sensors

MDPI

Article

Fusion Poser: 3D Human Pose Estimation Using Sparse IMUs and Head Trackers in Real Time

Meejin Kim † **and Sukwon Lee** *,†

Korea Electronics Technology Institute, Seongnam-si 13509, Gyeonggi-do, Korea; mj_kim1023@keti.re.kr
* Correspondence: sukwonlee@keti.re.kr
† These authors contributed equally to this work.

Abstract: The motion capture method using sparse inertial sensors is an approach for solving the occlusion and economic problems in vision-based methods, which is suitable for virtual reality applications and works in complex environments. However, VR applications need to track the location of the user in real-world space, which is hard to obtain using only inertial sensors. In this paper, we present Fusion Poser, which combines the deep learning-based pose estimation and location tracking method with six inertial measurement units and a head tracking sensor that provides head-mounted displays. To estimate human poses, we propose a bidirectional recurrent neural network with a convolutional long short-term memory layer that achieves higher accuracy and stability by preserving spatio-temporal properties. To locate a user with real-world coordinates, our method integrates the results of an estimated joint pose with the pose of the tracker. To train the model, we gathered public motion capture datasets of synthesized IMU measurement data, as well as creating a real-world dataset. In the evaluation, our method showed higher accuracy and a more robust estimation performance, especially when the user adopted lower poses, such as a squat or a bow.

Keywords: IMU; human pose estimation; real time; motion reconstruction; sensor fusion; inertial sensors

Citation: Kim, M.; Lee, S. Fusion Poser: 3D Human Pose Estimation Using Sparse IMUs and Head Trackers in Real Time. *Sensors* **2022**, *22*, 4846. https://doi.org/10.3390/s22134846

Academic Editor: Angelo Maria Sabatini

Received: 29 April 2022
Accepted: 21 June 2022
Published: 27 June 2022

1. Introduction

Reconstructing human poses with a 3D skeleton-based body model and recording the associated motions, commonly called motion capture (mocap), has significant influences on computer vision, animation, robotics, and biomechanics. For example, many fields use captured motion to create better visual effects, such as in games and movies. It is also used to analyze a subject's movement for medical or military purposes. Nowadays, there are demands for motion capture in virtual reality (VR) applications to interact with virtual objects in real time.

Typically, commercial full-body motion capture systems are optical, such as Vicon [1] and OptiTrack [2]. When a user moves while wearing a suit that is covered with markers, the motion is estimated by analyzing the 3D positions of the markers, which are projected on multiple calibrated cameras. This marker-based system can restrict space and mobility, depending on the installation of the cameras. Frequently, dedicated studios are carefully designed using pre-installed cameras to prevent occlusions and lighting disturbances. Furthermore, vision-based methods are affected by reflections and illuminations, which make it difficult to record motion outdoors. Despite these constraints, marker-based methods achieve high-quality results. However, depending on the financial and computational costs and installation difficulties, vision researchers have focused on obtaining pose estimation using a few RGB [3–5] or RGB-D [6–8] images. These methods are much simpler but allow for limited space due to the camera. Another approach uses finely controlled moving cameras [9–11], but its usability is still limited.

Another motion capture system uses multiple inertial measurement units (IMUs) that estimate the orientation by combining a gyroscope, magnetometer, and accelerometer.

In contrast to optical systems, motion capture using wearable inertial sensors is less affected by environmental constraints and occlusion. In addition, it does not require a complex facility setup or expensive costs, making it suitable for individual users in VR.

Xsens [12] and Perception Neuron [13], which are representative commercial inertial motion capture systems, estimate joint parameters using 17 IMUs. The equipment requirements of these systems decrease accessibility and make them unsuitable for personal use. Recently, von Marcard et al. [14] presented a method for full-body pose estimation using six IMUs. However, this study requires offline optimization, which is computationally expensive. In addition, Deep Inertial Poser [15] was the first study to apply a deep learning method to 3D pose estimation using six IMUs in real time. It proposed a bidirectional recurrent neural network (BiRNN) model, which was trained using captured and synthesized datasets. However, this method cannot estimate the position of the full body, only the local orientation of the joints.

Head-mounted displays (HMDs), which are widely used in VR fields, can track the position and rotation of the user's head with respect to the environment. By combining the head position with data from IMU sensors, our method can improve the pose estimation quality because the head's height provides more information than just data from IMUs. In addition, the hip position can be estimated from the position of the HMD. Because of these points, our method can robustly estimate full-body poses and its position within the virtual space.

This paper proposes a human pose estimation method that uses inertial sensors and a VR HMD that provides the positional information. We had three challenges to overcome for our method to be suitable for VR: (1) it had to be operated in real time; (2) the errors in the joint angles or distances had to be minimized; and (3) it had to track the global position of the user at the same time. To this end, we introduced a network that combines biRNN and convolutional long short-term memory (convLSTM) [16] layers to deal with these challenges using six IMUs and a head tracker. This network addresses the following constraints: (1) unlike optimization approaches, the learning-based method needs a relatively shorter time to produce the prediction, which could meet the real-time constraints; (2) because of the continuity of human motion, future movements depend on the current and past movements, so the suggested network improves accuracy and continuity by learning spatio-temporal properties from the datasets (despite an insufficient number of sensors); (3) because of the accuracy, the human poses in the training dataset are presented with body-centric coordinates, which are described Section 3.2.1, but they only show the pose of the user, not the location or the direction in which the user is looking. Out method merges the head pose from the HMD with the predicted pose, which is local information, to recover global information. Furthermore, the hip velocity, which is one of the network outputs, makes the hip trajectory more continuous.

Our proposed network is divided into 2 phases: joint position estimation and joint rotation estimation, as referred to as P1 and P2 in Section 3.2.2, respectively. The joint rotation is predicted after the joint position because the rotation estimation network takes the predicted position as its input recursively. The IMU sensors measure the angular velocity, acceleration, and magnetic field and calculate the user's orientation using the sensor fusion algorithm. Our method takes the orientation and acceleration data from the IMUs as the inputs for the position estimation network. In contrast to other inertial-based estimation methods, our approach also uses head height data from the head tracker, which is measured as the distance from the ground to the height of the HMD. This measurement makes the prediction more robust since it removes ambiguity by providing global information; for example, when the placement of the sensor is on the lower leg, the orientation of the sensor is the same in standing and sitting postures.

We acquired human motion data using OptiTrack [2] and Xsens DOT [12] to create a dataset to train the model. To obtain more datasets, we synthesized full-body joint data and the orientation and acceleration of inertial measurements from open datasets, such as the CMU 3D motion dataset [17] and TotalCapture [18].

We evaluated our model using effectiveness comparisons for network model variances and input types and using comparisons to other works, such as DIP [15]. We used two evaluation data metrics that are widely used in pose estimation studies: the mean per joint position error (MPJPE) and the joint angle error. The evaluation results showed that the proposed model could obtain the global position with a higher pose estimation accuracy. We also implemented real-time applications to show that our model could be applied to VR.

2. Related Work

Our proposed pose estimation model uses sparse inertial sensors and a head tracking sensor to determine human movement and full-body posture. Motion capture methods can be classified according to their input parameters, including method that use multiple sensors. Here, we introduce a related work analysis of vision-based methods that use cameras or markers, methods that only use inertial sensors, and methods for integrating signals from the sensors.

2.1. Vision-Based Motion Capture Methods

Vision-based motion capture, which is the classic method for obtaining human motion, has been the focus of various studies throughout its long history. In particular, commercial motion capture uses a large number of markers and multiple calibrated cameras. Several studies [19–24] have made efforts to overcome the shortcomings of the popular approaches that use single or multiple cameras. Many methods that require high estimation accuracy are conducted offline [25–33]. Recently, real-time studies have also been proposed. VNect [4] is a representative study on 3D kinematic human pose estimation in real time (30 Hz), which combines fully convolutional neural networks. As in previous studies, deep learning techniques have significantly improved the pose estimation method. Since DeepPose [3] was proposed, which was the first major 2D human pose estimation study to apply deep neural networks (DNNs), convolutional networks (ConvNets) that are based markerless motion capture analysis [34–39] have been generalized. In addition, other studies have used multi-view images [40,41] or single depth images to obtain high accuracy [6–8]. In contrast to vision-based methods, our approach uses a system that can be installed without significant restrictions. In order to overcome the problems with vision-based methods, such as occlusion (depending on where the cameras are installed), we combined sensor systems that are not highly affected by direction.

2.2. Full-Body Sensor-Based Motion Capture Methods

Methods that use inertial sensors are another broadly used approach to commercial motion capture. Typically, Xsens MVN [42] conducts six degrees of freedom (DOF) full-body motion tracking using 17 IMU sensors that take measurements from a combination of accelerometers, gyroscopes, and magnetometers. Compared to vision-based methods, IMU motion capture is easier to use in out-of-lab situations as it reduces spatial constraints. However, the large number of inertial sensors that is required has the problem of high costs and being time-consuming to set up. Therefore, existing studies have tried using a small number of sensors, despite the performance degradation. Some studies [43,44] have constructed human poses using only five accelerometers by retrieving pre-recorded poses with similar accelerations from a motion capture database. In these studies, the measurement instability of the sensors and the size of the database excessively affected the performance of the method. Recently, research has been conducted on reducing the number of sensors by using inertial sensors that can measure acceleration and orientation simultaneously. A pioneering work in this field, Sparse Inertial Poser (SIP) [14], presented a joint optimization model that reconstructs the pose of SMPL body model [45] using six IMUs but without relying on databases. To advance SIP, Deep Inertial Poser (DIP) [15] adopted a deep learning method for running in real time. DIP uses a BiRNN [46] with LSTM cells [47]. This approach has the potential for real-time 3D pose estimation in VR environments, which provided us with great motivation. However, DIP cannot estimate the global movement

of the user, which is an imperative component of tracking motion. Yi et al. [48] proposed TransPose to estimate global translations by using a supporting foot-based method and an RNN-based method. TransPose achieves a state-of-the-art performance in terms of pose estimation accuracy using only six IMUs. Our proposed model estimates the position of each joint and uses the 3D position of the head (which is obtained from the head tracking sensor) to increase the accuracy of the motion tracking. In addition, by using the head position, our study achieved real-time and full-body human motion estimation within real-world space by obtaining human movements that play important roles within VR.

