Using RGB-based Human Skeleton Recognition for CwC Project

Gururaj Mulay and CwC Team, Department of Computer Science, Colorado State University

Objectives

To replace the Kinect-based data collection module at the front-end of the Communicating with Computers project with an RGB camera-based module in a seamless manner.

- Retrieve the pose skeleton keypoints of a human subject from RGB images.
- Transmit keypoints & RGB images of hands, head to clients for gesture recognition.
- Write a wrapper for OpenPose library that does pose skeleton detection.

CwC Project Task

Get a structure of blocks build from a virtual AI agent (Diana) via multi-modal communication.

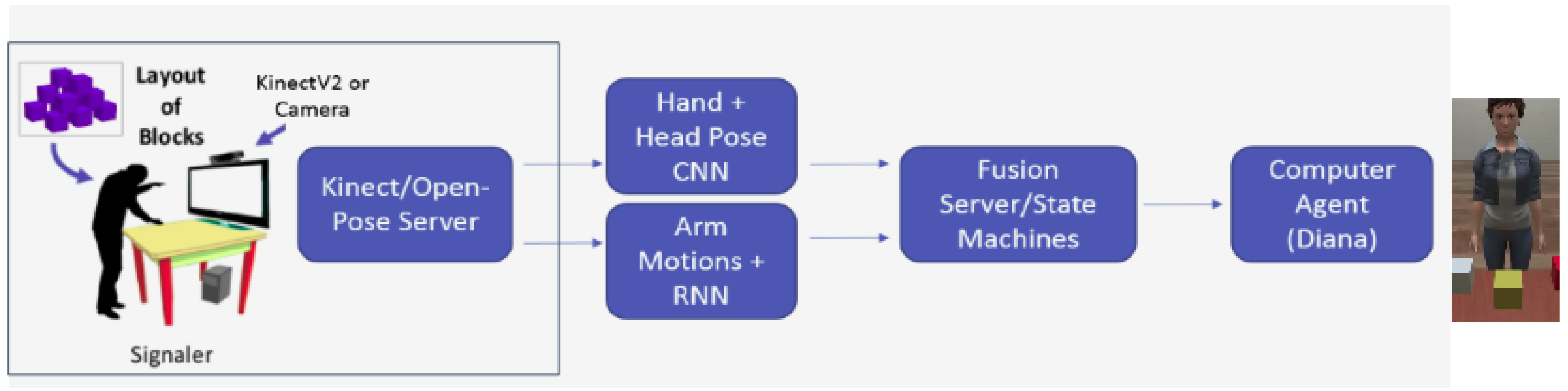


Figure 1: CwC project block diagram

Old: Kinect front-end

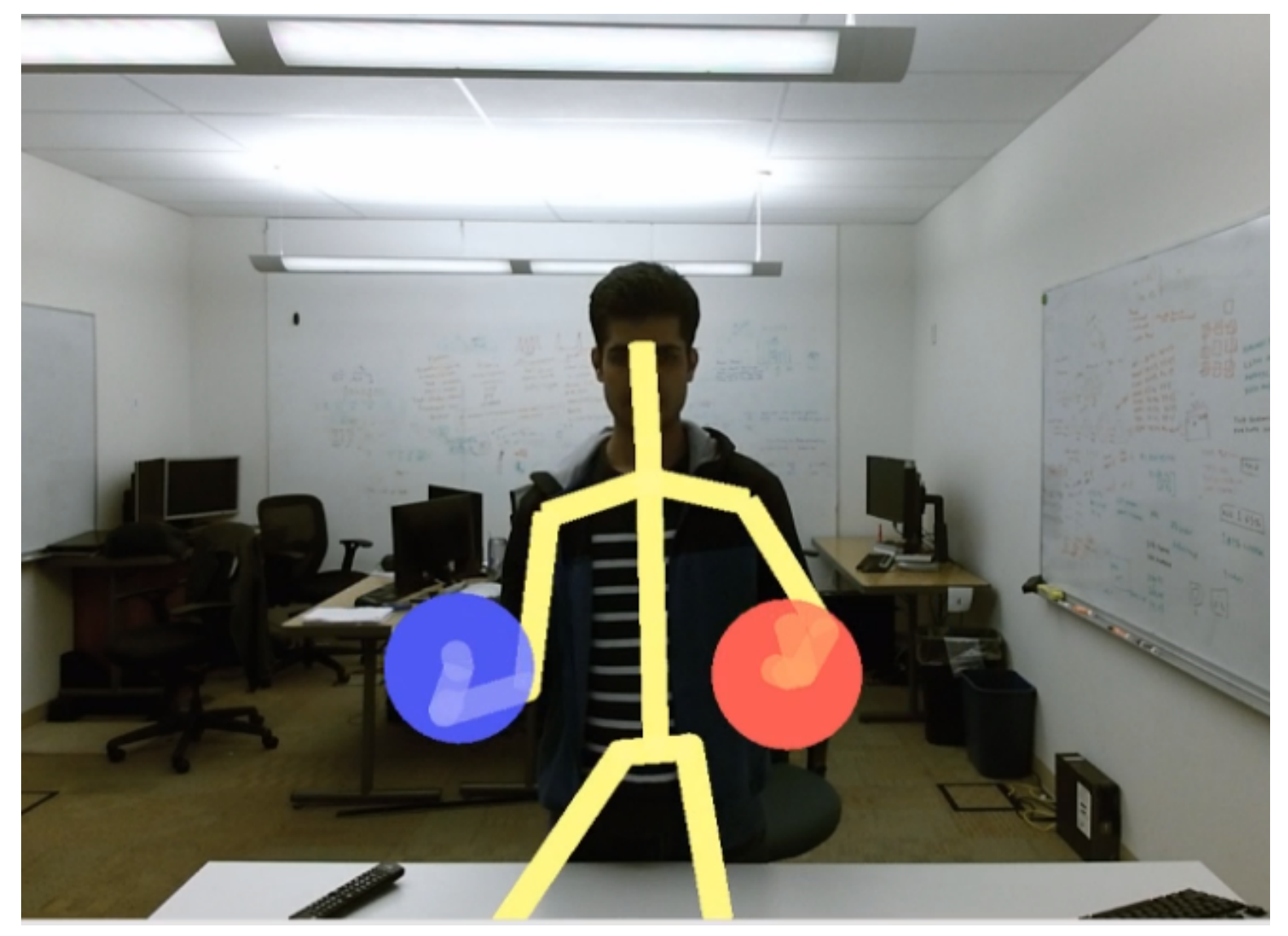


Figure 2: KinectV2

- Production of KinectV2 is stopping
- Kinect is relatively expensive
- It's observed to be noisy, fails in occlusions
- Proprietary, not open source
- Gives 3D skeletons with *Depth* information

