IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 2, 2019, accepted May 6, 2019, date of publication May 20, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917609

3D Human Motion Synthesis Based on Convolutional Neural Network

DONGSHENG ZHOU[1], (Member, IEEE), XINZHU FENG[1], PENGFEI YI[1], XIN YANG[2], (Member, IEEE), QIANG ZHANG[1,2], (Member, IEEE), XIAOPENG WEI[2], AND DEYUN YANG[3]

[1]Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, Dalian University, Dalian 116622, China
[2]College of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
[3]School of Information Science and Technology, Taishan University, Taian 271000, China

Corresponding authors: Dongsheng Zhou (donyson@126.com) and Qiang Zhang (zhangq@dlut.edu.cn)

This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 61425002, in part by the National Natural Science Foundation of China under Grant 91748104, Grant 61632006, and Grant 61877008, in part by the Program for ChangJiang Scholars and Innovative Research Team in University under Grant IRT_15R07, in part by the Program for the Liaoning Distinguished Professor, Program for Dalian High-level Talent Innovation Support, under Grant 2017RD11, and in part by the Science and Technology Innovation Fund of Dalian under Grant 2018J12GX036.

ABSTRACT Human motion synthesis technology has a very important position in computer animation, and it is widely used in medicine, film and television, motion analysis, games, and other related fields. The synthesis of human motion is the virtual of the action of the characters in the real world, the authenticity of the action, and the natural smoothness is especially important to the user's experience. Due to the complexity of human structure, how to generate a high-quality movement is a challenging task. The data used in this paper are all 3D human motion data in BioVision Hierarchical (BVH) format, which can be captured by optical, inertial, mechanical or other video-based motion capture devices. In this paper, first, a three-layer convolutional neural network was used to output mapping in the hidden unit of the input motion capture data. Then, a one-dimensional convolution auto-encoder was connected; meanwhile, the bone length constraint, position constraint, and trajectory constraint were added. It repaired the non-inertial joints of motion data and removed the motion artifacts. To achieve the synthesis of the two motions, we extracted the style transformation in the motion, added style and content constraints, and finally output the motion. To verify the feasibility of the algorithm, we obtained the animation effect of the synthesized motion by testing the input motion. The experimental results show that the motions synthesized by the proposed algorithm not only look natural smooth in visual effect but also reduce the time consumed by about 42.6% compared with the existing algorithms.

INDEX TERMS Convolutional neural network, convolution auto-encoder, human motion capture data, motion synthesis.

I. INTRODUCTION

3D Human motion data is acquired by state-of-the-art motion capture technologies which are widely employed to record and archive high quality motion trajectories of a character in three-dimensional space. Nowadays, there exist many different Mo-cap technologies including the optical based, the sensor based, the mechanical based, the depth image based, etc. Along with the strong demand from the emerging consumer field to the professional field, the booming technologies bring a lot of human motion data, which can be widely applied in many fields, such as animation production, film and television special effects production, robot control, sports training, medical rehabilitation, virtual reality, electronic games and so on. The biggest advantage of these data is that they can record the movement details of various parts of the human body faithfully, accurately and with high frequency. However, just as coins have two sides, due to the strong individualization of human movement, reusability of the data is lack of flexibility, which means the existing human motion data often cannot be applied directly to other objects, and resulting in an ever-increasing amount of new data. In this case, many researchers

