

Received February 10, 2020, accepted March 5, 2020, date of publication March 12, 2020, date of current version March 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980316

A Perceptual-Based Noise-Agnostic 3D Skeleton Motion Data Refinement Network

SHU-JIE LI, HAI-SHENG ZHU, LI-PING ZHENG, AND LIN LI 

Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei 230601, China
School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China

Corresponding author: Lin Li (lilin_julia@hfut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61877016 and Grant 61972128, in part by the Fundamental Research Funds for the Central Universities under Grant JZ2018HGTA0215, and in part by the National Natural Science Foundation of China under Grant 61972129 and Grant 61602146.

ABSTRACT In this paper, we demonstrate a perceptual-based 3D skeleton motion data refinement method based on a bidirectional recurrent autoencoder, called BRA-P. Three main technical contributions are made by the proposed network. First, the proposed BRA-P can address noisy data with different noise types and amplitudes using one network, and this attribute makes the approach more suitable for raw motion data with heterogeneous mixed noise. Second, due to the usage of perceptual loss, which measures the difference in high-level features extracted by a pretrained perceptual autoencoder, BRA-P improves the perceptual similarity between refined motion data and clean motion data, especially for the case where the noisy data and target clean data have different topologies. Third, BRA-P further improves the bone-length consistency and smoothness of the refined motion using the perceptual autoencoder as a postprocessing network. Ablation experiments verify the effect of the three technical contributions of our approach. The results of the experiments on synthetic noise data and raw motion data captured by Kinect demonstrate that our method outperforms several state-of-the-art methods in the cleaning of mixed-noise data by one network.


INDEX TERMS 3D skeleton motion data refinement, noise-agnostic, perceptual constraint, motion autoencoder, Kinect.

I. INTRODUCTION

Human motion data are widely used in virtual reality, human-computer interactions, computer games, sports and medical applications [1]–[4]. Human motion capture is a prevalent technique that aims to supply highly precise human motion data. Professional motion capture sensors such as Vicon [5] and Xsens [6] can offer motion data with high precision but are too expensive for home use. Furthermore, these mocap systems are not convenient to wear because users must wear capture suits. In recent decades, certain low-cost motion capture technologies, such as depth sensor-based and camera-based technologies, have been developed and can serve as alternatives for capturing human motion. However, the raw 3D skeleton motion data captured by these low-cost sensors are often noisy, occluded or incomplete for several reasons, such as calibration error, sensor noise, poor sensor resolution, and occlusion due to body parts or clothing. Therefore, raw

mocap data should be refined, i.e., missing data should be filled in and denoising should be performed, before the data are used [7]–[10].

With the rapid development of deep learning, the advantages of this method have been demonstrated in motion data refinement. However, motion data refinement based on deep learning is still an open problem. For example, if the optimization target of the algorithms is only the minimization of the mean square error (MSE) of the joint position between the refined motion and the label motion, i.e., the reproduction error [10], the kinematic information of the motion data is not fully exploited by the network, which causes the refined motion to lack perceptual similarity with the clean data. Mall *et al.* [11] noted that the result of encoder-bidirectional-decoder (EBD), which is only trained by reproduction error, is still somewhat noisy. Thus, these researchers trained an encoder-bidirectional-filter (EBF) network to postprocess EBD results. Holden [12] also used a smoothing step to filter jittery movements, but postprocessing steps are time-consuming and not suitable for real-time

The associate editor coordinating the review of this manuscript and approving it for publication was Jonghoon Kim .

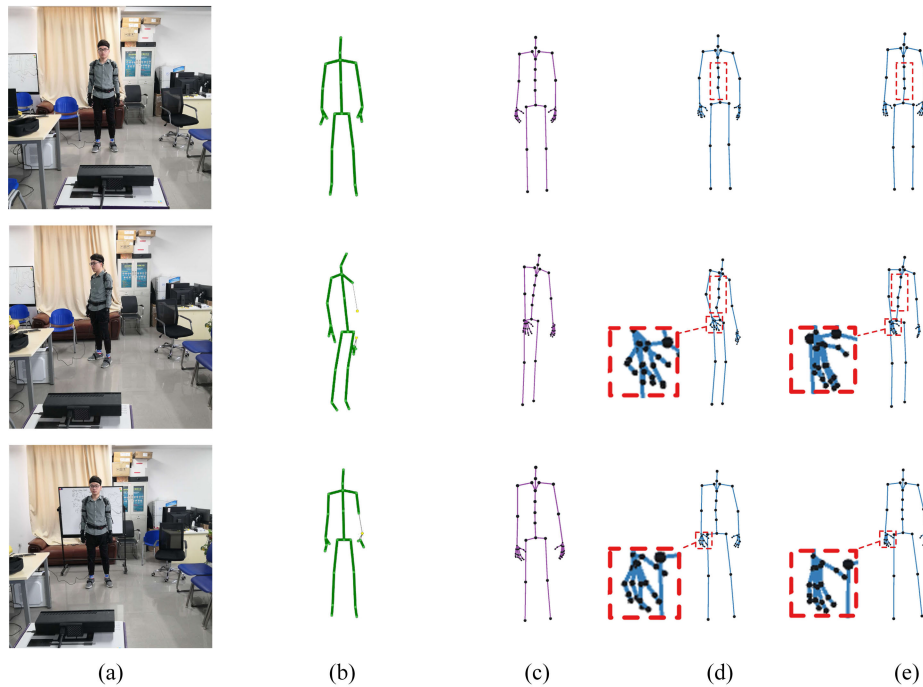


FIGURE 1. Examples of skeleton motion data captured by Kinect and the refined result of our proposed BRA-P. The first and second rows have the same background but different poses. The first and third rows have different backgrounds but the same pose. Obviously, the three different poses captured by Kinect indicate that different orientations and backgrounds can generate different types of noise in the motion data. Therefore, the human skeleton data captured by Kinect are mixed-noise data. (a) Original captured color images. (b) Pose captured by Kinect. (c) Pose captured by NOITOM Mocap. The skeletons in (b) and (c) have different skeleton topologies. (d) Pose refined by BRA [13] in which malformed or unnatural parts are squared. (e) Pose refined by the proposed BRA-P. The results show that BRA-P can improve the problems that BRA has when optimizing raw data captured by Kinect.

motion data acquisition systems. Li *et al.* [13] proposed a bidirectional recurrent autoencoder that can improve the kinematic information expression ability of the network by imposing smoothness and bone-length constraints. However, unfortunately, smoothness and bone-length constraints cannot satisfactorily maintain the kinematic information, and the noisy data and target clean data have different skeleton topologies. As shown in Fig. 1, the data captured by Kinect and the poses captured by Mocap have different skeleton topologies, and the BRA results have malformed or unnatural components. Furthermore, the raw mocap data, such as the skeleton motion data captured by Kinect, often contain mixed noise with different noise types and noise amplitudes due to changes in background or human posture orientation during capture. Fig. 1 shows that the raw data captured by Kinect contain different types of noise when the background or the body orientation changes. Hence, the refinement approach for raw mocap data should have the ability to remove the heterogeneous mix of noise through one network, i.e., the network should be noise-agnostic. Therefore, in summary, the objectives of this paper are to propose a network that is noise-agnostic and to further improve the kinematic information expression ability of the network.

