Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Mid Year Report

Project Number:

Project #65: Guide and

logistics robot system:

taking elevator with

interaction

Finn Tracey

Cale Ying

Ho Seok Ahn

JongYoon Lim

21/07/2023

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Name: Finn Tracey

Name: Cale Ying

ABSTRACT: Human robot interaction is a topic that has been around for a long time, and recent advances in technology brings us closer to perfection. Our goal is to decrease the interactivity boundary between humans and robots.

1. Research Problem

Human Robot Interaction is a complex field that facilitates many subcategories, one of which is motion. Whether it is gesturing directly or making idle movements while talking to people, the problem comes with stereotypical uncanny robot movement patterns. This is the problem we are trying to solve, attempting to make robot movements as fluid and humanlike as possible, as well as teaching the robot common human gestures to be able to recognize and make decisions based on context.

2. Research Plan

Our research plan has several distinct stages. In the beginning, we had to research different methods of getting a robot to mimic motions. We discovered a subtopic that was a vital step for translating physical motions into digital space, Human Pose Estimation. This is an area of computer vision that is underdeveloped at this stage and requires a lot of attention for our research, since we needed to consider our options carefully. This was followed by research into NAOqi and how to interface with the Pepper robot in a way that answers our problem the best i.e.: how to change angles, how to pass information between different python environments.

The next step in our research plan is finding ways to improve our current solution, whether it's increasing accuracy or making movements smoother. A new topic to introduce in our research agenda is prior art, which includes motion capture technology, and how we can apply this knowledge to our solution. We must also research chatbots and how to introduce the best solutions to older Python versions.

3. Current Work

So far, we have completed a motion-mimicking system on the Pepper robot using uncalibrated videos from a webcam. Our implementation is also capable of real-time motion mimicking; however, it is only supported on the simulator, not on the physical robot. Since our focus is to put the Pepper robot in a receptionist position in a healthcare environment, we created a set of meaningful motions that could be mimicked by the Pepper robot. Below is a list of the motions we have generated:

- Beckon
- Big wave
- Bow
- Celebrate
- Head shake
- High five give
- High five receive
- Seagull (yoga)
- Showing the way
- Wave

Our implementation pipeline consists of three stages: human pose estimation, joint angle calculation, and motion mimicking. The human pose estimation (HPE) stage is where we obtain the joint positions from the human actor performing the motion. We are using MediaPipe Pose, a computer vision model for detecting human body landmarks. We chose this HPE model because it is lightweight and optimised for embedded devices and has the best balance of performance (accuracy) and speed compared to other options: OpenPose which is computationally expensive requiring VRAM, DeepPose not providing accurate results. Our setup uses the Intel RealSense D435 on a tripod, and we record each gesture individually. The joint angle calculation stage takes the 2D joint coordinates obtained from MediaPipe Pose and uses trigonometry to estimate the depth of each joint and calculate the 3D joint angles. The program is run on a laptop using Python 3.8. We then take the list of joint angles along with the timestamps for each joint, package it in a comma-

separated variable (.csv) file and send it to the Pepper robot to perform the motions. This motion-mimicking program is run on Python 2.7 using the NAOqi Python SDK to interface with the various API's that control the robot. Below is an example of each stage to generating each motion:

Figure 1. Motion Mimicking Process (Top left: Recording human pose, Bottom left: Run MediaPipe Pose on recorded video, Top right: Test motion on Pepper simulation, Bottom right: Run motion on real Pepper robot)

The results from our current implementation perform quite well with a few exceptions on motions such as 'showing the way' and 'high-five-give'. We tested the performance of our implementation by evaluating the calculated joint angles and comparing them to the output joint angles in the Pepper robot simulator, Choregraphe. Below shows the results of the 'wave' motion.

Fig 2. Joint angle comparison of the calculated joint angles in estimation program vs output joint angles of Pepper robot sensors.

These graphs allow us to identify flaws with the robots movements and how to address them, as well as applying filters and observing a precise and readable response.

Fig. 3 is a system diagram that demonstrates our most recent working pipeline. We plan to add more to this as we continue to improve our interactivity with the overall system. However, this one is fit for our purpose of research and evaluation

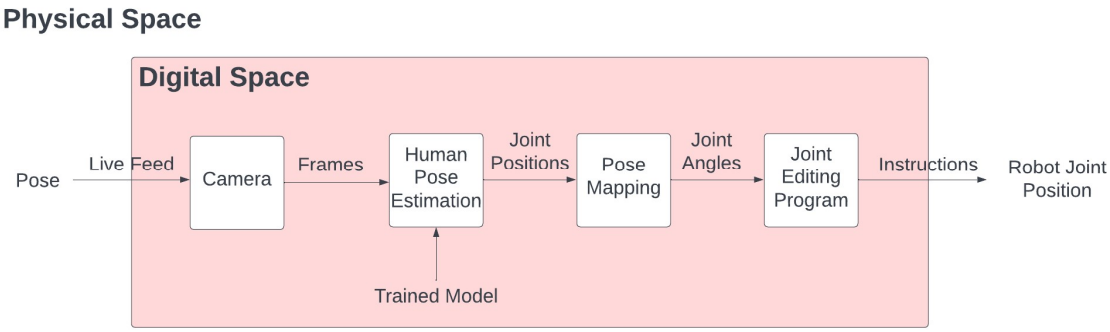

Fig. 3. System Diagram for Human Motion Mimicking.

4. Future Work

At this point in time, we have essentially completed the HPE section. However, we are constantly finding ways to improve our solution. We can increase the accuracy of the joint positions, which can make the motion-mimicking result more robust. We can also increase the smoothness of the joint movements with the help of filters and/or controllers as well as data post-processing, making the robot's movements more fluid and human-like. We can also improve the overall work

pipeline, making the application more efficient in terms of power and resource distribution while also improving ease of use from both a developer and a user perspective.

We have methods to improve our solution, primarily through evaluation. Currently, we have graphs of joint angles over a period which, when analysed, can help provide accuracy metrics to improve the process between the MediaPipe stage and the joint angle adjustment stage. Also, we will perform an evaluation of our outputs, gathering feedback that can potentially assist us in improving our solution even further.

Beyond this, we are also planning on implementing a chatbot in the pepper robot to improve the human-robot interaction and fitting it to a purpose i.e.: healthcare receptionist. We are also going to teach the robot to recognise different gestures to expand its ability to read the context and respond to situations in appropriate manners. We are also planning to implement real-time motion mimicking to improve ease of use for users and developers alike (Real-time motion mimicking can provide instant feedback.

5. Conclusions

The four months of work spent on this project consisted of researching the area of AI computer vision and human pose estimation, linking this research to our robot, and finding how to interface with the robot's software. After researching and producing our Literature Review, we began putting our research into practice.

As mentioned in the research plan, some of our future research can help us improve the overall improvements made to the solution, such as the smoothness being improved by researching different filters commonly used for robots or researching different post-processing procedures used for motion capture, or finding different methods of executing certain functions through the NAOqi documentation or research papers on how others used the Pepper robot.