2.3. Performance Optimization Based on IMUs

Reconstructing human poses from sparse IMUs to a high degree of accuracy is a challenging problem because the data from the sensors are insufficient for configuring human poses. Many researchers have studied the sensor fusion method using inertial sensors along with other sensors or cameras to increase the estimation quality. Some studies [49,50] have applied six inertial and ultrasonic sensors to obtain 3D positions and orientations. Liu et al. [49] proposed a method for online pose estimation that retrieves data with similar signal configurations from pre-defined motion databases. Another approach is to combine inertial sensors with videos [51–53], especially multi-viewpoint videos (MVVs) [18,54–56], depth cameras [57,58] or optimal markers [59]. Total Capture [18] fuses MVVs with inertial measurement units and applies a convolutional neural network (CNN) output layer to an LSTM model. The use of the sensor fusion model with cameras significantly increases estimation accuracy but still includes several challenges, such as occlusion, lighting problems, installation complexity, and limitations in mobility. Our proposed model is another sensor fusion model, which combines the signals from inertial sensors with signals from a head tracker. The head tracker records the 3D position of the head, is consciously used in VR to track the location of the HMD, and has a positive effect on estimating the global position and full-body pose of a human in real-world space. Therefore, our method increases the accuracy of human pose estimation while using fewer sensors compared to existing studies and facilitates its application in virtual environments.

3. Method

The proposed method estimates the full-body pose and pelvis position of the user via an HMD and IMU bands. A biRNN network with a ConvLSTM [16] layer is then used to predict joint position and rotation, which takes the orientation and acceleration sequences from the IMUs as its input. In addition, we integrate the estimated joint pose with the head pose from the HMD to calculate the pelvis position. In this paper, we first introduce our approach in Section 3.1. We explain the structure of the network in Section 3.2. We then describe the method for reconstructing global positions from local information in Section 3.3.

3.1. An Overview

Our approach requires two types of sensors: IMUs and global head trackers. Six IMU sensors that are placed on pre-specified body parts are used to predict the joint position and orientation of the user. Moreover, it is an underdetermined method because the number of input data is relatively sparse compared to the number of joints that need to be estimated. Thus, we developed a pose estimation network (Section 3.2) that predicts full-body poses based on training data from the measurements of sparse IMUs. We defined body-centric coordinates (Section 3.2.1) that describe every joint in the frame that is located on the root joint for learning efficiency and consistency. Using the body-centric coordinates, we can remove all global information, such as the global position and orientation, except for the height of the head, which changes with every movement. When predicting the pose as an output, the removed global information needs to be recovered to place the user within the virtual space. To this end, our approach uses an additional sensor: a global head tracker that can be an HMD or a motion capture device. In addition, the pose estimation network

takes head height as one of its inputs. The head height removes any ambiguity that comes from the sparse IMUs. At the end of the procedure, the tracked head pose from the HMD is combined with the predicted human pose from the network by locating the head pose of the HMD. When two head poses are identified without other processes, there could be a foot sliding problem: the foot could move in the air because of the incompetence of the prediction. To solve this problem, we introduced the velocity term at the network output to constrain the movement of the root joint. As another solution, we could use the velocity of the IMU that was located on the root joint, but a drift occurred as time passed. Thus, the predicted root joint velocity is used by averaging it with the velocity from the HMD. Figure 1 illustrates the entire process of our method and the more detailed parts are explained in the next section with formal definitions.

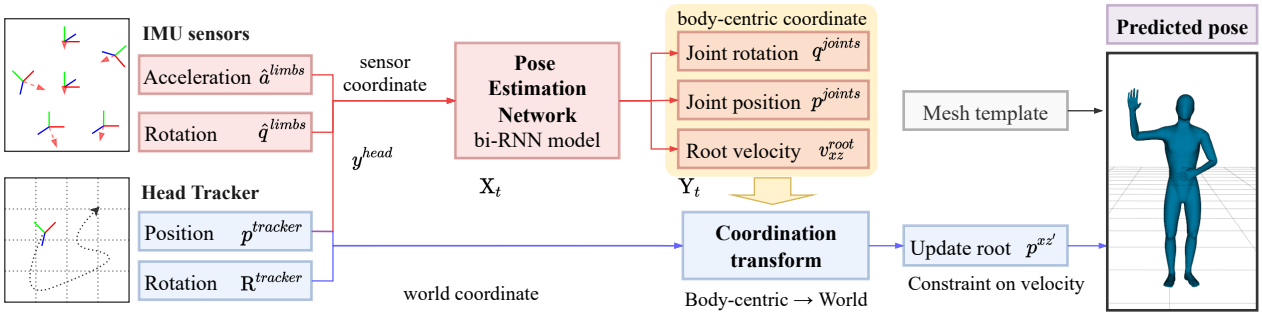

Figure 1. An overview of the proposed method. Our method uses two types of sensors to estimate human poses. IMU sensors are attached to each limb to measure their inertial data (orientation and acceleration), which are then input into the pose estimation network along with the head height to predict the user's joint position and rotation. By combining global head poses from the head tracker with the joint positions, our method can predict the global full-body pose of the user.

3.2. The Pose Estimation Network

We defined the input and output of the network at time t as X_t and Y_t, respectively. The input X_t is the sequence of the sensor measurement data x_f, where f is a neighbor frame of the time t. X_t and x_t are as follows:

$$X_t = [\, x_{f+t} \,|\, f = \{-15,\ -10,\ -7,\ -4,\ -2,\ -1,\ 0,\ 2,\ 4,\ 6\}\,], \tag{1}$$

$$x_t = [y^{head},\ \hat{a}^{limbs},\ \hat{q}^{limbs}] \in \mathbb{R}^{43}, \tag{2}$$

where $y^{head} \in \mathbb{R}$ is the height of the head (as measured by the tracker), $\hat{a}^{limbs}$ and $\hat{q}^{limbs}$ represent the acceleration and quaternion of the IMU on each *limbs*. For more clarity, each of the mathematical forms of $\hat{a}^{limbs}$ and $\hat{q}^{limbs}$ was defined as:

$$\hat{a}^{limbs} = [a_x^l, a_y^l, a_z^l] \in \mathbb{R}^{18} \quad \text{and} \quad \hat{q}^{limbs} = [q_x^l, q_y^l, q_z^l, q_w^l] \in \mathbb{R}^{24}, \tag{3}$$

where l is the list of limbs on which the IMU sensors are placed. In our experiment, the sensors were placed on the right hand, left hand, pelvis, head, right foot, and left foot, (cf. Figure 5). Next, we defined Y_t, which is the output of the network, as follows:

$$Y_t = [p^{joints},\ q^{joints},\ v_{xz}^{root}], \tag{4}$$

where p^{joints} is the concatenated positions of full-body joints in the motion data and, similarly, q^{joints} is the concatenated quaternions. Lastly, $v_{xz}^{root} \in \mathbb{R}^2$ is the velocity of the root joint with respect to the x,z plane.

3.2.1. The Body-Centric Coordinates

As mentioned above, Y_t is described in the body-centric coordinates as the frame that is placed on the root joint. Because the root joint describes every joint and only has a global

position and orientation, every joint loses its global information when the user moves and aligns the root joint with the origin. When the global data are not removed, Y_t can have different values, even when the user adopts the same poses in slightly different locations. We defined the local frame as follows:

$$\mathrm{T(t)} = [R_t^{root}(\theta_y); p_t^{root}]. \quad (5)$$

As the origin of the local frame is placed at the root joint position, p_{root}, the global translation can be removed. In addition, to remove the global orientation, the z-axis of the local frame $R^{root}(\theta_y)$ has the same direction as the z-axis of the root joint, which is the forward direction. Because the local frame is rotated about the y-axis, the rotation of the x- and z-axis can be preserved, which is essential for dealing with a more realistic pose; for example, a bowing or running pose requires the x-axis rotation of the root joint with respect to the real-world space. T depends on time t, so computing $\mathrm{T(t)}$ is described in Section 3.3.

3.2.2. Network Architecture

Using these definitions, we introduced a pose estimation network that predicts a single pose that corresponds to the input X at the current frame. Deriving high-accuracy poses from sparse acceleration and orientation data is a challenging task. Our proposed model focuses on two challenging solutions: (1) the naturalness of the motion and (2) the constraints of the human body structure. In a previous study on DIP [15], a biRNN [46] with LSTM [47] cells was proposed, which is suitable for use with time series learning to predict the SMPL pose parameters from IMU inputs. The biRNN model can access frames in two directions (past and future) and maintain the temporal and structural properties of motion in natural movements. Inspired by this work, we adopted the biRNN model pose reconstruction. However, the output of our method is directly composed of the joint position and rotation, which makes the usage of the output simpler, and the structural constraints are maintained well without SMPL. To reconstruct a pose, the rotational data q^{joints} must be assigned to every joint. Before estimating the rotation, the positional data p^{joints} are predicted using the measurement data from the $\hat{a}^{limbs}$ and $\hat{q}^{limbs}$ IMUs and the head height y^{head}, which then become the input for prediction of the rotation q^{joints}. Thus, the network architecture has a two-stage structure: the first stage infers the positional aspect of the pose; the second stage infers the rotational aspect by using the output of the first stage to predict the rotational pose of the IMU sensors.

We describe the architecture of the pose estimation network in Figure 2.