In line with [13], we also use the bidirectional long short-term memory recurrent neural network (B-LSTM-RNN) architecture [14], [15] to refine noisy motion data. Our previous work in [13] noted that the refinement network based on the B-LSTM-RNN architecture does not require noise amplitude as a priori knowledge. In this paper, we found that the B-LSTM-RNN architecture network also does not require the noise type as prior knowledge. As a result, the network can be noise-agnostic. At the same time, we improve the kinematic information expression ability of the network by imposing perceptual constraint based on a pretrained perceptual autoencoder. Perceptual loss functions, which are based on high-level features extracted from pretrained networks, are widely used in generative adversarial networks [16]–[21] to synthesize high-quality images or textures. Those works can generate high-quality images due to perceptual losses that measure image similarities more robustly than per-pixel losses [16]. Inspired by this idea, our strategy is to pretrain a perceptual autoencoder using clean motion data. Subsequently, we train a denoising autoencoder for refinement tasks using the perceptual loss, which is defined based on this perceptual motion autoencoder. As shown in Fig. 2, $H_R = E_p(X_R)$, which is the hidden unit of X_R calculated by the

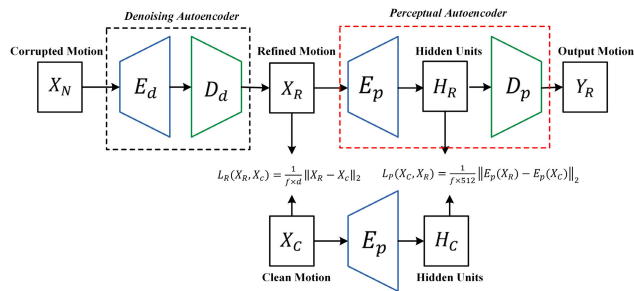


FIGURE 2. Pipeline of our approach. Our strategy is to train a denoising autoencoder consisting of E_d and D_d by noisy-clean motion pairs X_N and X_C based on a pretrained perceptual autoencoder trained by clean motion data and consisting of E_p and D_p . The loss function for the perceptual autoencoder consists of three elements, i.e., reproduction loss, smooth loss and bone-length loss, which are the same as the losses used in [13]. In contrast, the loss function for training the denoising autoencoder consists of four elements: perceptual loss, reproduction loss, smooth loss and bone-length loss. At training time, we optimize the output of the denoising network X_R by the loss function. At runtime, the output of the perceptual autoencoder Y_R is used as the final refined result.

perceptual autoencoder, has a close distance with the hidden units of the ground truth X_C due to the use of perceptual loss. The similarity in the hidden unit space of the clean data and refined data can help the denoising autoencoder learn additional kinematic information from noisy-clean motion pairs. On the other hand, Holden *et al.* [22] reported that the projection from the hidden unit space to motion space can generate a smooth and natural motion. Inspired by this idea and to further improve the quality of the refined motion, we use the perceptual autoencoder to postprocess X_R , i.e., we use $Y_R = D_p(H_R)$ as the final refined result. The experiment also shows that this postprocessing step is effective in improving bone-length consistency and smoothness. In previous work, the pretrained classification networks are only used for calculating perceptual loss, but in our approach, the pretrained network is an end-to-end autoencoder used to return a perceptual loss and to further postprocess the refined result. Fig. 2 shows the pipeline of our BRA-P approach.

We demonstrate the performance of our approach by training and testing it on a synthetic mixed-noise dataset generated by the CMU human motion dataset [23] and a raw skeleton dataset captured by Kinect. The experiments on the synthetic dataset can explain why the B-LSTM-RNN architecture is more suitable for mixed-noise data. On the raw motion dataset captured by Kinect, we validate each component of our approach via an ablation study and show the superior performance of our approach in the removal of mixed noise by comparing our approach with the state-of-the-art baseline.

In summary, our contributions are three-fold and are described as follows:

- 1) Our approach is noise-agnostic. The experiment on a synthetic mixed-noise dataset shows that our approach can address noisy data with different noise types and amplitudes using one network. This attribute makes our approach more suitable for raw motion data with mixed noise, as verified by experiments on the Kinect motion dataset.

- 2) Our approach improves the perceptual similarity between the refined motion data and clean motion data. By imposing perceptual loss during training, our network can better maintain the motion characteristics, especially for noisy and clean data with different topologies.
- 3) Our approach further improves the bone-length consistency and smoothness of the refined motion via the postprocessing step using the perceptual autoencoder.

The remainder of this paper is organized as follows. Section II gives a review of the related work and positions the proposed approach with respect to earlier work. Section III discusses the details of our proposed approach. The experimental results are presented and discussed in Section IV. Finally, in Section V, the conclusions of this work are presented and future research directions are discussed.

II. RELATED WORK

Many studies have been devoted to the refinement of corrupted motion data and have yielded encouraging results. Our approach is data-driven, and consequently, we mainly give a categorized overview of the related data-driven methods in this section.

A. FILTER-BASED METHODS

Standard signal denoising filters are the typical non-data-driven methods used in early research [13], but those non-data-driven filters [24]–[30] cannot preserve the spatial-temporal information embedded in human motion because these methods process each degree of freedom separately [31]. The groundbreaking work of data-driven filters was proposed by Lou and Chai [31] and can maintain the spatial-temporal patterns in human motion data. Their method can automatically train a series of spatial-temporal filter bases from prerecorded human motion data and use them along with robust statistical techniques to filter noisy motion data. However, this method cannot recover certain motion details because this method uses the Singular Value Decomposition (SVD) technique to choose only a set of orthogonal filter bases for filtering noisy motions. Another famous work by Akhter *et al.* [32] proposed a bilinear model that factors the basis into spatial and temporal variations and unifies the coefficients; hence, the model simultaneously exploits spatial and temporal regularity. This method cannot handle many different types of motion and noise altogether because the number of basis vectors should be determined based on the motion type before the denoising procedure.

B. SPARSE-REPRESENTATION-BASED METHODS

Sparse representation has become a hot research topic in the past decade and has been used to solve the problem of motion refinement. In 2011, Xiao *et al.* [33] proposed the prediction of missing markers in terms of finding an 11-sparse representation for the existing data of an incomplete pose. In 2015, Xiao *et al.* [34] and Feng *et al.* [35] divided each human

pose into five partitions and presented five dictionaries for those partitions to obtain a fine-grained pose representation, but this approach abandons the relationships between these partitions. In 2016, Xia *et al.* [36] point out that sparse coding and low-rank matrix completion only take the basic statistical properties of human motion into consideration, so the authors recover incomplete motions using sparse representations with smoothness and bone-length constraints, which includes the kinematic information in the recovery process. However, all of the above data-driven methods are action specific or noise specific; that is, each action type or noise type requires a separate refinement model.

C. DIMENSIONALITY-REDUCTION-BASED METHODS

Dimensional reduction can be used to eliminate noisy components of data and can be realized via principal component analysis (PCA) [37]. In 2006, Liu and McMillan [38] first modeled the motion sequences of a training set via principal component analysis and recovered a new sequence by finding the least squares solutions based on the available marker positions and the principal components of the associated model. In the same year, Tangkuampien and Suter [39] showed that the greedy KPCA (kernel PCA) algorithm can be applied to filter exemplar poses and build a reduced training set that optimally describes the entire sequence. Therefore, this approach has superior denoising qualities and lower evaluation costs compared with PCA. In 2007, Günter *et al.* [40] proposed a rapid iterative KPCA method that improved the convergence speed for denoising human motion capture data. However, dimensionality-reduction-based methods discard the temporal or spatial correlations of data, which leads to an overparameterization of the data [32], changing the structures in the original motion data.

D. REFINEMENT NEURAL NETWORKS

Recently, neural networks have displayed remarkable advantages in many machine learning tasks such as computer vision, image processing, pattern recognition, and natural language processing. Increasingly, neural networks have also been exploited for motion data refinement and have achieved state-of-the-art results. In 2007, Taylor *et al.* [41] used a restricted Boltzmann machine (RBM) to model the probability distribution of the observation vector at each time frame, and after training, the model could perform online filling of missing data during motion capture. In 2015, Fragkiadaki *et al.* [42] proposed an encoder-recurrent-decoder (ERD) model for predicting the mocap vector in the next frame from the past motion sequence. In 2017, Butepage *et al.* [43] proposed a fully connected network that could predict missing data of latter sequence from past information in the motion sequence. Mall *et al.* [11] trained a set of filters using a deep, bidirectional, recurrent framework for clean, noisy and incomplete mocap data. In 2019, Cui *et al.* [44] proposed a bidirectional attention network for missing data recovery, and their embedded attention mechanism can decide where to borrow information from