New: OpenPose front-end

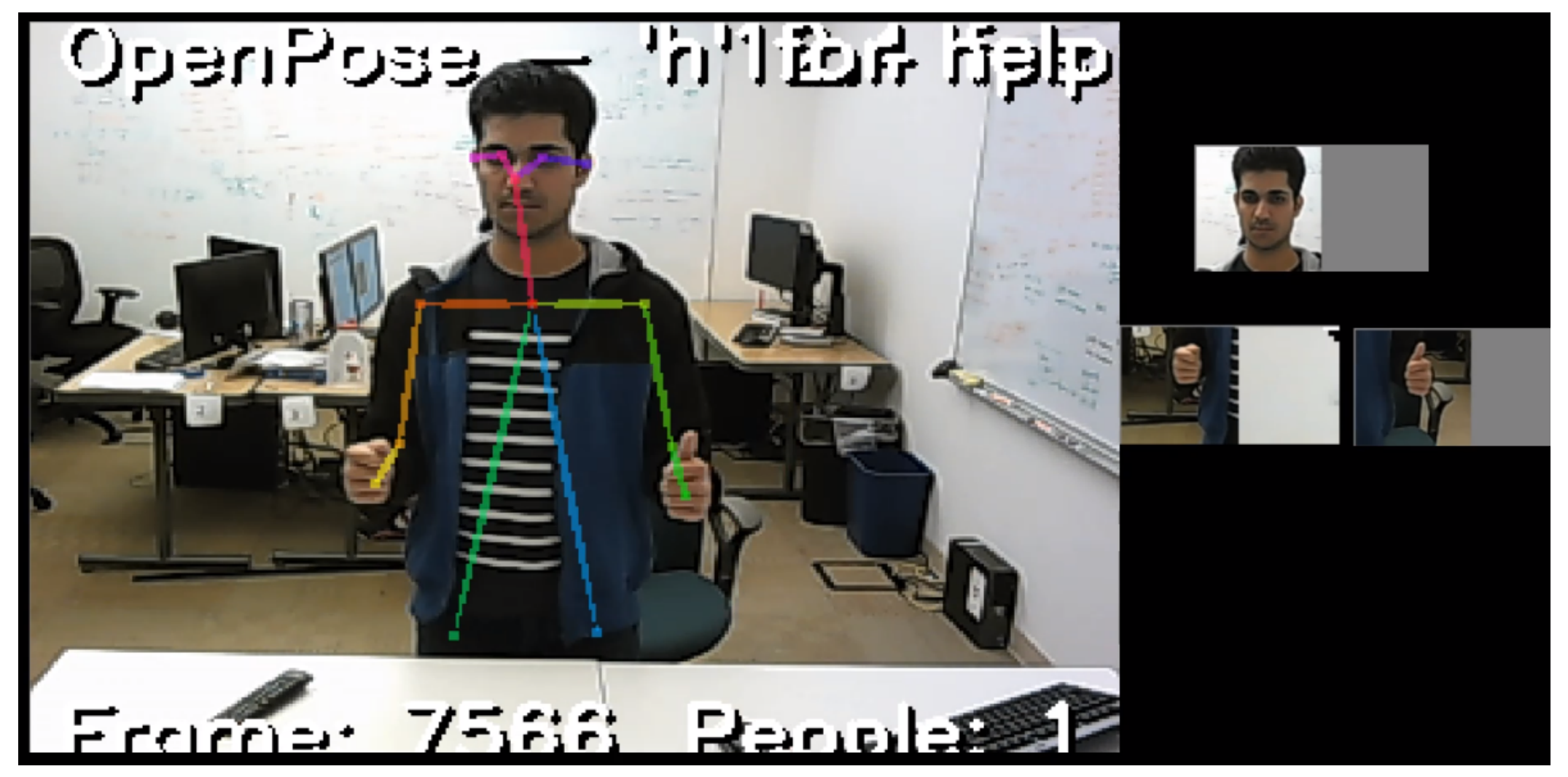


Figure 3: OpenPose Server

- Open source, cheaper, but gives 2D skeletons
- Stable outputs even with occlusion
- OpenPose, apart from 18 skeleton points, also gives detailed keypoints for hands and face. However, this add-on functionality runs at <7 fps which will potentially go up in future

Introduction

Recent work by CMU [1], [2] in human pose recognition from RGB images, produces state-of-the-art results on multiple datasets. Success of these methods can be primarily attributed to convolutional neural networks for joint detection and special techniques to associate detected joints with people in a frame. We incorporated this work to replace previous depth-based pose recognition. Task involved writing a C++ wrapper around the OpenPose library [1] and a Python server to interface this wrapper with the rest of the client's system, along with some modifications to previous clients.

