applied sciences

MDPI

Article

Upper Body Pose Estimation Using Deep Learning for a Virtual Reality Avatar

Taravat Anvari, Kyoungju Park * and Ganghyun Kim

Department of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Republic of Korea
* Correspondence: kjpark@cau.ac.kr

Abstract: With the popularity of virtual reality (VR) games and devices, demand is increasing for estimating and displaying user motion in VR applications. Most pose estimation methods for VR avatars exploit inverse kinematics (IK) and online motion capture methods. In contrast to existing approaches, we aim for a stable process with less computation, usable in a small space. Therefore, our strategy has minimum latency for VR device users, from high-performance to low-performance, in multi-user applications over the network. In this study, we estimate the upper body pose of a VR user in real time using a deep learning method. We propose a novel method inspired by a classical regression model and trained with 3D motion capture data. Thus, our design uses a convolutional neural network (CNN)-based architecture from the joint information of motion capture data and modifies the network input and output to obtain input from a head and both hands. After feeding the model with properly normalized inputs, a head-mounted display (HMD), and two controllers, we render the user's corresponding avatar in VR applications. We used our proposed pose estimation method to build single-user and multi-user applications, measure their performance, conduct a user study, and compare the results with previous methods for VR avatars.

Keywords: avatar; immersion; pose estimation; virtual reality

Citation: Anvari, T.; Park, K.; Kim, G. Upper Body Pose Estimation Using Deep Learning for a Virtual Reality Avatar. *Appl. Sci.* **2023**, *13*, 2460. https://doi.org/10.3390/app13042460

Academic Editor: Chaman Sabharwal

Received: 3 January 2023
Revised: 2 February 2023
Accepted: 11 February 2023
Published: 14 February 2023

1. Introduction

Immersion requires self-representation in the virtual environment, thus a virtual body [1]. The focus on presence and immersion requires connecting a user and a virtual body. It requires approximating the location of a human body to create a visualization that makes sense to the avatar's owner. This field has been growing recently at an impressive rate. Nevertheless, the exploitation of immersive virtual reality (VR) has enabled a reframing of whether and to what extent it is possible to experience the same sensations with a virtual body inside an immersive virtual environment as with a biological body [2].

For a user to feel ownership of the VR body, it is necessary to estimate and visualize the user's motion [3]. Therefore, pose estimation is among the most widely discussed topics in VR, with different techniques such as motion capture and inverse kinematics (IK) methods. Furthermore, we believe that considerable care is necessary for the upper body of a VR user. While VR users rarely glance at their legs, adding arms has a high potential to improve embodiment because the arms are often visible when interacting with the environment [4]. Thus, we focus on the upper body of the real-world user.

VR applications capture the user's movements in the physical world using various devices and synchronize the avatar's movements in a virtual world. Various devices include motion capture systems, depth sensors, trackers, etc. A conventional device is a motion capture system that is accurate. However, such motion capture devices require a significant amount of room, and the user's wearing a special suit. Moreover, motion capture devices are typically not affordable for personal users. In contrast, inexpensive depth sensors such as Microsoft Kinect do not have a sufficiently high resolution [5,6] or frame rate to compete with professional devices for motion capture. VR devices with

hand-held controllers and eye trackers provide limited motion information. Based on limited motion data, an IK method estimates the upper body pose, and a machine-learning classification method even identifies the user from human kinesiological movements [7].

For enhanced immersion in multi-user VR applications, a user must see other users' realistic motion at approximately 60 frames per second (fps). Personal VR users typically have limited space, but existing real-time motion capture equipment requires ten by ten meters. So, motion capture equipment is not possible for personal users. Due to differences in the hardware performance and the network speed of various users, it is challenging to guarantee real-time pose estimation and visualization, especially in low-performance VR devices operating with a relatively slow network speed. We achieve pose-mimicking in real-time, with lower computational requirements, using a data-driven approach based on a motion capture dataset and deep learning rather than pose estimation using the IK method.

We aim for the upper body animation of a VR avatar with low cost and high performance. One objective is to estimate and visualize the self-avatar following the user's poses with only a head-mounted display (HMD) and controller data. Another objective is performance so that our system produces an avatar with minimum latency while minimizing the delay even with low-performance hardware and slow networks of multi-user environments. Our method requires that a VR user has a HMD and controllers, a relatively low cost compared to a motion capture system, and shows the upper body of an avatar computationally more efficiently than an IK method. By employing our method, personal VR users are able to visualize the upper body of their self-avatars with minimum latency. However, VR developers are required to use expensive motion capture systems to acquire the pose data or use available public human pose data.

We use a commercial VR system to capture hand and head motions and a machine-learning approach to improve the presentation of the upper body features in the VR system. Our learning architecture is able to use either a public pose dataset or a collected pose dataset, and apply a modified neural network architecture. In VR experiments, we visualize the upper body of the corresponding avatar as the user moves based on the trained model. We then build single-user and multi-user VR applications, evaluate the performance and naturalness of our method by comparing it with an IK-based approach, and conduct a user study for embodiment.

Our study's main contributions are:

- We estimate and visualize the VR user's pose naturally and efficiently using a deep learning method.
- We apply the training results to a VR environment to generate upper body pose animation in real time by minimizing the latency, even for users with a low-performance machine and a relatively slow network.
- We design and test single-user and multi-user VR applications with upper body pose animated avatars that match the user's motion.
- We evaluate the performance of the proposed upper body pose estimation method and compare it with the performance of a pose estimation method using IK; we also assess user immersion.

The remainder of the paper is structured as follows. Section 2 reviews previous work related to our study, and Section 3 explains our proposed pose regression method for the VR environment. Section 4 describes pose regression results, including performance evaluation and accuracy. Section 5 presents the design and procedure of VR experiments. Section 6 evaluates VR experiments' results by analyzing user studies statistically. Section 7 discusses, Section 8 presents the limitations, and Section 9 concludes.

2. Related Works

Some humanities scholars have begun to use the term "presence" (from the scientific literature on VR), defined loosely as the feeling of being there. The terms immersion and presence are increasingly seen together, although both have been so loosely defined as interchangeable—which they often are [8]. Presence induced by technology-driven

immersion is a form of illusion because, in VR, stimuli are merely a form of energy projected onto our receptors. Examples include light from pixels on a screen or sound waves recorded at different times and places emitted from speakers [9]. Other researchers distinguish various forms of presence from unique perspectives. Presence is divided into four core components that are merely illusions of a non-existent reality. Based on our study's scope, we only discuss the illusion of self-embodiment and the illusion of social communication.

Self-embodiment is the perception that the user has a body within the virtual world. Presence is greatly strengthened and experienced more deeply if a user sees a visual object touching the skin while a physical object also touches the skin [10]. Gall et al. [11] prove that an illusion of embodiment intensifies the emotional processing of the virtual environment. Even if the VR body does not look similar to one's own body, the presence of a virtual body can be compelling. In contrast to physical attributes such as body shape and color, the importance of motion cannot be overstated. A discrepancy between visual body motion and actual physical activity can result in a breakdown of presence.

Social presence refers to the sense that a user is genuinely communicating, both verbally and through nonverbal cues such as body language, with other entities in the same environment, whether they be computer-controlled or operated by other users. Physical realism is not a prerequisite to social realism. Users have been found to express anxiety when they cause pain to low-fidelity virtual characters [12], and when users with a fear of public speaking must talk in front of a low-fidelity virtual audience [13]. While the level of social presence can increase as the realism of human behavior and objects (referred to as behavioral realism) improves, the mere tracking and rendering of a few key points on human users can effectively create a compelling experience [14].

Most of the research on the presence of VR uses expensive motion capture systems [15,16]. Consumer-grade depth sensors such as a Kinect detect body tracking [17], but do not provide the necessary accuracy. A Kinect v1.0 is limited: a low-resolution RGB-depth camera with a 320×240 16-bit depth sensor and a 640×480 32-bit color sensor, at a capture rate of 30 Hz. This low spatial and temporal resolution favors interactive gaming experiences over accurate pose reconstruction, resulting in the loss of crucial information for faster motion. A Kinect v1.0 device also limits the capture space to a small region of approximately two square meters [18].

Steed et al. [19] demonstrate that incorrect poses could decrease embodiment, suggesting that it might be more helpful not to display limbs if their poses are not sufficiently accurate. Consequently, the proposed method must obtain an accurate position to create a true sense of embodiment for a VR user. This study uses a data-driven solver method to predict upper body poses by integrating pose estimation in VR.

Some companies combine IK with VR, which can animate 3D characters of any shape or size in real time. Studies present an entire body using IK without running a user study to evaluate embodiment [20,21]. The result can be a virtual avatar that responds with natural movements, creating a stronger sense of VR immersion. However, many iterations are mandatory before converging on a solution that may ultimately generate improper joint orientations [22]. Nonetheless, the evidence for this is inconclusive.

Researchers have introduced deep learning methods to estimate human body poses. Zhou et al. [23] propose a body pose estimation approach using a regression model based on a deep neural network (DNN) with images as the input. Tekin et al. [24] embed a kinematic model into the deep learning architecture to impose kinematic constraints. They also propose a deep regression architecture combining a traditional convolutional neural network (CNN) for supervised learning and an auto-encoder that implicitly encodes 3D dependencies between body parts.