and use this information to recover corrupted frames. The above deep-learning-based methods are action agnostic but noise specific, i.e., these methods can be trained by large-scale data with a specified type of noise (such as Gaussian noise or missing data) and a heterogeneous mix of action types, and the network can refine any action with that noise type. In fact, the raw data captured by low-cost mocap sensors are often datasets with heterogeneous mixes of action types, noise types and noise amplitudes. As a result, the more suitable the approach is for mixed-noise data, the more suitable the approach is for raw data. For raw data refinement, in 2015, Holden *et al.* [10] used a convolutional autoencoder for denoising motion captured by Kinect. In 2018, Holden [12] used a deep residual network for mapping raw optical motion capture data to skeleton data. In the same year, Huang *et al.* [45] proposed a bidirectional recurrent framework for reconstruction of full body poses in real time from data captured by 6 IMUs. However, the results of these methods are still somewhat jittery, and [12] requires postprocessing to refine the results. In computer vision, certain works have addressed the problem of motion refinement. In 2018, Fieraru *et al.* [7] noted that even state-of-the-art models of human pose estimation from images or videos fail to correctly localize all the body joints, thus these researchers proposed a pose refinement network that takes both the image and a given pose estimate as input and learns to directly predict a refined pose by joint reasoning of the input-output space. Moon *et al.* [8] presented a model-agnostic pose refinement method to estimate a refined pose from a tuple of an input image and a pose. These two state-of-the-art works for pose refinement both require an image as clean information to refine the pose. In 2019, Li *et al.* [13] proposed an autoencoder based on B-LSTM-RNN for 3D motion data refinement and displayed its advantages regarding the visual quality of the refined motion. In this paper, we improve the performance of BRA using a perceptual constraint. The use of the perceptual constraint allows the network output to better maintain the motion characteristics, and during the runtime, the output of the perceptual autoencoder is used as the final refined result. Based on the two advantages of our approach mentioned above, we demonstrate its advantages over a variety of baselines via extensive experiments on both a synthetic mixed-noise dataset and a raw skeleton dataset captured by Kinect.

III. PROPOSED METHOD

A. DATA FORMULATION

Two datasets are used in this paper. The first is a synthetic dataset generated by the CMU human motion database [23], and the second is a raw motion dataset synchronously captured by Kinect and the NOITOM mocap system [46].

1) SYNTHETIC DATASET

We perform selected preprocessing steps on the CMU human motion database, similar to those in [22], including subsampling of all motion in the database to 60 frames

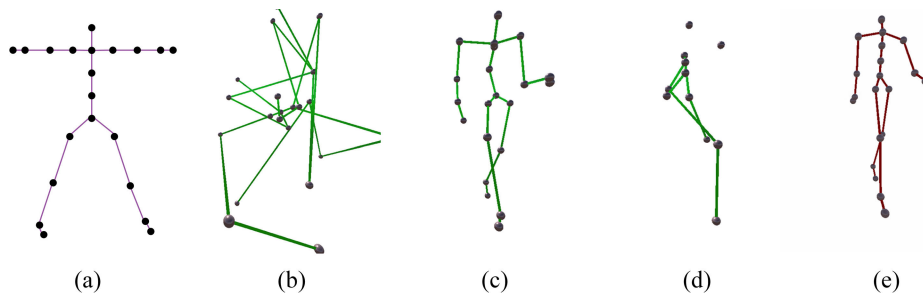


FIGURE 3. Data in the synthetic dataset based on the CMU human motion database. (a) T-pose of the CMU human motion data, containing 21 joints. (b)-(d) Three examples of synthetic noise data with high-amplitude Gaussian noise, low-amplitude Gaussian noise and randomly missing noise. (e) Clean data in the CMU motion database. The noisy and clean data have the same skeleton topology.

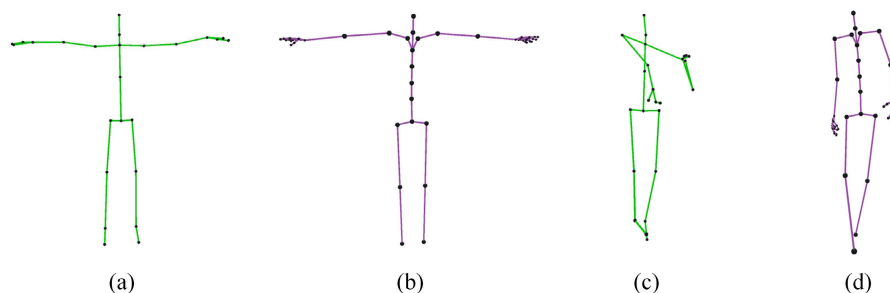


FIGURE 4. Data in the raw motion dataset. (a) T-pose of skeleton captured by Kinect. (b) T-pose of skeleton captured by NOITOM Mocap. (c) Noisy pose captured by Kinect. (d) Corresponding clean pose of (c) captured by NOITOM Mocap. The noisy and clean data in DS_{raw} have different skeleton topologies.

per second, conversion of the data from the joint angle representation in the original dataset to the 3D joint position format, and the transformation of the all joint positions to the local coordinate system, the origin of which is located on the ground and onto which the root position is projected. Only 21 of the most important joints are preserved, and thus the dimension of each posture is 63 ($21 \times 3 = 63$), where 3 is the number of channels of each joint (each joint contains three channels: X, Y, and Z). The mean pose is subtracted from the data and, and then the data are divided by the standard deviation to normalize the scale of the skeleton. However, the rotational velocity of the body around the vertical axis does not need to be removed from each frame because this preprocessing step is time-consuming and is not suitable for real-time use. The entire CMU database is separated into N_{CMU} overlapping clips of f frames (overlapped by $f/2$ frames), and all of these motion clips consist of DS_{CMU} . No fixed motion clip length is recommended, and we set $f = 120$. Let $X_{CMU} = [p_1, p_2, \dots, p_{120}]^T \in DS_{CMU}$ denote a motion chip, where $p_t = [x_{t,1}, y_{t,1}, z_{t,1}, \dots, x_{t,J}, y_{t,J}, z_{t,J}]$ represents the t -th frame, $J = 21$ is the number of skeleton joints, 120 is the length of the motion clip, and X'_{CMU} is used to represent the noise motion clip synthesized by X_{CMU} . All of the noisy-clean motion pairs consist of a synthetic dataset $DS_{syn} = \{[X'_{CMU}, X_{CMU}]\}$.

2) RAW MOTION DATASET

The raw motion datasets consist of many daily actions, similar to the CMU motion dataset (i.e., walking, jumping, dancing, basketball, box, etc.). Similar to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database [47], the data captured by two different sensors were synchronized using the network time protocol. The skeleton data captured by Kinect do not need preprocess. But the data in the bvh format captured by the NOITOM mocap system are converted to 3d joint position format using the actor skeleton, which has a height of 175 centimeters in a neutral pose. All the activity in the dataset is executed by one actor, and hence, the bone length of all the poses is treated as a constant.

The joint number of Kinect is 25, and hence, the dimension of each Kinect posture is 75 ($25 \times 3 = 75$). The number of skeleton joints in the NOITOM Mocap data is 59, including 40 hand joints, and hence, the dimension of each NOITOM Mocap posture is 177 ($59 \times 3 = 177$). The mean pose is subtracted from the data and then the data are divided by the standard deviation to normalize the scale of the skeleton. The Kinect skeleton data are treated as noisy data, and the NOITOM Mocap data are treated as clean data. Let X_{Kinect} and X_{Mocap} represent synchronal pairs, and all of these pairs consist of raw motion dataset $DS_{raw} = \{[X_{Kinect}, X_{Mocap}]\}$.

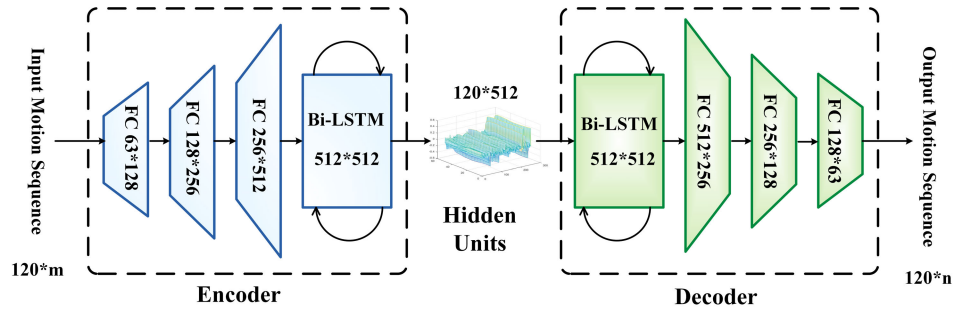


FIGURE 5. Architecture of motion autoencoder.

