2021 30th IEEE International Conference on Robot and
Human Interactive Communication (RO-MAN)
August 8 - 12, 2021 - Vancouver, BC, CA (Virtual Conference)

Human Motion Retargeting to Pepper Humanoid Robot from Uncalibrated Videos Using Human Pose Estimation*

Hisham Khalil[1], Enrique Coronado[2], Gentiane Venture[2]

Abstract— **Human motion retargeting to humanoid robots (i.e., transferring motion data to robots for human imitation) is a challenging process with many potential real-work applications. However, current state-of-the-art frameworks present practical limitations, such as the requirement of camera calibration and the implementation of expensive equipment for motion capture. Therefore, we propose a novel framework for motion retargeting based on a single-view camera and human pose estimation. Unlike previous works, our framework is cost and computationally efficient, and it is applicable both on prerecorded uncalibrated videos and web-camera live streams. The framework is composed of three modules: 1) 2D coordinate extraction from the integrated Google BlazePose, 2) 3D modeling by depth estimation using a geometrical algorithm, and 3) human joint angles computation and input process to Pepper robot. Pepper's imitation accuracy is evaluated qualitatively by direct motion similarity observation and quantitatively by comparison between output and input motion data to observe the effect of Pepper's physical limitations. Results suggest that our proposed framework is able to reproduce human-like motion sequences, however with some limitations due to the hardware.**

I. Introduction

Nowadays, humanoid robots play an important role in numerous applications which involve social and assistive interactions with humans. Such applications include education [1], therapy [2], public talk interaction based on imitation of human body language [3], and teleoperation for human assistance [4]. Therefore, human-robot interaction (HRI) is a major research pursue for improving and increasing the usability of humanoid robots so that they become a concrete interacting factor in human society. For humanoid robots to be efficiently optimized in interaction with humans, these robots must have some human features. Such resemblance includes the ability of humanoid robots to mimic human motion effectively. This can be achieved when human motion is transferred to humanoid robots directly, whether based on a real-time process or a pre-defined motion sequence. This is defined as human *motion retargeting*. Generally, the key methodology for motion retargeting is human motion capture, which is the modeling and analysis of human motion sequences based on motion data. Captured data is then transferred to the robot for human motion imitation.

Motion capture technologies that are widely in use are either marker-based (e.g., motion capture suits based on attached reflective markers) or marker-less (e.g., solutions where the human subject's motion is captured by a camera or a depth sensor). There are other types of marker-less technologies, such as wearable inertial measurement unit (IMU) motion capture systems. Integrating these various technologies in robotics research has been a major challenge considering their limitations [5]. Therefore, novel alternatives based on deep neural networks, which are explored in this paper, have been proposed to assist in the development of human pose estimation frameworks processed on images and videos without requiring physical measurement tools.

In this paper, our aim is to achieve upper body motion retargeting to humanoid robots based on a novel real-time human pose estimation library processed on uncalibrated single-view videos instead of using complex and costly marker-based or sensor-based solutions. Therefore, we validate the suitability of Google MediaPipe Pose (BlazePose) [6], [7] in a motion retargeting scenario. Our proposed framework is composed of three software modules or processes: 1) extraction of 2D body landmark coordinates from the integrated BlazePose, 2) depth estimation for 3D modeling using a geometrical algorithm, and 3) human joint angles computation by linear algebra and trigonometry and the data input process to the robot. Additionally, the joint angles output are extracted from the real robot's sensors for comparison purpose. The experiments were conducted on the Pepper robot [8], as it is an open-source and popular social humanoid robot that has human-like motion capabilities. Experimenting on such humanoid robot assisted to highlight the similarity of its imitated motion compared to the captured human motion.

The rest of the paper is organized as follows. Section II explains some related work in literature and a summary of our contribution. Section III describes implemented methods and algorithms. Section IV describes the experimental procedures conducted and the tools used in the experiment. Section V shows image results of the imitated motion, an evaluation of the output results from the real Pepper robot, and a discussion on the main points observed. Section VI states the conclusion of the research and the future improvements and work to be considered.

II. Related Work

A. Video-based Marker-less Motion Capture Technology

There has been a wide development of marker-less solutions for motion analysis that have been useful tools in research [9]. For instance, Microsoft Kinect is a depth

*This work was partially supported by the Japanese Society for the Promotion of Science Grant in Aid for Scientific Research (C) JP19K12167

[1] H. Khalil is with Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia mahisham@graduate.utm.my

[2] E. Coronado and G. Venture are with Tokyo University of Agriculture and Technology, Tokyo, Japan. venture@cc.tuat.ac.jp

978-1-6654-0492-1/21/$31.00 ©2021 IEEE

camera that can be used for kinematic analysis in a non-laboratory environment, unlike marker-based motion capture suits, which require extensive setup procedures and expensive equipment for data collection confined to a laboratory workspace [10]. Even though Kinect is a portable and cost-efficient alternative to motion capture suits [11], it is mostly applicable only in indoor environments due to its vulnerability to direct sunlight [12].

OpenPose [13] is a deep learning-based alternative framework for multi-person 2D human pose estimation. Researchers have used it to develop marker-less 3D human motion capture solutions [14]. However, OpenPose is often considered computationally inefficient [15] and requires high-performance computational resources, such as high-end graphical processing units (GPUs), to enable real-time human body pose detection. Moreover, using single camera systems, the OpenPose model can only extract 2D kinematic data from humans. OpenPose also provides a 3D keypoint reconstruction module [16] but with the use of a multiple-camera system and the calibration of each camera. This makes the use of pre-recorded videos infeasible where camera parameters are unknown. Therefore, there have been several pose estimation solutions developed as an alternative to OpenPose to propose a more efficient framework [15], [17]. A very recent alternative released in 2020 is BlazePose, which has a faster performance rate on low powerful processing units compared to OpenPose [7]. However, BlazePose can only detect a single person in the scene. To the best of the authors knowledge, BlazePose have been not used and evaluated before for enabling motion retargeting to humanoid robots. Therefore, achieving this is a main objective in our work.

