Bachelor Thesis

Humanoid Robot Control from Human Joint Angles via 2D Camera

by

M.Bahadir Kucuk

(mkk332)

First supervisor: Dr. Kim Baraka
Second reader: Prof. Dr. Koen Hindriks

July 18, 2022

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science

Humanoid Robot Control from Human Joint Angles via 2D Camera

M.Bahadir Kucuk
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
m.b.kucuk@student.vu.nl

ABSTRACT

Control over humanoid robots is necessary to facilitate them in many fields. With Robot Learning from Demonstration (LfD), humanoids can be taught and trained in the physical workloads that humans have to do. Here, it is the feasibility of controlling humanoid robots that holds utmost importance. The goal of this research is to investigate how feasible it is to operate a humanoid robot using human joint angles captured by a 2D camera. In order to handle embodiment mismatch, we use direct mapping of human joint angles on a humanoid robot. Our method does not require depth information on the 2D image to transfer the angle of abduction/adduction of human joint movements. We utilise a similar generic method to get the angle of flexion/extension of human joint movements. The external/internal rotations of human joint movements which stand for yaw control of humanoid robot joints are not included in this research. The evaluations are made by comparing angles obtained by our approach with ground truth angles and experimenting with 16 participants. The evaluations of our approach suggest that humanoid robots imitate abduction/adduction human movements quite successfully when compared to those of flexion/extension. Our work reveals that controlling a humanoid robot using human joint angles captured by a 2D camera is feasible. Finally, we implement an interface of our approach that provides control on our approach's output of head nod and tilting movement angles, open/closed hand status classifier, and human upper body joint angles based on human joint 3D coordinates captured by a 2D camera.

1 INTRODUCTION

Humanoid robots are being used in routine tasks of astronauts in space, helping with daily tasks of elderlies, providing companionship for the sick, interacting with customers, dangerous tasks, and heavy work instead of human labourers [2]. They are designed to imitate the behaviour and external appearance of humans. They have similar kinematics and reactions to humans including the way they move and respond. People need to control humanoid robots since they are used to fill the gap that people cannot reach and to make them involved in the workloads that humans have to do. To illustrate, an elderly wants his social humanoid robot to carry heavy furniture at home. Using complex devices for controlling humanoid robots in every environment is not feasible for everyone at any age of their life. Most elderly have trouble reading text and have an inadequate level of technical knowledge. The idea that the elderly control a humanoid robot to make it help in tasks which is hard to do for the elderly without a complex device has utmost significance in that sense. Moreover, people need to control humanoid robots in a straightforward way because people can teach robots to perform a task, for example via Robot Learning from Demonstration (LfD). It aims that users can teach robots a task without the help of a programmer or controller. For example, a mess attendant in a hotel wants to teach a robot order to prepare fruit juice. In each sub-task, the robot imitates an end-user who has no knowledge of programming and teaches the robot this task by demonstration. The mess attendant prepares fruit juice step by step and the robot imitates. In the end, the robot learns and does itself [11]. In both cases, it is obvious that the aim is to make robot control available for everyone to use robot capabilities for more extended fields and adapt for everyone in daily life.

Humanoid robots require motion control to perform a task that is proper to their creation aim. There are many ways to control a robot intuitively: using speech commands, keyboard control, gesture control, wearable suits, and vision-based control. All these methods have advantages and drawbacks according to their usage environment. An operator having a high knowledge of technology can control a sophisticated humanoid robot in critical duties such as defusing a bomb but controlling these robots for normal tasks should not require these skills if a person wants to let humanoid robots perform normal tasks in daily life. The human pose estimation based on 2D vision is one of the effortless methods to control a humanoid robot. In today's world, a wide range of people has access to smartphones or laptops, which have a built-in 2D camera. On the one hand, the idea of controlling humanoid robots via only these devices can make an expansive amount of people involved in teaching robots by demonstrating or controlling them for imitation learning [10]. On the other hand, this approach brings some limitations with it. For example, the user has to show a front view of himself, and detecting depth value with the 2D camera is not a flawless approach to obtain depth information from a frame compared to wearing body suits with sensors or using the 3D camera.

Obtaining body landmarks' coordinates of humans is not enough to operate humanoid robots. Existing models require analytical or mathematical modelling of the humanoid robots such as forward or inverse kinematics in order to operate humanoid robots. To achieve controlling humanoid robots, the embodiment mismatch problem has to be handled on the body landmark coordinates that are captured by bodysuits or cameras since the users may have different forms of bodies [16]. Regarding this, direct angle mapping of human joint angles on humanoid robot's joints solves the embodiment mismatch problem since human body joint angles do not differ from body form to form. This study does not come up with a new solution for capturing human body landmark coordinates but human body joint angles are calculated with different methods based on body landmark 3D coordinates via a 2D camera obtained by the existing skeleton tracking library. Having solved this problem, joint angle control has another advantage. It takes less time to

get from one position to another position compared to trajectory control [3].

The goal of this research is to investigate how feasible it is to operate a humanoid robot using human joint angles captured by a 2D camera. In this paper, we present how we obtained an open-closed hand status classifier and angles of the human head, shoulders, and elbows from the 3D coordinates of the MediaPipe Library on a 2D frame. Our immediate aim is to indicate the feasibility of controlling a humanoid robot via a 2D camera. To achieve that, we translated human joint angles to robot joint angles and transferred these angles on humanoid NAO and semi-humanoid Pepper robots in Pybullet simulation to see the functionality of these angles on a humanoid robot. We provide an open-source interface to have control on the output of our approach. We came up with 2 evaluation methods. A comparison module evaluates our approach's angles with the ground truth angles obtained by CMU Panoptic Dataset [1]. We experimented with 16 participants to measure to what extent our approach's angles work while controlling a humanoid robot with 16 people. Also, we present a documented open-source code of our work.