Recent studies have introduced a Transformer model, a sequence-to-sequence encoder-decoder model, to vision problems [25,26], and pose regression problems [27–29]. Yang et al. [27] propose a transformer network, Transpose, which estimates 2D pose from images. Lin et al. [28] combine CNNs with transformer networks in their method METRO (Mesh Transformer) to reconstruct the 3D pose and mesh vertices from a single image.

Zheng et al. [29] present a spatial–temporal transformer called PoseFormer to exploit the keypoint correlation of the human joints for each frame and the temporal correlations and to produce temporally coherent 3D human poses from videos.

In addition, researchers have used deep learning for multi-person pose estimation from an image-based standpoint. OpenPose [30] is one of the most popular bottom-up approaches for multi-person human pose estimation. DeepCut [31] is another bottom-up approach for multi-person human pose estimation, and AlphaPose, based on the regional multi-person pose estimation (RMPE) framework [32], is a popular top-down method of pose estimation. Whereas the datasets for these methods include images, this study investigates estimating the human upper body pose using a numerical dataset.

Moreover, researchers have used motion capture in data-driven deep learning methods. Toshev et al. [33] formulate pose estimation as a DNN regression problem by localizing the body's joint positions from images. The data-driven clarifier is based on motion capture data the researchers collect to identify a solution similar to the current pose [34–36].

This study proposes a pose regression model to predict human upper body joint poses using deep learning. We then apply our model in a VR environment to evaluate our results in real time.

3. Pose Regression Methods

3.1. System Overview

Figure 1 illustrates an overview of the proposed method. Our method comprises two parts, A and B. Part A is dedicated to the training step of pose regression, and part B focuses on the avatar animation step. Part B renders avatars in virtual reality following users' motions in the physical world.

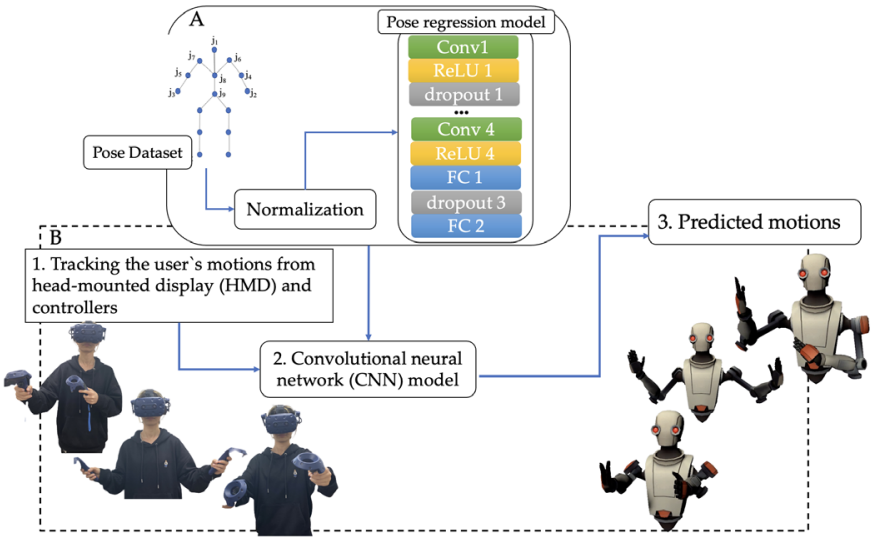

Figure 1. Our proposed pose estimation method for an avatar. Part A is network architecture for pose regression and Part B is the avatar animation step.

As depicted in Figure 1, part A illustrates the comprehensive view of the training procedure. To prepare the dataset, we collect the positions and orientations of the upper body joints of a user from a diverse range of gestures and movements in every frame. Subsequently, we perform preliminary noise reduction and normalization processing on the dataset. We consider the positions and orientations of the head and two hands as inputs to our pose regression model. Our model is based on a classical regression model, comprising four convolutional layers and two fully connected (FC) layers. The resulting hyper-parameters of our pose regression model are utilized in B to estimate the upper body pose based on the VR user's controllers and head-mounted display (HMD) data.

3.2. Data Acquisition

In traditional pose regression studies, researchers use images as inputs to train the neural networks and pose estimation is considered a classification problem. In contrast, our method diverges from traditional approaches by utilizing numerical 3D data to train our pose regression model. We acquire the numerical 3D data by attaching optical motion capture sensors to specific body joints, a standard method of obtaining human motion capture data. We focus more on VR interaction poses, considering that VR users stay in a limited space and perform actions such as twisting their bodies, talking with moving arms, sitting, standing, and selecting and manipulating objects using their arms. Public human datasets, such as CMU [37], PFNN [38], and MHAD [39], have walking, running, stepping, and sports motions. These public datasets are deficient regarding the selection and manipulation motions utilizing arms.

Our motion-capturing strategy is one way to achieve various human poses within a limited time frame. To focus more on VR interaction poses, we hire four choreographers to design moves to meet our requirements. We use an OptiTrack motion capture device while the subjects pose the moves according to the music. Our datasets expand further by capturing more human poses while a human performs activities in different environments and incorporating the public human pose datasets. The resulting data serves as a testbed for our VR pose regression model, providing a comprehensive and diverse representation of VR interaction poses.

From a total duration of approximately 2838 s, we select 60 fps out of 200 fps by filtering out similar poses. Consequently, our pose dataset includes 170,280 pose data samples divided into 80 to 20 between train and test data. We use nine upper body joints for our pose regression model with 54 parameters. For every joint, the dataset includes six values, local positions in the x-, y-, and z-coordinates, and local orientation angles about the x-, y-, and z-axes.

As depicted in Figure 2, we name each joint in order: head, left hand, right hand, left elbow, right elbow, left shoulder, right shoulder, spine, and pelvis, referred to as j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, and j_9, respectively. Let $\mathbf{p}_i = (px_i, py_i, pz_i)$ and $\mathbf{r}_i = (rx_i, ry_i, rz_i)$ be the position and orientation of a joint j_i in three dimensions. As depicted in Figure 2, our global coordinate system is defined as follows: the x-coordinate is from the left shoulder to the right shoulder, the y-coordinate is upward, and the z-coordinate is forward. Other than the head and the root joint, the other joints' orientations are relative to the corresponding parent joints. This approach correlates highly with Ben-Ari [40] and further supports the concept of IK [9] for orientation. We normalize our dataset before training.

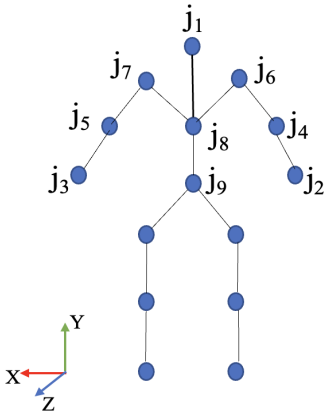

Figure 2. Upper body joints j_1, j_2, j_3, j_4, j_5, j_6, j_7, j_8, and j_9 indicate the head, left hand, right hand, left elbow, right elbow, left shoulder, right shoulder, spine, and pelvis.

3.3. Pose Regression Model

The entire process of a pose regression model is shown in Figure 3. Through comparison, we find CNN outperforms DNN in our problem. The hierarchical structures of

features in CNNs contribute the superior performance. Of paramount importance, our input data comprises two distinct types: position and orientation. We maintain minimal connectivity between these two data types to ensure optimal performance. The correlation between these two different data types is not relevant to our objective of inferring the position of the joints. Separately, treating the position and orientation data is optimal rather than trying to find a correlation between position and orientation values. In light of this, we leverage and adapt an existing CNN architecture to derive separate feature maps from the positions and orientations.

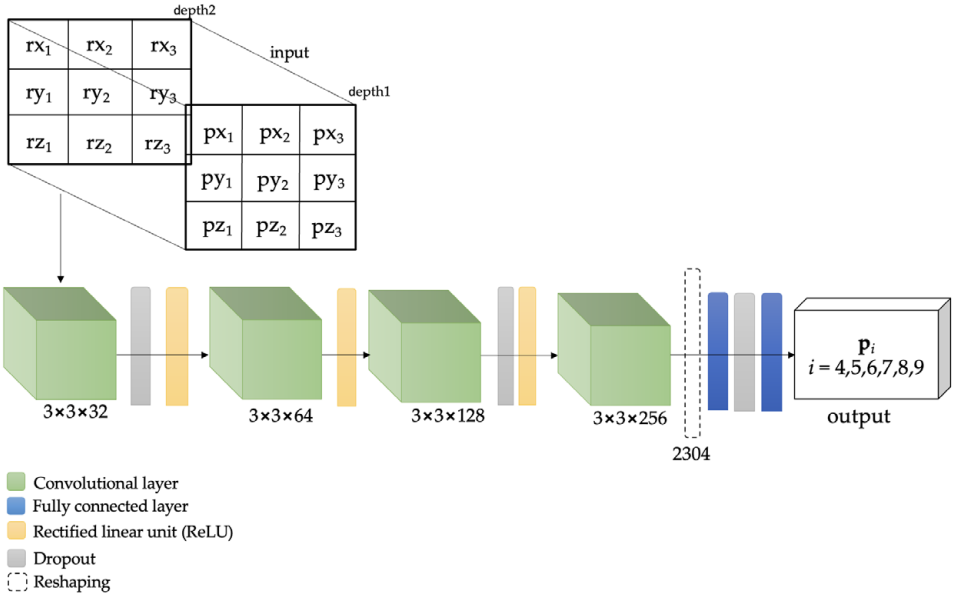

Figure 3. Our pose regression model.