B. 3D Human Pose Reconstruction

3D pose reconstruction of the human body from marker-less motion capture data is a novel approach to propose efficient solutions to improvise on the existing 2D pose estimation frameworks. Reconstruction of the human pose from uncalibrated videos for producing animation sequences was proposed in [18]. Similarly, Yiannakides et al. [19] used a monocular RGB camera to develop a method for 3D pose and motion reconstruction. Wang et al. [20], on the other hand, proposed a generalized 3D model complying with the modeled human subject while considering the target robot configurations of stability and joints without predefined joint mapping. The authors used Microsoft Kinect on the human subject to capture the RGBD sequences used in their human-robot (*HUMROB*) model. However, our framework is integrating pose estimation as most 3D reconstruction algorithms require a complex process for implementation. We used a derived method similar to the most suitable one in [18], although we experimented it on motion retargeting instead of animations.

C. Motion Optimization

In motion retargeting research, some methods were proposed that aim to optimize the motion of robots, either by relaxing constraints to narrow the gap of motion limitations between humans and robots or by preserving human motion features. One method of optimizing motion is constrained optimization, which was used to minimize pose and end-effector errors [21] and generate human-like continuous motion [22]. Geometric parameter identification, motion planning, and inverse kinematics (IK) solution were introduced in [23] based on quantitative analysis to imitate accurate motion while adapting with the variety of human subject body features. Kaplish and Yamane [24] focused on teleoperation based on physical human-robot interaction (pHRI) through a sensor-based controller that detects the contact forces and implements an optimization technique to minimize the discrepancy between the contact states of the robot and the human operator. The above-mentioned solutions implemented sensor-based technologies for kinematic or dynamic data extraction. This differs from our proposed framework which is based on an unwearable solution, and these optimization algorithms have not been tested on motion data from pose estimation libraries.

D. Inverse Kinematics

IK is commonly used in robot and human motion modeling by extracting joint angles from the end-effector position and orientation. Rapetti et al. [25] developed motion tracking algorithms based on dynamical IK and tested their performance through a human-robot motion retargeting scenario using Xsens motion capture suit. Darvish et al. [26] proposed a generalized framework for teleoperation of several humanoid robots using whole-body controllers. They implemented human subject measures as inputs to an IK model integrated with an optimization scheme to obtain the suitable robot input parameters for achieving whole-body motion retargeting. Even though it is an effective method for motion analysis, IK was not suitable for our approach as it requires the 3D orientation of the end-effector, which is difficult to extract from 2D pose estimation.

E. Training Robots to Imitate Humans

Various machine learning algorithms were developed to train robots to emulate human motion sequences. Generative Adversarial Networks (GAN) trained from motion capture data were implemented in [27] using Kinect and in [28] using a calibrated RGB camera. Martin and Moutarde [29] designed a controller that makes a robotic manipulator mimic a human subject through the implementation of Human Mesh Recovery (HMR), which is a trained model that estimates a 3D mesh model and camera configuration of a human from a single-view image. However, they concluded that HMR was not precise in detecting joint positions in their approach. Hwang and Liao [30] used stable movement classification by support vector machine in motion imitation by a humanoid robot. However, these methods require pre-training of data for implementation, which can reduce time efficiency for our approach.

F. Contribution

In this research, our contribution is summarized as follows:

1) Integration and evaluation of BlazePose for enabling motion retargeting to humanoid robots as it was not implemented in robotics research before.
2) Development of a cost and computationally efficient motion retargeting framework that is applicable, unlike previous works, in indoor, outdoor, and non-laboratory environments and operational on lower-end devices other than computers (e.g., mobile devices) [7].
3) Experimental validation of the proposed framework on humanoid robots using pre-recorded videos without the requirement of real-time physical presence of the human subject.

III. Methodology

A. 2D Coordinates of Body Landmarks

BlazePose is composed of a model of 33 landmarks, representing the whole body of a human including the facial landmarks. In the proposed algorithm, only the upper body and nose landmarks are used as highlighted in Fig. 1 since Pepper is not a bipedal robot.

The output of BlazePose landmark coordinates is composed of x and y coordinates normalized to a ratio of 0 to 1 multiplied by the image width and height respectively [31], which form the video resolution in pixels. The z coordinate relative to the camera frame is recently developed in BlazePose GHUM 3D model [31], but it was only released at the time of writing this paper. Therefore, an effective algorithm is utilized for estimating the depth coordinate of the detected landmarks instead.

B. Depth Estimation

To estimate the depth in pixels between two (x, y) coordinates for two landmarks, certain features from an initial pose template of the human in the video must be extracted first to be used as a reference. The input video of human motion must start with this initial pose, where the human is at resting position with all the arms stretched, torso straight, and the face looking directly to the camera as shown from the human initial pose in the image in Fig. 2. As the real arm segment lengths of the human in the video are unknown, the original length of each segment with reference to BlazePose coordinate system is obtained. Each original arm segment's length, l, is used in this equation:

$$dz = \sqrt{l^2 - ((x_1 - x_2)^2 + (y_1 - y_2)^2)}, \qquad (1)$$

which is derived from the equation presented in [32] and validated through related work [18]. dz in (1) represents the pixel depth between (x_1, y_1) and (x_2, y_2), which are two coordinates for two different landmarks. The scaling factor was set to $s = 1$ as a scaled orthographic projection is not assumed in this case. There is only one plane of reference, which is the frontal plane of human motion. When the arm segments move, their lengths change according to the orthographic projection of the person facing the camera as can be shown in the example of Fig. 3a.