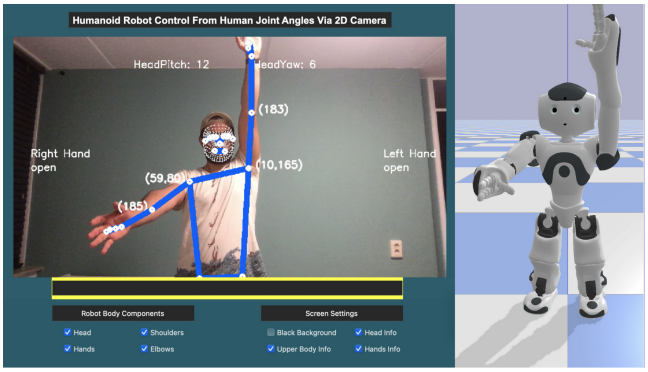


Figure 1: ***Humanoid robot Nao imitates a human position. The joint angles are given in parentheses. (abduction/adduction, flexion/extension) is for shoulder's angle and (flexion/extension) for elbow's angle.***

2 RELATED WORK

There is a wide range of work that is based on estimating human pose. These works are mostly accumulated in obtaining human body landmarks via wearable suits and via computer vision. In the computer vision part, there is a distinction between third-person and egocentric pose estimation, which literally describes the position of the camera with respect to the human. Moreover, the ways of controlling humanoid robot joints are also related to our research.

2.1 Human Body Landmarks via Wearable Suits

C. Stanton et al. [25] approach involves training a feed-forward neural network for each DOF. This manages humanoid robots to learn a mapping from capture suit to robot actuator in angular position. For each actuator, a neural network is allocated. The main advantage is that their methods can be applied to the calibration of any-human robot pairing thanks to the fact that they do not use forward or inverse kinematics. The formulation of Akhter and M. J. Black [7] for a prior human pose shows how their method restricts invalid poses in 2D to 3D human pose reconstruction and how they propose pose-conditioned joint angle limits, which can be applied to many problems in human pose estimation. The model of M. El-Gohary and J. McNames [14] provides cooperation between gyroscope and accelerometer random drift models and enforces physical constraints on the range of motion for each joint. Moreover, the model uses zero-velocity updates in order to smooth the effect of sensor drift. The majority of this work draws inspiration from the accuracy of landmark estimations captured by bodysuits versus cameras. Poitras et al. [22] present an evaluation of the validity and reliability of M/IMU for each body joint by screening different databases (Pubmed, Cinhal, Embase, Ergonomic abstract, and Compendex). They point out the variation of validity according to the complexity of tasks.

2.2 Human Body Landmarks via Computer Vision

Several computer vision algorithms are designed to obtain body landmarks' 3D coordinates from a frame, which are generally 2D.

Third-Person Pose Estimation G. Rogez et al. [23] present a work to localise and recognize the human pose by using a randomised hierarchical cascades classifier and an exemplar-based approach. Their work is capable of more challenging scenarios such as moving cameras and extensive viewpoint changes without any prior assumption. They highlight the efficiency of the cascade approach, which votes on different sets of features. E. Brau and H. Jiang [12] propose a deep convolutional network for 3D human pose and camera estimation from the 2D images of human joints. Their model learns a 3D prior over poses which have factors related to limitations of kinematic and self-intersection. They enforce camera projection operation and limitations by using a network layer. The output of this layer is designed to feed into the L2 norm loss function. H. Jiang [18] presents an exemplar method, which is based on the kd-tree and achieves real-time performance to estimate 3D human poses from single images by using only the joint correspondences. The challenge of their method is to estimate 3D poses from a monocular view in obtaining the depth ambiguity, which is a common problem in capturing body landmark coordinates based on computer vision. Their approach to the problem is to search through millions of exemplars for optimal poses. A. Guler et al. [15] highlight two new directions for automatic human gesture recognition and human joint angle estimation by human-robot interaction. The first one was investigated with a framework based on deep learning to tackle real-world noisy RGBD images. The second one has dense trajectory features for processing real-time videos in order to use in automatic gesture recognition.

Egocentric Pose Estimation Y. Yuan and K. Kitani [26] come up with a control-based approach to model human motion with physics simulation and use imitation learning to learn a video-conditioned control policy for ego-pose estimation. In their approach, the view of the camera is egocentric, which is different from traditional computer-vision-based approaches.

2.3 Controlling robot joints

Several works have been reported on controlling robot joints in the literature. The methods are generally based on the direct angle mapping method, cartesian control, forward kinematics, and inverse kinematics using iterative Jacobian and fuzzy logic. M. Zhang et al. [27] present work on chain-based inverse kinematics for real-time imitation of upper limbs. The joint angle configuration and end effector trajectory are built on each arm to provide motion similarity between them. C. Li et al [21] present a tracking system for the human head and arm via a Kinect v2 sensor in order to interact with Nao robots. Also, kinematics equations are applied to transfer Euler angles of the human head and arms from Kinect coordinates to the ground coordinate. J. de Lope et al. [13] demonstrate a method based on artificial neural networks which learn relative foot positions and orientations to solve inverse kinematics of humanoid robots. M. Bergerman and Y. Xu [9] provide a demonstration of the feasibility of designing a controller's robustness to parameter certainty. A. Roncone et al. [24] present a complete gaze control stabilisation architecture for a humanoid robot, which is focusing on the exploitation of the redundancy of the kinematic problem. A. Jaume-i-Capó et al. [17] present how to add image constraints to the inverse kinematics formulation to solve the lack of information for locating the internal joints.

In contrast, this research does not come up with a new technique for capturing human joint coordinates but it is based on using MediaPipe Library in a practical way to get the human angles to observe how feasible to control the humanoid robots based on estimating 3D human pose on the 2D frame.

3 METHODOLOGY

3.1 Overview

The aim is to provide 3D human joint angles and open/closed hand status from a single 2D camera that does not require calibration taken from a third-person view and then control these relevant joints of a humanoid robot. We obtain the human body landmarks' coordinates with an existing library and we calculate angles with the methods in the head, hands, shoulders, and elbows modules. After this process, we have an output of our approach which has related information on these body components. We tailor the angles to apply to Nao and Pepper robots in the simulation. A user can activate or deactivate the body components information on the output of our approach with the help of the interface that we implemented. Also, the user can open/close the black background of the camera frame with the interface.

3.2 Discussion of Each Component

3.2.1 *3D Coordinates of Human Body Landmarks.*

We obtain a 2D video frame from a single webcam of an ordinary laptop and convert this BGR image to an RGB image. Then we provide this image to Face, Hand, and Pose Mesh.

Face Mesh Two deep neural network models are used to estimate 468 3D face landmark coordinates in real-time. The first one is a BlazeFace [8] detector operating on this RGB image and computes face locations. Then, these locations are operated by a 3D face landmark model [20] which predicts the 3D surface with regression on the canonical face model [6].