We modify the existing Conv2D layer to incorporate the position and orientation data in different depths of our input matrix. Convolutional layers generate feature maps that separately have the position and orientation information. Without this consideration, one datatype could dominate the other after convolution, resulting in the loss of one of the features. Since our architecture has the position and orientation data in different depths, our convolutional layer keeps the position and orientation features rather than one fading away. Position and orientation data in our architecture works similarly to the red, green, and blue channels in the conventional Conv2D layer for image-related problems.

The input matrix components are as depicted in Equation (1). Given the 18 input values, which are the position and orientation values of j_1, j_2, and j_3, our pose regression model estimates the position values for the target joints j_4, j_5, j_6, j_7, j_8, and j_9. In the input matrices, the width is the joint index of the head, left-hand, and right-hand joints, the height is the values along x-, y-, and z-axes, respectively, and the depth is the position and orientation data.

$$\begin{bmatrix} \begin{bmatrix} px_1 & px_2 & px_3 \\ py_1 & py_2 & py_3 \\ pz_1 & pz_2 & pz_3 \end{bmatrix} \begin{bmatrix} rx_1 & rx_2 & rx_3 \\ ry_1 & ry_2 & ry_3 \\ rz_1 & rz_2 & rz_3 \end{bmatrix} \end{bmatrix} \quad (1)$$

The structure of our convolutional layers includes a convolution layer with a ReLU activation and two dropout layers, followed by two fully connected layers. The convolutional layer has a filter size of $3 \times 3 \times 2$ along the width and height; the stride in the horizontal and vertical directions is one. Our convolution is applied to the cells, and the same weights are applied to the position data in the first depth and orientation data in the second depth. This allows for treating position and orientation as a group and deriving features between joints. A zero-padding solution controls the shrinkage of the output dimension from each convolution layer, and we add zero-padding on the borders. The convolutional

layer derives the features from the input matrices. The first convolutional layer operates on the input matrix to perceive the corresponding features within each frame. The next convolutional layer detects the relevant features from the previous layer's feature map. After a convolution operation, the obtained feature map is equivalent to the corresponding feature detection results. After each convolutional layer, the dimension of the feature map is $\varphi^{\text{dim}}{}_{\text{layer}} = 3 \times 3 \times (2^{4+\text{layer}})$. In the end, the fully connected layer applies the obtained dependency of the input to the output. The model repeats regression steps for each of the outputs being estimated. The target values of our model are the positions of the six joints, $\mathbf{p}_4$, $\mathbf{p}_5$, $\mathbf{p}_6$, $\mathbf{p}_7$, $\mathbf{p}_8$, and $\mathbf{p}_9$, which are the left elbow, the right elbow, left shoulder, right shoulder, spine, and pelvis positions, respectively.

In order to mitigate the issue of overfitting, we employ two strategies: dropout and dataset augmentation. We apply a dropout solution to the first convolution layer and the first fully connected layer. Therefore, a neuron stops working at 0.7 probability—a 30% probability of dropping the neuron from the network. Furthermore, after acquiring the captured data, we re-generate the in-between pose data by applying linear interpolation of the position and orientation values of the joints.

Finally, we use the mean squared error (MSE) loss function and obtain a continuous numerical output. Equation (2) defines how we calculate the MSE in our model, where n is the number of iterations (150 for the target joint) in one epoch, and R_t and $\hat{R}_t$ are the ground truth and predicted values, respectively. Here, k is the counter for training steps for a specific target joint (from 0 to n). We train the network with five epochs and the Adam optimizer [41] with a learning rate of 0.001.

$$\text{MSE} = \frac{\sum_{k=0}^{n} \left(R_k - \hat{R}_k\right)^2}{n}, \tag{2}$$

3.4. Validation of the Pose Regression Model

Our pose regression model optimizes to minimize the overall MSE in Equation (2). During training our model for 5 epochs and 150 iterations, the calculated MSE decreases gradually, as depicted in Figure 4. As can be seen in Figure 4, the graph illustrates the relationship between the number of iterations in the fifth epoch (horizontal axis) and the calculated mean squared error (MSE) (vertical axis). The MSE decreases as the number of iterations increases, indicating that our model's choice of optimization method was appropriate. Figure 5 compares the output of the predicted values to the ground truth of the right shoulder. The horizontal axis represents the number of iterations, while the vertical axis illustrates the mean absolute error (MAE) between the y values of the joint j_7. From this comparison, it can be inferred that our model's prediction of joint j_7 values is in close alignment with the ground truth, further validating the effectiveness of the chosen optimization method. As Figure 5 shows, the MAE decreases within an acceptable range and is small enough to be unnoticeable in practice. Overall, these results demonstrate the validity and accuracy of the proposed model.

We validate our pose regression model by applying the pose regression model to both the training and test data and then calculate the MSE for the ground truth. The mean MSE of the joints is 0.0277 for the training data and 0.0448 for the test data. Table 1 shows the MSE of the joints; j_4, j_5, j_6, j_7, j_8, and j_9. Small MSEs for the test data indicate that our regression method yields accurate results.

To conduct a comprehensive comparison between our method and the IK method, we calculate the MSE between the ground truth position of the joints and the output of the Final IK [42] method. The results of this comparison are presented in Table 2 and demonstrate that our method results in lower errors with an MSE of 0.0448, compared to the Final IK method with an MSE of 0.8066. Furthermore, Table 3 compares the MAE of the right shoulder joint from the test dataset using both our method and the Final IK method. The results indicate that our method surpasses the performance of the Final IK method.

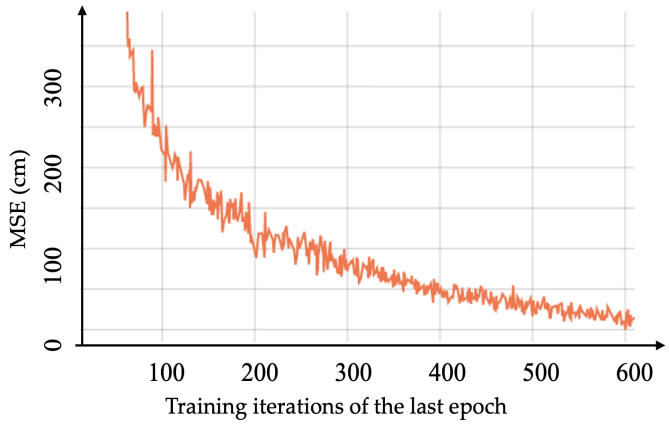

Figure 4. Mean squared error(MSE) of our pose regression model in the last (5th) epoch.

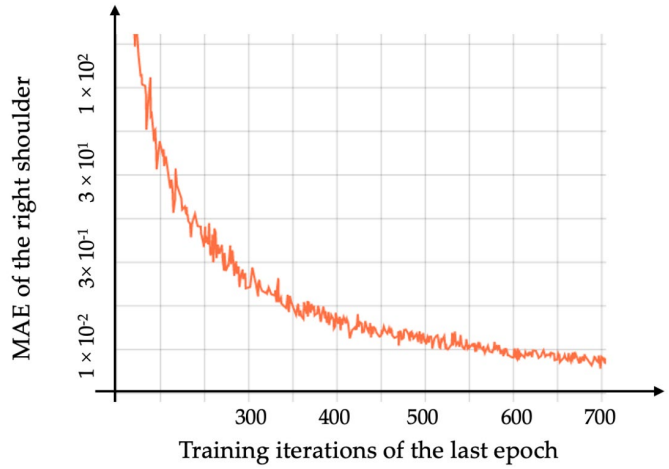

Figure 5. Mean absolute error(MAE) of the predicted values and the ground truth data of the right shoulder or joint j_7 of the last (5th) epoch.

Table 1. MSE values for j_4, j_5, j_6, j_7, j_8, and j_9.

Joints	MSE of Training Data (cm)	MSE of Test Data (cm)
j_4	0.0245	0.0342
j_5	0.0387	0.0456
j_6	0.0336	0.0448
j_7	0.0267	0.3870
j_8	0.0242	0.3940
j_9	0.0382	0.4270

Table 2. MSE values of the test data.

Methods	MSE of Test Data (cm)
Final IK	0.8066
Ours	0.0448

Table 3. MAE values of the right shoulder of the test data.

Methods	MAE of the Right Shoulder of Test Data (cm)
Final IK	0.1587
Ours	0.0610

For further evaluation, we employ the MPJPE (mean per joint position error) [43] metric, a widely accepted standard for evaluating human pose estimation. We evaluate the accuracy of the upper body tracking on a sample of 27,056 frames drawn from our train and test datasets each. Table 4 shows the calculated MPJPE metric by comparing the local positions of the joints in the reconstructed motion to those in the reference.

Table 4. Mean per joint position error (MPJPE) values for our method.