The dataset and the bone length information are provided publicly on GitHub.¹

B. NETWORK ARCHITECTURE

Our network is composed of two motion autoencoders, a perceptual autoencoder and a denoising autoencoder. In this paper, the two motion autoencoders share the same architecture. The network architecture is the same as that used in [13]. As shown in Fig. 5, the autoencoder has two components, the encoder and the decoder. The encoder receives the input motion clip X and outputs the encoded values H in the hidden unit space. The encoder operation is as follows:

$$H = E(X) = BiLSTM_1(W_3(W_2(W_1(X) + b_1) + b_2) + b_3), \quad (1)$$

where $W_1 \in R^{m \times 128}$, $b_1 \in R^{128}$, $W_2 \in R^{128 \times 256}$, $b_2 \in R^{256}$, $W_3 \in R^{256 \times 512}$, $b_3 \in R^{512}$. The decoder receives the hidden unit H , and outputs the reproduced motion clip Y . The decoder operation is as follows:

$$Y = D(X) = W_6(W_5(W_4(BiLSTM_2(H)) + b_4) + b_5) + b_6, \quad (2)$$

where $W_4 \in R^{512 \times 256}$, $b_4 \in R^{256}$, $W_5 \in R^{256 \times 128}$, $b_5 \in R^{128}$, $W_6 \in R^{128 \times n}$, $b_6 \in R^n$. In this work, $BiLSTM_1$ and $BiLSTM_2$ are bidirectional LSTM cells that both input and output sequences of size 120×512 . For X'_{CMU} and X_{CMU} training pairs, $m = n = 63$. For X_{Kinect} and X_{Mocap} training pairs, $m = 75$ and $n = 177$.

C. LOSS FUNCTIONS

For convenience, we uniformly use X_C to present the clean data and X_N to present the noisy data in the two training datasets DS_{syn} and DS_{raw} . Four loss functions are used during the training time.

1) REPRODUCTION LOSS

The autoencoder receives the input motion clip and outputs the reproduced motion clip, thus we expect that the output of the autoencoder is the clean ground-truth motion sequence. A joint-position wise loss function such as the mean square

loss (MSE) is the most commonly used loss function and guarantees that the reproduced motion has the minimum Euclidean distance with the clean motion clip. We define the reproduction loss by

$$L_R(Y, X) = \frac{1}{f \times d} \|Y - X\|_2, \quad (3)$$

where $\|\cdot\|_2$ denotes the l2-norm, f is the frame number of the motion clip, and d is the dimension of each posture.

2) PERCEPTUAL LOSS

Although the reproduction loss guaranteeing the reproduced motion has the minimum Euclidean distance with the clean motion clip, the solution of the MSE optimization problems often lacks smoothness and bone length, which results in perceptually unsatisfying solutions. We use a perceptual loss function to obtain perceptually satisfying solutions. Based on the aforementioned perceptual autoencoder, the perceptual loss of a reproduced motion clip X_R is defined as the Euclidean distance between the hidden units of itself and its corresponding clean data X_C :

$$L_P(X_C, X_R) = \frac{1}{f \times 512} \|E_p(X_R) - E_p(X_C)\|_2. \quad (4)$$

3) SMOOTHNESS LOSS

Smoothness loss has been used in many studies, such as data-driven method [36] and non-data-driven method [48], to yield natural motion sequences. These studies note that natural human motion should be smooth in the temporal direction. In addition to reproduction loss and perceptual loss, we also add spatial coherence regularizations to encourage neighboring frames to have continuity. The studies in [13], [36], [45] enforce C^2 continuity on each feature dimension of the motion clip via a smoothness penalty term. Repeating the border elements of X yields X' : $X'_{1,i} = X'_{2,i} = X_{1,i}$ and $X'_{n+1,i} = X'_{n,i} = X_{n,i}$, where $1 \leq i \leq d$. Let O be a symmetric matrix:

$$O = \begin{pmatrix} -1 & 1 & 0 & & & \\ 1 & -2 & 1 & & & \\ & & \ddots & \vdots & & \\ & & & \ddots & 1 & \\ & & & & 1 & -1 \end{pmatrix}_{(f+2) \times (f+2)}. \quad (5)$$

¹<https://github.com/vcc-zhu/BRA-P-Kinect2Mocap>

Given an input motion clip X , we define the smoothness loss as:

$$L_S(X) = \frac{1}{(f+2) \times d} \|OX'\|_2. \quad (6)$$

4) BONE-LENGTH LOSS

The skeleton of the character is a kinematic model composed of several bones and joints [29], [49]. A bone is a segment of a fixed length, and a joint is the end point of a bone. The bone length of such a kinematic model should maintain consistency among all the frames. We used similar bone-length loss in [13]. Let l_b denotes the bone length of b -th bone of skeleton. Given an input motion clip X , the cost in terms of a penalty for bone length can be written as follows:

$$L_B(X) = \frac{1}{f \times (J-1)} \sum_{i=1}^f \sum_{b=1}^{J-1} | \|p_b^{i,1}(X) - p_b^{i,2}(X)\|_2 - l_b |, \quad (7)$$

where $p_b^{i,1}(X)$ and $p_b^{i,2}(X)$ are the 3D positions of the two end joints of the b -th bone of frame i which is recorded in X , and J is the joint number of skeleton..

D. TRAINING DETAILS

The entire training procedure can be divided into two sub-procedures. The first training task is to train an encoder $E_p : X_C \rightarrow H_C$ and a decoder $D_p : H_C \rightarrow X'_C$ which consist of the perceptual autoencoder. The loss function for training is as follows:

$$L_p = \lambda_{p1} L_R(X_C, X'_C) + \lambda_{p2} L_S(X'_C) + \lambda_{p3} L_B(X'_C), \quad (8)$$

where the weights λ_{p1} , λ_{p2} and λ_{p3} balance the importance of each loss, X_C and X'_C are clean and reproduced data, respectively. Adding the smoothness loss and bone-length loss can improve the quality of the output motion clip. The choices for the three weights are not crucial. We set $\lambda_{p1} = 1$ and find that retaining the three losses at the same magnitude serves the purpose. In this paper, we use $\lambda_{p2} = 0.0001$ and $\lambda_{p3} = 0.0001$.

Second, we train E_d and D_d which consist of the denoising autoencoder. We let $X_R = D_d(E_d(X_N))$, and the denoising autoencoder minimizes the following:

$$L_d = \lambda_{d1} L_R(X_C, X_R) + \lambda_{d2} L_S(X_R) + \lambda_{d3} L_B(X_R) + \lambda_{d4} L_P(X_C, X_R). \quad (9)$$

As noted in [13], the smoothness constraint causes the reproduced motion to become static, and hence, the smoothness constraint keeps the reproduction error from decreasing. However, due to perceptual loss, we can slightly increase the weights of the smoothness loss and bone-length loss. In this paper, we set $\lambda_{d1} = 1, \lambda_{d2} = 0.001, \lambda_{d3} = 0.001$ and $\lambda_{d4} = 10$.

The implementation of our work is based on TensorFlow using a single GTX Tesla P100 GPU. Adam [50] is used to minimize the loss function of two networks. The mini-batch size is set to 16. Dropout wrapper is used on Bi-LSTM

layer and dropout rate is set to 0.2. The learning rate is set to 0.00001 when training the perceptual autoencoder and set to 0.001 when training the denoising autoencoder. Each of the two networks is trained by 300 epochs.

IV. EXPERIMENT AND ANALYSIS

Our experiment consists of two components. The experiments on the synthetic dataset are used to compare the performance of BRA-P with that of selected state-of-the-art baselines and analyze why the networks based on RNN architecture are suitable for noisy data with different noise types and amplitudes. Additionally, compared with BRA [13], BRA-P only has obvious superiority for the case in which the noisy data and target clean data have different topologies. Hence, we perform the ablation study only on the raw motion dataset to verify the effect of the three proposed characteristics of BRA-P.

