DOI: 10.1111/cgf.142631
EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

LoBSTr: Real-time Lower-body Pose Prediction from Sparse Upper-body Tracking Signals

Dongseok Yang, Doyeon Kim, and Sung-Hee Lee

Korea Advanced Institute of Science and Technology (KAIST)

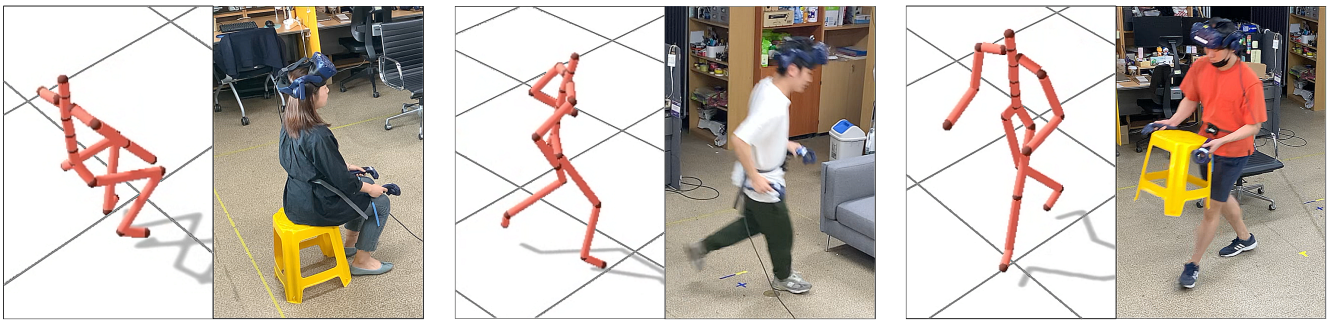

Figure 1: *Our system predicts the lower-body pose of the user from sparse tracking signals of the head, hands, and pelvis for a wide range of actions using off-the-shelf VR devices.*

Abstract

With the popularization of games and VR/AR devices, there is a growing need for capturing human motion with a sparse set of tracking data. In this paper, we introduce a deep neural network (DNN) based method for real-time prediction of the lower-body pose only from the tracking signals of the upper-body joints. Specifically, our Gated Recurrent Unit (GRU)-based recurrent architecture predicts the lower-body pose and feet contact states from a past sequence of tracking signals of the head, hands, and pelvis. A major feature of our method is that the input signal is represented by the velocity of tracking signals. We show that the velocity representation better models the correlation between the upper-body and lower-body motions and increases the robustness against the diverse scales and proportions of the user body than position-orientation representations. In addition, to remove foot-skating and floating artifacts, our network predicts feet contact state, which is used to post-process the lower-body pose with inverse kinematics to preserve the contact. Our network is lightweight so as to run in real-time applications. We show the effectiveness of our method through several quantitative evaluations against other architectures and input representations with respect to wild tracking data obtained from commercial VR devices.

CCS Concepts

• ***Computing methodologies*** → *Motion capture; Virtual reality; Mixed / augmented reality;*

1. Introduction

Human motion capture is gradually expanding its application areas. While precise motion capture with a dense set of sensory data is widely adopted for movie making, there is a growing need for capturing human motion with only a sparse set of data for mass-market, such as game and VR/AR, due to cost and usability reasons.

A reasonably minimal number of tracking points for the whole body motion capture would be six: head, hands, feet, and trunk. The whole body motion can then be reconstructed based on inverse kinematics. However, many commercial body-worn trackers (e.g., HTC VIVE and Oculus Rift) have limitations, especially for feet tracking.

First, trackers are vulnerable to impacts between the feet and floor during locomotion. The impact causes instantaneous deviation

of the trackers from the body, which induces a large sensory error. In addition, feet trackers have a high probability of failure due to the occlusion by nearby objects if the capture space is equipped with furniture and objects.

Except for the motions in which leg movement is important (e.g., kicking), the leg generally takes the role of moving the upper-body to some desired direction or supporting the posture of the upper-body. For this range of motions, is it possible to estimate the natural motion of the lower-body with only the sparse tracking data of the upper-body? Will such a lower-body motion estimator be valid in a wide range of motions, from locomotion that shows high correlations between upper and lower bodies to the motions in which such correlation is weak (e.g., in-place upper-body movement)?

To address these questions, we develop a novel deep neural network (DNN)-based architecture, named *LoBSTr* (**Lo**wer-**B**ody prediction with **S**parse **Tr**ackers), that predicts lower-body pose given the tracked position and orientation data for the head, hands, and pelvis. The goal of our network is challenging as it is severely underdetermined; sparse tracking information of upper-body joints can be mapped to many different lower-body poses, which are also affected by the body size and proportion. To tackle this challenge, we carefully designed the input representation and network structure, which are the major contributions of our work. First, to reduce the ambiguity of mapping from upper-body signals to lower-body pose, LoBSTr takes as input past temporal sequence of upper-body signals. Its recurrent neural network (RNN)-based architecture then outputs the lower-body pose at the current frame. Second, to model the correlation between upper-body and lower-body motions and increase the robustness against the diverse sizes and proportions of the user body, the input signal is represented in terms of the velocity. In addition, to remove foot-skating and floating artifacts, LoBSTr predicts feet contact states, which are used to post-process the lower-body pose with inverse kinematics to preserve the contact. Lastly, our network is designed to be lightweight so as to run in real-time applications (45 fps, < 22ms).

As a result, our method can generate plausible lower-body motions in a reasonable range for VR/AR applications from locomotion to those which have weak correlations with the upper body, e.g., upper-body gesture while walking or standing, allowing for a significantly wider range of capturable motions than previous 3-point tracking systems [Dee18; LO19] (Figure 1).

By removing feet trackers for full-body avatar motion generation, our system is free from artifacts caused by foot-floor impact and tracker occlusion in a cluttered environment, achieves a larger tracking area than those of conventional systems, and reduces the cost for devices; thereby making full-body tracking more accessible to the general VR/AR users.

In addition, the robustness of our method allows us to train the model with a single body size, using existing motion capture datasets such as CMU [13], PFNN [HKS17], and MHAD [OCK*13] datasets, yet to be applicable to different body proportions tracked with commercial trackers. Our network architecture successfully learns temporal features of human motions to reconstruct different motions and transitions between them without phase or contact labeling.

Our method is evaluated quantitatively by comparing average positional and rotational errors of different network architectures. Different input representations are evaluated in terms of toe-base distance error and average body movement [SZKZ20] to assess the model's robustness to user body shape change. An ablation study over loss terms is performed to see the effects of proposed training loss terms. A qualitative comparison is also conducted against a baseline 6-point tracking system.

2. Related Work

This section introduces previous work related to ours with respect to the problems of obtaining reduced space for human motion and estimating human motion from sparse sensor signals.