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

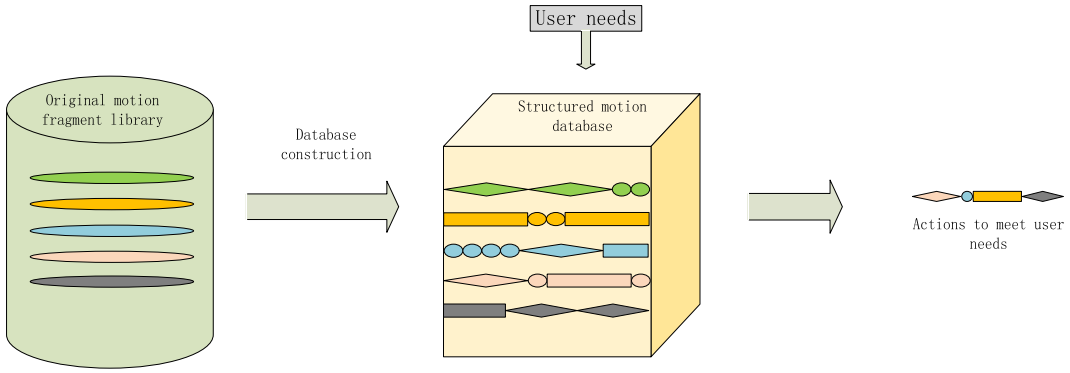


FIGURE 1. Pipeline of human motion data synthesis.

have begun to pay attention to solve the problem of data reuse, and tried to dig more potential information from human actions. At present, the research directions of human motion data reuse can be mainly divided into data cleaning, data retrieval, data editing and data synthesis.

Human motion data synthesis is a technology of generating new motion based on existing data. It combines science and art as well as reality and abstraction, which is a comprehensive and challenging leading-edge discipline. This technology can not only reduce the expensive cost of motion capture, but also extremely improve the reuse of motion data. In particular, this technology can establish a good foundation for human behavior cognition and prediction in one of the frontier research areas of artificial intelligence. At present, following the expansion of the application field, the technology has received continuous attention from many researchers.

II. RELATED WORKS

Based on motion segments produced by methods of motion segmentation [1] from motions that can be captured by multi-camera systems or computed by multi-sensor fusion method [2], human motion synthesis can obtain a new motion sequence from these existing segments, as shown in Figure 1. The references cited in the paper can be divided into three types: methods based on the autoregressive models [3]–[7], methods based on statistical learning [8]–[13], and methods based on the deep learning [16]–[25].

In recent years, many studies have been conducted on human motion synthesis. Kwon *et al.* [3] proposed an example-based on-line method. It models an unmarked example motion through labeled motion segments, so that users can perform motion mixing and motion transitions on-line. This method generates motions without artifacts and seamless transitions between actions. Xia et al. [4] proposed a real-time method which can automatically transform motion data into new styles. By constructing a series of local mixtures of autoregressive models (MAR), relationships between motions can be captured to generate high quality animations. To increase the total classes of motions in the motion database, and reduce the workload and the cost of the database, Yumer *et al.* [5] improved this algorithm by using spectrum space instead to calculate the similarity between motions and carry out the conversion between spectrum patterns. Min *et al.* [6] proposed a model adopting multi-linear analysis techniques for synthesizing, editing, and repositioning human motion. While it can speed up the process of synthesis and reduce the ambiguity of synthesis, it is not suitable for free-style human behavior, such as disco dancing, and requires the movements to be in a structured pattern, which should be similar and semantically matched. Holden *et al.* [7] introduced a method to quickly modify the movement path. which requires no data alignment and little manual intervention, but it is not easy to control the transfer of neural style.

The first type is to obtain the parameter model from the sample data learning, and then use different parameters to generate different motions, so as to achieve the control of the generated results. By constructing an example of a motion segment or a different autoregressive model, the motion data is processed and synthesized. The advantage of this method is that the synthesis of motion segments can be performed online in real time, but the motion data needs to be structurally similar, and has great limitations on some complicated actions, and is not suitable for the processing of huge motion data sets.

The traditional method of synthesizing human motion is the second type which is based on statistical learning. This method processes existing motion data and synthesizes new motion data by obtaining special information. It uses statistical models to learn motion data, and then summarizes and analyzes the motion data. Due to the regularity of human motion data in time, the temporal correlation can be expressed using a dynamic model. The motion data always was trained to obtain motion information contained therein. The commonly used statistical learning methods include principal component analysis (PCA) [8], [9], hidden Markov model (HMM) [10] and mixed Gaussian model (GPDM) [11], [12].