Hand Mesh Multiple models are used to predict 21 landmarks 3D coordinates of a hand in real-time. Firstly, a palm detection model is used to detect palm which is easier than detecting fingers. This model returns an oriented hand bounding box with the operation on the RGB image. Secondly, a hand detection model processes the cropped image which was taken by the palm model, and returns the high-fidelity 3D hand landmarks (see Figure 7 in Appendix B) [6]. Also, the output of this model provides a label of the left/right-hand classifier.

Pose Mesh Two machine learning models are used to estimate 33 body pose landmark coordinates in real-time. The first one is the person/pose detection model which is called the BlazePose detector. This model predicts the human body center, rotation, and the circle scale of the body. With the help of this information, the midpoints of a person's hips, the radius of a circle circumscribing the whole body, and the incline angle of the line between hip midpoints and shoulder are predicted by inspiring Leonardo's Vitruvian man. After that, the pose landmark model which is called BlazePose GHUM 3D predicts the coordinates of 33 body pose landmarks (see Figure 9 in Appendix D) by locating the pose region of interest [6].

Libraries We capture the video frame and process this frame with the OpenCV library. We used models of MediaPipe solutions to be able to obtain the coordinates of human landmarks on a 2D frame.

3.2.2 *Human Joint Angles.*

Stage 1 - Head Module

Having had head landmark 2D and 3D coordinates from Head Mesh, we obtain a rotation vector with the help of Perspective-n-Points in order to detect the rotational motion of the head based on the nose with respect to the camera. Here, we take image width as the default focal length in the calculation of the camera matrix and we take the default distortion coefficient which is zero. With the help of Rodrigues transformation, we convert the rotation vector to a rotation matrix. We pass the rotation matrix to the upper triangular matrix(R) and orthogonal matrix(Q) decomposer in order to get the Euler angle rotation around the x-axis and y-axis of the nose in the image plane by feeding the rotation matrix. After normalising the Euler angle to a degree, we have the nod angle of the head gesture between -45 and 45 which is head pitch in a humanoid robot, and tilting of the head between -90 and 90 degrees which is head yaw in a humanoid robot.

Implementation We use the OpenCV library for Perspective-n-Points solver to get the rotation vector, Rodrigues method to convert the rotation vector to a rotation matrix, and RQ decomposer to get the Euler angles.

Stage 2 - Hands Module

In this module, we classify both hands' open/closed status. We have the hand landmark coordinates and left-right handedness classifier labels that are obtained by Hand Mesh. In order to extract the status of whether a hand is open or closed, we calculate the angle of four fingers: index, middle, ring, and little finger. Since three points are needed to calculate the angle, we use three landmark coordinates for each finger. Two points for each finger are the tip, metacarpophalangeal (MCP) of the relevant finger and the third

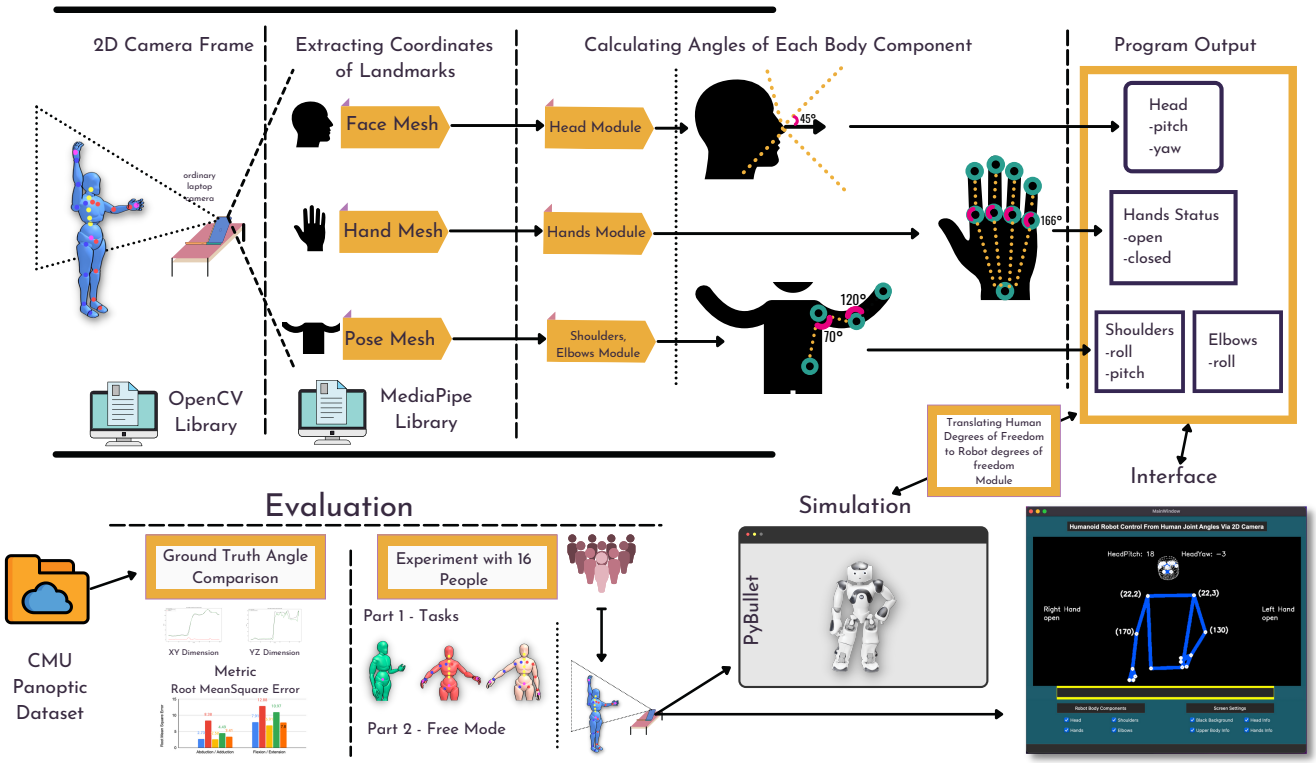


Figure 2: ***General overview of our methodology***

point is the wrist for the angle of each finger. To illustrate, for the index finger we use the landmarks 8 (tip), 5 (MCP), and 0 (wrist) (see Figure 7 in Appendix B).