First of all, the input X_t is divided and rearranged according to its meaning, instead of the time sequence. As the result, the measurement data are reshaped into a four-dimensional matrix, of which each dimension is $(\#\text{frames} \times \#\text{IMUs} \times (\hat{a}, \hat{q}) \in \mathbb{R}^7 \times 1)$. After reshaping, the output is fed into the convolutional LSTM layer to perform spatio-temporal learning, which enables the network to learn the relationship between the sensor data and time more effectively. In addition, this method shows a good ability to preserve pose stability rather than using the measurement data directly. The output of the convolutional LSTM is concatenated with the current frame x_t and the sequence of the head height data y^{head}, which is then used as the input for the bidirectional LSTM layer. The biRNN [46] and long short-term memory (LSTM) [47] cells are used to compute the optimal weights through continuous sequence learning. Since our method uses a sparse number of IMU sensors in relation to the size of the human body, the sensors have to provide sufficient information to generate full-body poses. Although frame-to-frame changes are applied using the acceleration values, as mentioned in the DIP study [15], the acceleration has less of an influence on the predicted results than the orientation. Therefore, we use bidirectional LSTM layers because of the continuity of the motion. We also needed to consider future consequences and distinguish between actions that have the same orientation and acceleration values as IMUs. The output of the joint position estimation phase was defined Y_t^{P1} as:

$$\mathrm{Y}^{P1} = [p^{joints}, v_{xz}^{root}]. \tag{6}$$

In the second phase, the subnetwork $P2$ in Figure 2 predicts the joint rotation $\hat{q}^{joints}$ of the output Y_t based on two inputs: the rotation data from the q^{limbs} measurements and the results of the subnetwork Y^{P1}. The subnetwork $P2$ consists of unidirectional RNNs with LSTM cells. We determined that the joint position data that are estimated in $P1$ provide enough information to reconstruct the rotational information q^{joints}. Note that, unlike the subnetwork, $P1$ requires a sequence of the frames, whereas $P2$ depends on the current frame t.

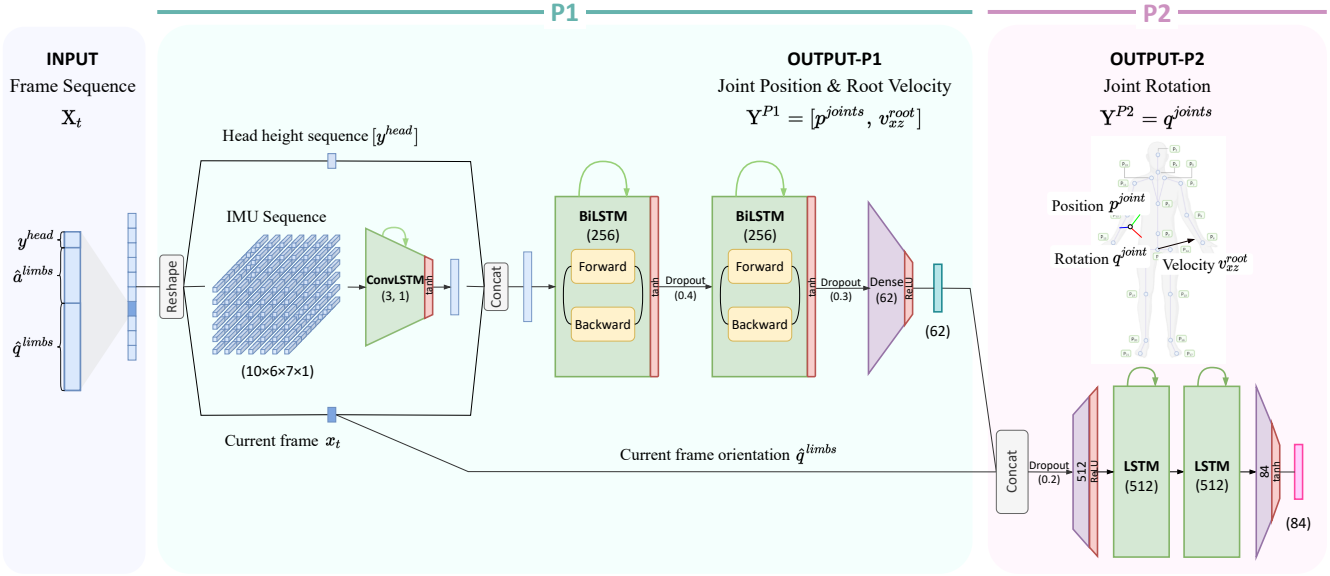

Figure 2. The Fusion Poser network architecture. The model inputs are the sequence of IMU sensor data and the height of the head. The length of the sequence is 10 time intervals, each of which has 43 features. The network consists of two stages: Phase 1 (P1) predicts the joint position using the IMU data and the head height sequences with the biLSTM layers, followed by the 4D convLSTM layers; Phase 2 predicts the joint orientation at the current time t using the output of P1 and the head height sequences with the LSTM layers.

3.3. Reconstructing Global Poses

Because the network output Y_t is with respect to the body-centric coordinates, the HMD position is combined with the output to reconstruct the root trajectory. To this end, the local frame $T(t)$ (Equation (5)) needs to be computed, but we could not obtain exact values for $[R_t^{root}$ and p_t^{root} because the positions of the sensors differ every time the user wears them. We introduced the calibration step to complete the parameters of $T(t)$, during which the user aligns the directions of the head and root; for example, the A-pose or T-pose. Firstly, we assumed that the z-axis direction of the tracker would coincide with the facing direction, the q_{head}^{w} of the IMU sensor would be represented as real-world coordinates, and that it can be easily satisfied by stacking the head tracker with an IMU sensor. Using this setup, the rotation aspect at calibration time c can be calculated as follows:

$$\phi_y^* \leftarrow \arg\min_{\phi_y} ||R_c^{tracker}\mathbf{z} - R(\phi_y)\hat{R}_c^{head}\mathbf{z}||^2 \tag{7}$$

where $R^{tracker}$ and $\hat{R}^{head}$ are the rotation matrices of the tracker and IMU sensor, respectively, and $\mathbf{z}$ is a unit vector $[0, 0, 1]$. After $R(\phi_y^*)$ is multiplied by $\hat{R}^{head}$, the projection of the transformed $\mathbf{z}$ has the same direction as one of the trackers. The rotation matrix of the root at time t can then be calculated by applying the results of Equation (7) to the measurement data from the IMU sensors:

$$R_t^{root} = \hat{R}_t^{root} \cdot (\hat{R}_c^{root})^{-1} \cdot R(\phi_y^*) \cdot \hat{R}_c^{root} \tag{8}$$

In this equation, $\hat{R}_c^{root}$ is the rotation matrix of the measurement data from the root joint at the calibration time c and thus, $\hat{R}_c^{root}$ and θ_y^* are stored to obtain R_t^{root}. To predict the rotational aspect of the transformation T, only the angle that rotates about the y-axis is needed, which can be solved in a similar way to Equation (7):

$$\theta_y^* \leftarrow \arg\min_{\theta_y} ||R_t^{root}\mathbf{z} - R(\theta_y)\mathbf{z}||^2 \tag{9}$$

After computing the rotational aspect of parameter of T, p_{root} can be obtained by applying the position of the head from Y_t (see Figure 3):

$$p_t^{root} = p_t^{tracker} - R(\theta_y^*)p_t^{head} \tag{10}$$

Because the root position follows the position of the head, the quality of the root trajectory depends on the quality of the estimation. However, the noise in the estimation cannot be removed. As a result, the root trajectory shows an unwanted jerk that lowers the motion quality. To solve this problem, we introduced the velocity term v_{xz}^{root} to the output Y_t, which constrains the velocity of the root joint using a simple weighted average:

$$p_t^{xz'} = p_{t-1}^{xz} + \alpha(p_t^{xz} - p_{t-1}^{xz}) + (1-\alpha)v_{xz}^{root} \tag{11}$$

For simplicity, the superscript of p is omitted in the above equation, which is related to the root joint. In our experiments, the value of 0.1 for α worked well, which meant that the results depended more on the prediction.

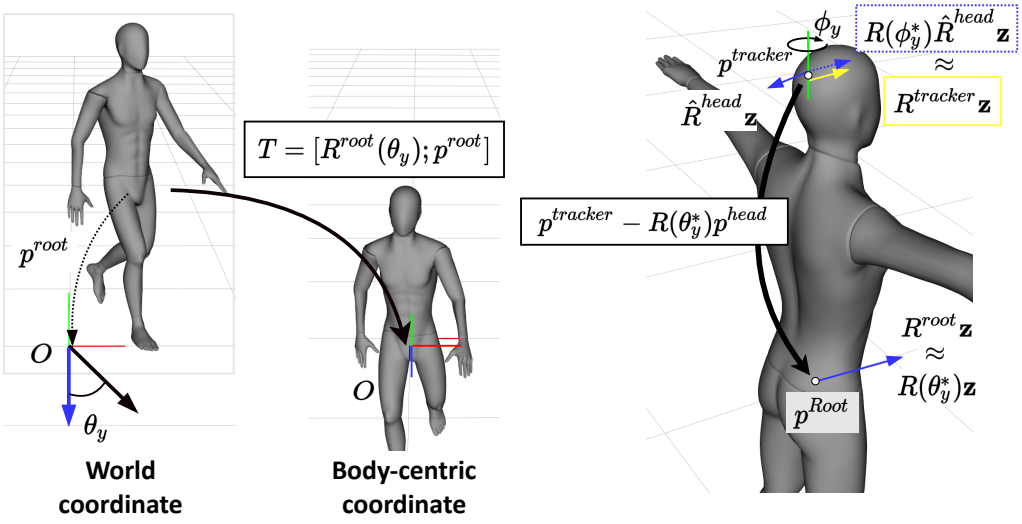

Figure 3. The coordination transformation: (**Left**) for training, we removed global information by transforming the real-world coordinates into the body-centric coordinates that identified the position of the frame at the root joint and the rotation about the y-axis was the direction of the root joint aligning to the z-axis; (**Right**) after the prediction, the global information had to be recovered, so the rotation ϕ_y^* was computed by matching the tracker's direction with the IMU that was attached to the head. With rotation $R(\theta_y^*)$, the position of the root joint could be computed from the head position using our network.

4. Datasets

This section describes the configuration of the datasets that were used for the model implementation in more detail. We introduce the skeleton structure that constructs the human pose data in Section 4.1, the detailed instructions for the motion capture and IMU

data in Section 4.2, the method for IMU calibration in Section 4.3, and the synthetic data in Section 4.4.

4.1. Skeleton Structure

The joint position p^{joints} provides the position of the human body joints using the body-centric coordinates. Figure 4 depicts a skeleton that consists of 21 joints (p^1 to p^{21}). The height of the avatar configuration that is set during motion capture is the size of the skeleton structure, within which the joint positions are determined in centimeters (cm). The joint placement of our skeleton structure was based on the full-body motion capture data that we collected.