A. EXPERIMENTS ON SYNTHETIC NOISE DATASET

The synthetic dataset DS_{syn} contains four types of noisy data: (a) Gaussian noise data: where 100% of the joint data are corrupted by Gaussian noise (SNR = 1 dB, 5 dB); (b) randomly missing data with Gaussian noise: where 30% and 40% of the joint data are randomly set to zero in every frame, and 100% of the reversed part data are corrupted by Gaussian noise (15 dB SNR). To improve the generalization ability of our network, each type of noisy data only contains $N_{CMU}/2$ motion clips, which are randomly selected from DS_{CMU} . As a result, the total number of noisy-clean pairs in DS_{syn} is $2 \times N_{CMU}$. All the data in DS_{syn} are used as the training dataset.

We use four quantitative measurements to quantify the refinement results of BRA-P and the baselines: the reproduction error (R), perceptual error (P), smoothness error (S) and bone-length error (B). The four errors are calculated via Eq. 3, Eq. 4, Eq. 6 and Eq. 7, respectively. All the reproduction errors are stored in centimeters. For ease of comparison of the precision with that of the experiments on raw data set, the CMU skeleton is also regularized with a height of 175 centimeters in neutral pose.

We compare the performance of our BRA-P on a synthetic dataset with those of four state-of-the-art baselines, including (1) the method proposed by Holden *et al.* [10] in 2015 SIG-Graph Asia, which is denoted CNN; (2) the method proposed by Holden *et al.* in 2016 SIGGraph [22], which can be used to optimize the results of CNN and is denoted CNN+Constrain; (3) an encoder-bidirectional-filter (EBF) model similar to [11], which is modified to the fit motion data represented by the joint position and denoted EBF; (4) the same network with EBF but with the addition of a bone-length constraint during training, which is denoted EBF+B. Because the key idea of EBF is to train a set of smooth filters to clean the noise, this method does not consider the bone-length constraint. Hence, we add the bone-length constraint while training EBF to make a fair comparison. Among these four baselines, EBF and EBF+B are both based on the RNN

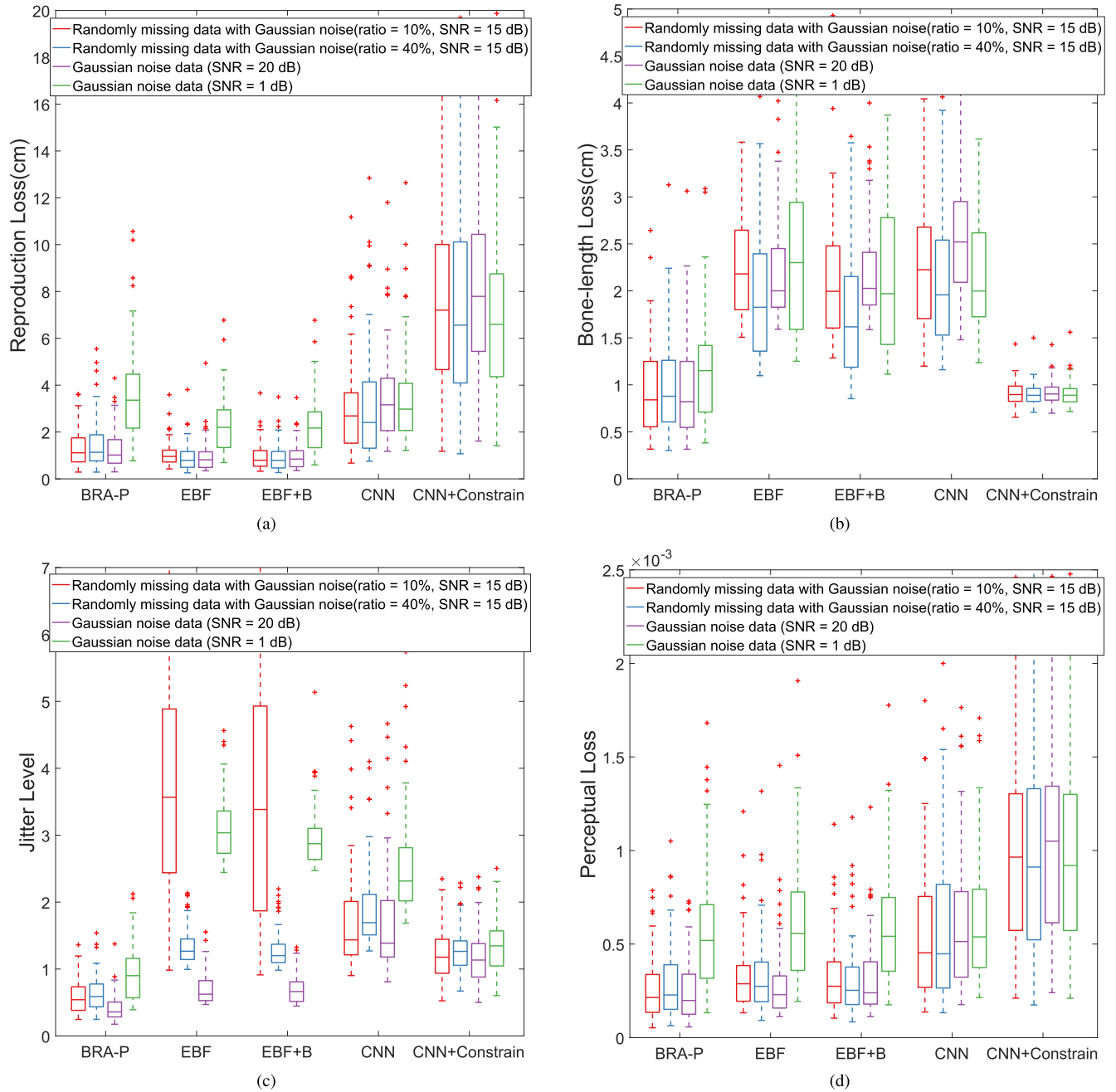


FIGURE 6. Comparisons between performances of BRA-P and four baselines on the testing dataset including 1000 motion clips randomly selected from *DSCMU* using box plots.

architecture, and CNN and CNN+Constrain are based on the CNN architecture.

For testing, we randomly selected 1000 motion clips from *DSCMU* and resynthesized four types of noisy data, including Gaussian noise data (SNR = 1 dB, 20 dB) and randomly missing data with Gaussian noise (missing ratio = 10%, 40%, Gaussian SNR = 15 dB) to compare the performance of five approaches. The Gaussian noise data (SNR = 20 dB) and randomly missing data (missing ratio = 10%) are not seen during training.

From Fig. 6, we conclude the following:

(1) BRA-P vs. EBF and EBF +B: The position error of EBF is slightly smaller than that of BRA-P because position loss is the sole optimization objective of EBF. However, the bone-length error of BRA-P is much better than that of EBF. Even when the bone-length constraint is used during the training of EBF+B, the bone-length error of EBF+B is still larger than that of BRA-P. EBF and EBF+B can produce a small smooth loss for the noisy data encountered in the training data, but their smooth losses become notably large

TABLE 1. Average values of four quantitative measurements obtained by the six approaches on the entire testing dataset, including the reproduction error (R, cm/channel), bone-length error (B, cm/bone), smoothness error (S) and perceptual error (P). BRA is the only approach that can maintain the four errors at less than 2.