All finger angles are almost 180° degrees at the open status of the hand. We put a threshold value to detect the closed status of the hand for each angle of these four fingers, which is 80°degrees. If all angles are less than this threshold then we label the hand as in the closed status.

Stage 3 - Shoulders and Elbows Module

We have a generic method to get the angle of abduction and adduction of human joint movements which stands for roll control in humanoid robots (see Figure 8 in Appendix C). There is no need to have depth information on the image for this movement. Thus, we use the x-y dimension to get the angle of the roll. We also utilise this method to get the angle of flexion and extension of human joint movements which stand for pitch control in humanoid robots. We use the y-z dimension to get the angle of these movements. The external/internal rotation of human joint movements which stands for yaw control of humanoid robot joints is ignored in this research.

These generic methods cannot be applied to all joint movements since some of the joints do not have a degree of freedom literally, for example, elbows do not have abduction/adduction movements but flexion/extension human joint movement and also external/internal rotation human joint movement. In this case, we accept flexion/extension human joint movement as elbow roll rather than elbow pitch since Nao and Pepper robots have only control of elbow roll and yaw, not pitch and we calculate elbow roll on the x-y dimension, we ignore all yaw controls.

Implementation of Shoulders Module To calculate shoulder angle, we need two other points which are the hips and elbows. After taking the relevant hip and elbow with the relevant shoulder on the same side, we calculate the angle on the x-y dimension for shoulder roll and the y-z dimension for shoulder pitch.

Implementation of Elbows Module To calculate the elbow angle, we need two other points which are the shoulder and wrist. After taking the relevant shoulder and wrist with the relevant elbow on the same side, we calculate the angle on the x-y dimension for elbow roll.

3.2.3 *Tailoring and Translating Angles to the Robot.*

The degrees of freedom of human and humanoid robots are not the same. To illustrate, humans can get arms up over head positions with the abduction/adduction movement or flexion/extension human movement. However, most humanoid robots cannot come to this position with abduction/adduction movement which is shoulder roll but flexion/extension human movement which is shoulder pitch. For example, Nao and Pepper robots have degrees of freedom in shoulder roll between -18 and 76 degrees on the left shoulder but they have degrees of freedom in shoulder pitch between -119.5 and 119.5 degrees [4]. Another problem is that there are some situations where angles are calculated as complemented. For example, after 180 degrees angle is not calculated as 181, 184 degrees but -179, -176, etc. This causes humanoid robots to take inaccurate positions.

Thus, we also have control over this kind of situation and translate our angles to the applicable one.

3.2.4 *Simulation and Interface*.

Qibullet library [1] on Pybullet simulation [2] is used for Nao and Pepper robots. The Nao robot is a humanoid robot that has 25 degrees of freedom, 7 touch sensors located on the head, hands, and feet, sonars, and an inertial unit to perceive its environment and locate itself in space. Also, it has 4 directional microphones and speakers to interact with humans and two 2D cameras [5]. The Pepper robot is a semi-humanoid robot that has 20 degrees of freedom, and perception modules to recognize and interact with the person talking to it. Touch sensors, LEDs, microphones, infrared sensors, bumpers, an inertial unit, 2D and 3D cameras, and sonars for omnidirectional and autonomous navigation [5]. We implemented a user interface to control our approach's output. (see Figure 10 in Appendix E) With the help of the interface, a user can open and close the desired body components' information in our approach's output that we feed the robot in simulation. We used the Python3 programming language and PyQT5 library [3].

3.2.5 *Comparison Module*.

CMU Panoptic Dataset includes HD videos and the 3D coordinates information of the human frame by frame in the video. In the comparison module, we parse the data of the CMU Panoptic Dataset and extract the human coordinates from the related frame to obtain ground truth 3D coordinates. Then, the coordinates of the left hip, shoulder, and elbow are processed to calculate the joint angle on the left shoulder, which we choose as the joint that we run our evaluation of ground truth angle comparison. On the other hand, we obtain the left shoulder angle of our approach from the same video, and then we provide a line graph, the root mean square error, and standard deviation. These steps are processed for both adduction/abduction and flexion/extension of human movements on the left shoulder.

4 EVALUATION

4.1 Study Design

We made two separate evaluations of our approach. The first evaluation is a comparison of human joint angles between our approach and the CMU Panoptic Dataset [19] that we accept as ground truth. The second evaluation is an experiment with 16 participants.

4.1.1 *Ground Truth Angle Comparison*.

This evaluation reveals the correctness of the human joint angles which are calculated based on the MediaPipe library's pose estimation solutions with respect to ground truth angles. In this evaluation, there are two different types of comparisons. These comparisons are to demonstrate the difference between abduction/adduction human movements (x-y dimension) which is not using depth and flexion/extension human movements (y-z dimension) which is using depth in a 2D camera. To achieve that we assume the human joint coordinates of the CMU Panoptic Dataset are ground truth since the videos in this dataset are captured based on 480 VGA cameras, 31 HD cameras, 10 Kinect Sensors, and 5 DLP Projectors. All sensors and cameras are synchronised among themselves using a hardware clock, timing aligned with each other [1].

We take 5 different videos, each with a hundred frames, from the CMU Panoptic Dataset. Each frame comes with human joint coordinates in a separate file. The body landmark points combinations of relevant joints to calculate angle are parsed with the help of our comparison module and we use our generic method to calculate human joint angles based on ground truth coordinates.

We demonstrate the difference between the human joint angle of the ground truth angle and our approach's angles. We take the root mean square error as a success metric for both comparisons separately. We make this comparison on only the left shoulder because we believe that this gives adequate insight into the comparison of the coordinates captured by Mediapipe Library [4], which our approach uses, and CMU Panopticon Dataset that we assume as ground truth coordinates.

4.1.2 *Experiment*.

In the second evaluation, we conducted an experiment with 16 participants in order to measure the feasibility of to what extent a humanoid robot can imitate participants based on their human joint angles and to see the limitations of this work.