2.1. Motion Synthesis from Reduced Space Representations

Obtaining a reduced space representation of human motion that captures only the valid scope of human motions enables generating natural motions as well as editing and optimizing motions faster.

Early methods used linear dimensionality reduction models. Chai and Hodgins [CH05] applied the principal component analysis (PCA) to construct locally linear pose spaces online and synthesized natural human pose with low dimensional control inputs. Liu et al. [LZWM06] proposed a PCA-based clustering method to map the full set of marker configurations to a lower-dimensional set while preserving the original captured poses. Liu et al. [LWC*11] constructed a dynamic linear motion space online by searching a set of similar motion clips from the dataset and estimating the current pose under the maximum a posteriori (MAP) framework. Safonova et al. [SHP04] synthesized realistic human motions by applying physics constraints to adjust poses obtained by PCA-based methods. Linear methods are simple and fast but can only model a narrow range of behavior-specific motions.

Nonlinear reduction methods can better represent human motion by reflecting the nonlinear characteristics of human motion. Grochow et al. [GMHP04] and Levine et al. [LWH*12] used Gaussian Process Latent Variable Model (GPLVM) to obtain a low dimensional latent space to reconstruct high dimensional motion data. The nonlinear models are generally effective when dealing with a small number of motion samples but their space and time complexities grow fast with the size of learned motion data.

Recently, DNN-based methods are developed to obtain a latent space for synthesizing and reconstructing motions. Holden et al. [HSK16] used a convolutional autoencoder to obtain a motion manifold, which is used to synthesize and edit motions with high-level control inputs such as locomotion paths. Jang and Lee [JL20] developed an RNN-based method to learn a motion manifold trained with a sequence-to-sequence architecture. Ling et al. [LZCV20] developed autoregressive conditional variational autoencoders that learn reduced space of plausible human motions for character motion generation. [HKPP20] applied DNN to the motion matching technique to reduce memory usage while retaining the quality of the controllable motion. Our work also learns a reduced motion space using a DNN architecture. However, unlike previous work that reduces the whole-body data to generate whole-body motions,

we obtain a reduced space from the upper-body data to generate the corresponding lower-body motions.

2.2. Real-time Pose Prediction from Sparse Tracking Signals

The task of predicting full-body pose from sparse signals is an underdetermined problem, where many different poses can satisfy the given inputs. To obtain the most plausible pose among them, researchers have developed methods that leverage additional knowledge or heuristics on human motion or learn to predict plausible poses from examples.

Recently deep learning approaches are used for pose prediction and show promising results. Wouda et al. [WGR*19] proposed a bi-directional Long Short-Term Memory (LSTM) model to estimate full-body pose from 5 IMUs and showed that their method is superior to a baseline shallow learning method with respect to preserving temporal coherency of motion. Huang et al. [HKA*18] developed a bi-directional RNN to learn the temporal pose priors and reconstructed human pose from 6 Inertial Measurement Units (IMUs) worn on the body. They first trained the network with synthetic IMU data and then fine-tuned it with a real dataset. These approaches predict full-body pose given sparse signals from sensors attached to all limbs and other body parts, whereas our method aims to predict the lower-body from upper-body tracking signals.

Retrieving full-body pose from sparse observations is also actively researched in computer vision. Cheng et al. [CYW*19] used 2D and 3D temporal convolutional networks (TCNs) to reconstruct 3D human pose from occluded monocular video. [RF20] showed that 3D mesh reconstruction can be improved by pre-training networks with cropped images in an annotated dataset.

Several studies proposed methods to predict the lower-body pose from the upper-body information. Jiang et al. [JYF16] developed a method to generate lower-body motion by blending 8 walking animations with different directions based on the body direction and velocity, which are predicted from off-the-shelf VR devices. As a blended motion, the resulting motion looks natural but its scope is limited by the predefined animations. A balance control-based method [TCW19] reconstructs static pose and locomotion of the lower-body according to the tracked upper-body joints. Specifically, the target Zero Moment Point (ZMP) trajectory is determined from the upper-body motion, and the full-body animation is generated to realize the ZMP trajectory. [Dee18] introduced a commercial 3-point tracking system with a physics-based character model, trained to satisfy physical and kinematic constraints using deep reinforcement learning. The resulting motion may take unnatural steps, different from the actual motion of the user. Recently, Lin [LO19] developed an LSTM-based pose estimator with 3 trackers worn on the user's head and hands. The trained model successfully estimates the upper-body pose but often fails to predict the lower-body pose. Compared with these studies, our method is capable of generating more natural lower-body motions for a wider range of actions.

3. Prediction of the Lower-Body Pose

Figure 2 shows the overview of our LoBSTr network architecture. It infers the lower-body pose and feet contact states at the current frame from a past sequence of tracking signals for the 4 upper-body joints, including the head, hands, and pelvis. The input sequence is mapped to a latent representation by an encoder and passed to two linear layers, one for lower-body pose prediction and the other for feet contact prediction. The output pose is post-processed with respect to the output contact state by inverse kinematics for better visual quality.

Pose Representation. We first describe the representation of pose in our work. A pose of a character $\mathbf{q}_i$ at frame i is represented with the world position p^{root} and orientation q^{root} of the root $\mathbf{r}_i = [p_i^{root}, q_i^{root}]$ and the local rotations of remaining joints $\mathbf{j}_i = [q_i^k]_{k=1}^{n_{joint}}$, i.e., $\mathbf{q}_i = [\mathbf{r}_i, \mathbf{j}_i]$. All rotations and orientations are represented in 6-DoF with forward (Z-axis) and up (Y-axis) vectors. To express motion in the user's egocentric space, we define a virtual reference joint, of which frame is located at the root joint and oriented to point to the world up vector with its Y-axis and to the frontal direction with its Z-axis. The frontal direction is obtained by projecting the frontal direction of the pelvis joint to the ground plane.

3.1. Velocity-based Prediction

Predicting pose from the position and orientation data of trackers is highly sensitive to the user's body shape, which causes variation in tracker configurations. For example, the pelvis tracker tends to be attached farther from the actual pelvis joint for an overweight user and has different initial rotations for every run; it requires careful calibration on tracker position and orientation to generate natural poses from those raw signals. To avoid this, we designed our input representation to be least affected by the variation of tracker configurations and user's body shape but to be directly acquired from the raw tracking signals.

To this end, we represent the input motion of the 4 joints (i.e., the head, hands, and reference joints) in terms of the velocities. Specifically, the velocities of sparse joints are described with respect to the defined reference coordinate frame, which is different from conventional body velocities of full-body joints [CAW*19]. We argue that this velocity representation is more robust to differences in tracker configuration and body shape of the users than position and orientation representation, and thus allows our network to produce robust output from different users with a simple calibration step.