In details, PCA is often used to reduce the dimensionality of motion data in order to reduce the complexity of inverse kinematics and motion editing. Shin *et al.* [9] used PCA to simplify motion, K-means clustering to collect similar

motions, and Markov model to simulate time constraints. To eliminate the limits in effectiveness caused bt linear assumption, a new type of PCA, nonlinear extended additive component analysis (ACA) [13]has been developed. It can effectively remove the noises and learn the local tangent space of data manifold and replace other algorithms directly. Therefore, ACA is suitable for dealing with large data sets. HMM uses hidden variables to parameterize sports styles. These hidden variables determined by learning can be used to generate new sports with different styles. Fox *et al.* [10] used Bayesian non-parametric methods to define HMM. This approach allows data to drive the complexity of the learning model while allowing for efficient inference algorithms. GPDM is a nonlinear hidden variable model, especially suitable for time series data. It considers the time correlation and also considers the structure of data in time. Wang *et al.* [11] regarded GPDM as a latent variable model. The model margin is parameterized in a closed form by using a Gaussian process prior to mapping the dynamics to the observations. This approach leads to a non-parametric model of the dynamic system, which solves the uncertainty in the model.

In recent years, deep learning has been widely used in image processing, speech recognition, human pose estimation etc. Deep learning has made great achievements in static images, and gradually expanded to time series human behavior recognition for dynamic video [14], [15]. Fortunately, human motion capture data have something in common with human motion video in terms of dynamics. This commonality makes it possible to use deep learning methods to deal with the motion capture data. The method based on deep learning can be mainly divided into two categories. One is the method based on automatic encoder which could train large-scale human motion data by improving different automatic encoders. It can generate a high quality motion data according to user needs. The other is a neural network based approach. The method is capable of synthesizing complex human motion sequences through training data.

Tan *et al.* [16] proposed a framework called mesh variational auto-encoders (mesh VAE), which can flexibly represent 3D animation sequences. The mesh VAE is represented by the invariant features of the discernible automatic encoder and mesh rotation, which can control the variation of potential variables. This framework can be used to analyze three-dimensional mesh problems, such as shape geometry, analysis and the generation of novel shapes. It is easy to be trained, and only a small amount of training data is needed to generate high quality deformable models with rich details. Habibie *et al.* [17] applied the general framework of mesh variational automatic encoder to human motion data, and established a long-term memory mesh variational auto-encoders (VAE-LSTM) model structure. The model can learn the diversity of human motion by training motion capture data. In the absence of existing sequence frames, high-quality motion can also be generated, allowing the user to generate animation from advanced control signals. Tan *et al.* [18] proposed a novel mesh-based variational auto-encoders architecture to deal with irregular topological grids. They added sparse regularization in the framework, which can be used to locate deformation with convolution operations. This framework can extract local deformation components from large-scale mesh data and is robust to noise.

In terms of human motion synthesis, deep learning techniques can be used to train and learn on existing motion capture data, and exercise models can be generated by training motion data to synthesize new sequences that meet the user's needs in a flexible manner. Zimo *et al.* [19] developed a new real-time training method to synthesize complex human motion, using an auto-conditioned recurrent neural network. Martinez *et al.* [20] adopted recursive neural networks to simulate human motion, where the network was used to learn tasks such as short-term motion prediction and long-term human motion synthesis. Human motion data can also be used to control a robot [26], Josh *et al.* [21] used a deep learning approach to train high-dimensional humanoid robots and extended the generation of antagonistic mimic learning to enable them to train general neural networks. Zhou *et al.* [22] used a deep convolutional neural network (CNN) to treat 2D joints as potential variables, and changed human data directly from 2D appearance to 3D geometry, demonstrating the ability to deeply learn 2D appearance features. Harvey *et al.* [23] used Recurrent Neural Networks (RNNs) based on Long Short Term Memory (LSTM) to identify and classify motion capture data. It is a semi-supervised learning process that can effectively reduce overfitting. Fragkiadaki *et al.* [24] proposed the Encoder-Recurrent-Decoder (ERD), which can be used to identify the pose of a human being in a video and in motion capture.

