Daily Log

Monday October 28

I was attempting to install tf-pose on my account on the server. I finished all the installation commands because I was missing certain packages. Began to test some videos at the end of class.

Tuesday October 29

Continued to test on different videos of some forms and dances. Then looked into 3D modeling using tf-pose.

Saturday November 2

Following Dr. Zacharias' direction, my partner and I decided to look into OpenPose. We found an example notebook on colab that runs OpenPose on youtube videos. We tested on wushu and dance YouTube videos and it works well. Although with flow-y clothing, it has a hard time detecting the skeleton.

Wednesday November 6

I began working on using our own videos for testing. Colab's API for using your Drive's files for testing is not the most intuitive. I also looked into some further capabilities of OpenPose, such as 3D modeling.

Thursday November 7

I began working on the 3D modeling aspect of OpenPose, which has its own camera calibration technique.

Timeline

Date	Goal	Met
10/28	Place skeletal model	No, tf-pose-estimator isn't the most compatible with colab
11/4	Place skeletal model using python environment on computer	Yes, using OpenPose
11/11	Work with partner to put data into neural network, tweak position data if necessary	No, but we will be able to get position data from the 3D modeling
11/18	Calibrated cameras and have calibration model ready for testing	
11/25	Tested calibration model and extracted position data	

Reflection

As previously mentioned, Dr. Zacharias' suggestion lead us to use OpenPose. It allows us to do what we need to for this project. OpenPose allows for multiple people without a significant slowdown, which would be useful for dances looking for synchronicity.

I was especially pleased to find the 3D modeling function. As I predicted, we will need at least three cameras on the subject. The calibration method seems to use a large chess or checkered board with known side lengths. I will probably use a 8x6 chessboard with box sizes at least 100mm, as the instructions recommend. It suggests 400 image photos for high quality calibration, although the run time grows exponentially. The github states that 450 photos takes about 9.5 hours to process.

I don't know if I will be exclusively using phone cameras. Over the summer when I was working with Photomodeler, it was not compatible with my phone camera due to the rotation of the pictures. I'm not sure if this will have the same issue, but I may use an actual camera for one of the cameras just to be safe.