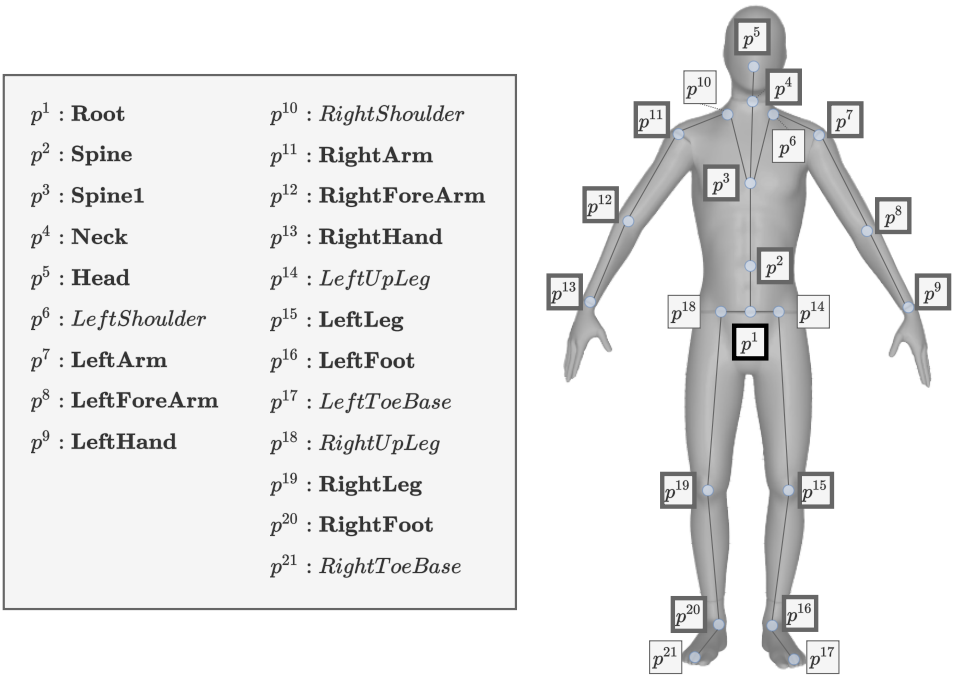

Figure 4. Skeleton structure: the skeleton for our study had 21 joints (p^1 to p^{21}). The 15 joints that were used to evaluate the errors in the experiments are highlighted in bold and thickly lined boxes. The joint list for evaluating the errors was as follows: Hip, Spine, Spine1, Neck, Head, LeftArm, LeftForeArm, LeftHand, RightArm, RightForeArm, RightHand, LeftLeg, LeftFoot, RightLeg, and RightFoot.

4.2. Motion Capture and IMUs

We utilize two types of sensor data for our data-driven model: motion capture data and IMU data. In the experiments, we recorded raw data from a subject who was wearing an OptiTrack [2] motion suit with 50 markers and 6 Xsens IMU sensors. As shown in Figure 5, six IMUs were mounted on the pelvis, the left and right hands, the left and right legs, and the head. The subject executed the calibration steps for the optical markers on the suit. After calibration, the participant performed actions following pre-defined scripts, such as locomotion, sitting, crawling, and other motions. The ground-truth motion capture data were recorded in 120 Hz and IMU data were recorded in 30 Hz, so data synchronization problems could occur due to the frequency differences. Therefore, we applied linear interpolation according to the timestamps of the two sets of data.

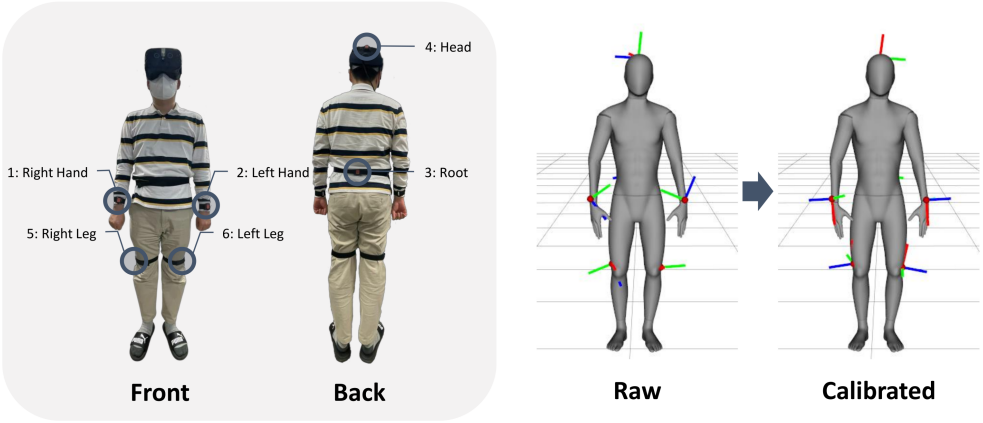

Figure 5. The IMU placement and calibration: (**Left**) the placement of the six IMUs that were attached to the body: right hand, left hand, pelvis (root), head, right leg, and left leg; (**Right**) the results of the IMU sensor calibration to the body-centric coordinates.

4.3. Sensor Calibration

We used Xsens DOT IMU sensors, which contain 3-axis accelerometers, gyroscopes, and magnetometers. The measurement of the IMUs was represented using the local coordinate system, which was defined as right-handed Cartesian coordinates, and thus, each IMU had different coordinates. To obtain the identified coordinate system between the sensors, we used the heading reset function on the IMU sensors that aligns magnetic north with the forward direction of the physical body. After the calibration step, we could obtain the measurement data from the six IMUs in terms of the inertial coordinate system. For training, we converted the measurement data into the body-centric coordinate system, which is described in Section 3.2.1. On the other hand, we used the inertial coordinate system for the predictions.

4.4. Generating Synthetic Data

To perform the predictions, the network requires a large amount of data, but when only the data from the motion capture are used, the cost of the data is unaffordable. To this end, many works [15,48] have generated synthetic data from existing motion capture datasets by simulating the measurement data from the IMUs, which is the method that we adopted to carry out the predictions. We generated synthetic data using the CMU 3D motion dataset [17] and TotalCapture [18] and uncommon behavior motions were excluded, such as sport, dance, and martial arts. To simulate the measurement data from the IMUs, we used the following steps: (1) we retargeted the motions from datasets onto our skeleton (Section 4.1) for consistency; (2) we placed virtual IMUs on the skeleton where the physical IMU sensors were placed; (3) lastly, we calculated the orientation $\hat{q}$ and acceleration $\hat{a}$ by synthesizing the motions of the virtual IMUs followed by smoothing with the B-spline curve to obtain the motion trajectory.

5. Experiments

Before this proposal, we conducted experiments to carry out a quantitative and qualitative evaluation of our pose estimation model. This section summarizes our experiments. First, we introduce the data and metrics that we used for the experiments in Section 5.1. We evaluate and compare the performance of our pose prediction model using real-time settings in Section 5.2. In Section 5.3, we introduce the implementation of a real-time application using our pose estimation network and global location tracking method. Finally, we describe the hardware settings for our work in Section 5.4.

5.1. Data and Metrics

The experiments were conducted using the real TotalCapture [18] dataset. The validation dataset consisted of 10 consecutive input frame sequences that were not used for training. We used the mean per joint position error (MPJPE), joint angle error, and location tracking error as the metrics for the quantitative evaluation. First, we calculated the mean value of the Euclidean distance between the expected position and the obtained position for 15 major joints that were representing a real-time pose. Then, we calculated the mean error per joint from the difference in degrees between the predicted and the actual movement and the difference in root distance between the reconstructed global position and the recorded position in real-world space.

5.2. Evaluation

5.2.1. Quantitative Evaluation

We evaluated the following variants to identify the model configuration that produced the best performance: (1) estimations with different components at the input and (2) estimations from reconstructing the network architecture.

Measuring the errors for comparison. To measure the positional errors of DIP and TransPose, we reconstructed an SMPL mesh model and the joint positions from the outputs of these works, which provided the SMPL parameters. To compare those results to ours, we used the same joints to measure the errors, as defined in Figure 4.

Influence of input components. To compare the effectiveness of the different input types, we experimented with three different types of networks: one network only used the current time t (only current), one network did not use the head height (non-head), and the other network did not use the acceleration of the IMUs (non-acc). Table 1 shows the results of these experiments. As the table shows, the positional error of the "only current" network was higher than that of "Ours", which used ten sequences. Moreover, when measuring the lower body error, the mean error of "Ours" was 49.18 mm ($\pm$29.50 mm) and that of the "only current" was 52.65 mm ($\pm$29.16 mm). This indicated that using past-to-future information led to an improved pose estimation performance. Furthermore, as shown in Figure 6, the model without head height data showed a significantly higher position error 55.45 mm ($\pm$27.19 mm). This result was because this model could not track extreme changes in full-body poses, such as bending over or sitting on the floor (cf. Figure 7). It can also be seen that the acceleration data from the inertial sensors improved the joint position estimation accuracy by including the relative position differences in the input layer. Therefore, we identified a solution for estimating a wider range of poses than previous studies by adding head height data to the input using a head tracker.

Table 1. The evaluation of our pose estimation network using different input variables and network architecture variants with the TotalCapture [18] dataset. The errors of each model are described as the mean (μ_{pos}) and standard deviation (σ_{pos}) of the joint position error in millimeters and the mean (μ_{ang}) and standard deviation (σ_{ang}) of the joint angle error in degrees (°).

	μ_{pos} (mm)	σ_{pos} (mm)	μ_{ang} (°)	σ_{ang} (°)
Ours	**50.51**	20.07	11.31	4.58
DIP	79.42	32.15	13.67	9.59
TransPose	68.51	41.43	12.93	6.15
FC (512)	53.54	21.58	**10.43**	4.36
Conv+RNN	53.38	22.08	11.13	4.72
biRNN	52.60	21.19	10.99	4.44
only current	52.55	21.44	11.51	4.56
non-head	55.45	27.19	11.31	4.87
non-acc	56.74	20.47	11.61	5.06

Influence of network architecture. The proposed network consists of a convolutional LSTM layer and bidirectional LSTM layers. These RNN layers were added to extend the capability of predicting more accurate full-body poses. In this evaluation, we compared three network variants: (1) a network consisting of a fully connected layer (FC); (2) a network using a unidirectional RNN layer, not bidirectional (Conv+RNN); and (3) a network with no convolution layers (biRNN). Figure 6 shows the influence of the network structure on positional errors using the validation dataset. The distributions of positional errors using the three tested cases indicated a higher error frequency than that using our network structure. Further, Table 1 shows that the joint angle errors in these cases had slightly lower values. However, the experimental results showed that our configuration would be more suitable for real-time application as it showed a higher performance for continuous frames.