Loss \ Approach	BRA-P	CNN	CNN+Constrain	EBF	EBF+B
R	1.9124	3.2249	7.7710	1.3062	1.2803
B	0.9975	2.3558	0.9116	2.2425	2.0974
S	0.6353	1.9458	1.2716	2.2666	2.1317
P	0.000343	0.000612	0.001037	0.000390	0.000385

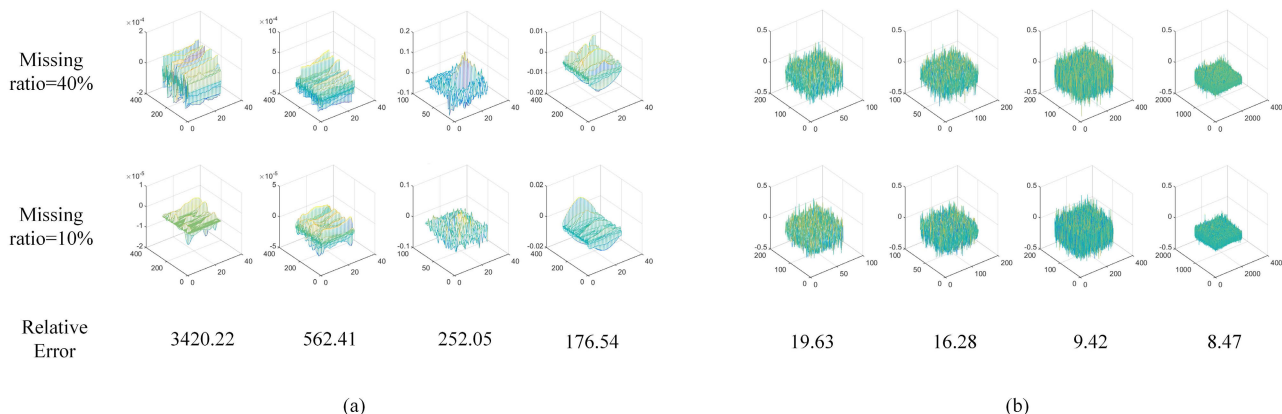


FIGURE 7. Subset of parameters of four autoencoders based on two architectures for two missing noise ratios. The first row represents the weights of the networks trained by randomly missing data (missing ratio = 40%). The second row represents the network trained by randomly missing data (missing ratio = 10%). The third row gives the relative error of two weights shown in the first row and second row. The relative error is calculated by $\|w_1 - w_2\|_2 / \|w_1\|_2$, where w_1 and w_2 are the weights in the first and second rows respectively, and $\|\cdot\|_2$ is the L2 norm. We visualize four weight pairs that have the top four relative errors for two types of autoencoders. Obviously, the relative errors of the weights of the network based on CNN are much larger than those of the network based on B-LSTM-RNN. (a) Four weight pairs from the two autoencoders based on CNN. (b) Four weight pairs from the two autoencoders based on B-LSTM-RNN.

TABLE 2. Different approaches used in the ablation study.

Methods \ Attributes	Perceptual constraint	Postprocessing by perceptual autoencoder	Bone-length constraint	Smoothness constraint	Dimensions of hidden units
EBD(512)	No	No	No	No	120 × 512
EBD-P(512)	Yes	Yes	No	No	120 × 512
BRA-P(256)	Yes	Yes	Yes	Yes	120 × 256
BRA(512)	No	No	Yes	Yes	120 × 512
BRA-P-D(512)	Yes	No	Yes	Yes	120 × 512
BRA-P(512)	Yes	Yes	Yes	Yes	120 × 512

for the noisy data not seen in the training data. BRA-P shows a stable smoothness loss for all types of noise including those seen or not seen in the training data.

(2) BRA-P vs. CNN and CNN+Constrain: The performance of BRA-P on four types of noise is better than that of CNN and CNN+Constrain, which means that the autoencoder based on the B-LSTM-RNN architecture is more suitable for mixed noise than the CNN architecture. CNN+Constrain can decrease the bone-length error and smoothness error of CNN but increase the position loss and perceptual loss at the same time.

We also display the average values of the four errors for the six approaches on the total testing dataset, which contains four types of noise. As reported in Table 1, BRA-P is the only approach that can maintain the four quantitative measurements at values smaller than 2.

To explain why autoencoders based on the B-LSTM-RNN architecture are more suitable for mixed noise than the CNN architecture, we use two types of randomly missing data (missing ratio = 10% and 40%) to train four denoising autoencoders based on two network architectures and compare the weights of these four networks. We visualize a portion of the parameters of the four networks, as shown in Fig. 7. Obviously, the two networks based on the B-LSTM-RNN architecture have more similar parameters when trained by the two missing-ratio noise data. The relative errors of the parameter pairs based on CNN are much larger than that of the network based on RNN. Hence, although the network based on B-LSTM-RNN is trained by mixed-noise data, its parameters converge more easily than those of the network based on CNN, which can explain why BRA-P is more suitable for mixed-noise data.

TABLE 3. Average values of four quantitative measurements obtained by the seven approaches on the entire Kinect testing dataset, including the reproduction error (R, cm/channel), bone-length error (B, cm/bone), smoothness error (S) and perceptual error (P). The minimum value of each error is highlighted. The proposed BRA-P(512) yields the lowest kinematic errors.

Loss	Approach	Approach						
		CNN	EBD(512)	EBD-P(512)	BRA-P(256)	BRA(512)	BRA-P-D(512)	BRA-P(512)
R		124.4595	21.9174	21.9753	25.6083	19.7190	20.2954	21.7332
B		1.1795	1.5766	0.7297	0.7237	0.6204	0.5733	0.4826
S		1.6054	1.2125	0.7631	0.5378	0.4717	0.4540	0.3984
P		0.000286	0.000141	0.000136	0.000157	0.000136	0.000127	0.000126

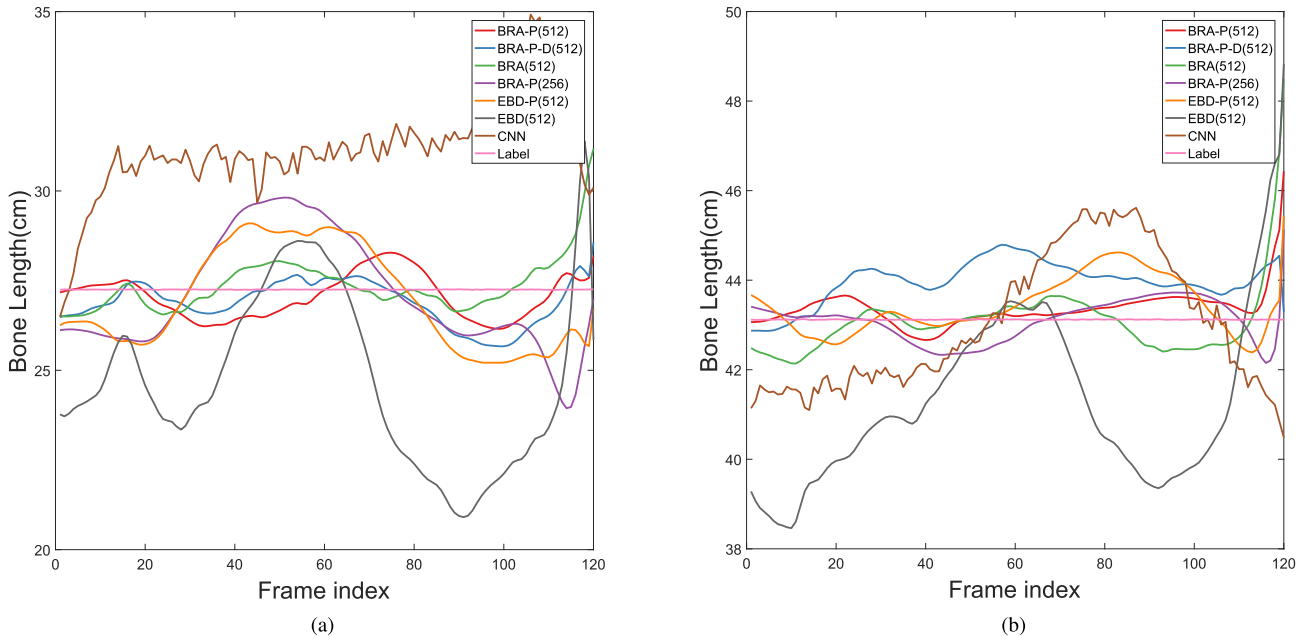


FIGURE 8. Bone-length variant curves for two bones in a walking motion sequence via different approaches. (a) Bone length variant curve for the left arm. (b) Bone length variant curve for the left leg. The bone-length variant curves of the networks based on B-LSTM-RNN are much smoother than those of the CNN.