The setup of the experiment is that we run our approach's interface besides the Pybullet simulation in MacBook Pro 15-inch (Late 2016). A participant stands in front of the 2D camera which we used as the built-in FaceTime camera for this MacBook. The best distance, especially for head and hands estimation, to the camera is at least one and a half meters to the camera but it is not a mandatory condition. Human pose estimations of MediaPipe can be detected even more than 3 meters as we observed. This experiment is conducted with an individual participant in front of the 2D camera. The participant looks at the screen and observes to what extent the humanoid robot in simulation imitates the participant while taking different positions. At the end of each part, the participant evaluates the imitation of the humanoid robot in the simulation. We used the Nao robot for 8 participants and the Pepper robot for 8 participants in the simulation. Our approach performs better under normal light conditions and separable background from the participant's clothes colour.

There are two separate parts in the experiment. In the first part, there are five different tasks (see Figure 11 in Appendix F) that a participant tries to get the position of the relevant task and observe the humanoid robot on the simulation screen. The first task positions of part 1 are based on abduction/adduction, and the second, third and fourth tasks are based on both abduction/adduction and flexion/extension. Task 5 is based on head movements and open/closed hand status.

The second part of the experiment is a free mode in which participants take positions without any instruction from the researcher. The only restriction in free mode is that the participant has to move upper body joints. After three minutes of observation, the participant evaluates the performance of the humanoid robot.

This data is collected according to the ethics review committee of the Faculty of Science (BETHCIE), Vrije Universiteit Amsterdam

[1] https://github.com/softbankrobotics-research/qibullet
[2] https://pybullet.org/
[3] https://pypi.org/project/PyQt5/
[4] https://google.github.io/mediapipe/

(see Appendix J), and the informed consent form of Vrije Universiteit Amsterdam in the example evaluation form. (see Appendix K). Participants' age, gender, and technology level are collected to make assessment evaluations of participants. Eight participants experimented with the Nao robot and the remaining participants experimented with the Pepper robot. In the first part (tasks), we asked participants to evaluate to what extent a humanoid robot imitates you after each task and, in the end, a general impression of humanoid robot movements. In the second part (free mode), participants assessed whether the humanoid robot moves rigidly or elegantly, to what extent it imitates participants' movements, to what extent it is easy to control, and the general impression of the humanoid robot's movements. All data collection is done in the range of one to ten linear scales. The success metrics of the evaluations are the average rate of the participants on the different types of movements in tasks.

4.2 Results

4.2.1 *Ground Truth Angle Comparison.*

Figure 3 shows the root mean square errors of human left shoulder's angle comparison between our approach and CMU Panoptic Dataset which we accept as ground truth angles. This figure highlights the root mean square error difference between abduction/adduction and flexion/extension human joint movements in each comparison. It is obvious to see the abduction/adduction movements have a lower root mean square error in each separate comparison, which does not require depth information in a 2D camera. The detailed root mean squared error and standard deviation of each comparison are presented (see Figure 12 in Appendix G).

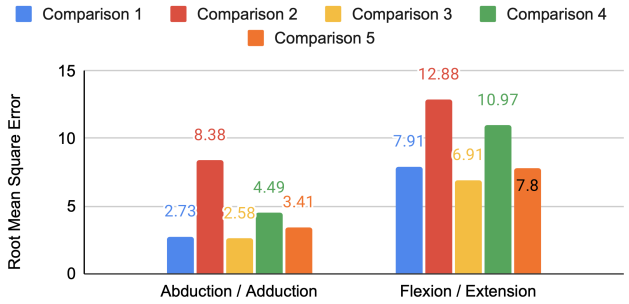


Figure 3: ***Evaluation I Results - The root mean square error of human left shoulder's joint angle comparison between our approach and ground truth. Each angle comparison is made in 100 frames.***

The figure [see Figure 13 Appendix H] demonstrates the angle comparison of the left shoulder's abduction/adduction movement between our approach and the CMU Panoptic Dataset for 100 frames. The root mean square error is 2.58 and the standard deviation is 1.43. It is important to see the overlaps between our approach and the ground truth angle. The figure [see Figure 14 Appendix I] shows the angle comparison of the left shoulder's flexion/extension movement between our approach and the CMU Panoptic Dataset for 100 frames. The root mean square error is 12.88 and the standard deviation is 10.31. It is obvious to see this movement type has more error than the abduction/adduction movement type since flexion/extension movement requires the depth information of the image. The graphs are examples to illustrate our approach. They are not actual meaningful results on their own.

4.2.2 *Experiment.*

The experiment reveals the feasibility of controlling humanoid robots using human joint angles via a 2D camera. Figure 4 demonstrates positions that are taken by participants and humanoid robots one by one. The body orientation of the participant is not taken into account. The camera is fixed to the front view of the participant. It is obvious that humanoid robots imitate humans accurately.

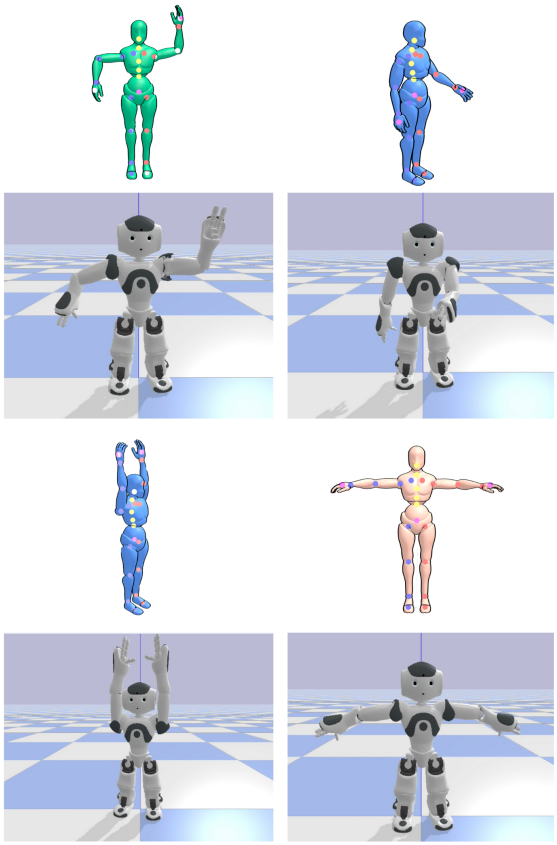


Figure 4: ***Humanoid robot Nao takes the participant's body position.The body orientation of the participant is not taken into account. The camera is fixed to the front view of the participant.***

After each task and part of the experiment, participants evaluated our approach. As it can be seen in Figure 5, there is no considerable difference between Nao and Pepper robots in each task other than in task 4 of part 1.