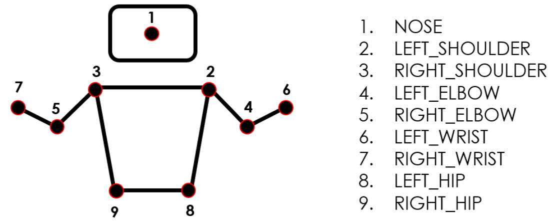


Fig. 1: Extracted Landmarks used in our algorithm from BlazePose landmark library [31].

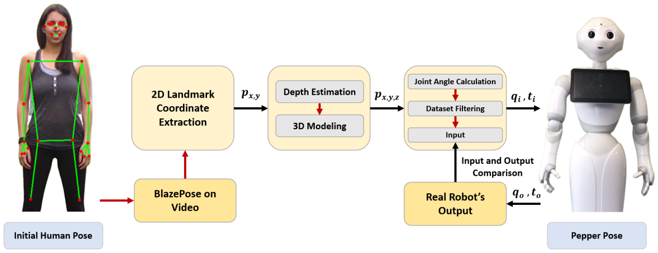


Fig. 2: The architecture of our proposed framework for motion retargeting to Pepper. $p_{x,y}$ represents the 2D positions of the body landmarks from BlazePose, $p_{x,y,z}$ represents the 3D coordinates including the depth parameter calculated. q_i and t_i are the input joint angles and motion time frame vectors respectively, while q_o and t_o are the output ones.

C. Calculation of Joint Angles

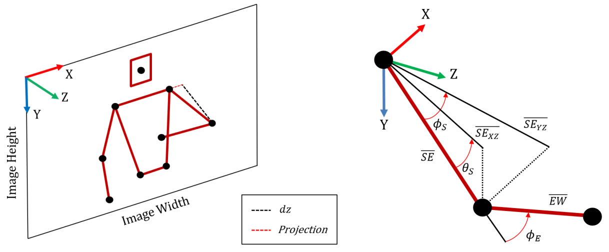
The notations *S*, *E*, *H*, and *HP* represent the shoulder, elbow, head, and hip respectively. *SE* stands for shoulder to elbow, or upper arm, and *EW* stands for elbow to wrist, or lower arm. The roll, pitch, and yaw angles are represented by ϕ, θ, and ψ respectively. These notations are used in the following equations and symbols for body segment and joint angle representations. The angles are all in radians (rad).

1) Arms: There are four degrees of freedom (DoFs) for each arm of Pepper: shoulder roll, shoulder pitch, elbow roll, and elbow yaw. In order to model the motion, a vector model of the human arm with three of those angles is represented (see Fig. 3b). The fourth angle, elbow yaw, cannot be calculated as the hand orientation in 3D space from BlazePose is unknown. Instead, its value is estimated based on the predicted hand pose at the moment the human performs a specific movement.

The upper arm vector illustrated in Fig. 3b, $\overline{SE}$, and its projection on the *YZ* plane are the two vectors that the angle between them will be the shoulder roll, ϕ_S. Upper arm *YZ* projection is represented by:

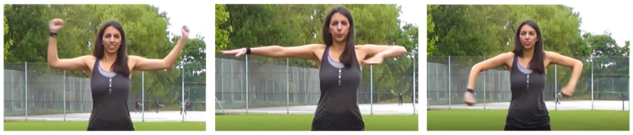
$$\overline{SE_{YZ}} = \begin{bmatrix} 0 & y_{SE} & z_{SE} \end{bmatrix}^T, \qquad (2)$$

while the shoulder roll is calculated using (2) with this



(a) (b)

Fig. 3: (a) A representation example of the projection of the human model by BlazePose, while the human is moving the left arm; (b) A model representing the shoulder roll (ϕ_S), shoulder pitch (θ_S), and elbow roll (ϕ_E) of the human arm in 3D space to be corresponded with Pepper joint angles.



(a) Supination (b) Neutral (c) Pronation

Fig. 4: The three hand pose conditions which correspond to (a) $\pm\frac{\pi}{2}$, (b) 0, and (c) $\pm\frac{\pi}{2}$ rad for the elbow yaw angle.

equation:

$$\phi_S = \arccos \frac{\overline{SE} \cdot \overline{SE_{YZ}}}{\|\overline{SE}\| \cdot \|\overline{SE_{YZ}}\|}. \tag{3}$$

Upper arm *XZ* projection is modeled as:

$$\overline{SE_{XZ}} = \begin{bmatrix} x_{SE} & 0 & z_{SE} \end{bmatrix}^T. \tag{4}$$

Equation (4) is then used with $\overline{SE}$ to compute the shoulder pitch, θ_S, by:

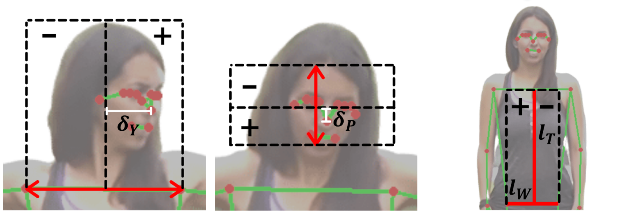
$$\theta_S = \arccos \frac{\overline{SE} \cdot \overline{SE_{XZ}}}{\|\overline{SE}\| \cdot \|\overline{SE_{XZ}}\|}. \tag{5}$$