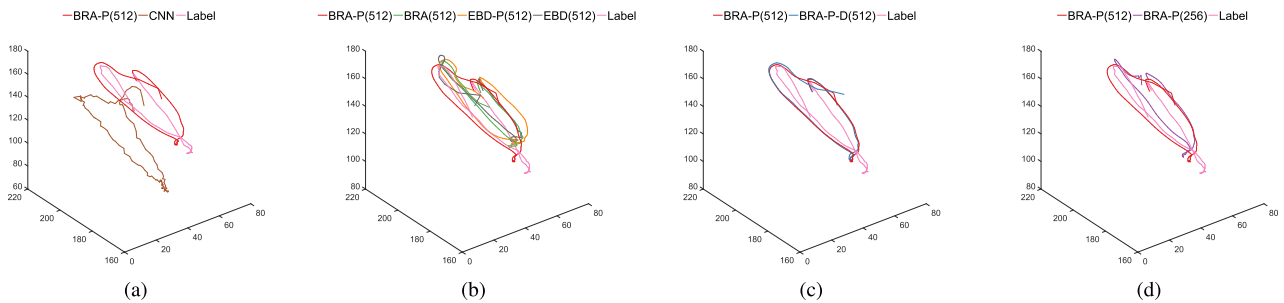


FIGURE 9. Moving trajectories of a finger joint in a walking motion sequence. The finger joint is marked with red in the last row of Fig. 10. All four subfigures show that the trajectories refined by BRA-P(512) are smoother than those of its competitors. (a) Effect of network architecture on smoothness. (b) Effect of the perceptual constraint on smoothness. (c) Effect of postprocessing by the perceptual autoencoder on smoothness. (d) Effect of the dimension of the hidden units on smoothness.

B. EXPERIMENTS ON RAW DATASET

Unlike synthetic noise, the noise type and amplitude of each motion clip captured by Kinect cannot be known in advance. The experiment on the synthetic dataset allows us to clearly examine the ability of each approach in working with noise of different types and amplitudes. However, only experiments on raw motion data can verify the ability of the different approaches in addressing mixed-noise data.

Among the baselines mentioned in the experiments on the synthetic dataset, CNN+Constrain requires precise information such as the footstep in the original data, and thus, this method cannot be used on raw data. The key idea of EBF is to train a set of smooth filters and subsequently use the trained filters to multiply the noise data to obtain the refined data; thus, this method is not applicable to the case in which the target refined data and noisy data are heterogeneous. As a result, we only compare BRA-P with CNN on the raw



FIGURE 10. Key frame sequence of the refinement results obtained via various approaches. The hand of each refined pose is circled and magnified for a clear comparison. The hand skeleton in raw data captured by Kinect is indistinct but should be clear in the refined pose, and the joint numbers of the two spines are also different. The results show that the spine and hand components of the results of BRA-P are more natural than its variants. CNN can produce a good skeleton topology, but the result is jittery. Readers can refer to the supplementary video for additional information.

motion dataset. We also perform ablation studies to verify each component of our approach on DS_{raw} . The different networks for the ablation studies are shown in Table 2. To distinguish the various methods in the ablation experiments, the proposed BRA-P is labeled BRA-P(512) in this section. We still use four quantitative measurements to quantify the refinement results of the BRA-P and the baselines: the

reproduction error (R), perceptual error (P), smoothness error (S) and bone-length error (B), in which P, S and B are denoted the kinematic errors. In addition to quantitative comparison, we also give qualitative analysis through, bone-length variant curves in Fig. 8, the moving trajectory of a joint in Fig. 9 and key frame sequences in Fig. 10. More comparisons of motion are shown in the supplementary video.

The effect of the B-LSTM-RNN architecture. In Table 3, the reproduction errors of all the networks based on B-LSTM-RNN are much better than those of the CNN. All three kinematic errors of the networks based on B-LSTM-RNN are smaller than those of the CNN, except that the bone length error of EBD(512) is slightly larger than that of CNN. In Fig. 8, two bones of a walking motion sequence are selected, and their bone-length variant curves are plotted. Obviously, the bone-length variant curves of the networks based on B-LSTM-RNN are smoother than those of the CNN. Specifically, we plot the moving trajectory examples of BRA-P(512) and the CNN in Fig. 9(a), and the trajectory of BRA-P(512) exhibits a smoother performance. These results indicate that the network based on the B-LSTM-RNN is more suitable for mixed noise, which is in line with the conclusion from the experiments on the synthetic dataset.

The effect of perceptual constraint. Among the different approaches in Table 2, the differences between the EBD(512) and EBD-P(512), BRA(512) and BRA-P(512) experimental pairs are only whether the perceptual constraint is used. As shown in Table 3, after adding the perceptual constraint, EBD-P(512) and BRA-P(512) perform better than EBD(512) and BRA(512) in three of the kinematic errors, respectively. We also find that the reproduction error is slightly increased, while the three kinematic errors decrease; perhaps some reproduction accuracy is lost to improve the quality of the motion. In Fig. 9(b), the moving trajectory of BRA-P(512) is smoother than that of BRA(512), and the trajectory of EBD-P(512) is also smoother than that of EBD(512). Therefore, we can conclude that the perceptual constraint can improve the kinematic expression ability of the network but with little sacrifice in the reproduction error.

The effect of the postprocessing step. We compare BRA-P(512) and BRA-P-D(512) in Table 2 because the only difference between these two methods is whether the perceptual autoencoder is used for postprocessing. In Table 3, the three kinematic errors of BRA-P(512) are better than those of BRA-P-D(512), which illustrates that the postprocessing step improves the kinematic information expression ability of the network. In Fig. 9(c), the moving trajectory of BRA-P(512) is smoother than that of BRA-P-D(512). Similarly, the reproduction error is also slightly increased while the three kinematic errors decrease, but the proposed BRA-P(512) achieves the best refinement performance. More clear comparisons are shown in Fig. 10 and the supplementary video.

The effect of the bone length and smoothness constraints. The comparisons of EBD(512) and BRA(512), EBD-P(512) and BRA-P(512) show that all four quantitative measurements decrease after imposing bone length and smoothness constraints during training. This result indicates that bone length and smoothness constraints can help improve the quality of reproduced motion even though perceptual constraints are used.

The effect of the dimension of the hidden units. As shown in Fig. 5, if the proposed BRA-P(512) has three

FC layers, the dimension of the hidden units is 120 plus 512. If only 2 FC layers are used, the dimension of the hidden units is 120 plus 256. Comparing BRA-P(256) and BRA-P(512) helps us to determine the effect of the dimension of the hidden units. We find that the four quantitative measurements can be improved by increasing the dimension of the hidden units. Furthermore, in Fig. 9(d), the moving trajectory of BRA-P(512) is smoother than that of BRA-P(256). Hence, we choose to increase the dimension of the data by three FC layers.

Fig. 10 summarize the difference of various approaches by the key frame sequences. We specifically compare the hand and spine of two skeletons, which are heterogeneous parts of the two different skeletons. The refinement results in Fig. 10 show that BRA-P(512) can yield best skeleton topology for those heterogeneous parts.

V. CONCLUSION

Refinement of raw motion data captured by a mocap device is an indispensable preprocessing step before the data are used, especially for low-cost yet noisy motion capture devices. In this paper, we propose a new refinement network based on a bidirectional RNN. The proposed BRA-P has the ability to remove noise of different types and amplitudes with one network because networks based on bidirectional RNN are more suitable for mixed noise than a network based on CNN. BRA-P also improves the kinematic information expression ability via the perceptual constraint, especially if the noisy data and target clean data have different skeleton topologies. Furthermore, because of the postprocessing step based on the perceptual autoencoder, the smoothness and bone-length consistency of the refined motion are further improved.

However, the proposed approach can be further improved. The reproduction accuracy is not improved, while the three kinematic errors decrease. Poor reproduction can also cause a refined motion that is still somewhat noisy. In the future, we plan to adjust the network and constraints to improve the reproduction accuracy. One possible improvement is the use of a residual network. Additionally, because Kinect can only detect a limited range of movement, we plan to extend the approach to a variety of motion capture systems, such as RGB cameras and inertia-based sensors. We also believe that motion refinement methods for those low-cost yet novel mocap systems will play a key role in emerging interactive technologies such as VR and AR.