		Nao Robot	Pepper Robot
Part 1	Task 1	9.625	9.125
	Task 2	8	8
	Task 3	7.875	7.125
	Task 4	8.125	6.875
	Task 5	8.375	8.625
	General Impression	8.25	7.625
Part 2	Rigidly-Elegantly	8.125	8.25
	Mirroring	8.125	7.375
	Easy to control	8.25	7.125
	General Impression	8.25	7.5

Figure 5: ***Average choice of participants between the range of one to ten in the experiment.***

5 DISCUSSION

In this research, we aimed to demonstrate the feasibility of controlling robots using human joint angles captured by the 2D camera. The evaluations of our approach suggest that humanoid robots imitate abduction/adduction human movements quite successfully than flexion/extension human movements. As demonstrated in Figure 3, abduction/adduction movements have less root mean square error in all angle comparisons between our approach and ground truth on 100 frames since our method does not use depth information in calculating human joint angles in these movements. The flexion/extension movements that require depth information come with more errors. Through analysing Figure 5, the humanoid robot in the simulation manages to imitate the human positions according to the evaluations of participants, with an approximate average of 8 out of 10. The fewer points of participants are related to flexion/extension movements. Moreover, there is less than one point difference between Nao and Pepper robots in each sub-part of the experiment. We conclude that these differences are related to different ranges of joint angles on both robots. While experimenting on imitating the open/closed hand status of participants on the humanoid robots, we were confronted that our approach did not work on different skin colours. It is significant to mention as an anecdotal remark that the MediaPipe library is not successful in estimating human hand landmarks in dark colour skins with respect to light colour skins. Our work reveals that controlling a humanoid robot using human joint angles captured by a 2D camera is feasible in most environments. Other researchers can build upon our approach and improve our method.

There are many limitations of our approach. The first and foremost one is the self-intersection problem of computer vision, for example, the occlusion of elbows as can be seen in Figure 6. This occurs due to the camera's taking only from the front view of a human who holds arms straight against the camera but it is detected as elbows are bent. Thus there becomes no difference between straight arms and bent arms if the human takes this position. The human joints' external and internal rotations of shoulders and elbows are ignored, which is yaw control in the joints of the humanoid robots. This limits the humanoid robot to take the position of hands up/down and the capability of elbow and shoulders movements. Robots' degree of movement is not enough to imitate humans. Thus, people can think that the robot does not imitate in some points however this is related to the maximum and minimum turning capability of robot joints. To illustrate, the Nao robot cannot roll the shoulder more than 76 degrees. When a human holds arms at 90 degrees in an abduction position, the Nao robot cannot get this position. However, the humanoid robot can get this human position with the help of yaw control but our approach ignores yaw control in the shoulders and elbows. Another limitation is the importance of the background of the person in front of the camera to distinguish the person, and the level of the light in the environment to get the best solutions.

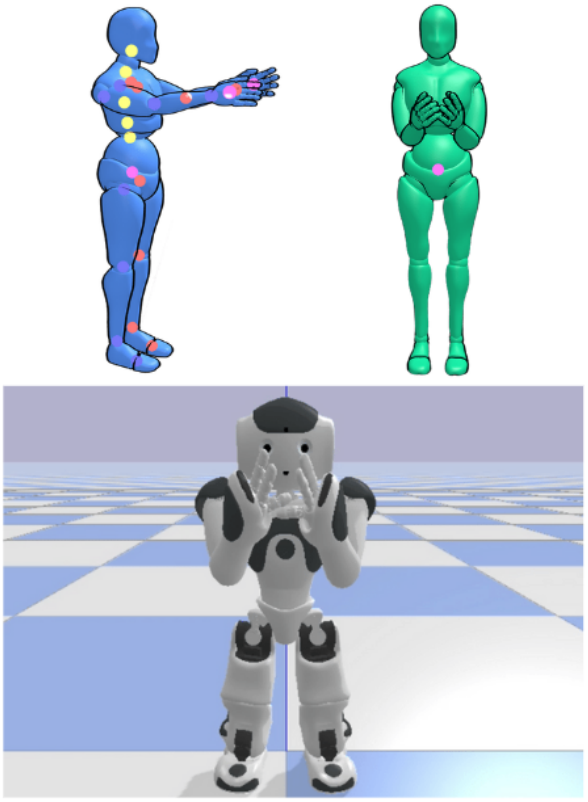


Figure 6: ***Occlusion of elbows***

6 CONCLUSION

We present research to measure the feasibility of operating humanoid robot control using human joint angles captured by a 2D camera. We implemented an interface of our approach that provides an output of head nod and tilting movement angles, open/closed hand status classifier, and human upper body joint angles based on human joint 3D coordinates captured via a 2D camera which is provided by existing libraries. Moreover, we tailored the output angles and applied our approach output to Nao and Pepper humanoid/semi-humanoid robots in a simulation. We evaluated

our approach on both ground truth angles and an experiment conducted with 16 people. We believe our work explicitly presents the feasibility of controlling humanoid robots via a 2D camera and contributes to showing the feasibility of controlling robots with the angles of human joints based on human landmark coordinates.

In future work, there are a number of improvements that could be easily implemented. The hips and legs module can be implemented based on our generic method that provides the human joint angle of abduction/adduction and flexion/extension human movements. The significant point here is to follow a different method between humanoid robots that have legs such as the Nao robot and those that do not have legs such as the Pepper robot. Especially, working on the walking movements of a humanoid robot using human leg joint angles would be interesting research. Other future works could be adding more settings to the interface, for example, there could be a record button on the interface so that users can easily record a video to use human joint angles taken from our approach in imitation learning of humanoid robots. A database having multiple categories of pre-recorded human joint angles from our approach could be generated for the purpose of Robot Learning by Demonstration. For example, a user wants a humanoid robot to learn how to stir ingredients while making a cake. This user can download a pre-recorded output of the human joint angles dataset on stirring a cake ingredients category and teach the robot with several different kinds of inputs rather than teaching this robot himself for hours.