The shoulder pitch range is assumed to be $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ rad instead of the indicated range in Pepper's joints documentation [33] as it is not possible to predict the arm's movement backward in the negative *Z* sagittal plane. $\overline{SE}$ and the lower arm vector, $\overline{EW}$, are used to calculate the elbow roll, ϕ_S, using:

$$\phi_E = \arccos \frac{\overline{SE} \cdot \overline{EW}}{\|\overline{SE}\| \cdot \|\overline{EW}\|}. \tag{6}$$

Equations (3), (5), and (6) are applicable for both left and right arms' joint angle calculations, with consideration of the respective sign conventions with reference to Pepper's motion coordinate frame [33].

The elbow yaw angles, ψ_E, are only assigned as either $\frac{\pi}{2}$, 0, or $-\frac{\pi}{2}$ rad depending on the sign conventions of Pepper coordinate frame. As in Fig. 4, supination, pronation, and neutral are the three types of poses that we refer to for assigning ψ_E. For right elbow yaw, supination matches $\psi_E = \frac{\pi}{2}$ rad when the right wrist *y* coordinate is higher in position, or less in magnitude referring to the coordinate frame in Fig. 3a, than the right elbow *y* coordinate by a relatively significant difference as in Fig. 4a. Pronation, on the other hand, matches $\psi_E = -\frac{\pi}{2}$ rad when the right wrist and right elbow relative *y* coordinate condition is vice-versa, where the right wrist is lower than the right elbow (see Fig. 4c). The same goes for right elbow yaw, however with supination matching $\psi_E = -\frac{\pi}{2}$ rad and pronation matching $\psi_E = \frac{\pi}{2}$ rad. The neutral pose in Fig. 4b, matches 0 rad in both right and left elbow yaw angles when the wrist and elbow are nearly on the same horizontal line with almost matching *y* coordinates.



(a) Head Yaw (b) Head Pitch (c) Hip Roll and Hip Pitch

Fig. 5: Head and hip motion representations. The sign conventions (+ or -) show the required sign of the estimated respective angles. δ_Y and δ_P are the head displacements proportional to head yaw and pitch angles respectively. l_T is the torso height, and l_W is the waist width.

2) Head: The head's joint angles to be estimated are the head yaw and pitch. For head yaw, θ_{HY}, even though it has a range of -2.0857 to 2.0857 rad, we assume that its range of motion is $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ rad so that human-like head movements are generated as humans usually do not have such extreme range of motion. We assume the following relation:

$$\psi_H \propto \delta_Y, \tag{7}$$

in which the *x* position of the nose landmark with respect to the initial pose reference template defines its left or right horizontal displacement, δ_Y in (7), with respect to the shoulders' *x* coordinates as can be seen in Fig. 5a. Similarly, the head pitch, θ_H, is represented by:

$$\theta_H \propto \delta_P, \tag{8}$$

where δ_P in (8) is the vertical up or down displacement of the nose's *y* coordinate with respect to the shoulders' *y* coordinates (see Fig. 5b). θ_H range of motion, -0.7068 to 0.6371 rad, is smaller than that of ψ_H, which is the usual case for human head pitch movement as well. The sign conventions of ψ_H and θ_H are represented by the signs of δ_Y in Fig. 5a and δ_P in Fig. 5b respectively.

3) Hip: The hip joint angles are the hip roll, ϕ_{HP}, and hip pitch, θ_{HP}. ϕ_{HP} is represented by the tilt angle of the waist width segment, l_W, shown in Fig. 5c with sign conventions and a range of motion of -0.5149 to 0.5149 rad. It is calculated by the following equation:

$$\phi_{HP} = \arccos \frac{l_W{}'}{l_W}, \tag{9}$$

TABLE I: Pepper joint angles with a description of their respective motion, the axis which each rotates around, their filtered range of motion with reference to Pepper joints documentation [33], and the maximum angular velocities achieved by the joint actuators obtained from Pepper's system log after experimentation on the real robot. X', Y', or Z' represents the new frame axis that the rotation is around when Pepper moves and changes its initial position.

Joint Angle	**Motion**	**Rotation Axis**	**Range (rad)**	**Max. Angular Velocity (rad/s)**
LShoulderPitch	Left shoulder joint front and back	X	-1.5708 to 1.5708	7.340
LShoulderRoll	Left shoulder joint right and left	Y	0.0087 to 1.5620	9.228
LElbowRoll	Left elbow joint flexion and extension	Y'	-1.5620 to -0.0087	9.228
LElbowYaw	Left shoulder joint twist	Z'	-1.5708 to 1.5708	7.340
RShoulderPitch	Right shoulder joint front and back	X	-1.5708 to 1.5708	7.340
RShoulderRoll	Right shoulder joint right and left	Y	0.0087 to 1.5620	9.228
RElbowRoll	Right elbow joint flexion and extension	Y'	-1.5620 to -0.0087	9.228
RElbowYaw	Right shoulder joint twist	Z'	-1.5708 to 1.5708	7.340
HeadYaw	Head joint twist	Y	-1.5708 to 1.5708	7.340
HeadPitch	Head joint front and back	X	-0.7068 to 0.6371	9.228
HipPitch	Hip joint front and back	X	-1.0385 to 1.0385	2.933
HipRoll	Hip joint right and left	Z	-0.5149 to 0.5149	2.270

where ${l_W}'$ in (9) is the changed l_W during motion. θ_{HP} in our case is assumed always to be in positive direction only, or leaning forward following the sign conventions of Pepper since we cannot predict the direction of hip pitch movement. In addition, humans usually do not tend to lean backward while moving naturally. Therefore, θ_{HP} range is 0 to 1.0385 rad. θ_{HP} is then represented by the tilt angle of the torso height segment, l_T, or the segment representing the distance between the shoulders and the hip, in the following equation:

$$\theta_{HP} = \arccos \frac{{l_T}'}{l_T}, \tag{10}$$

in which ${l_T}'$ in (10) is the changed l_T.