As an effective method treating temporal data, RNN can be also used to process motion data with little efforts. While this model always treat input motions as a whole, changes will occur everywhere in output motions, which is not suitable for cases that only minor changes is allowed, such as motion editing. For this reason, in this paper, the Convolutional Neural Network (CNN) is adopted and improved to generate motion. The motion capture data is treated as data stored in time series, and at each point in time, the pose of the character can be described by the angle of each joint in the skeleton, and the data is input into the multi-channel of the convolution model. Each channel represents the angle of a joint with respect to an axis. Through the network model constructed by training, the unique characteristics of the motion data can be effectively learned. Therefore, this paper will use the method of constructing convolutional neural network model to realize the synthesis of human motion data.

III. ALGORITHM DESCRIPTION

When a motion is input, a feature representation can be obtained through the hidden unit layer, that is, the automatic encoder generates a Motion Manifold. In the algorithm, a three-layer convolutional neural network was added to the automatic encoder to generate a mapping of high-level control parameters T and hidden units. When the parameter T

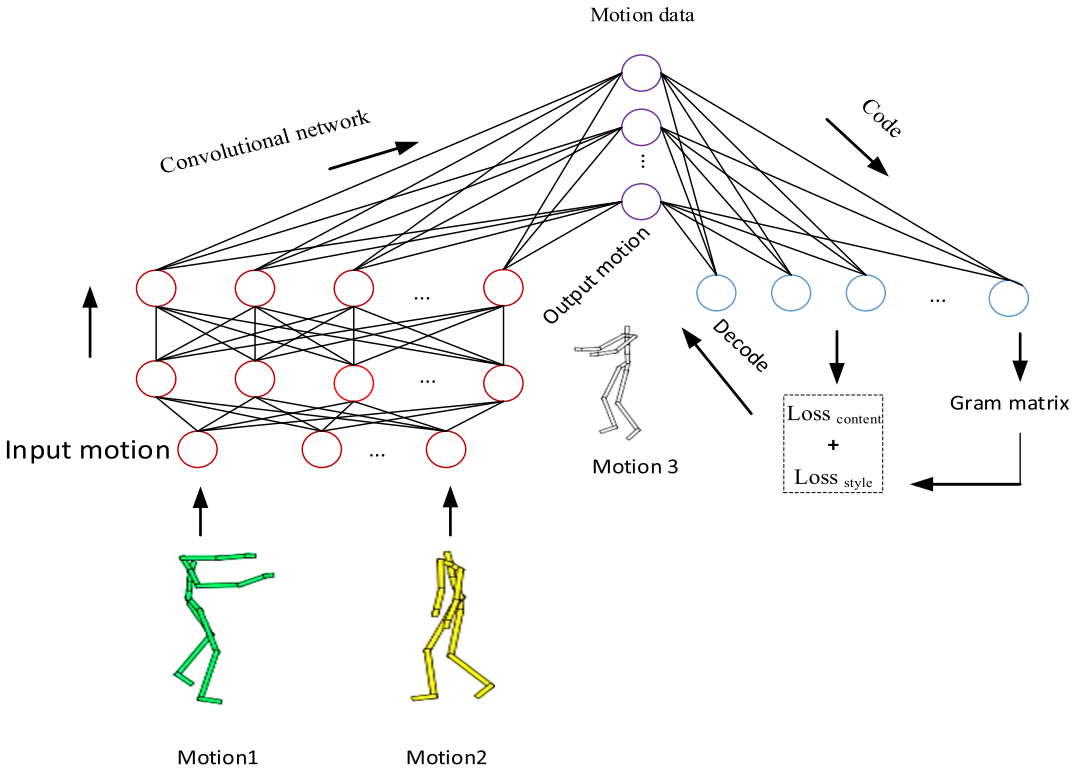


FIGURE 2. The overall structure of the network for 3D human motion synthesis.

was mapped to the hidden unit, it corresponded to the motion. In order to control the range of motion generation, three constraints including bone length, position, and trajectory were added to achieve motion constraints. And, two more constraints of motion style and motion content were used to achieve a synthesis of stylized motion data. The overall structure is as follow:

In Fig.2, there are two networks used in the synthesis system. The left (red) transform network is a three-layer convolutional neural network that performs the synthesis of motion. The right (blue) network is a convolution auto-encoder with three features: the first is to help training the loss between the motion content and the style, the second is to repair the non-inertial detail error that exists in the motion style conversion process, and the third is to remove the artifact problem that exists in the movement.