7 REFLECTION

I encountered several challenges in this research. Firstly, the human joint angle calculation difference between abduction/adduction and flexion/extension human movements was one of the challenges that we needed to come up with a solution, which is our generic method for calculating angles of these different movement types. I ignored the external/internal rotation of human movements. If I started over again to the research, I would try to calculate the joint angle of ignored external/internal rotation, which is the yaw control of humanoid robots. Another point is that the user looks at the camera only from the front view. This brings some problems in imitating the elbow joint angle when the elbow falls behind the hand with respect to the camera. I would work more on that issue to solve.

Acknowledgments We appreciate all the support of Dr. Kim Baraka and Prof. Dr. Koen Hindrinks in this work.

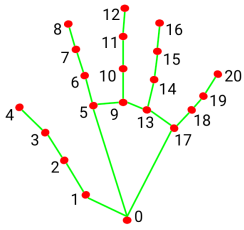
REFERENCES

[1] [n.d.]. CMU panoptic dataset. http://domedb.perception.cs.cmu.edu/index.html. Accessed: 2022-7-8.
[2] [n.d.]. Humanoid Robots. https://www.automate.org/a3-content/service-robots-humanoid-robots. Accessed: 2022-7-8.
[3] [n.d.]. Joint angle control versus trajectory control - MATLAB & Simulink. https://www.mathworks.com/help/supportpkg/robotmanipulator/ug/joint-angle-vs-trajectory-control.html. Accessed: 2022-7-8.
[4] [n.d.]. Joint control — NAO Software 1.14.5 documentation. http://doc.aldebaran.com/1-14/naoqi/motion/control-joint.html. Accessed: 2022-7-8.
[5] [n.d.]. SoftBank Robotics. https://www.softbankrobotics.com/emea/en. Accessed: 2022-7-8.
[6] [n.d.]. Solutions. https://google.github.io/mediapipe/solutions/solutions.html. Accessed: 2022-7-8.
[7] Ijaz Akhter and Michael J Black. 2015. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA, USA). IEEE.
[8] Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. 2019. BlazeFace: Sub-millisecond neural face detection on mobile GPUs. (July 2019). arXiv:1907.05047 [cs.CV]
[9] Marcel Bergerman and Yangsheng Xu. 1996. Robust joint and Cartesian control of underactuated manipulators. *J. Dyn. Syst. Meas. Control* 118, 3 (Sept. 1996), 557–565.
[10] Aude Billard and Daniel Grollman. 2012. Imitation Learning in Robots. In *Encyclopedia of the Sciences of Learning*. Springer US, Boston, MA, 1494–1496.
[11] Aude Billard and Daniel Grollman. 2013. Robot learning by demonstration. *Scholarpedia J.* 8, 12 (2013), 3824.
[12] Ernesto Brau and Hao Jiang. 2016. 3D human pose estimation via deep learning from 2D annotations. In *2016 Fourth International Conference on 3D Vision (3DV)* (Stanford, CA, USA). IEEE.
[13] Javier de Lope, Rafaela González-Careaga, Telmo Zarraonandia, and Darío Maravall. 2003. Inverse kinematics for humanoid robots using artificial neural networks. In *Computer Aided Systems Theory - EUROCAST 2003*. Springer Berlin Heidelberg, Berlin, Heidelberg, 448–459.
[14] Mahmoud El-Gohary and James McNames. 2015. Human joint angle estimation with inertial sensors and validation with A robot arm. *IEEE Trans. Biomed. Eng.* 62, 7 (July 2015), 1759–1767.
[15] Alp Guler, Nikolaos Kardaris, Siddhartha Chandra, Vassilis Pitsikalis, Christian Werner, Klaus Hauer, Costas Tzafestas, Petros Maragos, and Iasonas Kokkinos. 2016. Human joint angle estimation and gesture recognition for assistive robotic vision. In *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 415–431.
[16] Eddy Hudson, Garrett Warnell, Faraz Torabi, and Peter Stone. 2022. Skeletal feature compensation for imitation learning with embodiment mismatch. In *2022 International Conference on Robotics and Automation (ICRA)* (Philadelphia, PA, USA). IEEE.
[17] Antoni Jaume-i Capó, Javier Varona, Manuel González-Hidalgo, and Francisco J Perales. 2009. Adding image constraints to inverse kinematics for human motion capture. *EURASIP J. Adv. Signal Process.* 2010, 1 (Dec. 2009).
[18] Hao Jiang. 2010. 3D human pose reconstruction using millions of exemplars. In *2010 20th International Conference on Pattern Recognition* (Istanbul, Turkey). IEEE.
[19] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2015. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *2015 IEEE International Conference on Computer Vision (ICCV)* (Santiago, Chile). IEEE.
[20] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann. 2019. Real-time facial surface geometry from monocular video on mobile GPUs. (July 2019). arXiv:1907.06724 [cs.CV]
[21] Chunxu Li, Chenguang Yang, Peidong Liang, Angelo Cangelosi, and Jian Wan. 2016. Development of Kinect based teleoperation of Nao robot. In *2016 International Conference on Advanced Robotics and Mechatronics (ICARM)* (Macau, China). IEEE.
[22] Isabelle Poitras, Frédérique Dupuis, Mathieu Bielmann, Alexandre Campeau-Lecours, Catherine Mercier, Laurent Bouyer, and Jean-Sébastien Roy. 2019. Validity and reliability of wearable sensors for joint angle estimation: A systematic review. *Sensors (Basel)* 19, 7 (March 2019), 1555.
[23] Grégory Rogez, Jonathan Rihan, Carlos Orrite-Uruñuela, and Philip H S Torr. 2012. Fast human pose detection using randomized hierarchical cascades of rejectors. *Int. J. Comput. Vis.* 99, 1 (Aug. 2012), 25–52.
[24] Alessandro Roncone, Ugo Pattacini, Giorgio Metta, and Lorenzo Natale. 2016. A Cartesian 6-DoF gaze controller for humanoid robots. In *Robotics: Science and Systems XII*. Robotics: Science and Systems Foundation.
[25] Christopher Stanton, Anton Bogdanovych, and Edward Ratanasena. 2012. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In *Proc. Australasian Conference on Robotics and Automation*, Vol. 8. 51.
[26] Ye Yuan and Kris Kitani. 2018. 3D Ego-Pose Estimation via Imitation Learning. In *Computer Vision – ECCV 2018*. Springer International Publishing, Cham, 763–778.
[27] Ming Zhang, Jianxin Chen, Xin Wei, and Dezhou Zhang. 2018. Work chain-based inverse kinematics of robot to imitate human motion with Kinect. *ETRI J.* 40, 4 (Aug. 2018), 511–521.