The velocities $\mathbf{d}_i$ of tracked joints at the i^{th} frame consist of the linear and angular velocities $d_i^{ref} = [v_i^{ref}, w_i^{ref}]$ of the reference joint and the linear and angular velocities $d_i^{joint} = [v_i^{joint}, w_i^{joint}] \in R^{(3+6)}$ of the three joints, expressed with respect to the current reference joint frame. Refer to Appendix 1 for the equations to compute $\mathbf{d}_i$.

The input X_i to our network at the i^{th} frame includes the velocity vector $\mathbf{d}$ and the height p_y^{ref} of the reference joint above the ground, in the frame range of $[i-(\Delta i-1), i]$:

$$X_i = [x_{i-(\Delta i-1)},\ x_{i-(\Delta i-2)},\ \dots,\ x_i] \in R^{\Delta i \times \dim(x)} \quad (1)$$

$$x_i = [\mathbf{d}_i, p_{y,i}^{ref}] \in R^{(4\times 9+1)} \quad (2)$$

$$\mathbf{d}_i = [d_i^{ref}, [d_i^{joint,k}]_{k=1}^{3}] \quad (3)$$

Then our network outputs the information to predict the lower-

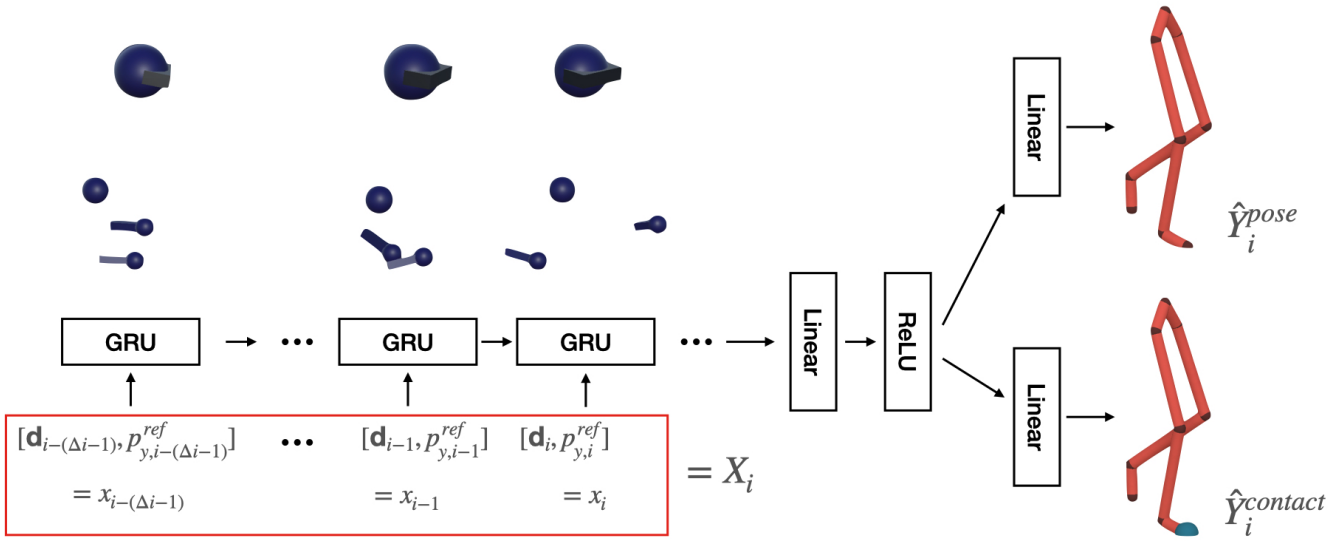

Figure 2: *Our LoBSTr network architecture to predict the lower-body pose given a sequence of upper-body tracking signals.*

body pose at the current frame i. These are the local rotations of 8 lower-body joints (the hip joints, upper-legs, lower-legs, and feet) $\hat{Y}^{pose}_i \in R^{(8\times6)}$ and contact probabilities of feet $\hat{Y}^{contact}_i \in R^{(2\times2)}$. Feet contact states are determined from the output contact probabilities and used to remove the foot-skating artifact in the post-processing step.

3.2. Network Architecture

We use an RNN with Gated Recurrent Units (GRUs) [CGCB14] as the latent encoder model, which has shown good performance in preserving temporal continuity of human motions [WGR*19]. GRUs have the remember-forget function that is capable of learning implicit priorities over frames to form a hidden space from the input frame sequence. Furthermore, GRU structure requires a fewer number of parameters than LSTM [HS97] to achieve similar performance, thereby allowing for faster computation time, an important quality for real-time applications.

In our network architecture, one layer GRU network with a hidden dimension of 1024 encodes the input vectors to a latent representation, conditioned by the input vectors of previous frames. Specifically, an input vector x_i of the i^{th} frame is encoded as follows: $h_i = GRU(h_{i-1}, x_i|\theta_g)$, where h_{i-1} is the hidden state of the $(i-1)^{th}$ frame and θ_g is the hidden layer parameter. After the input vector x_i at the current frame is encoded to h_i, h_i passes through a linear layer with parameter θ_{l_1} and ReLU activation, generating a latent vector of 128 dimensions. Finally, the latent vector passes through two linear layers with parameters θ_{l_p} and θ_{l_c} to produce the local rotations of the 8 lower-body joints $\hat{Y}^{pose}_i$ and the contact probability $\hat{Y}^{contact}_i$, respectively. In our experiment, the dimensions of each parameter are as follows: $\dim(\theta_g) = (4\times 9+1)\times 1024$, $\dim(\theta_{l_1}) = 1024\times 128$, $\dim(\theta_{l_p}) = 128\times(8\times6)$, $\dim(\theta_{l_c}) = 128\times(2\times2)$.

The size of the time window Δi is set to 45 (= 1 second for 45 fps data), which was found empirically to be appropriate to capture characteristics of different motion categories and maintain temporal continuity of output lower-body poses when played sequentially in real-time.

3.3. Network Training

We train the network in a supervised manner. In order to include various actions ranging from locomotion to in-place upper body motion, we built a dataset combining parts of several motion capture datasets. Specifically, a training set of (X_i, Y_i) is obtained from PFNN, MHAD, and CMU motion datasets and the network is trained to output as close as to Y_i given X_i. Refer to Appendix 2 for the details on preparing and augmenting the motion dataset for the training.

Following the previous studies, we optimize our network to minimize four loss terms: pose loss, forward kinematics loss [PGA18; VYCL18], feet velocity loss [PFAG19; KPKH20], and contact prediction loss [SAA*20]. The pose loss is our target loss which directly affects the accuracy of the predicted lower-body pose. The remaining three terms work as regularizers to promote the naturalness of feet trajectories and reduce artifacts such as foot-skating and floating.

