)
    - - Linear interpolation (10%); Slerp Interpolation (15%)
    - - Report the different performance by giving different numbers (5%)
- part2_concatenate (35%)
    - - Define the search window (10%); Calculate the sim_matrix (10%); Find the real_i and real_j (10%); The shifting on the root joint position (5)
- part3_motion_matching (25%)
    - System works (10%) + variable terms (15% <- 22% - your_variable_num)
- Report (8%) + 2 videos (2%)
    - Including necessary experiment results by different parameters (4%) and your thinking(4%) for how to produce high quality motions