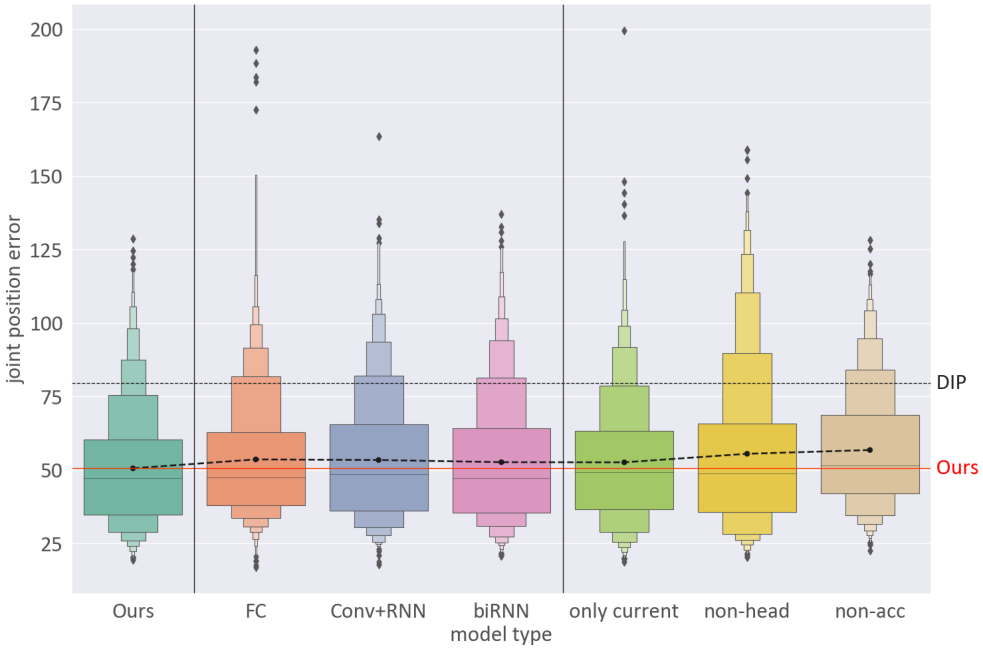

Figure 6. An evaluation of the mean per joint position error. The graph shown here is based on Table 1. We recorded the errors in each frame and showed that our network achieved a better performance than the other variants and DIP [15].

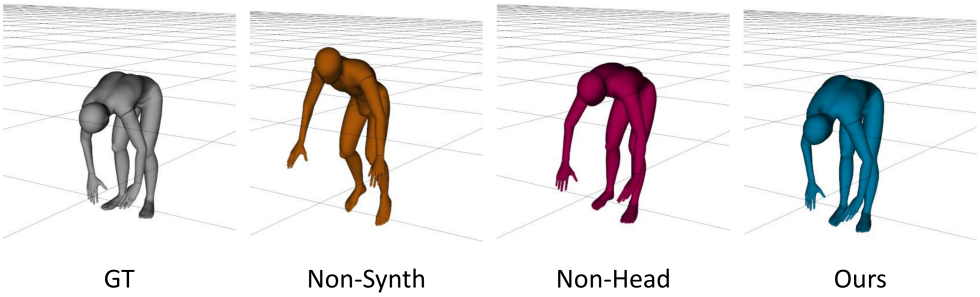

Figure 7. A comparison of the 3D model results. The figure shows the ground-truth and the real-time results of three types of models: the model that was trained excluding synthetic data (Non-Synth); the model that was trained excluding head height data (Non-head); and our model.

Location tracking error. Figure 8 shows the root trajectory that was estimated using our method and that of the ground-truth. The mean location error was 9.4148 cm (±7.9058). For the evaluation, we used the walking motions of Subject 4 in the TotalCapture dataset because it was suitable for demonstrating the root trajectory. Figure 9 shows the discrepancy

between the full-body poses by overlaying the ground-truth and reconstructed results. It shows that the reconstructed path achieved the intended result with high accuracy.

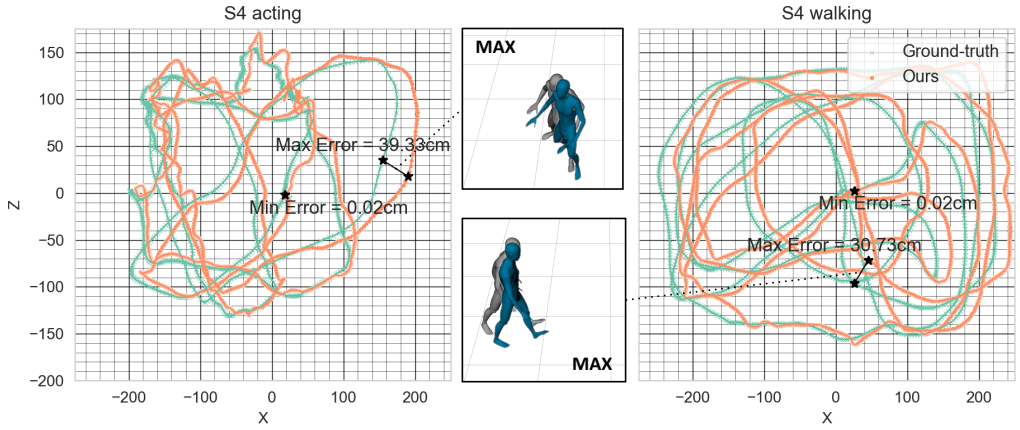

Figure 8. The trajectory of S4 (Subject 4) from the TotalCapture dataset. The overlapping frames in the middle of figure show the maximum value of the location tracking error in each motion sequence.

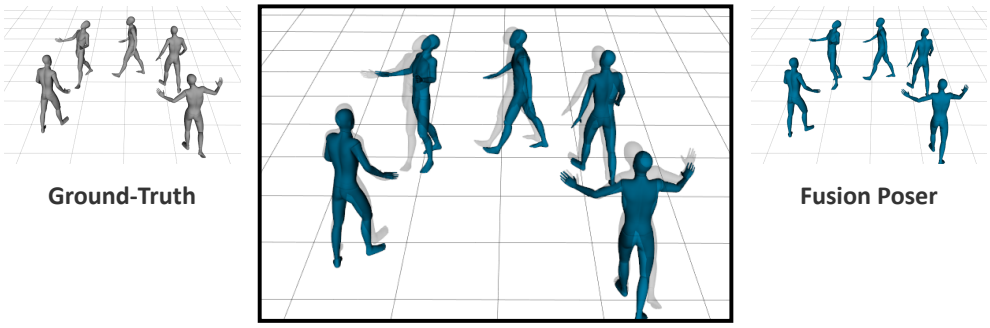

Figure 9. A comparison between the predicted and ground-truth poses using TotalCapture data (S4). By overlaying the predicted and ground-truth results, we could visualize the differences.

5.2.2. Qualitative Evaluation

As described in Section 5.2.1, we experimentally determined the best performance configuration. This section presents the poses that were estimated using our network as 3D body models and compares these poses to the ground-truth. We provided this comparison using our mocap dataset and the TotalCapture dataset [18]. In this section, we also describe the differences between the poses in terms of the evaluation variants that were difficult to express numerically.

Figure 7 shows the differences between two models that were trained in different ways. The first model was trained without synthetic data and the second model was the network variant that did not use the head height data. This figure shows that the proposed approach was more promising than the other variants. Moreover, in full-body exercises such as stretching or jumping, the height of the head provides reliable information for distinguishing poses and can be easily acquired from HMDs.

Comparison to ground-truth poses. Figure 10 shows some example prediction results using different poses from our mocap dataset and the TotalCapture dataset [18]. The ground-truth (GT) pose on the left was captured using a large number of optical markers from both datasets and the pose on the right was estimated with our method using six IMUs and one head tracker. Although there were challenging issues, such as hand pose (as detailed in Section 6.1), we could use our method for human pose estimation in real time.

Comparison to previous works. Figure 11 shows a visual comparison between the online pose estimation results of our method and those of other state-of-the-art works. The figure shows a qualitative comparison between our predicted poses and the SMPL poses that

were estimated using DIP and TransPose for example frames from the TotalCapture dataset. Our approach estimated more accurate results for relative joint positions in the upper and lower body.

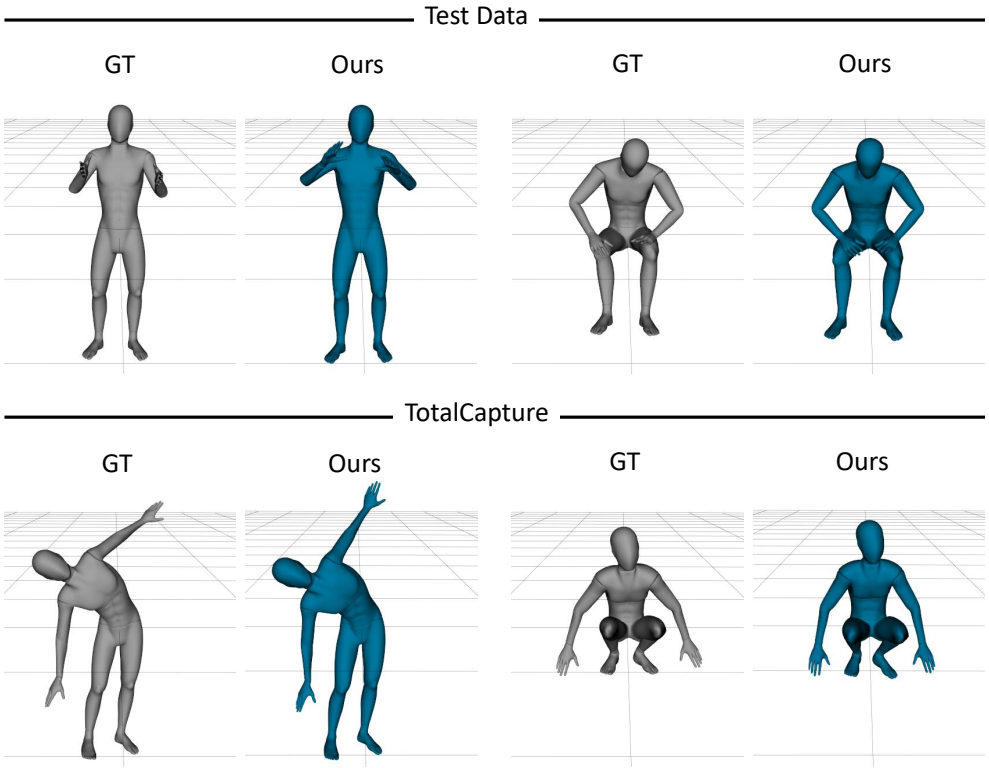

Figure 10. Example results using our mocap data and TotalCapture data.