A. STRUCTURE OF CONVOLUTIONAL NEURAL NETWORK

The feedforward convolutional neural network is used to achieve the regression between high-level parameters T and human motion X. The high-level parameters defined here represent the ones that are abstracted to describe the motion trajectory.

The construction of the feedforward convolutional network advanced parameter T to the hidden layer self-encoding network, such that the final system outputs of the motion characteristic is $X \in R^{n\times d}$. The deep feedforward network will use a three-layer convolution network, and the core formula is as follows:

$$\Gamma = RELU(\Psi(RELU(RELU(\gamma(\mathrm{T}) * W_1 + b_1) * W_2 + b_2) * W_3 + b_3)) \quad (1)$$

where $W_1 \in R^{h_1\times l\times w_1}, b_1 \in R^{h_1}, W_2 \in R^{h_2\times h_1\times w_2}, b_2 \in R^{h_2}, W_3 \in R_2^{h_2 xm\times w_3}, b_3 \in R^m, h_1, h_2$ are hidden units, w_1, w_2, w_3 are three filter widths, l is the degree of freedom of the high-level parameters, the parameters are set to 64, 128, 45, 25, 15 and 7, m is the number of hidden units set to 256, and $\Phi = \{W_1, W_2, W_3, b_1, b_2, b_3\}$.

In order to train the regression mapping between high level parameters and output motions, we used the same stochastic gradient descent method to minimize the loss function. The cost function is defined as follows, consisting of two terms:

$$Loss(T, X, \Phi) = \left\| X - \Phi^{+}(\Gamma) \right\|_2^2 + a \left\| \Phi \right\|_1 \quad (2)$$

The first is to calculate the mean square error of regression, and the second is a sparse item to ensure that the minimum numbers of hidden units are used to perform regression, a set to 0.1.

In Figure 3, the black boxes in the figure indicate the network structure. Each layer of the convolutional network structure corresponds to the internal structure of the color box below it. In the figure, there are three layers of convolutional networks corresponding to three internal structures, and the last layer of the network is the output layer. The white squares represent the motion features extracted by each layer of the network. As the number of network layers increases,

