# Body Imitation: Human Body Movement Recognition and Robotic Emulation

Alekhya Kuchimanchi
Email: akuchima@utexas.edu

Ruchi Bhalani
Email: ruchi.bhalani@utexas.edu

Emily Pietersz
Email: emily.pietersz@utexas.edu

***Abstract* - This paper discusses the development of a system that translates human body movement and replicates that movement on a humanoid simulation. This builds on previous ideas of using joint data to recognize poses, such as hand gestures or body language, and adds the implementation of robotic emulation through simulation. The approach we took to replicate human body gestures on a humanoid simulation consists of recognizing movement from the human by analyzing the changes in linear positions and quaternion rotations, then comparing these changes to our library of movements to determine which movement the human-made in a given set of frames. After recognizing the human's movement, the humanoid simulation is able to imitate the human by adjusting its limbs accordingly. After running our experiments, we found that the humanoid simulation was able to replicate the human's body movements with correct timing and a high degree of accuracy. Our system performs well for the current movements we have stored in our library, but this work can progress by expanding the number of movements the robot can recognize.**

## I. Introduction

People have long had the desire to achieve natural-seeming interaction with robots, and many efforts have been made towards advancing this field of human-robot interaction. Humans subconsciously use many non-verbal cues, such as body language, to communicate intent with one another, and it would be advantageous for robots to be able to understand them. Body-movement communication would require little to no user adjustment and would establish a channel of communication that is intuitive and understandable by both humans and robots.

We developed a system that detects a given body movement made by the human and imitates it on a humanoid robot. First, a series of body movements is captured on camera and the system uses the linear axes and quaternion rotations of joints to recognize it as one of the body movements from the library. After it catalogs the gesture, the same body movement is mimicked on the robotic simulation. Our library of body movements consists of the following: raising arms, lowering arms, squatting, and upward pushing (pictured in Figure 2).

This not only allows humans to communicate with robots but also allows robots to communicate back. Thus, a bi-directional communication channel is formed, facilitating human-robot interaction.

## II. Background

Body gesture and pose recognition have been explored by multiple groups using neural networks, the Microsoft Kinect System, ROS, and depth sensors. A group at the Foundation for Research and Technology Hellas utilized two Multi-Layer Perceptron/Radial Basis Function (MLP/RBF) neural network classifiers in a 9-dimensional configuration space to track the upper body. Classifiers were fed into a neural network known as the MLP Neural Network which had been trained to recognize 5 gesture states. The modeling and classification of each gesture were implemented through the use of joint angles and were able to determine the position, shape, size, speed, and duration of hand movements [1].

As opposed to recognizing movements, many projects focus on tracking the joints of the human body using camera and depth image analysis. The team at Microsoft developed the Microsoft Kinect camera to track the skeletal motions of the human body and gain human kinematic motion data. The real-time skeletal tracking provided by the Microsoft Kinect SDK is based on the depth data using body part estimation algorithm based on random decision forest [2]. Kinect is an instance of a structured light depth sensor and it has an IR projector that projects a known light speckle pattern into the 3-D scene which is invisible to the color camera but can be viewed by the IR camera. This allows the matching between local dot patterns, and the depth of a point is deduced by the translation of the dot pattern. This is different from the Azure Kinect SDK which implements the Amplitude Modulated Continuous Wave (AMCW) Time-of-Flight (ToF) principle to generate a depth map and a clear IR reading [3]. Other projects utilize the joint data provided by the Microsoft Kinect SDK to recognize body gestures and movements [4].

Depth image analysis allows for depth data to be used to extract body poses and gain relative body pose estimates. The group at Saarland University & MPI Informatik utilizes depth cameras to capture depth images. With these images, depth data is used to extract features of poses and hence reconstruct that pose in a background model. In addition, through the use of a pose database, the depth data was utilized to look up poses that the image may be making. The time-of-flight (ToF) cameras were used to capture 2.5D scene geometry at video frame rates. To capture depth/distance data the camera

measured the round trip time of infrared light emitted into and reflected from the scene. A pose was determined through the use of a kinematic change that models the motion of the actor and contains the position and orientation of the root joint as well as a set of joint angles of the actor. Furthermore, to obtain feature representation, a large number of geodesic extrema was computed which allows for the effective detection of effector positions of the body. [5]

While previous projects implemented human-to-robot communication, in which humans would communicate instructions to a robot via body gestures, our project provides human-to-robot-to-human communication. That is, the human relays a body gesture to the robot, who understands and imitates it back in a human-recognizable form.

This technology could later be implemented in efforts to aid with robot body gesture communication as the robot can emulate gestures it recognizes. As the robot has gestures it recognizes and can reproduce, this can be used for further work in human-robot interactions as the robot could learn to respond to certain gestures that it recognizes using the database of gestures it knows.