D. Robot Input

The sets of calculated joint angles are named as in Table I, with each of them represented as:

$$\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T. \tag{11}$$

Since the length of each $\boldsymbol{q}$ vector is large, each is scaled by a set scaling factor, which is determined by trial and error applicable for any input video. The input data size to the robot is reduced to avoid any errors while executing the motion retargeting process. Such errors include commanding kinematic data that cause the input joint angular velocity to Pepper to exceed the maximum angular velocity that the robot can achieve by its actuators (see Table I). This can occur if there is a small change in time between two commanded joint angle inputs. Furthermore, the angles are filtered based on their range of motion shown in Table I. To correspond the time frame of captured motion to the robot motion one, a time series vector of the same length as $\boldsymbol{q}$, expressed in seconds, is represented as:

$$\boldsymbol{t} = \begin{bmatrix} t_1 & t_2 & \dots & t_n \end{bmatrix}^T, \tag{12}$$

where n in (11) and (12) is the scaled number of extracted frames from BlazePose since the $\boldsymbol{t}$ vector is also scaled by the same scaling factor as $\boldsymbol{q}$. All joint angle vectors and their corresponding time series ones are combined into a single data set, which is the input to the robot as can be summarized in Fig. 2.

IV. Experiment

We implemented BlazePose Python Application Programming Interface (API) running on a PC with Intel Core i7-8550U central processing unit (CPU) @1.80 GHz, NVIDIA GeForce MX150 GPU, and 16 GB of random access memory (RAM). The model is operating with an output of more than 20 frames per second (FPS), and it is using TensorFlow Lite integrated with XNNPACK library.

For the input to BlazePose, we utilized a suitable video from YouTube[1] as the upper body is fully appearing in the scene and the human subject is front facing the camera without rotation. The projection of the detected segment annotations is accurate, and occlusions are minimal. This allows us to use our proposed algorithm for estimating depth coordinates since the human subject starts the motion with the initial pose that is shown in Fig. 2. The human was performing multiple variations of arm movements and head twists, either slow and steady or fast and complex, and this helped us to experiment and evaluate a variety of generated robot movements.

In order to retarget the generated data set of joint angles and corresponding time series, we used the method *angleInterpolation* from the Joint Control API from NAOqi 2.5 Python Software Development Kit (SDK) [33]. The *angleInterpolation* method receives the joint angles and the corresponding time series vectors as inputs in order to command the robot to move in a set time frame. We connected the robot with Choregraphe 2.5 [34], which also includes a real-time simulation of Pepper's motion. For extracting the actual joint angles from the sensor output of the real robot,

[1] We used this video of a person performing an exercise session: `https://youtu.be/MBh14pJ6MU0`

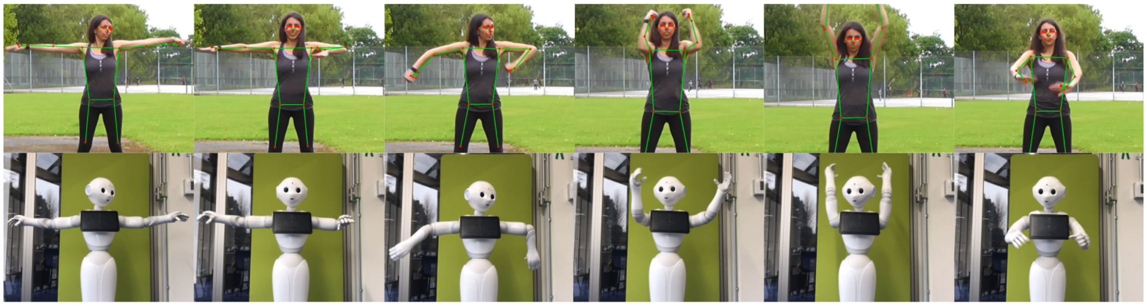


Fig. 6: Pepper mimics different movements from the exercise video.