Joints	MPJPE of Training Data (cm)	MPJPE of Test Data (cm)
Our method	2.4026	1.7301

3.5. Avatar Pose Regression

In VR applications, we can obtain the position of the VR user's specific body joints in the real world. The HMD and controllers track the user's head and both hands. We then apply our pose regression model to estimate the other body joints. The input to the pose regression model is the positions $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, and orientations $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ of a HMD, left controller, and right controller, respectively. It appraises the values of $\mathbf{p}_4$, $\mathbf{p}_5$, $\mathbf{p}_6$, $\mathbf{p}_7$, $\mathbf{p}_8$, and $\mathbf{p}_9$ as outputs for the avatar. After estimating the positions of the joints, we calculate the orientation of the joints of the entire upper body of the VR user.

We use the avatar's parent–child hierarchy to estimate the orientation angles of joints. We compute the range of orientation angles of the neck, shoulders, pelvis, and spine joints using the estimated positions of the joints and the known orientation angles of the head and wrist joints. The neck joint is oriented using the values of the head joint and its position. Let the directional vector from the neck and head joints be the y-axis of the neck. We then determine the orientation angles of the neck by rotating the head coordinate system to align its y-axis with that of the neck.

Similarly, we determine the orientation angles of the shoulders, pelvis, and spine joints. For the shoulder joints, the positions of the left and right shoulders are used to define the x-axis of the shoulder coordinate system. The neck coordinate is rotated to align its x-axis with the shoulder coordinate system. The coordinates of the pelvis and spine joints are defined using both the coordinates of the neck and shoulders.

After these steps, we calculate the orientation angles of the elbow joints. With known position and orientations of the wrists and shoulders, the orientation values of the elbow joints are calculated as follows. The y-axis is computed as the unit vector along the line from the elbow to the wrist joint locations, an x-axis is a unit normal vector to the plane defined with three points, including the elbow, wrist, and shoulder joint locations, and the z-axis is a cross product of the x- and y-axes. We point an imaginary line from the shoulder to the hand, compute the distance to determine the threshold value of the elbow using the cosine rule [44], and produce the most probable state of the elbow.

4. Pose Regression Results

In VR systems, our method obtains input data of the head and two controllers of the user from the physical world, estimates the upper body pose, and visualizes the corresponding pose of the avatar in VR. We implement our algorithm in Python using TensorFlow to evaluate our neural network. The neural network model requires 22 MB of memory. We speed up the neural network operation using a GPU (NVIDIA GeForce GTX1080) with an HTC VIVE VR device and Unity version 2019.3.11f1.

The resulting upper body pose estimation of the avatar is depicted in Figure 6. The left image is the self-avatar estimated using our method in VR, and the right image is the user in the physical world. As users move their heads and controllers, the self-avatar follows the users' motion. Our proposed method can successfully generate the upper body of the avatar in a consumer VR system with reasonable accuracy in a short time from the HMD and controllers, maintaining temporal consistency between frames. Since Parger et al. [4]

previously confirmed that using an IK system with a fair price achieves results comparable to a full motion capture system, we do not compare our model with a motion capture method.

Figure 6. Comparison of the virtual reality(VR) user (**right**) and the corresponding self-avatar using our method (**left**).

To compare our pose regression results with IK results, we test several poses of a user and estimate the upper body poses using both the IK and our methods. We use the Final IK [42] tool from RootMotion in Unity to estimate the upper body pose using an IK method. Figure 7 illustrates the example comparison of our results and IK results for interaction motions such as left-hand blocking in Figure 7a, right-hand blocking in Figure 7b, swaying in Figure 7c, lifting in Figure 7d, right-hand in front in Figure 7e, and folding arms motion in Figure 7f. The red lines are IK results, and the blue lines are our results from the users in the physical world. Figure 8 displays the avatar pose estimation results in VR using two methods; our results in the middle row and IK in the bottom row as the VR user in the top row grabs and throws a ball in the physical world. As depicted in Figure 8, both our pose-estimated avatar and the IK-based avatar follow the motion of the VR user similarly. The IK-based avatar throws a ball naturally, and our pose-estimated avatar also throws a ball naturally. The MSE of our method is about 0.041 cm, and that of the IK method is about 0.042 cm for 6736 frames while playing the standalone VR applications.

We also compare the performance of our regression method and that of IK in sample VR applications. We measure the computation times per frame while users play sample VR games and analyze the minimum, maximum, and average computation times. We use the Unity Profiler, a CPU usage profiler module that tracks the time spent on the application's main thread per frame. For computation time calculation, we accumulate time spent on rendering, scripts, physics, and animation categories. We conduct these experiments with standalone and multi-user VR applications over the network.

Table 5 for a standalone VR application, presents the computation times of our method and the IK method. The computation time to render a self-avatar using our method is between 3.19 and 14.75 milliseconds for 6736 frames (70 s). As in Table 5, the average computation time to generate a frame is 6.73 milliseconds using our method and 10.23 milliseconds using the IK method. Moreover, our method's maximum computation time is

almost five milliseconds faster than IK. Consequently, our approach is generally faster with lower latency than the IK method.

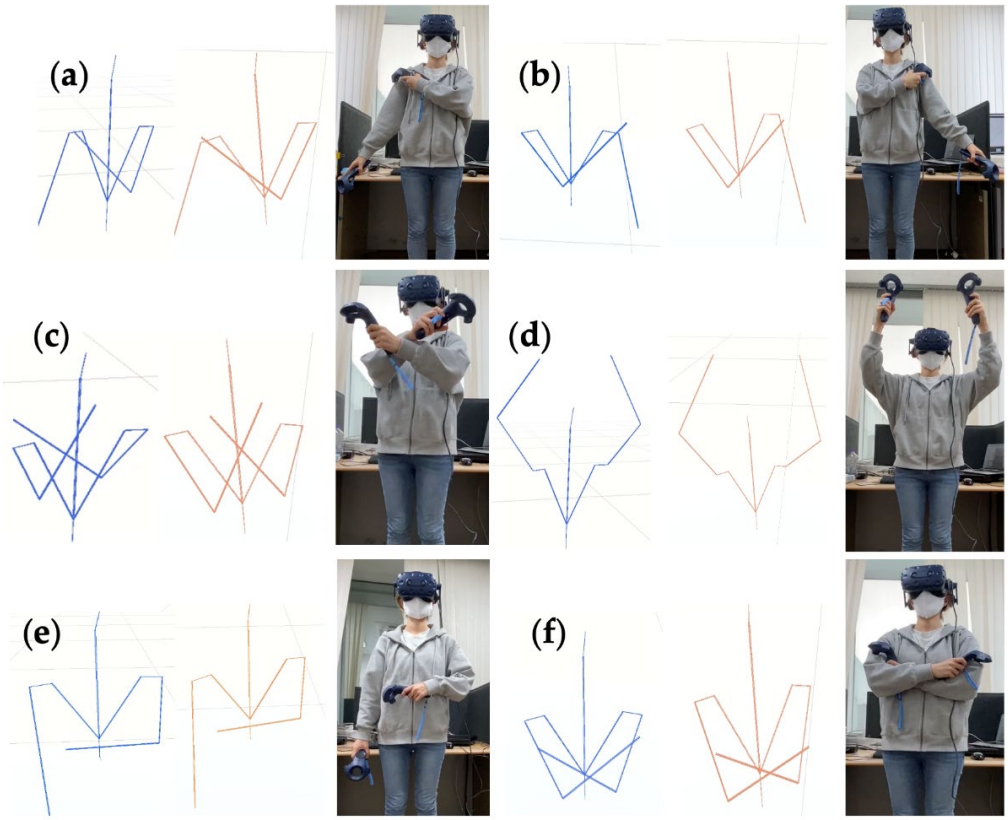

Figure 7. Comparison of our results in blue (**left**) and inverse kinematics (IK) in red (**middle**) from the user (**right**) (**a**) left-hand blocking, (**b**) right-hand blocking, (**c**) sway motion, (**d**) lift motion, (**e**) right-hand in front motion, and (**f**) folded arms motion generation.

Figure 8. Comparison of our results (**middle**) and IK results (**bottom**) from the user (**top**).

Table 5. Performance evaluation in a standalone VR application.

Solution	Minimum (ms)	Maximum (ms)	Average (ms)
Our method	3.19	14.75	6.73
IK	3.81	19.65	10.23

The computation time of multi-user VR applications includes rendering and networking time. Table 6 compares the performance of our method and IK in a multi-user application. Our approach minimizes the computational time between 3.36 and 17.1 milliseconds for 8661 frames (90 s) in a multi-user game environment. The average computational time is 9.21 milliseconds using our method and 16.78 milliseconds using the IK method. Our approach is faster than an IK system—our method minimizes latency even for a multi-user VR system.

Table 6. Performance evaluation in a multi-user game.

Solution	Minimum (ms)	Maximum (ms)	Average (ms)
Our method	3.36	17.01	9.21
IK	4.28	22.41	16.78

Our pose estimation method predicts the upper body from the head and two hand-held controllers, similar to the IK method. Moreover, the performance evaluation of computations shows that our process yields lower latency than the IK method.