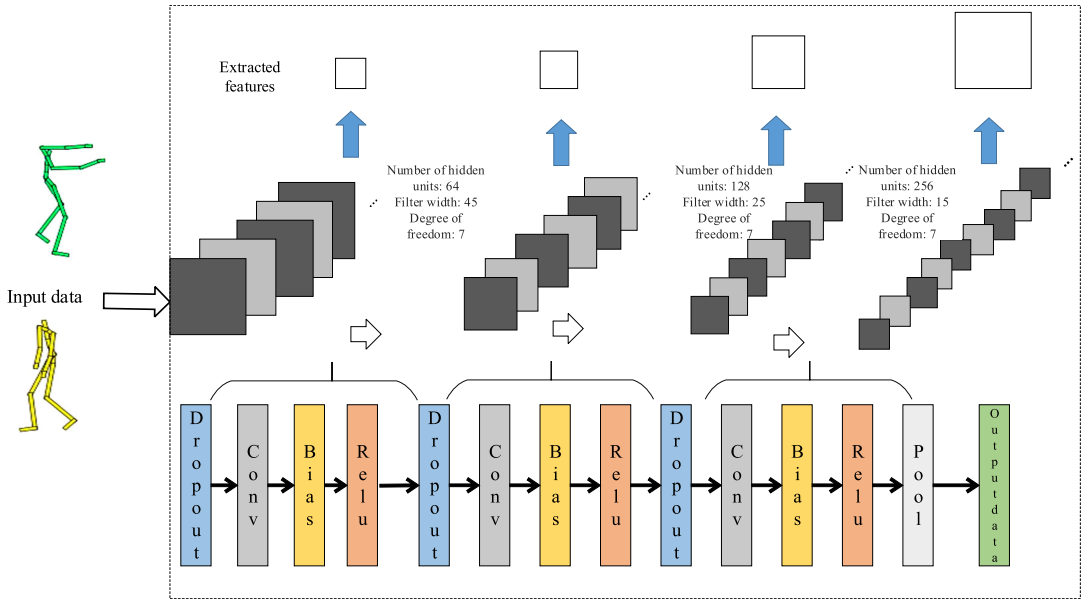


FIGURE 3. The internal structure of a convolutional network.

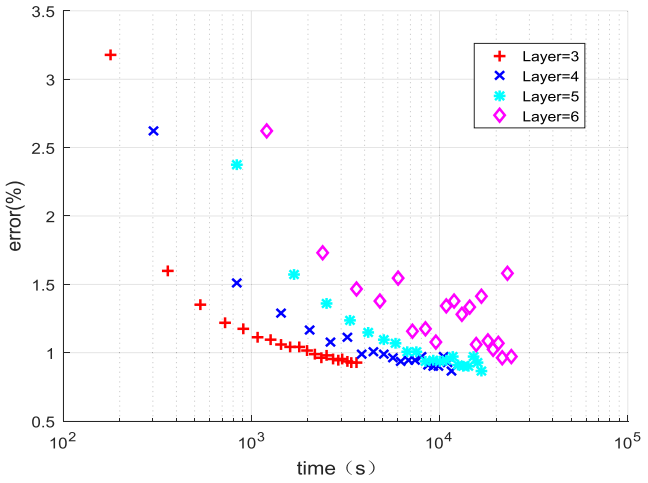


FIGURE 4. Comparison of training result of different layer networks from 3 layers to 6 layer.

the motion characteristics become more and more obvious, and the more motion information is extracted.

In the experiment, the length of the input sample was set to 240. Since the number of hidden units is directly related to the complexity of the problem, in theory, it should be an integer power of 2, such as 64, 128 and 256. In order to verify the effectiveness of the three-layer network, the network models were increased to four, five, and six layers and perform experimental verification. The performance of the training results of the four models is compared as shown in Figure 4. When setting epochs $=$ 20, Figure 4 shows the results of training time and error rate in different network layers. Obviously, more network layers will lead to more training time. And, when the number of network layers is 3, 4, and 5, the error convergence speed is equivalent, but the value gradually fluctuates. When the number of network layers reaches 6, the error distribution appears disordered and does not converge. So, the three-layer's error is the smallest, and under the same error, the three-layer network takes the least training time. Therefore, as a result, a three-layer network was chosen in our model.

B. MOTION MANIFOLD NETWORK

The feedforward network is trained together with the motion manifold network. The motion manifold in this section was constructed by encoding the input motion data by equation (3):

$$\Phi(X) = PRELU(\Psi(X * W_4 + b_4)) \tag{3}$$

where $(*)$ is the convolution operation, $W_4 \in R^{m\times d\times W_4}$ represents the weight matrix, W_4 is the filter width, m is the number of hidden units in the automatic coding layer, $b_4 \in R^m$ indicates bias, Ψ represents the maximum pool operation, and $PRELU$ is the activation function.

The output of the encoding of input data X is H, which is decoded as follows:

$$\Phi^+(H) = \left(\Psi^+(H) - b_4\right) \times W_4' \tag{4}$$

where H is the hidden unit, Ψ^+ is the inverse pool operation, b_4 is the offset, and W_4' is the weight matrix. Training the data by (5):

$$\cos(X, \theta) = \left\|X - \phi^+(\phi(X))\right\|_2^2 + \alpha\|\theta\|_1 \tag{5}$$

where $\theta = \{W_4, b_4\}$, the first term represents the squared error, the second term represents an additional sparse term; α is a constant, set to 0.1, and θ is a network parameter.