5.3. Real-Time Application

In this study, we implemented a real-time application using a VR HMD, as described in Figure 12. Our application received the measurement data from six IMU sensors and continuously estimated the user's pose. At the same time, the pose of the head served as the input and global information for reconstructing the root trajectory. Our real-time application ran in the Unity 3D environment. Since the biLSTM layer required future data from the predicted time t, we stacked 10 sequences before the prediction, causing the application to have a delay of around 0.3 s. Because of the delay, there could be a discrepancy between the head position and the predicted pose when the user moved relatively fast.

5.4. Hardware Configurations

We trained our model using an Intel(R) Core(TM) i9-10900K CPU and an NVIDIA RTX 3090 graphics card. The real-time application ran on another PC with an Intel i5-10600 CPU and an NVIDIA RTX 2060 graphics card. We used Xsens [12] DOT IMU sensors to record the IMU data, both for training and real-time data. The ground-truth motion data were captured using an OptiTrack [2] Prime Camera and a motion suit with 50 markers. In addition, we used the Antilatency [60] tracking system to track the head position in real-world space.

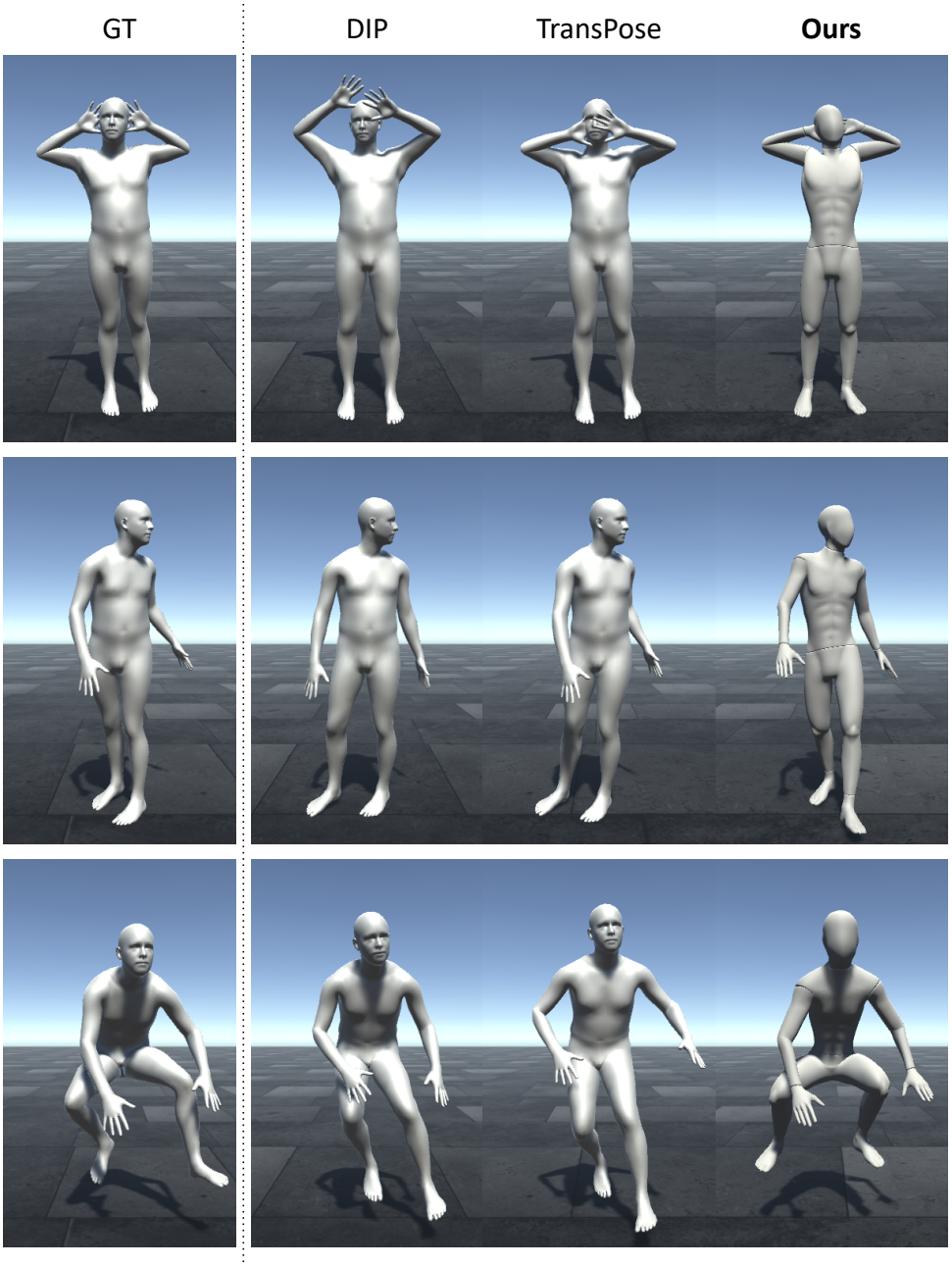

Figure 11. A qualitative comparison between the online pose estimation results of our method and those of previous works: the first column shows the ground-truth of the selected frame from the TotalCapture dataset, then the reconstructed SMPL poses from the estimation results of DIP are in the second column and those of TransPose are in the third column.

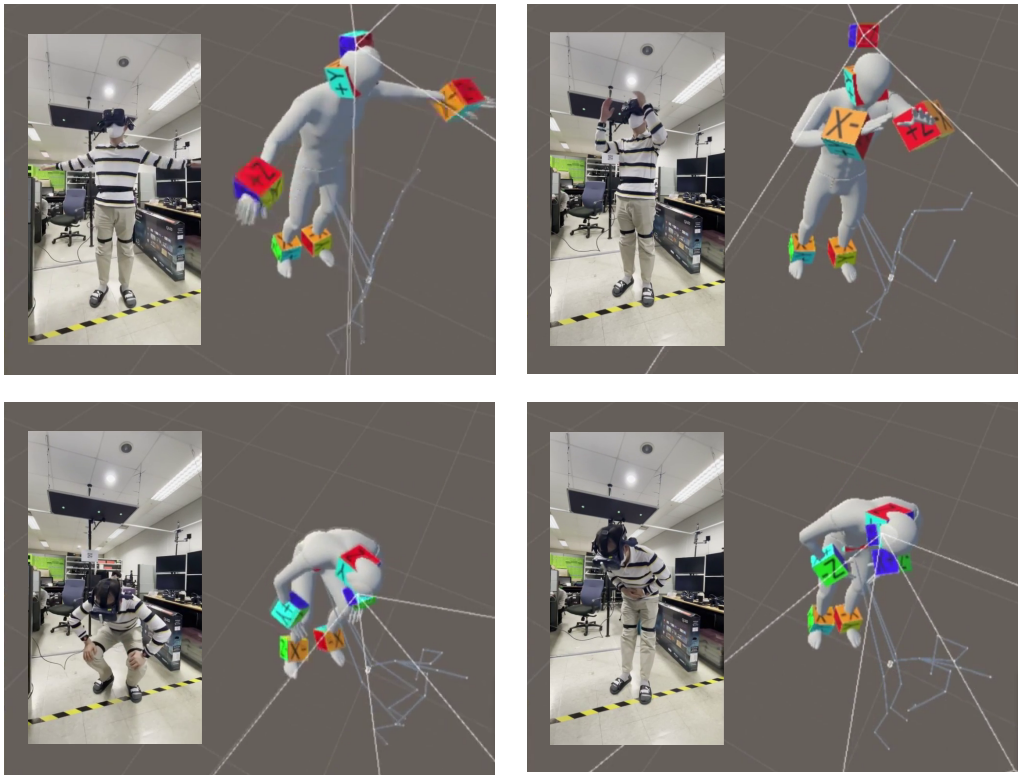

Figure 12. The real-time application using sample frames in Unity 3D. Our implemented application took the HMD sensor as input and reconstructed the 3D modeling body after a slight delay.

6. Discussion

This paper introduced a pose estimation method that uses six inertial sensors and a head tracker to reconstruct human poses and global body positions in real time. Our model has the following novelties: (1) an improvement in human pose estimation by adding head position data; (2) the provision of a reliable global position, which is essential for VR applications; and (3) the acquisition of a higher accuracy for pose estimation by combining spatio-temporal layers and body-centric coordinates. We showed better quantitative results using the head position data and model configuration (cf. Section 5.2). Moreover, although we adopted an economically efficient type of sensor, the method had fewer restrictions on action and mobility. Nevertheless, the noise accumulation problem over time when using IMUs and some other limitations remain a challenge (Figure 13).

6.1. Limitations

The motion capture dataset that was collected to train our model included various actions, but it could not respond to extreme changes in the position of the pelvis, for example, when crawling or lying down. When the waist and the floor were parallel, as shown in Figure 14, the predicted pose and the body rotation were not similar to the ground-truth. We posit that the pelvis rotation caused the pose errors as the IMU data that were used for training were transformed into body-centric coordinates.