5. VR User Experiments

We design and conduct a VR user study to investigate whether our estimated avatars enhance the feeling of presence in VR comparable to the IK method. A motion capture system and an IK system were compared by Parger et al. [4] for an upper body pose in VR, and an IK system achieves significantly different results than a full motion capture system in user studies. Therefore, we compare our model with an IK method excluding a motion capture method. Considering the pose estimation results that our method is accurate and more efficient, we construct three hypotheses:

H1. *Our method generates a naturally looking avatar comparable to the IK method.*

H2. *Our method's display of the upper body in VR provides the illusion of self-embodiment, similar to the IK method.*

H3. *Our method's lower latency than the IK method results in less break-in-presence and enhances user experience.*

We used a within-subject design and divided the process into two tasks. The first task includes self-embodiment in a single-user VR application, for which we implement an archery game. The second task is a multi-user environment application where users throw and catch balls. Please also refer to the Supplementary Video S1 for additional visual findings.

5.1. VR Game Design

We create a single-user VR system: an archery game using a virtual bow and arrow that a single user uses to shoot at a target in VR, as shown in Figure 9. Our scene contains four marks at different distances from the user's position. In this game, a user begins at the center of a specific zone and shoots the targets. The targets are at distances of 4.5, 6.5, 8.5, and 10.5 m from the user and are worth 5, 10, 15, and 20 points, respectively. The further the target is, the more points the user earns. Each user can shoot up to five arrows at the targets within a limited time of 70 s. Figure 9 shows an archery game with our pose-estimated avatar in the leftmost column, the IK-based avatar in the middle column, and the VR user in the rightmost column. As a user manipulates hand-held controllers, avatars mimic the user's motion to shoot the arrows.

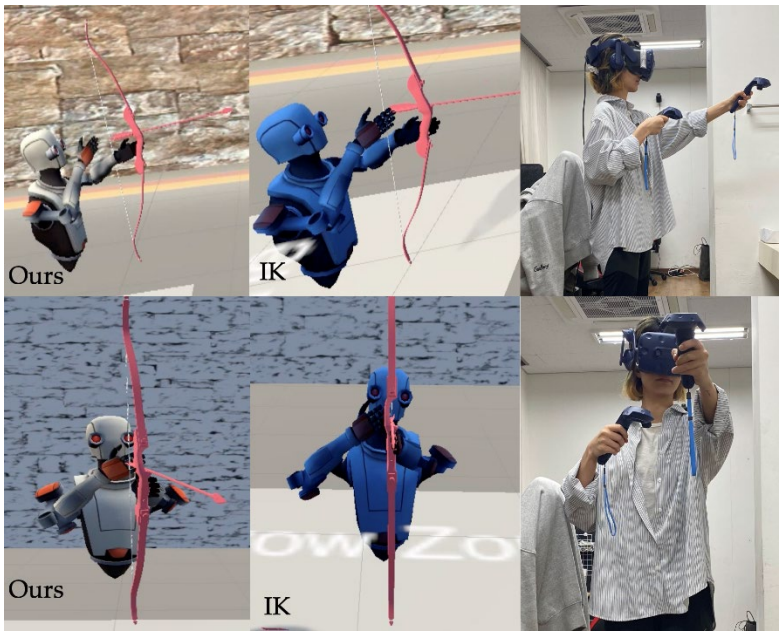

Figure 9. Archery game in two modes. Camera view of the user playing the archery game (**right**) using our method (**left**) and the IK method (**middle**).

We design a ball-grabbing VR game as a user study of the proposed pose regression method in a multi-user environment. In a ball-grabbing VR game shown in Figure 10, a user throws a ball toward the basket, and another receives a ball using the holding basket. Figure 10 shows the avatars in a ball-grabbing game using our method at the top, an IK method in the middle, and VR users at the bottom. Ball grabbing and throwing poses of avatars match the poses of the users in both IK and our method modes. A user throwing a ball needs to pay attention to the pose of another user with a basket. Vice versa, a user receiving a ball needs to pay attention to the pose of the throwing user. We asked 15 participants to form five groups of three users randomly. As depicted in Figure 10, two users grab the balls and throw them toward the target basket held by the third user from a distance of 5 m. Users have a total of 20 balls, a successful ball-grabbing score is 1, and the game time is limited to 90 s. Consequently, 20 is the maximum score for each group.

5.2. Questionnaires

We ask participants to answer seven questions for the archery game and five questions for the ball-grabbing game. Each question has to be answered on a five-point Likert scale from one (very poor) to five (excellent). For both tasks, most questions are about presence, embodiment, sense of moving, accuracy, and being natural. Tables 7 and 8 indicate our questions after the archery and ball-grabbing tasks.

5.3. Experimental Procedure

We recruited 15 healthy college students to play under three different modes including a tutorial in hands-only mode, our method mode, and an IK mode. At first, we asked for a demographic questionnaire. The demographic questionnaire includes the gender, age, major field of study, and information on the participant's prior experience with virtual environments (VEs). Among the participants, 9 (60%) were male, 12 (80%) were right-handed, and all were majoring in computer science. The average age of our participants was 26.1333, with a standard deviation (STD) of 1.9223. Four participants indicated that they had used the HTC VIVE system. Three subjects stated that they had used the Oculus system. Furthermore, 40% of the selected participants had impaired vision but could either wear contact lenses or customize the VR HMD to have good eyesight.

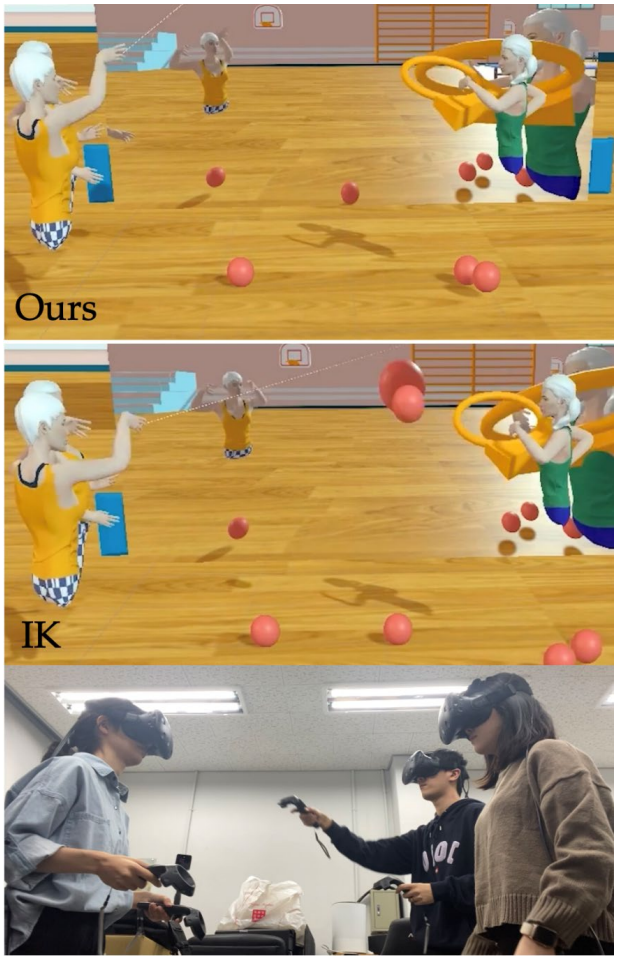

Figure 10. Ball-grabbing game in two modes. Camera view of three users (**bottom**) playing the ball-grabbing game using our method (**top**) and the IK method (**middle**).

Table 7. Questionnaire for the archery game.

Number	Question
Q1	How strongly did you feel a sense of presence while using this method?
Q2	How strongly did you feel a sense of embodiment while using this method?
Q3	How compelling was your sense of moving around inside the virtual environment?
Q4	How accurate was the environment for actions you are in?
Q5	To what extent did you observe a delay between actions and the expected outcomes?
Q6	How natural did your interactions with the environment seem?
Q7	Do you prefer this method overall (for a single-user virtual game)?

Table 8. Questionnaire for the ball-grabbing game.

Number	Question
Q8	How strongly did you feel a sense of presence while using this method?
Q9	How strongly did you feel like your virtual body was your own while using this method?
Q10	How well were you able to predict the outcome of your actions in the virtual environment?
Q11	How easily were you able to move and manipulate objects in the virtual environment?
Q12	Would you say you prefer this method over others for a multi-user virtual game?

The participants' first task was to play a single-user game, the archery game. The single-user experiments were conducted in two sets, beginning with a hands-only mode, followed by an IK mode and our method mode in random order. Participants were not informed of the modes they were playing in advance. The hands-only mode was a trial, so a participant could get used to the game's rules. IK and our methods are the modes we evaluate and analyze. This task was tested on a PC with one HTC VIVE, two controllers, and two base sensor stations. The participants initially stood in an arrow zone of 110 cm by 100 cm in the set-up room and were guided to stay inside the arrow zone during the game. They then wore the HMD and stated any discomfort to ascertain the best experience. They posed similar to an archer with their hands and got ready to shoot toward the target. They moved one hand to shoot the arrow and the other to adjust the bow. The game was over when all the arrows were used, or when the time limit was exceeded. After finishing the first set of two modes, participants took a rest for the 60 s. Participants answered seven questions (Table 7) in seven minutes. This procedure was repeated for the other set.