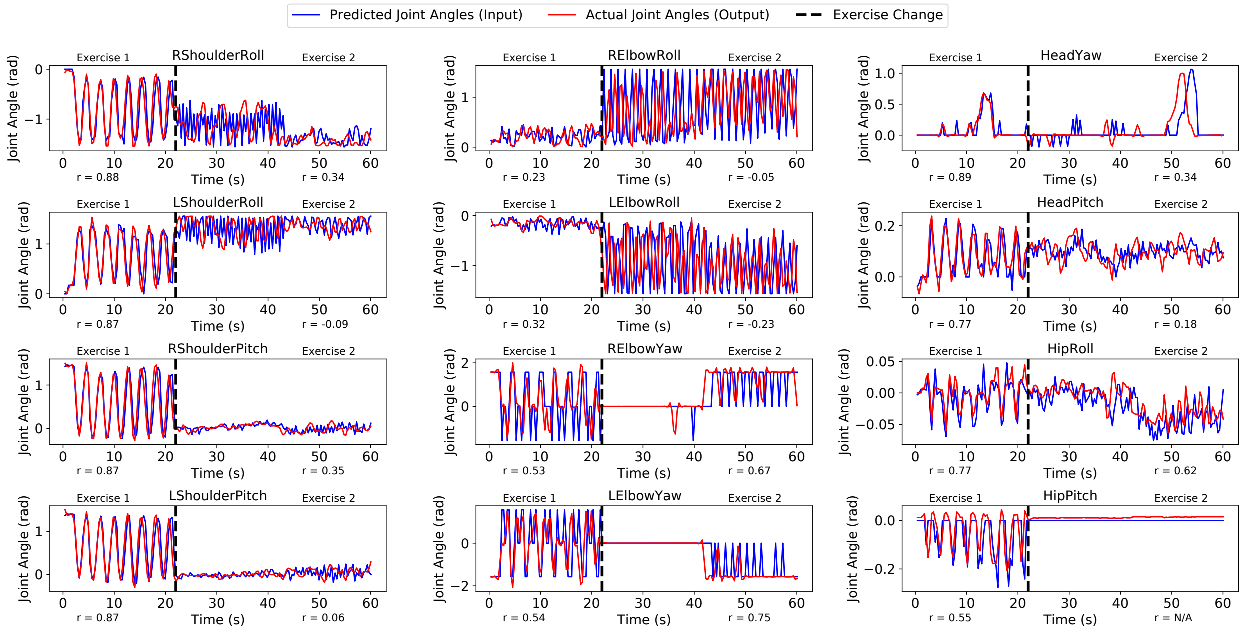


Fig. 7: The output joint angle data (red) from the sensors of Pepper compared with the filtered human motion data extracted from BlazePose (blue), which is the input to the robot controller for two types of exercises: Exercise 1 (slow movement) and Exercise 2 (fast movement). r represents the correlation coefficient of the two signals.

we integrated the Node Primitives (NEP) framework [35] in which we created a subscriber based on the *getAngles* method in the Joint Control API. The subscriber operates in real-time while the *angleInterpolation* method (the publisher) is processing and the robot is moving. A new output data set composed of joint angles and the executed time series of motion is generated, and we compared it with our input data set.

V. Results & Discussion

The result analysis applies sample data of 60 seconds of the original video for enhanced visualization. The first 22 seconds (Exercise 1) contain mainly slow and steady arm movements, and the remaining 38 seconds (Exercise 2) are faster and more complex exercise movements. Various head and hip movements are embedded in all exercises.

For performance evaluation of the whole proposed framework, we only consider evaluating the second and third framework components, as the first component's evaluation of BlazePose 2D landmark detection is performed in [7] and indicates that the 2D pose estimation is accurate.

Our major framework evaluation, namely of the third framework component, of the performed motion by Pepper is divided into two parts: 1) qualitative analysis based on direct observation of the motion sequences and 2) quantitative analysis based on the comparison between the input joint angle data to the robot's system and the output joint angle data from the real robot's sensors using the experimental setup described in Section IV.

We can conclude from the observed motion that Pepper is able to mimic the movements of the arms, head, and hips in correspondence to the performed motion in the video (see Fig. 6). The robot can correctly imitate the arm orientations of the flexion and extension as well as the abduction and adduction movements, and approximately imitate the head yaw and pitch and hip roll and pitch movements in different movement scenarios. However, the minor noisy or inaccurate detections and occlusions of the body landmarks

TABLE II: The mean absolute error between the predicted and actual joint angles of the robot shown in Fig. 7 during Exercise 1 and Exercise 2.

Joint Angle	Mean Absolute Error (rad)	
	Exercise 1	Exercise 2
LShoulderPitch	0.21	0.08
LShoulderRoll	0.18	0.23
LElbowRoll	0.09	0.64
LElbowYaw	0.80	0.23
RShoulderPitch	0.22	0.07
RShoulderRoll	0.19	0.24
RElbowRoll	0.12	0.64
RElbowYaw	0.80	0.33
HeadYaw	0.04	0.13
HeadPitch	0.04	0.04
HipPitch	0.06	0.01
HipRoll	0.01	0.02

in BlazePose, especially during fast movements, affect the accuracy of the calculated joint angles, specifically for arm movements. Thus, there are some inaccuracies in performing fast and complex movements by Pepper. For example, the supination and pronation movements, since the elbow yaw for both arms is an estimated value, are correctly performed but only in naturally performed steady movements, where the human does not twist the arms complexly.

During Exercise 1, the *RShoulderRoll* and *LShoulderRoll* subplots in Fig. 7 show high correlation between the predicted and actual movements (see Fig. 7), resulting in less error than in Exercise 2 (see Table II). It is also shown that in Exercise 2, the shoulder roll and elbow roll subplots display the actual robot movement (red) to be of lower frequency between 23 and 60 seconds (i.e., the robot is not able to maintain the same rapid change in joint angular motion in the common time frame as was commanded to it) compared to the movement in Exercise 1. The larger error in Exercise 2 for *RElbowRoll* and *LElbowRoll* in Table II also supports this hypothesis. In the indicated joint angles of arms, a lower frequency of motion is observed due to the friction and physical joint and actuator limitations of Pepper. The robot's limitations are also the cause of errors in various movements for both exercises in all other measured joint angles (see Table II). The discussed results can be validated by experimenting with the framework's open-source repository[2] on the real Pepper robot. Moreover, a video[3] of imitation by Pepper is uploaded for further support of the framework's performance evaluation.