Pose loss. As predicting the lower-body pose at the current frame is the goal of our network, it is trained to minimize the L^1 norm between the predicted lower-body pose and the ground truth Y^{pose}_i.

$$\mathcal{L}_{pose} = \left\| \hat{Y}^{pose}_i - Y^{pose}_i \right\|_1$$

Forward kinematics (FK) loss. Minimizing 3D positional errors of the feet results in better perceptual quality by decreasing skating and floating artifacts of the feet. The loss is defined as the L^2 norm between the toe-base joint positions computed by FK from

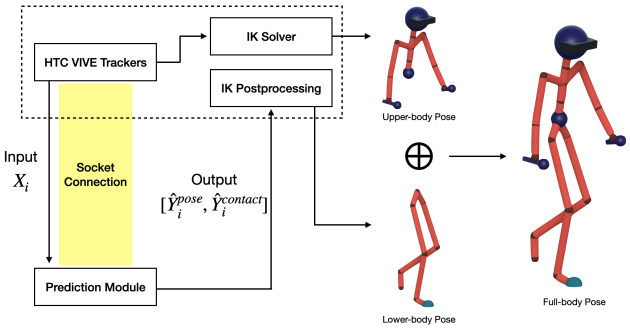

Figure 3: *Overview of our full-body avatar system.*

the predicted and ground truth lower-body poses.

$$\mathcal{L}_{FK} = \left\| FK(\hat{Y}_i^{pose}) - FK(Y_i^{pose}) \right\|_2$$

Feet velocity loss. The jiggling of the resulting feet trajectory is one of the major artifacts which severely degrades the quality of output motion. We designed a velocity loss, which drives our network to produce the ground truth velocity of toe-base joints.

$$\begin{aligned}\mathcal{L}_{velocity} &= \|(FK(\hat{Y}_i^{pose}) - FK(Y_{i-1}^{pose})) \\ &\quad -(FK(Y_i^{pose}) - FK(Y_{i-1}^{pose}))\|_2\end{aligned}$$

Contact Prediction Loss. In addition to the lower-body pose, our model predicts the contact state of each toe-base joint from the latent vector of the input sequence. The contact prediction layer is a binary classification layer trained to minimize the binary cross-entropy between output label probability and ground truth contact label.

$$\mathcal{L}_{contact} = CrossEntropy(\hat{Y}_i^{contact}, Y_i^{contact})$$

Combining the four loss terms, the final loss function is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{pose} + \lambda_2 \mathcal{L}_{FK} + \lambda_3 \mathcal{L}_{velocity} + \frac{\lambda_4}{2}(\mathcal{L}_{contact}^{left} + \mathcal{L}_{contact}^{right}),$$

where hyper-parameters for experiments are set as follows: $\lambda_1 = 1$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$, and $\lambda_4 = 10^{-6}$. Refer to Appendix 3 for the details on network training.

3.4. Real-Time Motion Generation

By using our lower-body pose prediction network, we develop a real-time full-body avatar system that uses off-the-shelf VR trackers as shown in Fig. 3. Specifically, HTC Vive Pro set is used as a tracking device: HMD, two hand-held controllers, and a tracker on the pelvis. Input to the network is computed from the world transformations of trackers. Note that there may be a large deviation from the tracker positions and the joint positions. For example, the pelvis tracker worn in front of the waist is far from and the pelvis joint. However, we do not take any special treatment to compensate for this difference, which is challenging due to the user's shape variations. Instead, our input representation of velocities robustly handles the deviation.

Our system is implemented with Unity3D engine and SteamVR platform and runs at a fixed framerate of 45 fps. The system requires a warm-up time of 1 second to generate an initial input of 45-frames. The input sequence is sent to the lower-body pose prediction module every frame via TCP socket connection. The average inference time is around 2.5ms, which is fast enough to maintain the running rate of 45 fps. The upper-body pose is computed to match the tracked end-effector transformations by an IK solver [Roo17] and combined with the lower-body pose to animate the virtual character skeleton.

3.4.1. Contact Post-processing

While our network outputs plausible poses of the legs, continuous animation of the leg shows some extent of foot-skating and floating artifacts. These artifacts are more visible when users take a vigorous motion like running or sharp turn, which induces high-frequency shaking of the pelvis tracker.

To resolve those artifacts that occur during runtime, we train our model to output contact probabilities of two toe-base joints and post-process the leg pose in the contact state. At the moment when the contact state changes from false to true, the toe-base position is set as the target contact position and a Jacobian-based inverse kinematics solver is used to maintain the contact while the contact state remains true.

To maintain a smooth transition when the toe-base joint loses contact, a simple interpolation is performed for a fixed time window (10 frames in our experiment); the toe-base joint position at the contact-losing frame and that computed from the network output are linearly interpolated to determine the target toe-base joint position. The corresponding leg pose is computed from a Jacobian-based inverse kinematics solver. The interpolation parameters are computed from a shifted sine function to implement a slow-in-slow-out transition.

4. Experiment

We assess the effectiveness of our method through several offline and online evaluations as shown below.

4.1. Offline Evaluation

4.1.1. Network Architecture Comparison

To the best of our knowledge, our work is the first DNN-based method trained with existing motion capture datasets to predict lower-body pose only from the input sequence of upper-body joints. Therefore, we compare models with baseline network architectures of fully connected layers (FC), convolutional layers (CNN), and different types of recurrent unit layers (RNN, LSTM, and GRU). We assess the ratio of correct contact labels out of the whole frames, average rotational error per joint (8 lower-body joints), and average positional error of two end-effectors (left and right toe-bases). The networks only vary in the latent mapping module for input sequence; the encoder is replaced with different network architectures while linear layers for predictions remain identical. Hidden and latent dimensions are fixed to 1024 and 128, respectively. The FC encoder consists of 6 layers of FC layers [Hol18] and the CNN encoder consists of 1D convolution and

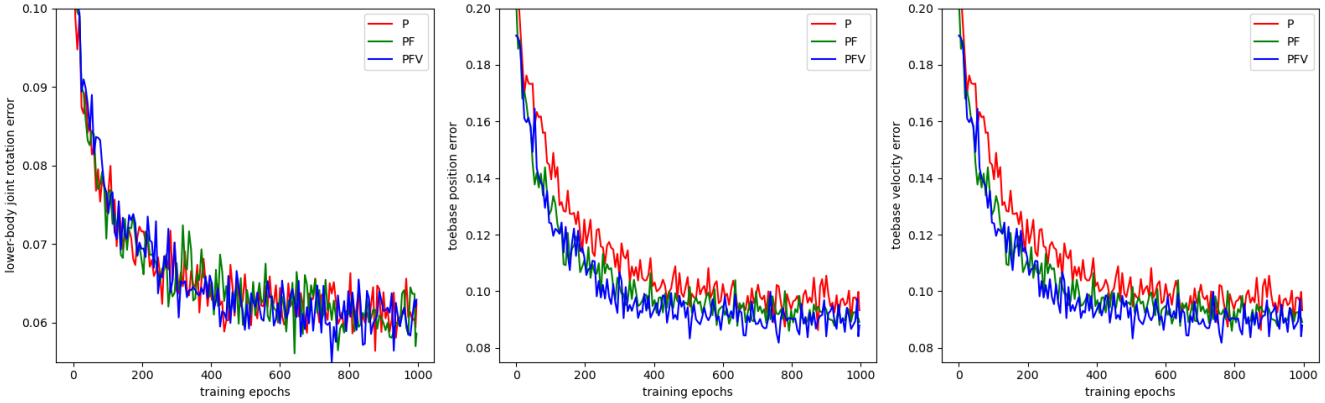

Figure 4: *Errors of different loss term combinations according to training epochs.*

max-pooling layers [HSKJ15]. Table 1 shows the errors of different architectures for the test dataset.