Then, groups of participants gathered to play a multi-user game, a ball-grabbing game. In the ball-grabbing game, three participants in a group throw and grab a ball at each other in a virtual shared space while they are physically located in different places. One room, where two participants played, was equipped with two PCs, two HMD devices, four controllers, and two base sensor stations, and the other room, where one participant played, was equipped with one PC, one HMD, two controllers, and two base sensor stations. Our experiment uses a client-server network. In other words, PCs with the HMDs are clients and communicate through a local server that a game engine provides over the game engine's network. So, network communication among VR participants begins with one HMD, goes through the connected PC, network router, and the VR virtual server of our game engine, and comes back to the network router and PCs connected to the other HMDs. Five teams of three participants played the ball-grabbing game. Each team played the ball-grabbing game in a hands-only mode to get used to the game. After 60 s of rest, participants were randomly assigned to play one of the two modes without knowing what it was and played the game in the assigned mode for 90 s. Participants rested for 60 s and filled out the questionnaire (Table 8) in seven minutes. Then, participants played the ball-grabbing game in the other mode for 90 s, rested, and filled out the questionnaire.

We obtain 45 observations each for the single-user and multi-user game environments. Our observations include the performance measurements, the rendering time per frame, game scores, and the user study of the IK method and our method.

6. VR User Experiments' Results

6.1. Single-User Archery Game Evaluation

We summarize the archery game's questionnaire results using our proposed method and an IK method in Figure 11. On a scale from one (very poor) to five (excellent), participants rate their experiences in the single-user VR game. We review the means of questionnaire results for each question. Figure 11 illustrates that the majority of the participants voted for our suggested method in Q1 (mean = 4.80), Q2 (mean = 4.86), Q3 (mean = 4.53), Q4 (mean = 4.73), Q5 (mean = 4.93), Q6 (mean = 4.80), and Q7 (mean = 4.93). Participants favored our method in all the questionnaires.

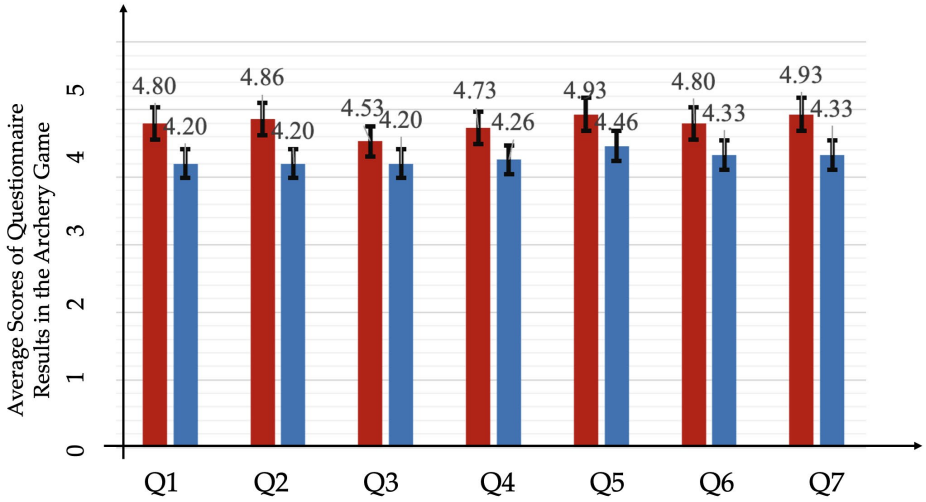

Figure 11. Questionnaire results in the archery game. ■ Our proposed method and ■ IK.

We compare the post-questionnaire results using a paired *t*-test with significant importance of 0.05 for either accepting or rejecting the null hypothesis. The paired *t*-test results with $p < 0.05$ indicate that our method significantly differs from an IK method. Table 9 is the statistical analytics of the archery game questionnaire under our method and the IK method. Table 9 illustrates that the difference between our method and IK in Q1 (t = 4.582, $p = 0.003$), Q2 (t = 3.567, $p = 0.003$), Q4 (t = 2.167, $p = 0.047$), Q5 (t = 2.449, $p = 0.028$), Q6 (t = 2.167, $p = 0.047$), and Q7 (t = 3.674, $p = 0.002$) are substantial. The paired *t*-test results show that the *p*-value of Q3 is high, so the difference between our method and IK for a sense of moving is insignificant.

Table 9. Statistical analysis of questionnaire of the archery game under our method and the IK method. * marks the questions with insignificant differences.

Question	Group (Mean)		Ours-IK	
	Ours	IK	t	*p*
Q1	4.80	4.20	4.582	0.003
Q2	4.86	4.20	3.567	0.003
Q3	4.53	4.20	1.434	0.173 *
Q4	4.73	4.26	2.167	0.047
Q5	1.06	1.46	2.449	0.028
Q6	4.80	4.38	2.167	0.047
Q7	4.93	4.33	3.674	0.002

The paired *t*-test reveals significant differences in the presence (Q1) and embodiment (Q2). This validates hypothesis H2 that our method of displaying the upper body in a single-user VR environment increases the feeling of presence more than the IK method. The *t*-test result of accuracy (Q4) and naturalness (Q6) tells that our method is significantly favorable with an IK method with $p < 0.05$ and confirms hypothesis H1 that our method generates natural-looking avatars in VR. The *t*-test result of the Q5 shows that our method has less delay with a significant importance of 0.028, and therefore confirms hypothesis H3 that our lower latency method enhances user experiences and presence. The *t*-test result of Q3 shows that our and IK's methods are similar. This result makes sense, considering that our experiments include the minimal moving of participants. Therefore, participants feel less sense of moving around in a virtual environment no matter what.

Furthermore, we compare the points participants earned after playing the archery game under two modes. The bar chart in Figure 12 illustrates all participants' average points for two modes. With our pose regression solution, participants obtained average points of 42.66 (85.32% of the highest possible points) and 35.73 (71.46% of the highest

possible points) using an IK solution. Participants obtained higher average points with our proposed pose regression method.

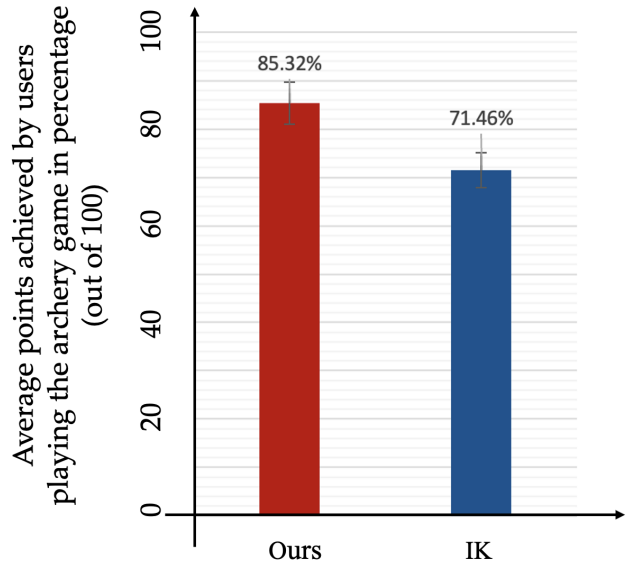

Figure 12. Average of total points in percentage in the archery game. Our proposed method and IK.

6.2. Multi-User Ball-Grabbing Game Evaluation

Figure 13 shows the questionnaire results of the multi-user VR experiences. Participants favor our pose regression model for the Q8 (mean = 4.66), Q9 (mean = 4.73), Q10 (mean = 4.53), and Q11 (mean = 4.73). Figure 13 illustrates that most participants prefer our model with Q12 (mean = 4.66) rather than IK.

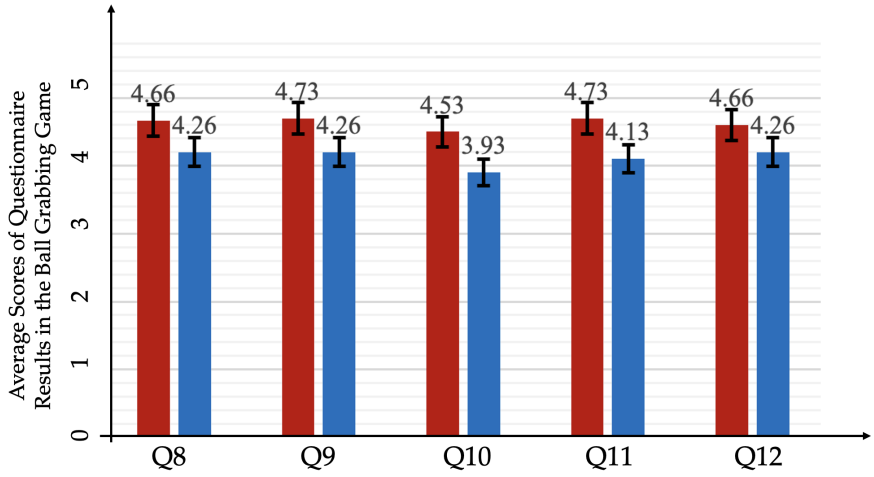

Figure 13. Questionnaire results in the ball-grabbing game. Our proposed method and IK.