For evaluation of the second framework component, BlazePose GHUM 3D model's depth component (z coordinate) is compared with the depth value time series obtained by the algorithm in Section III-B, based on our model's reference frame. The results are shown in Fig. 8. The GHUM 3D model developed by Google Research, which the novel BlazePose model is based on, is presented and evaluated in [36]. It is observed that the elbows' and wrists' depth changes in BlazePose GHUM 3D model's data are larger, specifically in Exercise 1, compared to our calculated depth values. However, Exercise 1 does not show such large changes in elbow and wrist depth movements when observing the original video. Moreover, our depth estimation results were observed to produce correct imitations by the robot as was observed in the third framework component evaluation. The discrepancy between the two methods' results can be the cause of GHUM 3D model's baseline of obtaining depth values by synthetic data fitting to the 2D pose annotation [36], which can lead to value prediction inaccuracy. Currently, BlazePose team is working on improving the accuracy of z coordinate estimation in BlazePose GHUM 3D model [31].

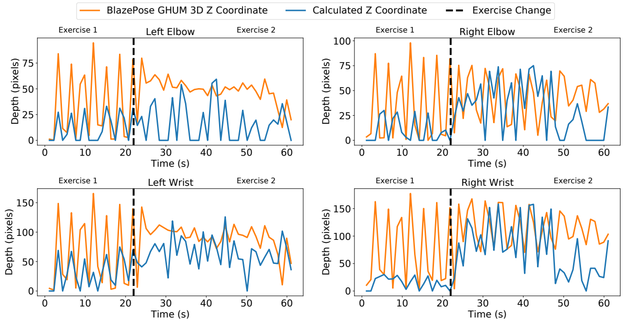


Fig. 8: Comparison between our calculated depth (z) coordinate of the elbows and wrists (blue) and the extracted values from BlazePose GHUM 3D model (orange) for Exercise 1 and Exercise 2.

VI. Conclusion & Future Work

In this paper, we presented a human-to-humanoid robot motion retargeting framework using BlazePose model on uncalibrated videos of human motion. This proposed novel work leads to a further step in human-robot interaction (HRI) research, where pose estimation solutions can be implemented to tackle the limitations of sensor-based and marker-based motion capture technologies. Efficient motion retargeting solutions help advance more real-life applications in which humanoid robots are more engaged with humans. Because of its computational efficiency, our framework can be integrated with robotic vision systems. Moreover, our framework is open-source and can assist people in education, research, and other fields to expand human motion interactive applications using simple video recordings. However, it is optimal for use only for humanoid robots with a close number of DoFs to those of humans, and its dependency on the initial pose input template constrains its functionality to single-view videos with a specific orientation. Moreover, in BlazePose, direct osculation on the detected human body by other people or other objects in the scene sometimes causes inaccurate pose detection, which limits our framework's performance in such scenarios.

Further improvements based on motion optimization can

[2] The source code of our framework on GitHub: https://github.com/GVLabRobotics/pepper-blazepose

[3] Pepper in Choregraphe simulation imitates gestures from another random video: https://youtu.be/BxJbxjFeQko

be introduced in future research in order to eliminate noise in motion and to make it more human-like. The novel BlazePose GHUM 3D model can be tested further for motion retargeting after its improvement and compared with our approach as well as other 3D motion capture solutions. Newer pose estimation solutions that can detect the position and orientation of both hands and arms, such as MediaPipe Holistic [37], can be tested to apply IK for computing joint angles. Moreover, we will work to implement online real-time video teleoperation of humanoid robots using the experimented NEP framework [35] integrated with our model.

Acknowledgment

The authors would like to thank Raquel, a qualified Zumba and fitness instructor based in Liverpool, UK for allowing us to use her workout YouTube video.

References

[1] M.-L. Bourguet, Y. Jin, Y. Shi, Y. Chen, L. Rincon-Ardila, and G. Venture, "Social Robots that can Sense and Improve Student Engagement," in *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Takamatsu, Japan, 2020, pp. 127–134.

[2] I. Fujimoto, T. Matsumoto, P. R. S. De Silva, M. Kobayashi, and M. Higashi, "Mimicking and Evaluating Human Motion to Improve the Imitation Skill of Children with Autism Through a Robot," *International Journal of Social Robotics*, vol. 3, no. 4, pp. 349–357, 2011.

[3] M.-L. Bourguet, M. Xu, S. Zhang, J. Urakami, and G. Venture, "The Impact of a Social Robot Public Speaker on Audience Attention," in *Proceedings of the 8th International Conference on Human-Agent Interaction*, Virtual Event USA, 2020, pp. 60–68.

[4] M. A. Goodrich, J. W. Crandal, and E. Barakova, "Teleoperation and Beyond for Assistive Humanoid Robots," *Reviews of Human Factors and Ergonomics*, vol. 9, no. 1, pp. 175–226, 2013.

[5] M. Field, D. Stirling, F. Naghdy, and Z. Pan, "Motion capture in robotics review," in *2009 IEEE International Conference on Control and Automation*, Christchurch, New Zealand, 2009, pp. 1697–1702.

[6] C. Lugaresi *et al.*, "MediaPipe: A Framework for Building Perception Pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[7] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose tracking," *arXiv preprint arXiv:2006.10204*, 2020.

[8] A. K. Pandey and R. Gelin, "A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind," *IEEE Robotics & Automation Magazine*, vol. 25, no. 3, pp. 40–48, 2018.

[9] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall, "A Survey on Human Motion Analysis from Depth Data," in *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, Berlin, Heidelberg, 2013, pp. 149–187.