In addition, we tested a GRU encoder with an autoregressive structure, which was reported to improve motion prediction accuracy and reduce visual artifacts in the output sequence [MBR17]. Our tested autoregressive model forms a latent vector by concatenating a latent vector output from a GRU encoder, which takes the upper-body tracking signals as input, and a linear layer (parameter dimension of 48×128), whose input is the lower-body pose of the previous time step. This autoregressive model leads to the explosion of the resulting lower-body pose after approximately 120 frames of running, presumably because of the accumulation of errors in the lower-body pose.

The comparison shows that the models with recurrent structure and remember-forget function (LSTM and GRU) produce the best performance on all three measures and the difference between the two is not significant. We choose to use GRU for its higher prediction accuracy for contacts as well as having fewer cell parameters, which allows for faster training and inference time.

Table 1: *Network architecture comparison on the test dataset.*

Model	Contact Accuracy	Rotational Error	Positional Error
FC-6	84.22%	9.37°	7.2cm
CNN-1D	72.33%	13.86°	13.3cm
RNN	82.27%	9.43°	7.1cm
LSTM	83.85%	8.50°	6.63cm
GRU (Ours)	**85.77%**	8.53°	6.63cm

In addition to the quantitative analysis, we evaluate the visual quality of the output animation. Architectures with recurrent units (RNN, LSTM, and GRU) produce smooth and continuous motions while others suffer from jittery output motions.

4.1.2. Loss Term Ablation Study

We examine the effect of proposed loss terms ($\mathcal{L}_{FK}$ and $\mathcal{L}_{velocity}$) by doing an ablation study for the terms. The proposed model of GRUs is trained by three different loss term combinations: 1. $\mathcal{L}_{pose}$, 2. $\mathcal{L}_{pose} + \mathcal{L}_{FK}$, 3. $\mathcal{L}_{pose} + \mathcal{L}_{FK} + \mathcal{L}_{velocity}$, denoted as **P**, **PK**, and **PKV**, respectively. Figure 4 shows that adding $\mathcal{L}_{FK}$ and $\mathcal{L}_{velocity}$ does not affect the joint angle error but decreases the average position and velocity errors of the feet. Table 2 shows that the average positional error of the feet is minimum with **PKV**. We evaluate the visual quality of output lower-body motion from the models trained with the three loss term combinations, among which **PKV** gives significantly more stable lower-body motions, especially when the user is performing in-place upper-body motions.

Table 2: *Errors of different loss term combinations on the test dataset.*

Loss Term Combination	P	PK	PKV
Positional Error (cm)	6.95	6.64	**6.63**

4.1.3. Input Representations Comparison

The robustness of the proposed velocity input feature is evaluated by two measures. First, we measure toe-base distance error, the difference of the distances between two toe-bases from the output pose and ground truth pose captured with trackers. Secondly, we compare body movement [SZKZ20], the sum of absolute joint angle updates in output pose, generated from recorded inputs. Body movement metric examines whether the network is overfitted and produces over-smoothed motion for sub-domains represented by sparse training samples. The test set of wild signals is captured from subjects with 157, 171, and 184 cm heights. The subjects are asked to perform a sequence of actions; walk, run, sit/stand up, carry and move, static gestures, and free movement. The free movement contains motion categories that are not in training sets (e.g., walking backward, cross-steps, and jumping). Three identical models are trained with inputs of position-orientations, velocities,

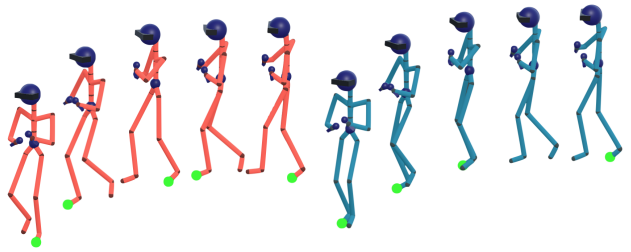

Figure 5: *Styled running (arms swinging in front of the chest) motion by a 157cm subject. Compared with velocity input (red), position-orientation input (green) occasionally generates static poses.*

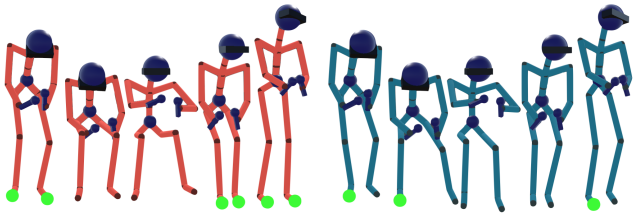

Figure 6: *Sitting-standing up motion by a 176cm subject. Compared with velocity input (red), position-orientation input (green) shows less accurate contact state prediction.*

and a combination of both representations. For position-orientation representation, the input for reference joint is given in velocity representation.

Table 3 shows the average toe-base distance error per frame over different user heights. For the whole sequence, velocity representation outperforms the other representations in average error for every subject. When divided into motion categories, position-orientation representation shows smaller errors for a majority of the subjects (2 out of 3), only for the sit/stand up category. The combined representation shows significantly larger errors for VR tracker data, although it outperformed the others for motion capture data. This result suggests that the model with combined input is trained to fit the distribution of the training domain (motion capture data) more tightly and thus becomes more sensitive to the body dimensions, tracker configurations, and wild noise than the velocity or position-orientation representation. From the result, we conclude that velocity representation generates much closer leg swings to the ground-truth than position-orientation representation.

Figure 7 shows that velocity representation surpasses position-orientation representation in body movement measure for all users. While comparing visual quality of output lower-body motions, we found that the model with velocity representation learned larger motion space including sharp motions but that with position-orientation representation was overfitted to a smaller range of locomotion and sitting motion.

We observe that velocity representation generates continuous

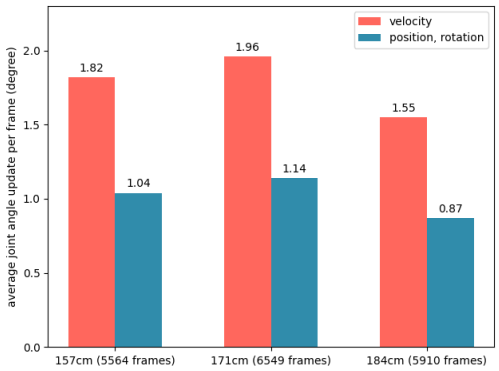

Figure 7: *Average per-frame joint angle update of output poses from users with different heights.*

lower-body motion for the recorded test inputs, while position-orientation representation sometimes produces unnatural static poses during motion (Figure 5). In addition, the model maintains higher precision of contact state prediction with the velocity representation (Figure 6). From the results, we conclude that velocity representation outperforms position-orientation representation with respect to the robustness to unobserved input data (e.g., different body size and motion style), the accuracy of feet contact prediction, and visual quality.