It is also challenging to determine hand poses when using a small amount of IMU data. In this paper, the only data that could determine the hand poses were from a pair of sensors that were worn on the wrists, but these were insufficient data to track the wrist rotation. The right-hand side of Figure 15 shows an example of different hand poses for which the wrist rotation was not predicted correctly during the motion of putting the hands together.

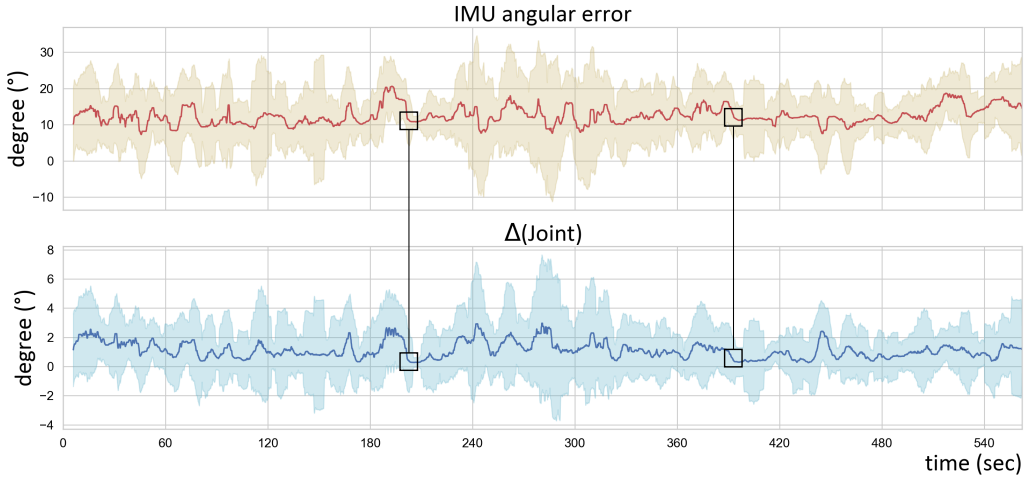

Figure 13. The time-dependent changes in the angular errors of the inertial sensors and the angles of the joints to which the sensors are attached. The errors represent the angle differences between the IMU measurement/synthesized data and the motion capture data within same frame. The errors increased with the continual movement of the sensors and returned to the initial error value when the movement stopped.

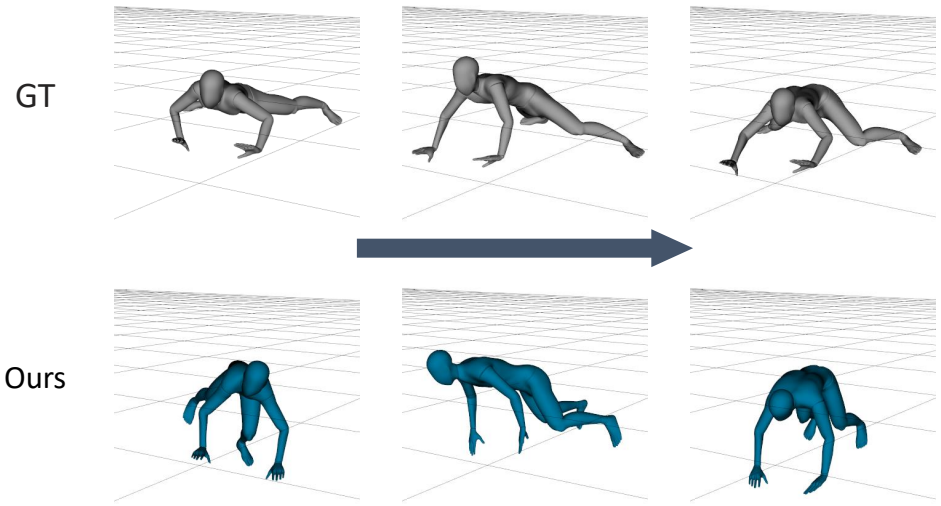

Figure 14. Examples of failure cases for the crawling motion in the TotalCapture dataset: (**top**) the ground-truth motion; (**bottom**) our predicted motion. It was a continuous motion with 15 frame intervals.

We extended the dataset by adding synthetic data to our mocap data to improve the performance of our network (cf. Figure 7). We built a 3D body model to simulate the measurement data from the IMUs and generate the output Y. We manually retargeted all of the datasets for prediction accuracy, but the retarget process was carried out per subject, which required a small amount of labor compared to what would be required for a whole dataset. However, we observed that the distortion of the retargeted body was due to the limitations of the retargeting method. For example, on the left-hand of Figure 15, the problem of the shape twisting according to the movement of the body can be seen. Although this distortion was seen in a low proportion in the overall data, it could be analyzed as the cause of low accuracy for specific poses.

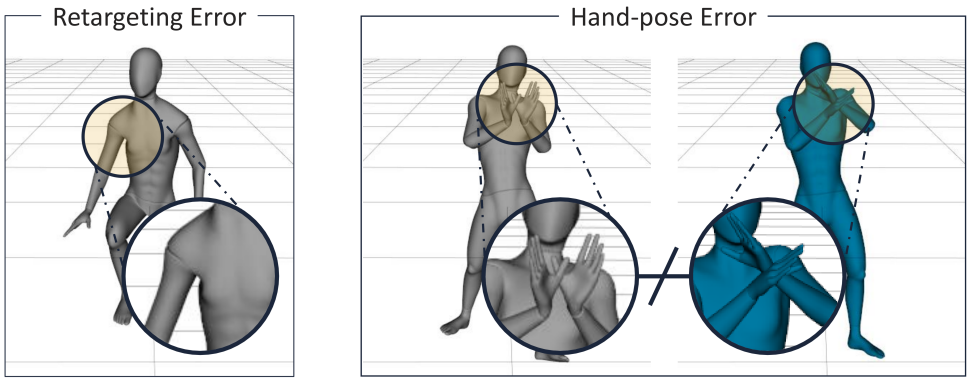

Figure 15. Examples of failure cases for retargeting the body and estimating the hand pose: (**Left**) retargeting error; (**Right**) incorrectly estimated hand pose. This is an example of an incorrect prediction during the motion of putting the hands together.

7. Conclusions

In this paper, we introduced Fusion Poser, which estimates the pose of a user who is wearing six IMUs and translates the world coordinates of a head tracker in real time. The orientation and acceleration of the inertial sensors and the head height data are used as network inputs to estimate joint position, joint rotation, and root velocity. Our network architecture mainly adopts biLSTM layers to maintain the spatio-temporal relationship between the joints. The convLSTM layer is applied to the IMU sequence before the biLSTM layer to improve the prediction quality. In addition, the LSTM layer shows higher accuracy for estimating the orientation of a joint than the fully connected layer, as shown in Table 1. This method requires a large dataset to train the proposed network, which is cost-intensive to gather using motion capture. For cost-effectiveness, synthesized data can also be used by simulating the measurement data of virtual IMUs and models from open datasets, such as CMU and TotalCapture. For pre-processing, the coordinates of the output Y are converted into body-centric coordinates, which enables effective learning by removing global information. For the estimations, the translation and orientation of the root joint are recovered using the head tracker. In our experiments using the TotalCapture dataset, our method achieved a mean per joint position error of about 50 mm and a mean per joint angle error of about 11.31°, which was a better performance than those of the compared works [15]. Our approach requires a head sensor to track the pose, but this is commonly implemented using HMDs.

Author Contributions: Conceptualization, M.K. and S.L.; funding acquisition, S.L.; investigation, M.K.; methodology, M.K. and S.L.; validation, M.K.; formal analysis, M.K. and S.L.; data curation, M.K. and S.L.; software, M.K. and S.L.; writing—original draft preparation, M.K.; writing—review and editing, S.L.; visualization, M.K.; supervision and project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. Written informed consent was obtained from the patient(s) in order to publish this paper.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://github.com/LuzyCat/FusionPoser (accessed on 20 June 2022). This paper used third party data to train and evaluate the model. Restrictions apply to the availability of these data. The training data were obtained from the CMU Graphics Lab Motion Capture Database and are openly available from http://mocap.cs.cmu.edu/ (accessed on 14 April 2022). The training data were also obtained from the TotalCapture dataset and are available from https://cvssp.org/data/totalcapture/data/ (accessed on 14 April 2022) with the permission of Andrew Gilbert.

Acknowledgments: This work was supported by the Korean Evaluation Institute of Industrial Technology (KEIT), which is funded by the Korean government (MOTIE) (project number: 20008948; name: "Development of virtual training system for cooperating flight crew team response to abnormal flight conditions").