## III. Method

Our approach consists of two main components, namely body gesture recognition and imitation via the robotic humanoid simulation. As a part of both our recognition and imitation of body movements, we recognized and imitated the following: raising arms, lowering arms, squatting, standing back up straight, and upward pushing. We use an Azure Kinect camera to film body movements, the user mode SDK to track joint states and quaternion rotations, and the DARwIn-OP humanoid robot simulation using ROS to imitate the movements.

### A. Movement Recognition

The Azure Kinect SDK tracks the coordinates and quaternion rotation of 32 joints for each frame captured. We wrote a system that takes in a frame rate and calculates the differences in the body positions between a specified interval of frames to detect decipherable patterns in joint movements from one section to another. Through observing patterns in the changes of coordinates, we deduced that any significant motions (body movements that should be recognized and imitated) should take place within 10 frames. Thus, the difference in joint state between every 10 frames was recorded, and we put the differences into a database. To represent each difference, we calculated the change in the positions x, y, z as well as the change in the quaternion rotations qx, qy, qz, qw. We then analyzed the data to find which joints had the most significant change for each section of 10 frames, which are pictured in Figure 1.

From the analysis of the linear differences and also the angular differences, we determined which changes typically applied to a given movement in our library of gestures (see algorithm 1).

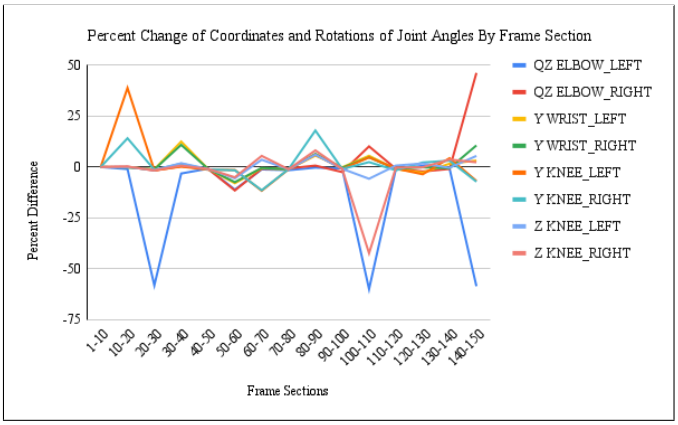


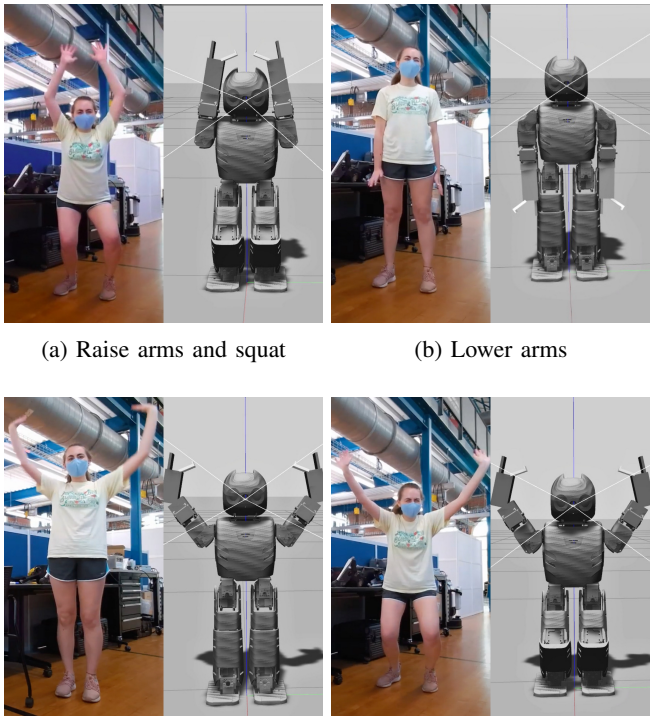

Fig. 1: Percent change of coordinates and rotations of joint angles by frame section

**Algorithm 1:** Simplified version of body recognition system

```
Result: Body movements from library recognized
retrieve joint state differences for each frame section;
while still have frame sections to iterate through do
    if qz of elbows indicates half rotation then
        raise the roof movement;
        if y of wrists indicates significant positive
         change then
            high level of upward movement in arms
    else if y of wrists indicates medium positive
     change then
        medium level of upward movement in arms;
    else if y of wrists indicates low negative change
     then
        low level of downward movement in arms;
    if y of knees indicates downward movement and z
     of knees indicates significant forward movement
     then
        squat;
end
```

### B. Robotic Humanoid Imitation

Collections of joints on the robotic simulation were mapped to the corresponding values tracked with the Azure Kinect. The system we developed processes the differences in joint states for each frame section and moves the robot's limbs accordingly, to account for an increased duration of a movement spanning multiple frame sections or multiple occurrences. We analyzed the differences in joint states observed through the Azure Kinect SDK and determined a series of ranges for the x, y, and z values for the hand, hand tip, and wrist to determine the level of movement and degree of motion. The "arms up" and "arms down" movements in the libraries have three levels of movement - high, medium, and low. These ranges were then mapped to one of the levels of movement to produce an accurate imitation of the user's body.



