CSCI 5551- Intro to IRS

"CopyCat" by Group 6

Samaddar,Anwesha samad037@umn.edu
Tsang, Vivian tsang065@umn.edu
Yadav, Mohit yadav171@umn.edu

Introduction:

Our objective was to make Baxter imitate arm and head movements performed by a person. Real-time transfer of pose information to Kinematic Evaluator (KinEval) was done utilizing BlazePose[1], Flask and Socket.IO. Servo Controls in KinEval were used for smooth transitions between poses.

Input and Output:

Live video input will be taken through the webcam to capture pose information by Blazepose. Blazepose is a deep CNN network consisting of a Human Detector and Estimator. Angles between limbs will be extracted using 3d keypoints estimated by Blazepose and these will be passed as configuration for Baxter's joints. Baxter is given controls to move to the specific configuration in Simulation.

Methodology:

We use Flask as our web framework to serve our KinEval web application. Alongside Flask, we use the Socket.IO library to send angle values to the KinEval environment with Baxter. To run the Socket.IO server concurrently with the pose detection code, we used threading. We compute the angles in our Python code using the data extracted from the webcam feed. After sending the angles through Socket.IO, we use them to update Baxter's joint angles.

Environment:

We used the Kinematic Evaluator(KinEval) developed by Prof. Chad Jenkins as our simulation environment. Baxter robot was imported to the simulation for imitating the poses.

Experiments:

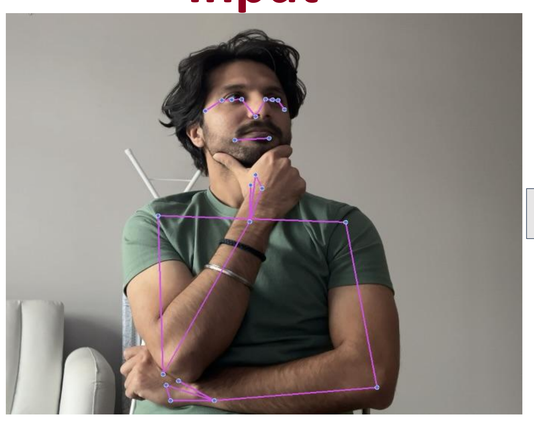
- Explored methods for finding joint angles of Baxter robot based on position of human joints.
- Tested forward kinematics and PID control to apply joint angles to the Baxter robot.
- Flask and KinEval's Built-in Rosbridge module were tried to establish real-time connectivity between Python script and KinEval.

References:

[1] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann, "BlazePose: On-device Real-time Body Pose Tracking," 2020. arXiv:2006.10204.

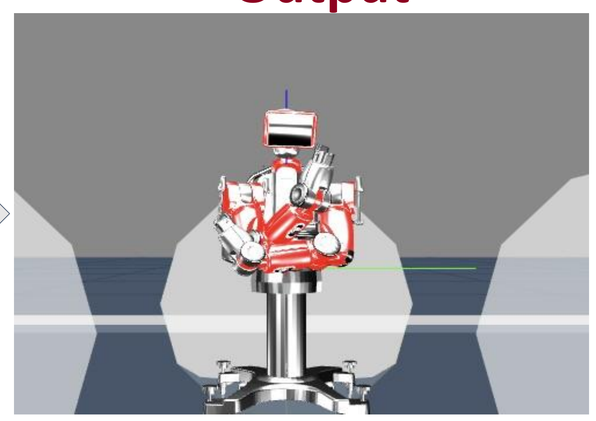
Mimicking human's body pose on robot in real time.