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

References

1. Vicon. Available online: https://www.vicon.com/ (accessed on 25 April 2022).
2. OptiTrack. Available online: https://optitrack.com/ (accessed on 25 April 2022).
3. Toshev, A.; Szegedy, C. DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Los Alamitos, CA, USA, 2014; pp. 1653–1660. [CrossRef]
4. Mehta, D.; Sridhar, S.; Sotnychenko, O.; Rhodin, H.; Shafiei, M.; Seidel, H.P.; Xu, W.; Casas, D.; Theobalt, C. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
5. Mehta, D.; Sotnychenko, O.; Mueller, F.; Xu, W.; Elgharib, M.; Fua, P.; Seidel, H.P.; Rhodin, H.; Pons-Moll, G.; Theobalt, C. XNect: Real-Time Multi-Person 3D Motion Capture with a Single RGB Camera. *ACM Trans. Graph.* **2020**, *39*, 82:1–82:17. [CrossRef]
6. Ye, M.; Wang, X.; Yang, R.; Ren, L.; Pollefeys, M. Accurate 3D pose estimation from a single depth image. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: Los Alamitos, CA, USA, 2011; pp. 731–738. [CrossRef]
7. Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A. Real-time human pose recognition in parts from single depth images. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; IEEE: Los Alamitos, CA, USA, 2011; pp. 1297–1304. [CrossRef]
8. Wei, X.; Zhang, P.; Chai, J. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph. (TOG)* **2012**, *31*, 1–12. [CrossRef]
9. Xu, L.; Liu, Y.; Cheng, W.; Guo, K.; Zhou, G.; Dai, Q.; Fang, L. FlyCap: Markerless Motion Capture Using Multiple Autonomous Flying Cameras. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2284–2297. [CrossRef]
10. Nägeli, T.; Oberholzer, S.; Plüss, S.; Alonso-Mora, J.; Hilliges, O. Flycon: Real-time environment-independent multi-view human pose estimation with aerial vehicles. *ACM Trans. Graph. (TOG)* **2018**, *37*, 1–14. [CrossRef]
11. Saini, N.; Price, E.; Tallamraju, R.; Enficiaud, R.; Ludwig, R.; Martinovic, I.; Ahmad, A.; Black, M.J. Markerless Outdoor Human Motion Capture Using Multiple Autonomous Micro Aerial Vehicles. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; IEEE: Los Alamitos, CA, USA, 2019; pp. 823–832.
12. Xsens. Available online: https://www.xsens.com/ (accessed on 25 April 2022).
13. Perception Neuron Motion Capture. Available online: https://neuronmocap.com/ (accessed on 25 April 2022).
14. von Marcard, T.; Rosenhahn, B.; Black, M.; Pons-Moll, G. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Comput. Graph. Forum* **2017**, *36*, 349–360. [CrossRef]
15. Huang, Y.; Kaufmann, M.; Aksan, E.; Black, M.J.; Hilliges, O.; Pons-Moll, G. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *ACM Trans. Graph.* **2018**, *37*, 185:1–185:15. [CrossRef]
16. Xingjian, S.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 802–810.
17. CMU Graphics Lab Motion Capture Database. Available online: http://mocap.cs.cmu.edu/ (accessed on 25 April 2022).
18. Trumble, M.; Gilbert, A.; Malleson, C.; Hilton, A.; Collomosse, J. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In Proceedings of the 28th British Machine Vision Conference (BMVC), London, UK, 4–7 September 2017.
19. Moeslund, T.B.; Granum, E. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.* **2001**, *81*, 231–268. [CrossRef]
20. Moeslund, T.B.; Hilton, A.; Krüger, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.* **2006**, *104*, 90–126. [CrossRef]
21. Sarafianos, N.; Boteanu, B.; Ionescu, B.; Kakadiaris, I.A. 3d human pose estimation: A review of the literature and analysis of covariates. *Comput. Vis. Image Underst.* **2016**, *152*, 1–20. [CrossRef]
22. Poppe, R. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst.* **2007**, *108*, 4–18. [CrossRef]
23. Gong, W.; Zhang, X.; Gonzàlez, J.; Sobral, A.; Bouwmans, T.; Tu, C.; Zahzah, E.H. Human Pose Estimation from Monocular Images: A Comprehensive Survey. *Sensors* **2016**, *16*, 1966. [CrossRef]
24. Chen, Y.; Tian, Y.; He, M. Monocular human pose estimation: A survey of deep learning-based methods. *Comput. Vis. Image Underst.* **2020**, *192*, 102897. [CrossRef]
25. Starck, J.; Hilton, A. Model-based multiple view reconstruction of people. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; IEEE: Los Alamitos, CA, USA, 2003; pp. 915–922.

26. Bregler, C.; Malik, J. Tracking people with twists and exponential maps. In Proceedings of the 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231), Santa Barbara, CA, USA, 25–25 June 1998; IEEE: Los Alamitos, CA, USA,1998; pp. 8–15. [CrossRef]
27. Rosales, R.; Sclaroff, S. Combining generative and discriminative models in a framework for articulated pose estimation. *Int. J. Comput. Vis.* **2006**, *67*, 251–276. [CrossRef]
28. Sidenbladh, H.; Black, M.J.; Fleet, D.J. Stochastic tracking of 3D Human Figures Using 2D Image Motion. In Proceedings of the European Conference on Computer Vision, Dublin, Ireland, 26 June–1 July 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 702–718.
29. Sanzari, M.; Ntouskos, V.; Pirri, F. Bayesian image based 3d pose estimation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerlands, 2016; pp. 566–582.
30. Balan, A.O.; Sigal, L.; Black, M.J.; Davis, J.E.; Haussecker, H.W. Detailed Human Shape and Pose from Images. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; IEEE: Los Alamitos, CA, USA, 2007; pp. 1–8. [CrossRef]
31. Luvizon, D.C.; Picard, D.; Tabia, H. 2d/3d pose estimation and action recognition using multitask deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Los Alamitos, CA, USA, 2018; pp. 5137–5146.
32. Kanazawa, A.; Black, M.J.; Jacobs, D.W.; Malik, J. End-to-end recovery of human shape and pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Los Alamitos, CA, USA, 2018; pp. 7122–7131.
33. Kocabas, M.; Athanasiou, N.; Black, M.J. VIBE: Video Inference for Human Body Pose and Shape Estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Los Alamitos, CA, USA, 2020.
34. Elhayek, A.; de Aguiar, E.; Jain, A.; Thompson, J.; Pishchulin, L.; Andriluka, M.; Bregler, C.; Schiele, B.; Theobalt, C. MARCOnI—ConvNet-Based MARker-less motion capture in outdoor and indoor scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 501–514. [CrossRef] [PubMed]
35. Yang, W.; Ouyang, W.; Wang, X.; Ren, J.; Li, H.; Wang, X. 3d human pose estimation in the wild by adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Los Alamitos, CA, USA, 2018; pp. 5255–5264.
36. Zhou, X.; Sun, X.; Zhang, W.; Liang, S.; Wei, Y. Deep kinematic pose regression. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerlands, 2016; pp. 186–201.
37. Sun, X.; Xiao, B.; Wei, F.; Liang, S.; Wei, Y. Integral human pose regression. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerlands, 2018; pp. 529–545.
38. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: Los Alamitos, CA, USA, 2019; pp. 5693–5703.
39. Güler, R.A.; Neverova, N.; Kokkinos, I. Densepose: Dense human pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Los Alamitos, CA, USA, 2018; pp. 7297–7306.
40. Liu, Y.; Stoll, C.; Gall, J.; Seidel, H.P.; Theobalt, C. Markerless motion capture of interacting characters using multi-view image segmentation. In Proceedings of the CVPR 2011, Providence, RI, USA, 20–25 June 2011; IEEE: Los Alamitos, CA, USA, 2011; pp. 1249–1256. [CrossRef]
41. Rhodin, H.; Spörri, J.; Katircioglu, I.; Constantin, V.; Meyer, F.; Müller, E.; Salzmann, M.; Fua, P. Learning monocular 3d human pose estimation from multi-view images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Los Alamitos, CA, USA, 2018; pp. 8437–8446.
42. Roetenberg, D.; Luinge, H.; Slycke, P. Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial sensors. *Xsens Motion Technol. BV Tech. Rep.* **2009**, *1*, 1–7.
43. Slyper, R.; Hodgins, J.K. Action Capture with Accelerometers. In Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08), Dublin, Ireland, 7–9 July 2008; Eurographics Association: Goslar, Germany, 2008; pp. 193–199.
44. Tautges, J.; Zinke, A.; Krüger, B.; Baumann, J.; Weber, A.; Helten, T.; Müller, M.; Seidel, H.P.; Eberhardt, B. Motion Reconstruction Using Sparse Accelerometer Data. *ACM Trans. Graph.* **2011**, *30*, 18:1–18:12. [CrossRef]
45. Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M.J. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graph.* **2015**, *34*, 248:1–248:16. [CrossRef]
46. Schuster, M.; Paliwal, K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
47. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
48. Yi, X.; Zhou, Y.; Xu, F. TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors. *ACM Trans. Graph.* **2021**, *40*, 1–13.

49. Liu, H.; Wei, X.; Chai, J.; Ha, I.; Rhee, T. Realtime Human Motion Control with a Small Number of Inertial Sensors. In Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D '11), San Francisco, CA, USA, 18–20 February 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 133–140. [CrossRef]
50. Schwarz, L.A.; Mateus, D.; Navab, N. Discriminative human full-body pose estimation from wearable inertial sensor data. In Proceedings of the 3D Physiological Human Workshop, Zermatt, Switzerland, 29 November–2 December 2009; pp. 159–172.
51. Malleson, C.; Gilbert, A.; Trumble, M.; Collomosse, J.; Hilton, A.; Volino, M. Real-time full-body motion capture from video and imus. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; IEEE: Los Alamitos, CA, USA, 2017; pp. 449–457.
52. Von Marcard, T.; Pons-Moll, G.; Rosenhahn, B. Human pose estimation from video and imus. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1533–1547. [CrossRef] [PubMed]
53. von Marcard, T.; Henschel, R.; Black, M.J.; Rosenhahn, B.; Pons-Moll, G. Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerlands, 2018.
54. Zhang, Z.; Wang, C.; Qin, W.; Zeng, W. Fusing Wearable IMUs With Multi-View Images for Human Pose Estimation: A Geometric Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Los Alamitos, CA, USA, 2020.
55. Huang, F.; Zeng, A.; Liu, M.; Lai, Q.; Xu, Q. DeepFuse: An IMU-Aware Network for Real-Time 3D Human Pose Estimation from Multi-View Image. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020; IEEE: Los Alamitos, CA, USA, 2020.
56. Gilbert, A.; Trumble, M.; Malleson, C.; Hilton, A.; Collomosse, J. Fusing visual and inertial sensors with semantics for 3d human pose estimation. *Int. J. Comput. Vis.* **2019**, *127*, 381–397. [CrossRef]
57. Helten, T.; Muller, M.; Seidel, H.P.; Theobalt, C. Real-Time Body Tracking with One Depth Camera and Inertial Sensors. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; IEEE: Los Alamitos, CA, USA, 2013.
58. Zheng, Z.; Yu, T.; Li, H.; Guo, K.; Dai, Q.; Fang, L.; Liu, Y. Hybridfusion: Real-time performance capture using a single depth sensor and sparse imus. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 384–400.
59. Andrews, S.; Huerta, I.; Komura, T.; Sigal, L.; Mitchell, K. Real-time physics-based motion capture with sparse sensors. In Proceedings of the 13th European Conference on Visual Media Production (CVMP 2016), London, UK, 12–13 December 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1–10.
60. Antilatency. Available online: https://antilatency.com/ (accessed on 25 April 2022).