(a) Raise arms and squat (b) Lower arms

(c) Upward push (d) Upward push and squat

Fig. 2: The library of recognizable and imitable body movements

## IV. Evaluation

To test the system, a recording was taken with the Azure Kinect camera of a user performing all of the movements in the library at varying intervals, sometimes simultaneously. The joint states were obtained from the Kinect camera for each frame and then passed to our system, where it could be parsed to determine the differences for each frame section for a frame rate of 10. The system then ran through each frame section to determine which movements were performed and then proceeded to execute them on the robotic simulation. The humanoid simulation was run to compare its movements to the body movements of the original user and test if they were recognized and emulated correctly, with accurate timing, duration, and degrees of movement.

## V. Results

We found that the humanoid simulation was able to imitate the human's body movements with a high degree of accuracy as pictured in Figure 2. Not only was the humanoid simulation able to imitate certain movements from the human, but it was able to imitate multiple movements at a given time. For example, as seen in Figure 2a, the humanoid simulation was able to properly imitate both the squatting movement along with the raising of the arms and in Figure 2d in which the robot was able to squat while simultaneously pushing upwards. When comparing the video of the human to the humanoid simulation, the timing and accuracy of the movements were nearly exact.

## VI. Discussion

We found that if specific clusters of joints could have their movements measured in different axes, then the movement of a person can be interpreted into a body gesture, which can be mapped to a robotic humanoid simulation. This can be helpful in training robots to perform specific movements, as people wouldn't have to program the robot's motions themselves, they would simply be able to execute the movement and have the robot copy them and build its own library of recognizable motions. Facilitating the learning of service bots would also take the burden off the user to make the robot execute specific motions.

Additionally, determining body movements and positions has significant implications for VR. The automatic recognition of body movements with a camera means that that the user does not have to be outfitted with numerous sensors and controllers, since a camera with a depth sensor can simply map their joint states to a recognizable body movement in its library. The ability to emulate these movements means that the VR system can also render the movement in its environment, making the virtual reality more realistic, since people would be able to interact with their virtual environment without a lot of heavy equipment.

Finally, body positioning helps facilitate human-robot interaction. The recognition component of this system allows the human to communicate with the robot, since the robot can recognize and interpret the body movement, taking the burden off the user to adapt to its language. The robot can process human movements and react appropriately. However, the system also allows the robot to communicate with the human, through the imitation of the movement. Since the movements are human-understandable body gestures, this allows robots to perform gestures that humans can understand. The robot can process input and convey human-understandable output, allowing for a bi-directional channel of communication to be established.

## VII. Conclusion

We developed a system containing two main components: (1) to identify human body movements and be able to categorize them as an element of our body movement library, and (2) to mimic the given body movement on the humanoid robot simulation. The Azure Kinect tracks the joint positions and rotations of the body movements, and our system analyzes the significant clusters of joints to map them to one of the body movements in our library. Once the motion is recognized, the system processes the body-tracking data by the frame rate and uses the original recording to determine the duration, frequency, and degree of movement for the body movements. The humanoid simulation was able to demonstrate the imitation of the body movements in our library with a high degree of accuracy, even being able to imitate movements that occurred simultaneously.

The ability to recognize and emulate these body movements holds many implications for the training of service bots, VR, and the establishment of a bi-directional channel of communication that allows humans and robots to understand

each other with little to no user adjustment. The robot's capability to emulate the body movements that it recognizes eases the burden on the user in many different fashions. Body movements of humanoid service robots no longer have to be programmed manually, since they can be taught by watching a human execute the same motion. The amount of VR equipment decreases since sensors and various controllers are no longer required for a user's body to be mapped to the virtual reality; the depth sensor and camera can analyze their body movements well enough for them to physically interact with the virtual world. Humans do not have to learn an entirely new language of specific commands just to communicate with robots since body movements are understandable by both the system and humans.

While the system performs well for the movements currently in the library, it does not recognize all possible body movements. Future work includes further generalizing the algorithm to include many more body movements in the library, analyzing the movement and rotations for all joints to determine which body movement the human is executing.

## REFERENCES


[1] M. Sigalas, H. Baltzakis, and P. Trahanias, "Gesture recognition based on arm tracking for human-robot interaction," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5424–5429.

[2] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR 2011*, 2011, pp. 1297–1304.

[3] P. Meadows, Y. Wang, and T. Sych, "Azure kinect dk documentation." [Online]. Available: https://docs.microsoft.com/en-us/azure/kinect-dk/

[4] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, "Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect," in *2015 International Conference on Healthcare Informatics*, 2015, pp. 380–389.

[5] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *2011 International Conference on Computer Vision*, 2011, pp. 1092–1099.