4.2. Online Evaluation

We tested our real-time full-body avatar system for a wide range of motions performed by several participants with varying heights. Recordings of avatar and participant motions are provided in the supplemental video.

Before capturing, each participant takes T-pose in the calibration stage to compute rotational offsets between trackers and corresponding joints in the skeleton. Rotational offsets are multiplied to tracking signals to compensate for individual differences in tracker configuration, which may result from different controller grip and different wearing positions of trackers; the offsets not only vary among users but also for the same user at each session. Input joint velocities are then computed from the tracked world positions and orientations, compensated by the offsets. In addition, a height scale is obtained as the ratio of the pelvis heights between the participant and the standard skeleton used for the network training. The height of the participant's reference is multiplied by the height scale to map to the standard skeleton before feeding to the prediction network. Then our system outputs the whole-body pose of the standard skeleton, which is visualized after scaling with the height scale to reflect the participant's height.

A total of four people with heights of 157, 171, 180, and 184 cm participated in the test. The participants are first asked to perform basic actions, including walking, running, sitting, standing up, and static gestures. After that, they are asked to do actions that are not in the training set, including walking backward, moving objects to another place, and walking with greetings. Finally, they are asked to move freely with any actions they want to test. The capture envi-

Table 3: *Average toe-base distance errors (cm) for users with different heights.*

User Height	Representation	Walk	Run	Sit/Stand up	Carry and move	Static gestures	Free	**Total**
157 cm	velocity	4.8	1.6	3.5	5.3	4.0	9.5	**4.4**
	pos-ori	6.9	4.2	1.7	7.5	0.2	11.0	6.0
	combined	17.0	10.7	6.3	13.1	5.2	20.5	13.3
171 cm	velocity	3.7	1.3	1.3	8.3	0.7	7.5	**4.6**
	pos-ori	8.3	5.3	2.1	6.3	2.3	7.4	5.0
	combined	20.8	12.2	6.2	10.6	1.2	17.0	11.9
184 cm	velocity	0.03	1.4	2.4	7.0	9.2	3.3	**2.1**
	pos-ori	2.9	1.2	1.6	7.7	14.4	6.4	5.2
	combined	14.6	12.9	9.7	17.8	2.3	16.3	13.0

ronment is designed to represent the play area of general users; the room is 3.6m x 3.6m and is equipped with objects and furniture.

Figure 8 shows the snapshots of participants capturing various actions. Our network successfully reconstructed lower-body pose corresponding to the input upper-body motion and maintained temporal continuity between output poses. Interestingly, our model successfully learned gait patterns without using any phase or contact labels as input, matching leg movement and the feet contact states with the upper-body motion of the participants. Furthermore, our network successfully created the lower-body pose from unobserved inputs of diverse motion categories, different motion styles, and body shapes.

Comparison against 6-Point Tracking. Figure 9 compares avatar animation obtained by our method and by a 6-point tracking (4 trackers as ours and 2 additional trackers at feet) method. The lower body motion of the 6-point tracking method is created by inverse kinematics. One can see that the resulting animation of our method is similar to that from the 6-point tracking.

Due to the infrared occlusion by existing objects, the feet tracking often failed and caused the 6-point tracking avatar to make bizarre poses (Figure 10). In contrast, our method does not suffer from such tracking loss and outputs lower-body motion robustly.

5. Limitation and Future Work

Our method has several limitations, which should be overcome in order to enable users to freely capture the unbounded range of high-quality motions with sparse tracking data.

First, we fixed the Y-axis of the reference joint to the world up-vector to clearly capture the ground-plane translation and rotation. As a result, the reference joint can only rotate about Y-axis but not about the frontal or lateral axes, which are major axes for some rotational actions like rolling and windmill. To deal with these rotations, an additional step of predicting frontal and lateral rotations of the pelvis by using all four tracking signals would be necessary.

Second, while our method shows competitive results for contact prediction in the majority of cases, prediction accuracy drops when the user makes sharp turns or complex leg-crossing motions, which may cause inappropriate fixation of feet or skating/floating behavior. It remains a future goal to develop more robust contact predictors and corresponding leg animation methods.

Third, the output lower-body animation sometimes contains jittering and sliding artifacts. In most cases, the artifacts are due to the unintended movement of the pelvis tracker, which is attached in front of the abdomen and thus moves differently from the actual pelvis joint. An additional source of artifact is the ambiguity occurring when a stationary user starts moving; it is difficult to determine whether the user begins locomotion or moving in-place only from past observations. For this reason, artifacts are observed during dynamic in-place motions such as making boxing or walking gestures. Utilizing short future information [HKA*18] and exploring the autoregressive approach [MBR17] are promising future directions to reduce this ambiguity.

Lastly, a very challenging goal with high impact is to capture the full-body motion by using only a minimal 3 trackers on the head and hands, without using the pelvis tracker, which is cumbersome to wear on top of clothing. However, considering the great amount of ambiguity in human poses given only three tracker signals, we conjecture that it would be extremely difficult to develop a 3-point tracking system that can distinguish a wide range of human poses. A feasible direction might be to combine with additional sensory modals, e.g., attaching a vision sensor on HMD, that reinforce the full-body motion capture.

6. Conclusion

We presented a novel method to reconstruct lower-body motion from sparse tracking data of upper-body joints, using a GRU-based network architecture. Our method does not require to use feet trackers, which are error-prone due to IR occlusion and foot-ground impact. Our GRU-based structure successfully learns the temporal characteristics of human motion. The velocity-based prediction scheme is robust against variations in tracker attachment and users' body shape. Overall, our system generates lower-body motion that is visually competitive to the sequence obtained from the baseline 6-point tracking system with additional feet trackers.

Acknowledgement

This work was supported in part by KEIT, Korea (20011076) and Korea Creative Content Agency, Korea (R2020040211).