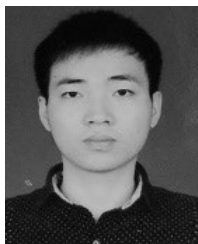
REFERENCES

- [1] M. Koohzadi and N. M. Charkari, "Survey on deep learning methods in human action recognition," *IET Comput. Vis.*, vol. 11, no. 8, pp. 623–632, Dec. 2017.
- [2] S. Xia, L. Gao, Y.-K. Lai, M.-Z. Yuan, and J. Chai, "A survey on human performance capture and animation," *J. Comput. Sci. Technol.*, vol. 32, no. 3, pp. 536–554, May 2017.
- [3] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D human pose estimation: A review of the literature and analysis of covariates," *Comput. Vis. Image Understand.*, vol. 152, pp. 1–20, Nov. 2016.
- [4] S. Guo, R. Southern, J. Chang, D. Greer, and J. J. Zhang, "Adaptive motion synthesis for virtual characters: A survey," *Vis. Comput.*, vol. 31, no. 5, pp. 497–512, May 2015.

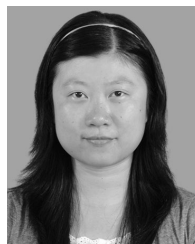
- [5] (2018). *Vicon Systems*. [Online]. Available: <https://www.vicon.com>
- [6] (2018). *Xsens Systems*. [Online]. Available: <https://www.xsens.com>
- [7] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele, "Learning to refine human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 205–214.
- [8] G. Moon, J. Y. Chang, and K. M. Lee, "PoseFix: Model-agnostic general human pose refinement network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7773–7781.
- [9] Z. Liu, L. Zhou, H. Leung, and H. P. H. Shum, "Kinect posture reconstruction based on a local mixture of Gaussian process models," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 11, pp. 2437–2450, Nov. 2016.
- [10] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *Proc. SIGGRAPH ASIA Tech. Briefs (SA)*, 2015, p. 18.
- [11] U. Mall, G. R. Lal, S. Chaudhuri, and P. Chaudhuri, "A deep recurrent framework for cleaning motion capture data," 2017, *arXiv:1712.03380*. [Online]. Available: <http://arxiv.org/abs/1712.03380>
- [12] D. Holden, "Robust solving of optical motion capture data by denoising," *ACM Trans. Graph.*, vol. 37, no. 4, p. 165, Jul. 2018.
- [13] S. Li, Y. Zhou, H. Zhu, W. Xie, Y. Zhao, and X. Liu, "Bidirectional recurrent autoencoder for 3D skeleton motion data refinement," *Comput. Graph.*, vol. 81, pp. 92–103, Jun. 2019.
- [14] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.
- [17] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 694–711.
- [18] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 702–716.
- [19] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 262–270.
- [20] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," 2015, *arXiv:1511.05666*. [Online]. Available: <http://arxiv.org/abs/1511.05666>
- [21] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
- [22] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, no. 4, p. 138, Jul. 2016.
- [23] CMU. (2013). *Carnegie-Mellon Mocap Database*. [Online]. Available: <http://mocap.cs.cmu.edu/>
- [24] A. Bruderlin and L. Williams, "Motion signal processing," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, 1995, pp. 97–104.
- [25] J. Lee and S. Y. Shin, "General construction of time-domain filters for orientation data," *IEEE Trans. Vis. Comput. Graphics*, vol. 8, no. 2, pp. 119–128, Aug. 2002.
- [26] C.-C. Hsieh and P.-L. Kuo, "An impulsive noise reduction agent for rigid body motion data using B-spline wavelets," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1733–1741, Apr. 2008.
- [27] S. Tak and H.-S. Ko, "A physically-based motion retargeting filter," *ACM Trans. Graph.*, vol. 24, no. 1, pp. 98–117, Jan. 2005.
- [28] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, "Dynammo: Mining and summarization of coevolving sequences with missing values," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 507–516.
- [29] L. Li, J. McCann, N. Pollard, and C. Faloutsos, "Bolero: A principled technique for including bone length constraints in motion capture occlusion filling," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation. Aire-la-Ville, Switzerland: Eurographics Association*, 2010, pp. 179–188.
- [30] M. Burke and J. Lasenby, "Estimating missing marker positions using low dimensional Kalman smoothing," *J. Biomech.*, vol. 49, no. 9, pp. 1854–1858, Jun. 2016.
- [31] H. Lou and J. Chai, "Example-based human motion denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 5, pp. 870–879, Sep. 2010.
- [32] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh, "Bilinear spatiotemporal basis models," *ACM Trans. Graph.*, vol. 31, no. 2, pp. 1–12, Apr. 2012.
- [33] J. Xiao, Y. Feng, and W. Hu, "Predicting missing markers in human motion capture using l1-sparse representation," *Comput. Animation Virtual Worlds*, vol. 22, nos. 2–3, pp. 221–228, Apr. 2011.
- [34] J. Xiao, Y. Feng, M. Ji, X. Yang, J. J. Zhang, and Y. Zhuang, "Sparse motion bases selection for human motion denoising," *Signal Process.*, vol. 110, pp. 108–122, May 2015.
- [35] Y. Feng, M. Ji, J. Xiao, X. Yang, J. J. Zhang, Y. Zhuang, and X. Li, "Mining spatial-temporal patterns and structural sparsity for human motion data denoising," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2693–2706, Dec. 2015.
- [36] G. Xia, H. Sun, G. Zhang, and L. Feng, "Human motion recovery jointly utilizing statistical and kinematic information," *Inf. Sci.*, vol. 339, pp. 189–205, Apr. 2016.
- [37] N. N. Schraudolph, S. Günter, and S. Vishwanathan, "Fast iterative kernel PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1225–1232.
- [38] G. Liu and L. Mcmillan, "Estimation of missing markers in human motion capture," *Vis. Comput.*, vol. 22, nos. 9–11, pp. 721–728, Sep. 2006.
- [39] T. Tangkuampien and D. Suter, "Human motion de-noising via greedy kernel principal component analysis filtering," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2006, pp. 457–460.
- [40] S. Günter, N. N. Schraudolph, and S. V. N. Vishwanathan, "Fast iterative kernel principal component analysis," *J. Mach. Learn. Res.*, vol. 8, pp. 1893–1918, Aug. 2007.
- [41] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1345–1352.
- [42] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4346–4354.
- [43] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6158–6166.
- [44] Q. Cui, H. Sun, Y. Li, and Y. Kong, "A deep bi-directional attention network for human motion recovery," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 701–707.
- [45] Y. Huang, M. Kaufmann, E. Aksan, M. J. Black, O. Hilliges, and G. Pons-Moll, "Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time," *ACM Trans. Graph.*, vol. 37, no. 6, p. 185, Dec. 2018.
- [46] (2018). *Noitom Mocap System*. [Online]. Available: <https://www.noitom.com.cn/perception-neuron-series.html>
- [47] CMU. (2018). *Carnegie Mellon University Multimodal Activity Database*. [Online]. Available: <http://kitchen.cs.cmu.edu/index.php>
- [48] Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J. J. Zhang, and R. Song, "Exploiting temporal stability and low-rank structure for motion capture data refinement," *Inf. Sci.*, vol. 277, pp. 777–793, Sep. 2014.
- [49] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 186–201.
- [50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>



SHU-JIE LI received the B.Sc. degree in information and computing science and the M.Sc. degree in computer science from the Hefei University of Technology, Hefei, China, in 2004 and 2008, respectively, and the Ph.D. degree in pattern recognition and intelligence systems from the University of Science and Technology of China, Hefei, in 2012. She is currently a Lecturer with the School of Computer and Information, Hefei University of Technology. Her research interests include human motion analysis and pattern recognition.



HAI-SHENG ZHU received the B.E. degree in ammunition engineering from the Anhui University of Science and Technology, Huainan, China, in 2016. He is currently pursuing the M.E. degree in computer technology with the Hefei University of Technology, Hefei, China. His research interests include motion data analysis and generation, and machine learning.



LIN LI received the master's and Ph.D. degrees in computer application technology from the Hefei University of Technology, Hefei, China, in 2014 and 2016, respectively. She is currently an Assistant Professor with the School of Computer and Information, Hefei University of Technology. Her research interests include computer graphics and computer animation.

...



LI-PING ZHENG received the master's and Ph.D. degrees in computer science from the Hefei University of Technology, Hefei, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Computer Science and Information Engineering, Hefei University of Technology. His research interests include visualization and computer simulation.