For statistical analysis, we conducted a paired *t*-test with a *p*-value borderline of 0.05. Table 10 shows the paired *t*-test's statistical analytics under our and IK methods. The results illustrate that the differences between our method and IK are significant in Q8 ($t = 3.055$, $p < 0.01$), Q9 ($t = 3.500$, $p < 0.005$), Q10 ($t = 3.674$, $p < 0.005$), Q11 ($t = 2.304$, $p < 0.005$), and Q12 ($t = 2.449$, $p < 0.05$). The Q10 about accuracy of actions and Q11 about manipulation, with $p < 0.005$, support hypothesis H1 strongly that the avatar's motions are natural. Moreover, Q8 about presence and Q9 about ownership, with $p < 0.01$, proves hypothesis H2 that displaying the upper body in a multi-user VR environment increases the feeling of presence. The questionnaire about overall user satisfaction reveals that the

users preferred our proposed model over the IK method. A significant difference exists between our proposed method and IK ($p = 0.028 < 0.05$) for this satisfaction factor.

Table 10. Statistical analysis of the questionnaire after the ball-grabbing game.

Question	Group (Mean)		Ours-IK	
	Ours	IK	t	*p*
Q8	4.66	4.26	3.055	0.008
Q9	4.73	4.26	3.500	0.003
Q10	4.53	3.93	3.674	0.002
Q11	4.73	4.13	2.304	0.002
Q12	4.66	4.26	2.449	0.028

The five groups obtained different scores while playing the ball-grabbing game in two modes. The bar chart in Figure 14 illustrates the average number of balls grabbed. Participants scored an average of 19 points (95% of the highest possible points) using our pose regression method and 16.6 points (83% of the highest possible points) using an IK solution. These results confirm hypothesis H3 that our lower latency method enhances the participants' performance.

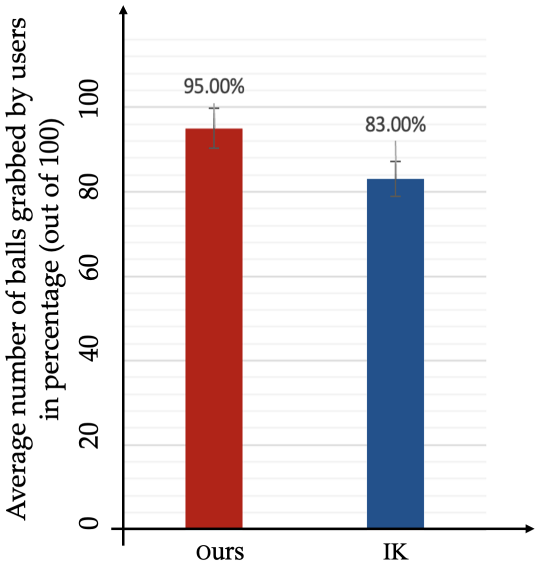

Figure 14. Average of total balls grabbed by the users in percentage in the ball-grabbing game. ■ Our proposed method and ■ IK.

7. Discussion

Our study investigates the effectiveness of a deep learning method to render the mimicked upper body avatar in VR that follows the motion of a user in the physical world. The proposed learning-based method predicts the location of the joints of the upper body, including the neck, pelvis, elbows, and shoulders, from the data of the HMD and two controllers. Our pose learning method produces reliable inference of the upper body joints, validated for test data in Section 3.4. Our method predicts the avatar's upper body given a VR user's HMD and controller data, as in Section 3.5. We compare our pose regression results with the IK results in Section 4 regarding visual accuracy and performance. Our resulting avatar looks visually similar to the avatar made from an IK method for the motion test that a user blocks, sways, and lifts. This visual resemblance tells us that our method may be a substitute for a widely used IK method in visualizing an upper body self-avatar of a VR user. Section 4 also compares our and the IK methods' performance evaluation. The computation time of our method is shorter than an IK method by approximately 60%. VR applications require 60–80 frames per second and should

guarantee the rendering time of each frame under a specific time limit to prevent possible motion sickness. Therefore, our low-latency method is desirable to avoid the potential decrease in the feeling of presence due to break-in-presence.

We conducted VR user experiments on single-user and multi-user VR applications in Section 5 and analyzed the user experiments' results in Section 6. We paid attention to reducing bias by randomly allocating the experimental units across the participant groups. These evaluations show that our result significantly differs from an IK result in most aspects, including presence, embodiment, manipulation, accuracy, delay, and naturalness. The sense of moving is the only factor whose difference is insignificant. Statistical analysis of these user evaluations reports that our hypotheses are accepted. We hypothesized in H1 and H2 that our method could be similar to an IK method but better. Our testing verifies that our method generates a natural avatar and yields self-embodiment similar to and even better than an IK method. In addition, H3 testing confirms that our lower latency enhances presence with less break-in presence. Our user evaluation supports that our method significantly differs from the IK method.

Surprisingly, our user evaluation confirms that the learning-based upper body is better than the IK upper body. The IK upper body is a typical method to visualize an avatar in VR corresponding to the pose of a user due to its ease of use and naturalness. Comparable to an IK, we tested our learning method for an upper body expecting that both ways produced similar results. The resulting statistical analysis of user evaluation shows that our learning method is better in most factors than an IK method. We guess that the lower latency of our method, as shown in Section 4, plays a crucial benefit because it minimizes the possible break-in presence and embodiment during VR experiences. In our multi-user VR experiments, the users are close to each other in the same building; therefore, they share the same network server. In practice, users communicate over the network worldwide during VR experiences, increasing latency. Our proposed pose estimation method is faster with lower latency than the IK method, which yields less interference with the feeling of presence, especially in multiuser VR over a worldwide network. We infer that being visually similar looking and with lower latency are the reasons for user preferences.

We aimed for a low-cost and high-performance method to estimate a user's upper body and animate the corresponding avatar in VR. Our method applies to any user with only essential VR equipment and therefore is low-cost compared to VR experiences with onsite motion capture equipment and operational staff. However, our learning method requires motion capture datasets acquired using an expensive capturing system. As such, our method is not low-cost in terms of acquiring datasets. VR developers may use the public human pose datasets or obtain datasets by renting a motion capture studio for a couple of days to minimize the cost of acquiring datasets. As for VR users, our method is low-cost and high-performance.

Our pose learning method is stable enough to apply to other datasets, including famous ones. These public human pose datasets mainly cover locomotion such as walking, running, kicking, climbing, etc. Moreover, we can extend our datasets, including the necessary ones, whenever a new VR application requires specific poses that are not typical in general VR applications. We made our pose datasets focus on the dynamic arm gestures in the limited space because most VR users stay in a small room and use their upper body and arms for interaction. We used rhythmic choreographies, but others may use sports action, game action, and other possible actions. Our datasets are a testbed to see that our proposed system works. Our proposed method can extend and replace the datasets if necessary.

8. Limitations

Still, our method has limitations in estimating large and weird postures, especially the poses not in our learning dataset. One impossible posture could be the twisted legs and arms observed in yoga. These postures are challenging to acquire even with real-time motion capturing due to the obstruction of the sensors from the cameras. Our proposed

learning method has limitations in predicting the accurate pose of these twisted poses. A lot of human pose datasets may overcome this limitation. If VR developers want to acquire lots of human poses, data acquisition becomes more expensive due to renting a costly motion capture system. VR developers may consider that our method is not low-cost, especially if they capture many poses to estimate even the weird poses. More public datasets of human poses should help reduce the efforts and cost of acquiring datasets.

In VR, the avatar with hand and finger movements is important for presence and embodiment. Researchers have investigated the problem of predicting the finger movements of the user in the physical world and rendering the corresponding hand in VR. To build the hand and finger motion datasets to learn, motion capture devices for the body are improper devices because of their inaccuracy for the small finger joints. Special devices, including the leap motion and wearable gloves, are necessary to capture finger motions. Since these devices have different capturing rates and timings with a motion capture system, combining the finger and body data requires challenging synchronization. Therefore, hand and finger motion studies have been developed separately from body motion studies for decades. Combining the body and hand motions is an interesting and open problem. In the future, we would like to extend our study to hand motion, which is crucial in gesture-based interactions, and the whole body, including the lower body, for better embodiment.

9. Conclusions

This work enhances the presence and embodiment that VR experiences display the mimicked upper body avatar of a user. To track a user's motion and visualize a corresponding avatar in VR, on-site motion capture equipment is the most accurate method. Motion capture equipment, however, requires 10 m by 10 m space and an operating engineer. Therefore, on-site motion capture equipment is a choice for theme parks and training centers with spacious areas and staff. An IK method is an alternative if VR experiences track and visualize only the upper body. An IK method is feasible because it has no minimum space and staff requirements. A learning-based method is a promising alternative to motion capture equipment for the upper body avatar, similar to an IK method.

According to the statistical analysis of user evaluation, our proposed learning model achieves significantly higher than the IK method. Performance evaluation of our learning method is better than the IK method, and it yields significant differences in user evaluation. In addition, our learning method has the potential to improve its accuracy by training the network with targeted datasets for specific VR experiences. Therefore, our learning method is a substitute for the IK method and a new method with endless potential for the upper body avatar in VR.

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/app13042460/s1.