Figure 8: *Avatar motions generated in real-time from users with various heights, 157cm (top), 171cm (middle), and 180cm (bottom).*

Figure 9: *Comparison of real-time avatar motion against 6-point tracking (Red: Ours, Green: 6-point tracking).*

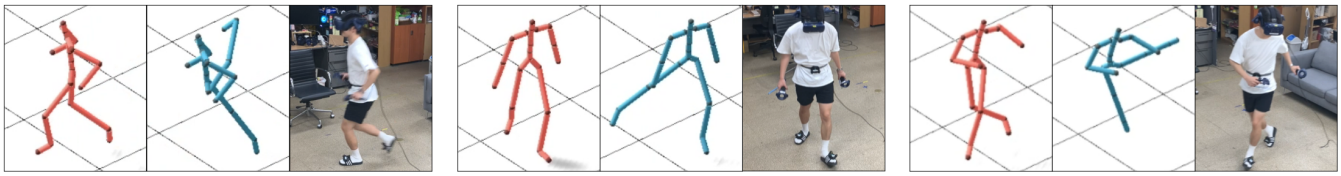

Figure 10: *Comparison of robustness to feet tracking loss against 6-point tracking (Red: Ours, Green: 6-point tracking).*

Appendix

1. Velocity Computation

Let (q_i^{ref}, p_i^{ref}) and $(q_i^{joint}, p_i^{joint})$ denote the world orientation and position of the reference joint and descendent joints at the i^{th} frame, respectively. Then the equations to compute velocity $\mathbf{d}_i = [d_i^{ref}, [d_i^{joint,k}]_{k=1}^3]$ are given below:

$$
\begin{aligned}
d_i^{ref} &= [v_i^{ref}, w_i^{ref}] \\
v_i^{ref} &= (q_i^{ref})^{-1} * (p_i^{ref} - p_{i-1}^{ref}) \\
w_i^{ref} &= (q_i^{ref})^{-1} * (q_{i-1}^{ref})^{-1} * q_i^{ref}
\end{aligned}
$$

$$
\begin{aligned}
d_i^{joint} &= [v_i^{joint}, w_i^{joint}] \\
v_i^{joint} &= (p_i'^{joint} - p'^{joint}_{i-1}), \\
p' &= (q^{ref})^{-1} * (p^{joint} - p^{ref}) \\
w_i{}^{joint} &= (q'^{joint}_{i-1})^{-1} * q_i'^{joint} \\
q' &= (q^{ref})^{-1} * q^{joint}.
\end{aligned}
$$

2. Dataset Formation

The goal of this research is to reconstruct general human motions from sparse tracking signals. We built a dataset combining parts of three different motion capture datasets. PFNN dataset [HKS17] is selected as the main dataset for containing various locomotion styles and transitions. We only used general locomotion animations of the PFNN dataset and excluded animations of climbing up and jumping. Selected animations consist of different motion categories, including stand, walk, jog, and run; the categories are randomly placed in an animation sequence and various transitions, such as normal turn, side steps, and back steps, exist between motion categories.

To cover in-place motions such as sit, stand up, and upper-body gestures, we selected animations in the CMU motion capture dataset [13] and MHAD dataset [OCK*13], and synthesized sequential animations for the combined dataset by connecting, clipping, and looping the selected animations.

All raw motion capture animations were down-sampled to 45 fps and the root position and joint offsets were edited (1. scaling parameter: 0.0594, 2. root height shift: -0.05m) to match the root height of 1m in T-pose. Table 4 shows the ratio and the total number of frames of different action categories. Tables 5 and 6 show the sources and names of animations consisting of the training and test sets, respectively.

Table 4: *Ratio of action categories in the dataset.*

Category	Frames	Duration (min)	Ratio
Locomotion	141,340	52.35	84.0%
Sit/Stand	7,548	2.80	4.5%
Upper-body motion	19,402	7.19	11.5%
Total	**168,290**	**62.34**	**100%**

Tracking Signal Augmentation. To augment imaginary tracking signals from motion data, we first computed the world transformations for the 4 joints of the head, finger-bases, and root that correspond to Head-Mounted display (HMD), two hand-held controllers, and pelvis tracker, respectively. The VIVE Lighthouse tracking system has an expected accuracy of 2mm [Kre16]. To apply this drifting behavior of the trackers to the training data, we added random vectors with scale sampled from a normal distribution $\mathcal{N}(0, 0.01)$, to the position of joints. For rotational noise, random rotations with a maximum angle of 1.5 degrees were applied to the joint world rotations.

Table 5: *Training set composition.*

Category	Source	Name
Locomotion	PFNN	NewCaptures01_000 NewCaptures02_000 LocomotionFlat01_000 LocomotionFlat02_000 LocomotionFlat03_000 LocomotionFlat04_000 LocomotionFlat05_000 LocomotionFlat06_000 LocomotionFlat07_000 LocomotionFlat08_000 LocomotionFlat10_000
Sit/Stand	MHAD	skl_s01_a10&a11_r01 skl_s01_a10&a11_r02 skl_s01_a10&a11_r03 skl_s01_a10&a11_r04 skl_s01_a10&a11_r05
Upper-body motion	CMU	15_05 111_22

Table 6: *Test set composition.*

Category	Source	Name
Locomotion	PFNN	LocomotionFlat02_001 LocomotionFlat06_001 LocomotionFlat08_001 LocomotionFlat11_000
Sit/Stand	MHAD	skl_s02_a10&a11_r01 skl_s02_a10&a11_r02
Upper-body motion	CMU	32_11 29_18

Contact Labeling. Each frame of a motion clip has binary contact labels for two toe-bases. We simply labeled that a toe-base is in contact if the height of the toe-base joint is below 1cm, which gave us a reasonable result for the dataset.

3. Details on Network Training

The number of training epoch is 1,500. We use Adam optimizer with an initial learning rate of 10^{-3} with a decaying rate of 0.999 every epoch. We use a batch size of 256; to form a batch, 256 animations are randomly selected from the set with the probability of $p_{anim} = \frac{f_{anim}}{\sum f}$, where f is the number of frames of the animation. Finally, a 45-frame chunk is randomly picked from each selected animation to form a batch of dimension $R^{256\times45\times(4\times9+1)}$.

References

[13] "Carnegie-Mellon Motion Capture Database". (2013). URL: %5Curl%7Bhttps://mocap.cs.cmu.edu%7D 2, 10.

[CAW*19] CHIU, HSU-KUANG, ADELI, EHSAN, WANG, BORUI, et al. "Action-agnostic human pose forecasting". *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2019, 1423–1432 3.

[CGCB14] CHUNG, JUNYOUNG, GULCEHRE, CAGLAR, CHO, KYUNGHYUN, and BENGIO, YOSHUA. "Empirical evaluation of gated recurrent neural networks on sequence modeling". *arXiv preprint arXiv:1412.3555* (2014) 4.

[CH05] CHAI, JINXIANG and HODGINS, JESSICA K. "Performance animation from low-dimensional control signals". *ACM SIGGRAPH 2005 Papers*. 2005, 686–696 2.

[CYW*19] CHENG, YU, YANG, BO, WANG, BO, et al. "Occlusion-aware networks for 3d human pose estimation in video". *Proceedings of the IEEE International Conference on Computer Vision*. 2019, 723–732 3.

