DS5220 Project Proposal

Jaee Oh

July 3, 2025

1 Participants

Jaee Oh

2 Description

I want to create a real-time motion capture system that can run on a low budget system, which in this case will be using a single webcam and a pc. There are many useful tools that can read out skeletal structures from a webcam view, such as MediaPipe, but this is only a skeletal structure data in 2D. I want to train a model that can accurately predict 3D positional data from 2D skeletal data using publicly available motion capture data. This was performed in many previous literature [1][2][3], but they were all done in recorded video. I want to extend these results in real-time images while keeping it computationally as cheap as possible. The motivation of this project is from various VR games and live streaming of V-tubers. If this simple setup can accurately capture 3D pose of a user without requiring often expensive and complicated optical and/or inertial motion capture systems with markers, it may lower the bar for accessing VR related contents and contribute to development of metaverse.

3 Literature Review

- ‘A Simple Yet Effective Baseline for 3D Human Pose Estimation’ by Martinez et al.[2] (2017): Neural network with linear and RELU layers with residual connections.
- ‘3D Human Pose Estimation in Video with Temporal Convolutions and Semi-Supervised Training’ by Pavllo et al. (2019)[3]: Convolution model using unlabeled videos. Performance is tested by projecting output 3D data into 2D data and compare it with the initial 2D data.
- MPL: Lifting 3D Human Pose from Multi view 2D Poses by Seyed Abolfazl Ghasemzadeh, Alexandre Alahi, and Christophe De Vleeschouwer(2024)[1]: Transformer network taking 2D poses from all vies as inputs.

All three recent publication use a 2D pose estimation to find 2D join positions, and then use the 2D joint positions to estimate 3D join positions. This is computationally cheaper than directly training from original 2D images like pictures to 3D images. Also, it is possible due to development of highly efficient and accurate 2D joint position.

4 Algorithms

- Linear, nonlinear and logistic regression models. This is a regression problem, not a classification problem. We're not classifying motion's label, but predicting closest possible 3D pose from 2D pose.
- PCA, tSNE and Autoencoder for dimensionality reduction. This may or maynot be used on the final product depending on its advantage in efficiency vs disadvantages in accuracy. Number of joint positions data is most likely going to be two digits number, so it can be done without it.
- Neural network and transformer models. Possibly extend the problem into some subjects that may or may not be covered in the class.

5 Data sets

- CMU MoCap: This dataset is provided in ASF format which provides skeletal structure. I will generate 2D images of corresponding 3D data using the ASF file format, which then will be used as training dataset that converts 2D to 3D. Although, skeletal data is formatted slightly different from that of MediaPipe.
- Human3.6M: More popular dataset which is a standard for many motion capture applications. However, it requires registration.

May use either one of them or both for the project.

6 Libraries and tools

- MediaPipe: 2D pose estimation library. Will be used to convert real time 2D image from a webcam to skeletal structure data.
- OpenCV: library for real-time image processing.
- Scikit-Learn: machine learning library.
- PyTorch: In-depth machine learning library.
- Unity: 3D modeling tool.
- Claude AI: Used as a tool for searching resources and clarification of tools.

7 Results

- Use a webcam to capture a user's motion and make a 3D avatar to mimic the motion real-time.

- I will be using various different supervised machine learning techniques to figure out which one is the most accurate and efficient in motion capture. I will be splitting training set and test set to compare error rate. In addition I want to compare the actual outcome, which will be the video showing both a webcam view and a 3D avatar view.

- If I run out of time, I might not be able to implement non-machine learning related parts, such as creation of a 3D avatar and a code to connect the avatar to trained model and real-time webcam view. In this case, I will simply compare different evaluation metrics with different models.

References

[1] Seyed Abolfazl Ghasemzadeh, Alexandre Alahi, and Christophe De Vleeschouwer. "MPL: Lifting 3D Human Pose from Multi-view 2D Poses". In: *arXiv e-prints* (2024), arXiv–2408.

[2] Julieta Martinez et al. "A simple yet effective baseline for 3d human pose estimation". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2640–2649.

[3] Dario Pavllo et al. "3d human pose estimation in video with temporal convolutions and semi-supervised training". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 7753–7762.