Author Contributions: Conceptualization, K.P.; methodology, T.A. and K.P.; software, T.A.; validation, T.A. and K.P.; formal analysis, T.A. and K.P.; data curation, T.A., K.P. and G.K.; writing—original draft preparation, T.A.; writing—review and editing, K.P.; supervision, K.P.; funding acquisition, K.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Mid-Career Research Program through an NRF Grant Funded by the Korea MEST under Grant NRF-2021R1A2C1014210, and in part by the Chung-Ang University Young Scientist Scholarship in 2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Slater, M.; Wilbur, S.A. Framework for immersive virtual environment (FIVE): Speculations on the role of presence in virtual environments. *Presence Teleoper. Virtual Environ.* **1997**, *6*, 603–616. [CrossRef]
2. Kilteni, K.; Groten, R.; Slater, M. The sense of embodiment in virtual reality. *Presence Teleoper. Virtual Environ.* **2012**, *21*, 373–387. [CrossRef]
3. Jerald, J. *The VR Book: Human-Centered Design for Virtual Reality*, 1st ed.; Morgan & Claypool Publishers and ACM Books: San Rafael, CA, USA, 2015.
4. Parger, M.; Mueller, J.H.; Schmalstieg, D.; Steinberger, M. Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality. In Proceedings of the 24th ACM Symposium on VRST, Tokyo, Japan, 28 November–1 December 2018.
5. Khoshelham, K.; Elberink, S.O. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors* **2012**, *12*, 1437–1454. [CrossRef] [PubMed]
6. Yeung, L.F.; Cheng, K.C.; Fong, C.H.; Lee, W.C.; Tong, K.Y. Evaluation of the Microsoft Kinect as a clinical assessment tool of body sway. *Gait Posture* **2014**, *40*, 532–538. [CrossRef]
7. Olade, L.; Fleming, C.; Liang, H. BioMove: Biometric User Identification from Human Kinesiological Movements for Virtual Reality Systems. *Sensors* **2020**, *20*, 2944. [CrossRef]
8. Wolf, M.J.P.; Perron, B. *The Video Game Theory Reader*, 1st ed.; Routledge: New York, NY, USA, 2003; pp. 89–108. [CrossRef]
9. Roth, D.; Lugrin, J.; Büser, J.; Bente, G.; Fuhrmann, A.; Latoschik, M.E. A simplified inverse kinematic approach for embodied VR applications. In Proceedings of the IEEE Virtual Reality (VR), Greenville, SC, USA, 19–23 March 2019.
10. Botvinick, M.; Cohen, J. Rubber hands 'feel' touch that eyes see. *Nature* **1998**, *391*, 6669. [CrossRef]
11. Gall, D.; Roth, D.; Stauffert, J.P.; Zarges, J.; Latoschik, M.E. Embodiment in virtual reality intensifies emotional responses to virtual stimuli. *Front. Psychol.* **2021**, *12*, 674179. [CrossRef] [PubMed]
12. Slater, M.; Antley, A.; Davison, A.; Swapp, D.; Guger, C.; Barker, C.; Pistrang, N.; Sanchez-Vives, M.V. A Virtual Reprise of the Stanley Milgram Obedience Experiments. *PLoS ONE* **2006**, *1*, e39. [CrossRef] [PubMed]
13. Slater, M.; Pertaub, D.P.; Barker, C.; Clark, D.M. An Experimental Study on Fear of Public Speaking Using a Virtual Environment. *CyberPsychol. Behav.* **2006**, *9*, 627–633. [CrossRef] [PubMed]
14. Guadagno, R.E.; Blascovich, J.; Bailenson, J.N.; McCall, C. Virtual humans and persuasion: The effects of agency and behavioral realism. *Media Psychol.* **2007**, *10*, 1–22. [CrossRef]
15. Spanlang, B.; Normand, J.M.; Borland, D.; Kilteni, K.; Giannopoulos, E.; Pomés, A.; González-Franco, M.; Perez-Marcos, D.; Arroyo-Palacios, J.; Muncunill, X.N.; et al. How to Build an Embodiment Lab: Achieving Body Representation Illusions in Virtual Reality. *Front. Robot. AI* **2014**, *1*, 9. [CrossRef]
16. Spanlang, B.; Normand, J.M.; Giannopoulos, E.; Slater, M. A first person avatar system with haptic feedback. In Proceedings of the 17th ACM Symposium on VRST, Hong Kong, China, 22–24 November 2010. [CrossRef]
17. Lee, D.I.; Baek, K.Y.; Lee, J.H.; Lim, H. A Development of Virtual Reality Game utilizing Kinect, Oculus Rift and Smartphone. *Int. J. Appl. Eng. Res.* **2016**, *11*, 829–833.
18. Dong, Y.; Aristidou, A.; Shamir, A.; Mahler, M.; Jain, E. Adult2child: Motion Style Transfer using CycleGANs. In Proceedings of the on Motion, Interaction and Games, New York, NY, USA, 16–18 October 2020. [CrossRef]
19. Steed, A.; Frlston, S.; Lopez, M.M.; Drummond, J.; Pan, Y.; Swapp, D. An 'In the Wild' Experiment on Presence and Embodiment using Consumer Virtual Reality Equipment. *IEEE TVCG* **2016**, *22*, 1406–1414. [CrossRef] [PubMed]
20. Jiang, F.; Yang, X.; Feng, L. Real-time full-body motion reconstruction and recognition for off-the-shelf VR devices. In Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry(VRCAI), Zhuhai, China, 3–4 December 2016. [CrossRef]
21. Tan, Z.; Hu, Y.; Xu, K. Virtual Reality Based Immersive Telepresence System for Remote Conversation and Collaboration. In Proceedings of the International Workshop on Next Generation Computer Animation Techniques, Bournemouth, UK, 22–23 June 2017; pp. 234–247. [CrossRef]
22. Mahendran, S.; Ali, H.; Vidal, R. 3D Pose Regression Using Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 22–29 October 2017. [CrossRef]
23. Zhou, X.; Sun, X.; Zhang, W.; Liang, S.; Wei, Y. Deep kinematic pose regression. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–10 & 15–16 October 2016. [CrossRef]
24. Tekin, B.; Katircioglu, I.; Salzmann, M.; Lepetit, V.; Fua, P. Structured prediction of 3d human pose with deep neural networks. In Proceedings of the British Machine Vision Conference (BMVC), York, UK, 19–22 September 2016. [CrossRef]
25. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Shahbaz Khan, F.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
26. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
27. Yang, S.; Quan, Z.; Nie, M.; Yang, W. Transpose: Keypoint localization via transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021.

28. Lin, K.; Wang, L.; Liu, Z. End-to-end human pose and mesh reconstruction with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition (CVPR), Nashville, TN, USA, 19–25 June 2021.
29. Zheng, C.; Zhu, S.; Mendieta, M.; Yang, T.; Chen, C.; Ding, Z. 3D human pose estimation with spatial and Temporal Transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021.
30. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef] [PubMed]
31. Pishchulin, L.; Insafutdinov, E.; Tang, S.; Andres, B.; Andriluka, M.; Gehler, P.V.; Schiele, B. DeepCut: Joint Subset Partition and Labeling for Multi Person Pose Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
32. Fang, H.; Xie, S.; Tai, Y.; Lu, C. RMPE: Regional Multi-person Pose Estimation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
33. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014. [CrossRef]
34. Artemiadis, P.K.; Katsiaris, T.P.; Kyriakopoulos, K.J. A biomimetic approach to inverse kinematics for a redundant robot arm. *Auton. Robot.* **2010**, *29*, 293–308. [CrossRef]
35. Asfour, T.; Dillmann, R. Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2003. [CrossRef]
36. Mousas, C. Performance-Driven Dance Motion Control of a Virtual Partner Character. In Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Reutlingen, Germany, 18–22 March 2018. [CrossRef]
37. Carnegie-Mellon Motion Capture Database. Available online: http://mocap.cs.cmu.edu/ (accessed on 31 January 2023).
38. Holden, D.; Komura, T.; Saito, J. Phase-functioned neural networks for character control. *ACM ToG* **2017**, *36*, 1–13. [CrossRef]
39. Ofli, F.; Chaudhry, R.; Kurillo, G.; Vidal, R.; Bajcsy, R. Berkeley MHAD: A comprehensive Multimodal Human Action Database. In Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV), Clearwater Beach, FL, USA, 15–17 January 2013. [CrossRef]
40. Ben-Ari, M.; Mondada, F. Kinematics of a Robotic Manipulator. In *Elements of Robotics*; Springer: Cham, Switzerland, 2018; pp. 267–291. [CrossRef]
41. Kingma, D.P.; Ba, L.J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
42. RootMotion. Available online: http://root-motion.com (accessed on 31 January 2023).
43. Ionescu, C.; Papava, D.; Olaru, V.; Sminchisescu, C. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Trans. PAMI* **2014**, *36*, 1325–1339. [CrossRef] [PubMed]
44. Müller-Cajar, R.; Mukundan, R. Triangulation: A new algorithm for inverse kinematics. *Proc. Image Vis. Comput.* **2007**, 181–186. Available online: https://ir.canterbury.ac.nz/bitstream/handle/10092/743/12607089_ivcnz07.pdf;sequence=1 (accessed on 6 February 2023).