Input



Video feed through webcam

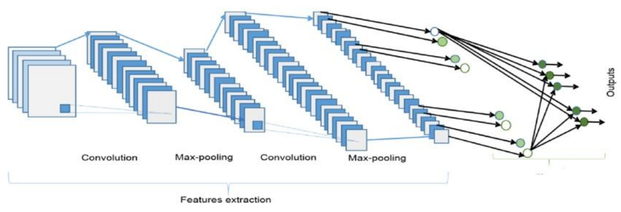
Output



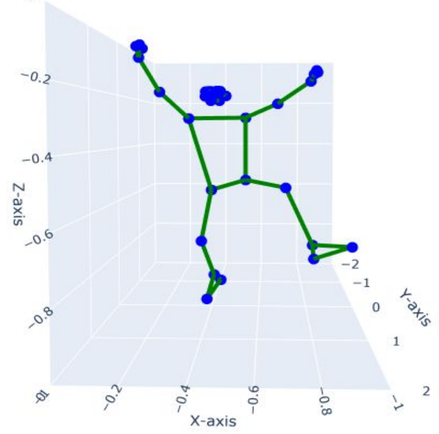
Baxter imitating the Pose

Step 1: Extract Coordinates

Input Frame



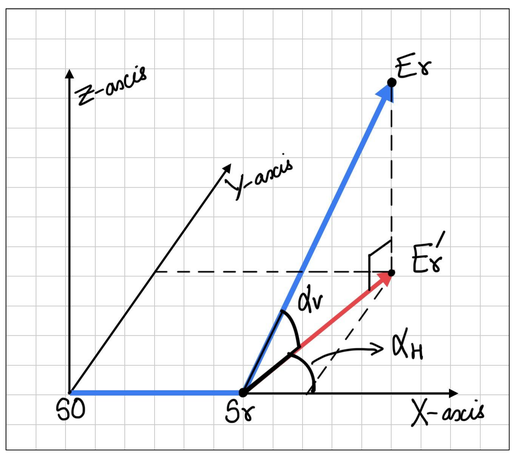
BlazePose



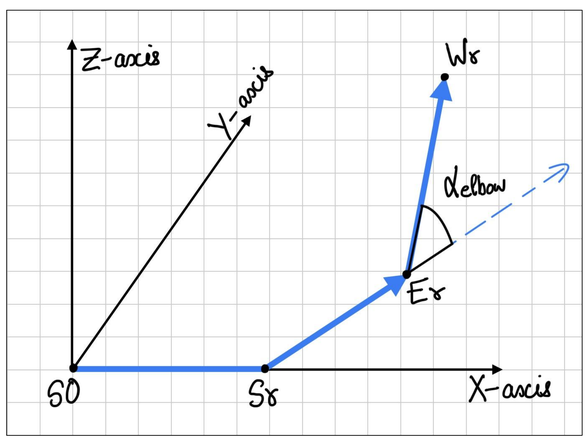
3D Keypoints estimation on the Frame

Step 2: Calculate angles to transfer human pose to Baxter

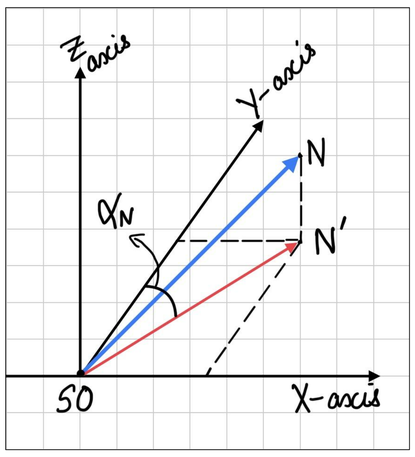
- 3D points obtained by BlazePose model are rotated about X-axis & then Z-axis to make the alignment as used below.
- The points are centered at midpoint of shoulders and then rotated to make X-axis align with lateral dir. of person.



Shoulder orientation in Vertical & Horizontal Plane



Elbow Flex angle.



Headpan angle.

S0 - M.P. of b/w Shoulders	Wr- Rt. Wrist Joint	N' - Projection of N in xy	αN- Nose angle from y-axis
Sr - Rt. Shoulder Joint	N - Nose	αH- Angle Shoulder in Hor. Plane	αelbow- Elbow flex angle
Er - Rt. Elbow joint	Er' -Projection of Er in xy plane	αV- Angle Shoulder in Ver. Plane	

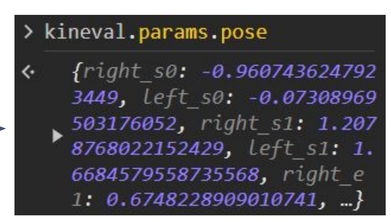
Note: Similar analysis was done to find the rotation angle of shoulder and movement of wrist.

Step 3: Send angles to KinEval & Apply Servo Control on Baxter

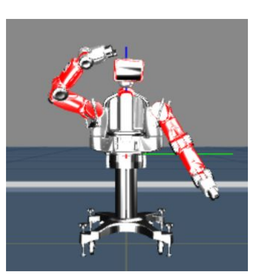
{'right_s0': -0.9607436247923449, 'left_s0': -0.07308969503176052, 'right_s1': 1.2078768022152429, … , 'left_e0': -2.004004914840902}

Calculated angles from webcam pose in Python

Socket.IO



Received angles in KinEval



Final pose executed by Baxter

computer science & engineering

MINNESOTA ROBOTICS INSTITUTE
UNIVERSITY OF MINNESOTA