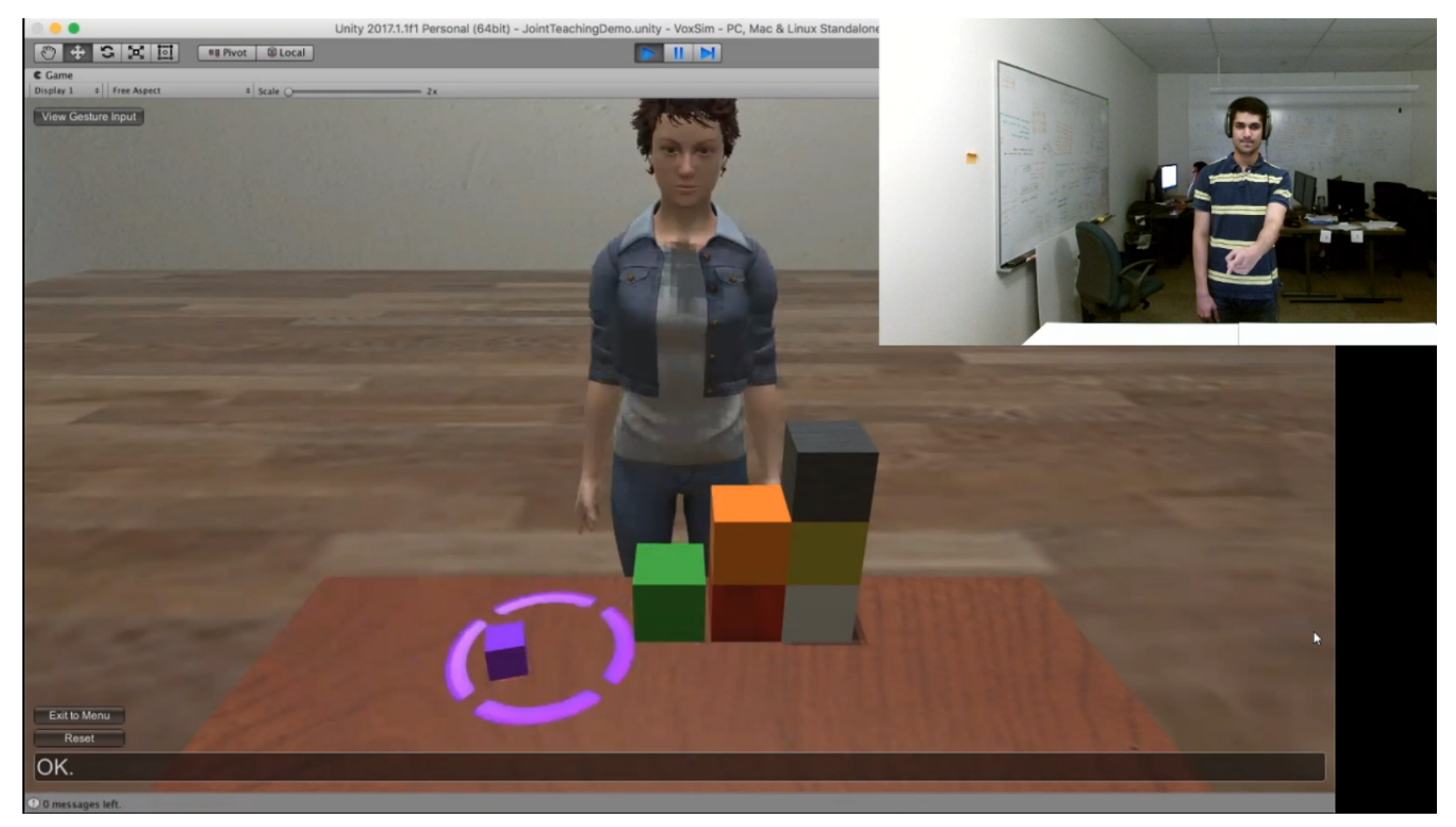
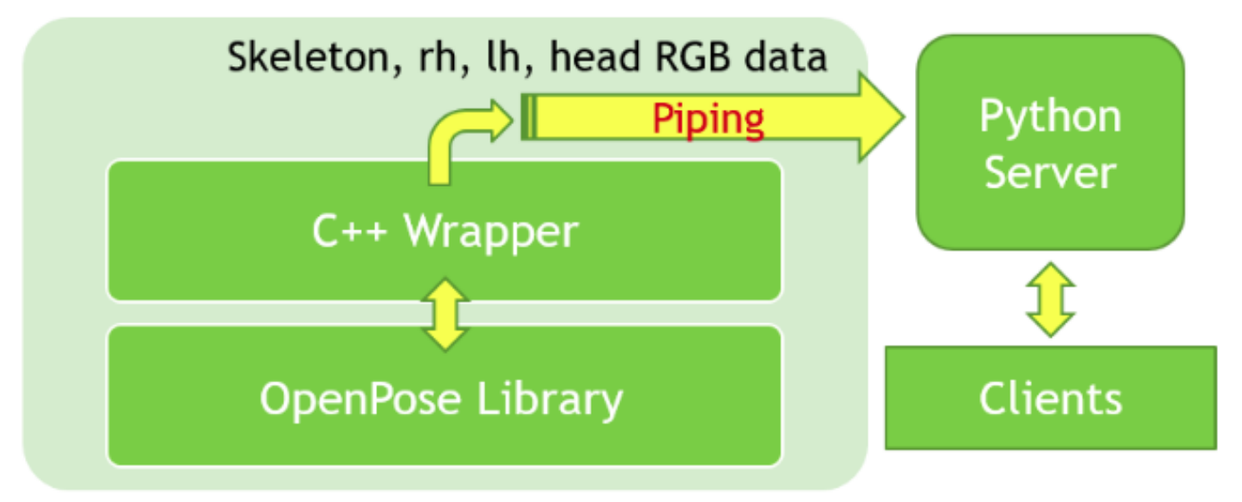