[10] L. G. Wiedemann, R. Planinc, I. Nemec, and M. Kampel, "Performance evaluation of joint angles obtained by the Kinect V2," in *IET International Conference on Technologies for Active and Assisted Living (TechAAL)*, London, UK, 2015, pp. 1–6.

[11] T. Dutta, "Evaluation of the Kinect™ sensor for 3-D kinematic measurement in the workplace," *Applied Ergonomics*, vol. 43, no. 4, pp. 645–649, 2012.

[12] M. Tölgyessy, M. Dekan, L. Chovanec, and P. Hubinskỳ, "Evaluation of the Azure Kinect and Its Comparison to Kinect V1 and Kinect V2," *Sensors*, vol. 21, no. 2, p. 413, 2021.

[13] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, 2019.

[14] N. Nakano *et al.*, "Evaluation of 3D markerless motion capture accuracy using OpenPose with multiple video cameras," *Frontiers in Sports and Active Living*, vol. 2, 2020.

[15] D. Groos, H. Ramampiaro, and E. A. Ihlen, "EfficientPose: Scalable single-person pose estimation," *Applied Intelligence*, pp. 1–16, 2020.

[16] G. Hidalgo, "OpenPose," Accessed: Mar. 29, 2021. [Online]. Available: https://github.com/CMU-Perceptual-Computing-Lab/openpose

[17] L. J. Silva, D. L. S. da Silva, A. B. Raposo, L. Velho, and H. C. V. Lopes, "TensorPose: Real-time pose estimation for interactive applications," *Computers Graphics*, vol. 85, no. 1, pp. 1–14, 2019.

[18] U. Güdükbay, I. Demir, and Y. Dedeoğlu, "Motion capture and human pose reconstruction from a single-view video sequence," *Digital Signal Processing*, vol. 23, no. 5, pp. 1441–1450, 2013.

[19] A. Yiannakides, A. Aristidou, and Y. Chrysanthou, "Real-time 3D human pose and motion reconstruction from monocular RGB videos," *Computer Animation and Virtual Worlds*, vol. 30, no. 9, 2019.

[20] S. Wang, X. Zuo, R. Wang, and R. Yang, "A Generative Human-Robot Motion Retargeting Approach Using a Single RGBD Sensor," *IEEE Access*, vol. 7, pp. 51 499–51 512, 2019.

[21] T. Tosun, R. Mead, and R. Stengel, "A General Method for Kinematic Retargeting: Adapting Poses Between Humans and Robots," in *ASME 2014 International Mechanical Engineering Congress and Exposition*, Montreal, Quebec, Canada, 2014.

[22] M. J. Gielniak, C. K. Liu, and A. L. Thomaz, "Generating human-like motion for robots," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1275–1301, 2013.

[23] K. Ayusawa and E. Yoshida, "Motion Retargeting for Humanoid Robots Based on Simultaneous Morphing Parameter Identification and Motion Optimization," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1343–1357, 2017.

[24] A. Kaplish and K. Yamane, "Motion Retargeting and Control for Teleoperated Physical Human-Robot Interaction," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, Toronto, ON, Canada, 2019, pp. 723–730.

[25] L. Rapetti *et al.*, "Model-Based Real-Time Motion Tracking Using Dynamical Inverse Kinematics," *Algorithms*, vol. 13, no. 10, p. 266, 2020.

[26] K. Darvish *et al.*, "Whole-Body Geometric Retargeting for Humanoid Robots," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, Toronto, ON, Canada, 2019, pp. 679–686.

[27] U. Zabala, I. Rodriguez, J. M. Martínez-Otzeta, and E. Lazkano, "Learning to gesticulate by observation using a deep generative approach," in *International Conference on Social Robotics*, Madrid, Spain, 2019, pp. 666–675.

[28] L. Gui, K. Zhang, Y. Wang, X. Liang, J. M. F. Moura, and M. Veloso, "Teaching Robots to Predict Human Motion," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018, pp. 562–567.

[29] J. B. Martin and F. Moutarde, "Real-time gestural control of robot manipulator through Deep Learning human-pose inference," in *International Conference on Computer Vision Systems*, Thessaloniki, Greece, 2019, pp. 565–572.

[30] C. Hwang and G. Liao, "Real-Time Pose Imitation by Mid-Size Humanoid Robot With Servo-Cradle-Head RGB-D Vision System," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 181–191, 2019.

[31] "MediaPipe Pose," Accessed: Apr. 3, 2021. [Online]. Available: https://google.github.io/mediapipe/solutions/pose.html

[32] C. J. Taylor, "Reconstruction of articulated objects from point correspondences in a single uncalibrated image," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, Hilton Head, SC, USA, 2000, pp. 677–684 vol. 1.

[33] "Pepper (NAOqi 2.5)," Accessed: Mar. 20, 2021. [Online]. Available: https://developer.softbankrobotics.com/pepper-naoqi-25

[34] E. Pot, J. Monceaux, R. Gelin, and B. Maisonnier, "Choregraphe: a graphical tool for humanoid robot programming," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, Toyama, Japan, 2009, pp. 46–51.

[35] E. Coronado and G. Venture, "Towards IoT-Aided Human–Robot Interaction Using NEP and ROS: A Platform-Independent, Accessible and Distributed Approach," *Sensors*, vol. 20, no. 5, p. 1500, 2020.

[36] X. Hongyi, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2020, pp. 6183–6192.

[37] "MediaPipe Holistic," Accessed: Apr. 3, 2021. [Online]. Available: https://google.github.io/mediapipe/solutions/holistic