A APPENDIX

Open source code[5]

B APPENDIX



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Figure 7: ***Human hand landmarks. Source: MediaPipe library***

C APPENDIX

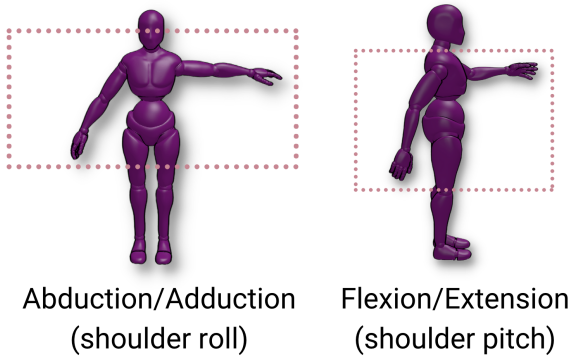
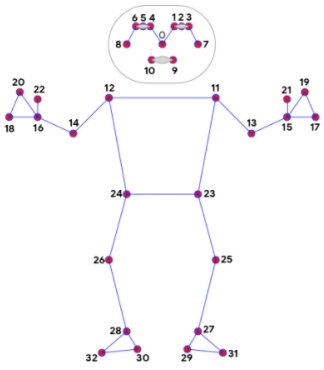


Figure 8: ***Human abduction/adduction (shoulder roll) and flexion/extension (shoulder pitch) movements***

D APPENDIX



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Figure 9: ***Body Pose Landmarks. Source: MediaPipe library***

[5] https://github.com/kucukbahadir/humanoidRobotControl

E APPENDIX

Figure 10: ***The interface of our approach***

F APPENDIX

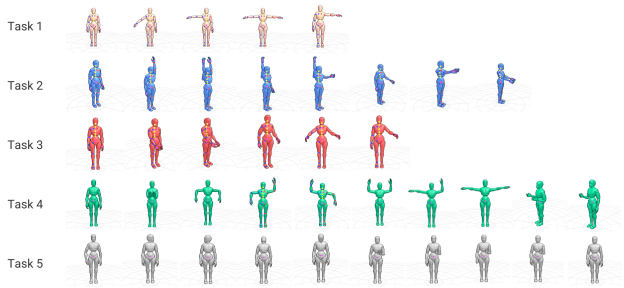


Figure 11: ***Part 1(tasks) of the experiment. Source: Human figures were sketched via*** [6]

G APPENDIX

			Abduction / Adduction	Flexion / Extension
Left Shoulder	Comparison 1	Root Mean Square Error	2.73	7.91
		Standart Deviation	1.33	5.6
	Comparison 2	Root Mean Square Error	8.38	12.88
		Standart Deviation	7.09	10.31
	Comparison 3	Root Mean Square Error	2.58	6.91
		Standart Deviation	1.43	4.42
	Comparison 4	Root Mean Square Error	4.49	10.97
		Standart Deviation	2.07	7.2
	Comparison 5	Root Mean Square Error	3.41	7.8
		Standart Deviation	2.1	5.19

Figure 12: ***Angle comparison results between our approach and ground truth on both abduction/adduction and flexion/extension movement.***

H APPENDIX

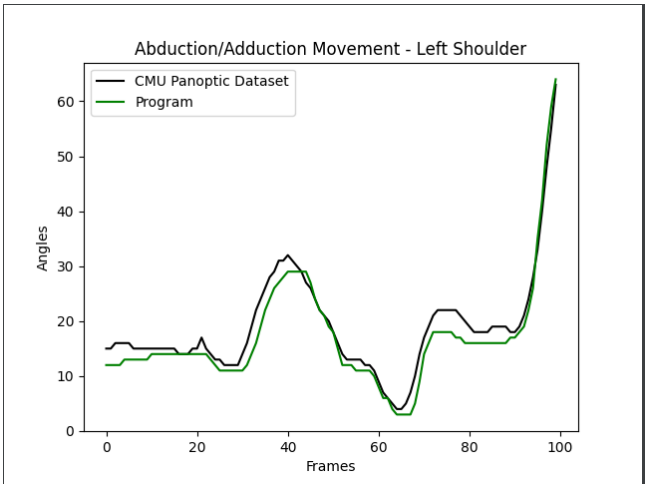


Figure 13: ***The angle comparison of left shoulder abduction/adduction movement. The root mean square error is 2.58.***

I APPENDIX

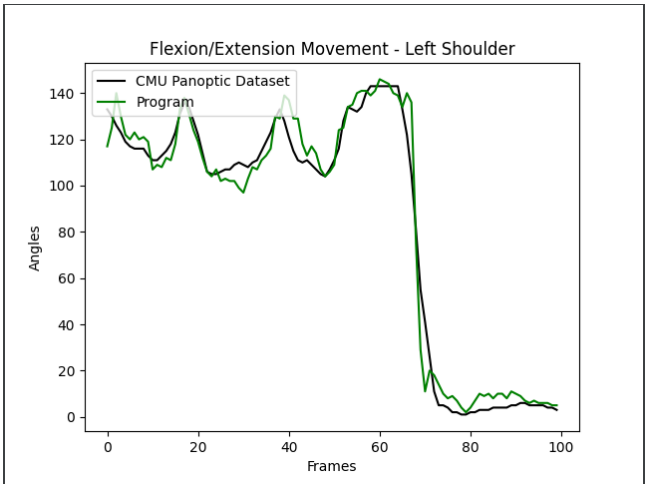


Figure 14: ***The angle comparison of the left shoulder flexion/extension movement. The root mean square error is 12.88.***

J APPENDIX

The ethics review committee of the Faculty of Science (BETHCIE), Vrije Universiteit Amsterdam. [7]

K APPENDIX

Experiment consent form of participants included in example experiment form [8].

[7] https://drive.google.com/file/d/1N6O6aUaU3GWBtA4RPJ7pNX9uNT56Kx7f/view?usp=sharing
[8] https://docs.google.com/forms/d/1WRDyR7ft-cn9N-ogd3dDiHAjMwRTEYkSJYG7QRhBf0A/copy