Figure 4: CwC in actions

System Design: C++ wrapper and Python server

- The C++ wrapper extracts pose skeletons from frames captured by a web-cam and uses these keypoints to crop the RGB images of hands and the head of the primary person in the video.
- The Python server is responsible for communicating with the wrapper process to collect the skeleton keypoints, hands and head images for primary (engaged) person in the frame.
- Server accepts four clients: skeleton, right hand, left hand, and head client.



- For every frame processed by OpenPose, the server sends corresponding data to four clients.

System Details

- Primary person is the closest person in the center of frame. With only 2D skeletons, it's challenging.
- We calculate the average expanse (length) of the limbs of candidate persons present in a central window and choose the closest person based on a fixed threshold of limb length (51 in our case)
- Outputs 18 keypoints (x, y pixel locations of body parts + confidence) for people in the image frame
- ~15 FPS on 320x240x3 input on NVidia GTX980
- Server sends keypoints to arm motion recognition
- Wrapper crops a 64x64x3 image for left hand, right hand, and head; Server sends them to hand and head clients for pose recognition

Conclusion

- We successfully implemented & tested the system by seamlessly switching from Kinect-based front end. Hands-on testing with the AI agent proved that it's possible to get complex block structures build from the agent even with a RGB sensor.
- The system ran with no significant lag, but it needed specific conditions (e.g., bright light)
- We demoed the system at DARPA in Dec 2017
- We hope that it can possibly replace the Kinect sensor, making our project demo more portable.
- In future, we aim to improve the performance in terms of fps and input image resolution.

References

[1] Openpose Library: `github.com/CMU-Perceptual-Computing-Lab/openpose`

[2] Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, Z Cao, T Simon, S Wei, Y Sheikh, arXiv:1611.08050, 2016

Acknowledgements

I would like to acknowledge my colleagues and advisers from Computer Vision Lab for providing guidance during this work.