[Dee18] DEEPMOTION. "How To Make 3 Point Tracked Full-Body Avatars in VR". (Apr. 2018). URL: https://blog.deepmotion.com/2018/04/30/how-to-make-3-point-tracked-full-body-avatars-in-vr/ 2, 3.

[GMHP04] GROCHOW, KEITH, MARTIN, STEVEN L, HERTZMANN, AARON, and POPOVIĆ, ZORAN. "Style-based inverse kinematics". *ACM SIGGRAPH 2004 Papers*. 2004, 522–531 2.

[HKA*18] HUANG, YINGHAO, KAUFMANN, MANUEL, AKSAN, EMRE, et al. "Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time". *ACM Transactions on Graphics (TOG)* 37.6 (2018), 1–15 3, 8.

[HKPP20] HOLDEN, DANIEL, KANOUN, OUSSAMA, PEREPICHKA, MAKSYM, and POPA, TIBERIU. "Learned motion matching". *ACM Transactions on Graphics (TOG)* 39.4 (2020), 53–1 2.

[HKS17] HOLDEN, DANIEL, KOMURA, TAKU, and SAITO, JUN. "Phase-functioned neural networks for character control". *ACM Transactions on Graphics (TOG)* 36.4 (2017), 1–13 2, 10.

[Hol18] HOLDEN, DANIEL. "Robust solving of optical motion capture data by denoising". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 1–12 5.

[HS97] HOCHREITER, SEPP and SCHMIDHUBER, JÜRGEN. "Long short-term memory". *Neural computation* 9.8 (1997), 1735–1780 4.

[HSK16] HOLDEN, DANIEL, SAITO, JUN, and KOMURA, TAKU. "A deep learning framework for character motion synthesis and editing". *ACM Transactions on Graphics (TOG)* 35.4 (2016), 1–11 2.

[HSKJ15] HOLDEN, DANIEL, SAITO, JUN, KOMURA, TAKU, and JOYCE, THOMAS. "Learning motion manifolds with convolutional autoencoders". *SIGGRAPH Asia 2015 Technical Briefs*. 2015, 1–4 6.

[JL20] JANG, DEOK-KYEONG and LEE, SUNG-HEE. "Constructing Human Motion Manifold With Sequential Networks". *Computer Graphics Forum*. Wiley Online Library. 2020 2.

[JYF16] JIANG, FAN, YANG, XUBO, and FENG, LELE. "Real-time full-body motion reconstruction and recognition for off-the-shelf VR devices". *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*. 2016, 309–318 3.

[KPKH20] KIM, SANGBIN, PARK, INBUM, KWON, SEONGSU, and HAN, JUNGHYUN. "Motion Retargetting based on Dilated Convolutions and Skeleton-specific Loss Functions". *Computer Graphics Forum*. Vol. 39. 2. Wiley Online Library. 2020, 497–507 4.

[Kre16] KREYLOS, OLIVER. "Lighthouse tracking examined". (May 2016). URL: http://doc-ok.org/?page_id=6 10.

[LO19] LIN, JAMES and O'BRIEN, JAMES. "Temporal IK: Data-Driven Pose Estimation for Virtual Realiity". (2019) 2, 3.

[LWC*11] LIU, HUAJUN, WEI, XIAOLIN, CHAI, JINXIANG, et al. "Realtime human motion control with a small number of inertial sensors". *Symposium on interactive 3D graphics and games*. 2011, 133–140 2.

[LWH*12] LEVINE, SERGEY, WANG, JACK M, HARAUX, ALEXIS, et al. "Continuous character control with low-dimensional embeddings". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 1–10 2.

[LZCV20] LING, HUNG YU, ZINNO, FABIO, CHENG, GEORGE, and VAN DE PANNE, MICHIEL. "Character controllers using motion vaes". *ACM Transactions on Graphics (TOG)* 39.4 (2020), 40–1 2.

[LZWM06] LIU, GUODONG, ZHANG, JINGDAN, WANG, WEI, and MCMILLAN, LEONARD. "Human motion estimation from a reduced marker set". *Proceedings of the 2006 symposium on Interactive 3D graphics and games*. 2006, 35–42 2.

[MBR17] MARTINEZ, JULIETA, BLACK, MICHAEL J, and ROMERO, JAVIER. "On human motion prediction using recurrent neural networks". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, 2891–2900 6, 8.

[OCK*13] OFLI, FERDA, CHAUDHRY, RIZWAN, KURILLO, GREGORIJ, et al. "Berkeley mhad: A comprehensive multimodal human action database". *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE. 2013, 53–60 2, 10.

[PFAG19] PAVLLO, DARIO, FEICHTENHOFER, CHRISTOPH, AULI, MICHAEL, and GRANGIER, DAVID. "Modeling human motion with quaternion-based neural networks". *International Journal of Computer Vision* (2019), 1–18 4.

[PGA18] PAVLLO, DARIO, GRANGIER, DAVID, and AULI, MICHAEL. "Quaternet: A quaternion-based recurrent model for human motion". *arXiv preprint arXiv:1805.06485* (2018) 4.

[RF20] ROCKWELL, CHRIS and FOUHEY, DAVID F. "Full-body awareness from partial observations". *arXiv preprint arXiv:2008.06046* (2020) 3.

[Roo17] ROOT-MOTION. "FINAL-IK". (2017) 5.

[SAA*20] SHI, MINGYI, ABERMAN, KFIR, ARISTIDOU, ANDREAS, et al. "MotioNet: 3D Human Motion Reconstruction from Monocular Video with Skeleton Consistency". *ACM Transactions on Graphics (TOG)* 40.1 (2020), 1–15 4.

[SHP04] SAFONOVA, ALLA, HODGINS, JESSICA K, and POLLARD, NANCY S. "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces". *ACM Transactions on Graphics (ToG)* 23.3 (2004), 514–521 2.

[SZKZ20] STARKE, SEBASTIAN, ZHAO, YIWEI, KOMURA, TAKU, and ZAMAN, KAZI. "Local motion phases for learning multi-contact character movements". *ACM Transactions on Graphics (TOG)* 39.4 (2020), 54–1 2, 6.

[TCW19] THOMASSET, VINCENT, CARON, STÉPHANE, and WEISTROFFER, VINCENT. "Lower body control of a semi-autonomous avatar in Virtual Reality: Balance and Locomotion of a 3D Bipedal Model". *25th ACM Symposium on Virtual Reality Software and Technology*. 2019, 1–11 3.

[VYCL18] VILLEGAS, RUBEN, YANG, JIMEI, CEYLAN, DUYGU, and LEE, HONGLAK. "Neural kinematic networks for unsupervised motion retargetting". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 8639–8648 4.

[WGR*19] WOUDA, FRANK J, GIUBERTI, MATTEO, RUDIGKEIT, NINA, et al. "Time Coherent Full-Body Poses Estimated Using Only Five Inertial Sensors: Deep versus Shallow Learning". *Sensors* 19.17 (2019), 3716 3, 4.