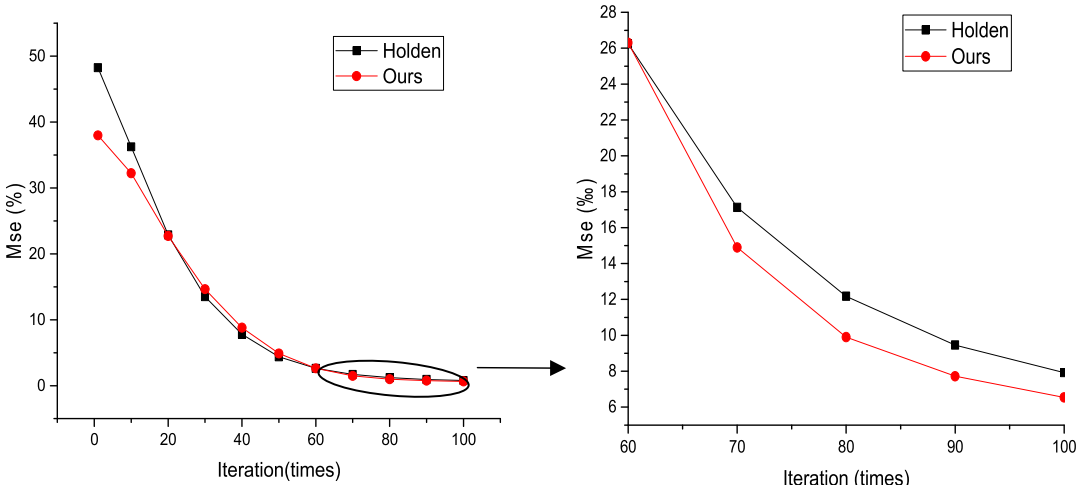


FIGURE 5. Comparison of motion errors.

Position Constraint: Given the initial input motion in hidden cell space H, the cost function for calculating the violation position is as follows:

$$Pos\,(H) = \alpha \sum_{j} \left\| V_r^H + W^H \times P_j^H + V_j^H - V_j^{'} \right\|_2^2 \quad (6)$$

where $V_j^{'} \in R^{n\times 2}$ is the target speed of joint j in the human coordinate system, V_r^H, P_j^H, $V_j^H \in R^{n\times 2}$, $W^H \in R^n$ are the velocity of the root, the position and velocity of the joint, and the angular velocity of the body around the axis, respectively.

Bone length constraint: The cost function is as follows:

$$Bone\,(H) = \beta \sum_{i} \sum_{b} \left| \left\| P_{bj1}^{Hi} - P_{bj2}^{Hi} \right\| - l_b \right|^2 \quad (7)$$

where b is the bone index of the human body, P_{bj1}^{Hi} and P_{bj2}^{Hi} are the three-dimensional position of the joint reconstruction at both ends of coordinate b in the coordinate system i , and l_b is the length of the b bone;

Trajectory constraints: The constraint objective function is as follows:

$$Traj\,(H) = \gamma \left\| W^H - W^{'} \right\|_2^2 + \left\| V_r^H - V_r^{'} \right\|_2^2 \quad (8)$$

The motion produced by the auto-encoder is adjusted in the space of the hidden unit by the gradient descent until the total constraint converges to a threshold:

$$H^{'} = \arg_H^{\min} Pos\,(H) + Bone\,(H) + Traj\,(H) \quad (9)$$

C. MOTION STYLE CONSTRAINTS

In this section, the character style constraint network was trained before the output of the three-layer convolutional neural network, and minimized the loss function to extract the motion content and style.

The method used in this section is a parametric approach to constraints on human motion, including human motion content constraints and style constraints. The Gram matrix was used to represent the sum of the inner product of the motion in the hidden unit on the time axis i, which is:

$$Gram(H) = \sum_{i} H_i H_i^T \quad (10)$$

Style Constraints: To ensure that the output style of the motion contains the style of the input, the constraint cost function is:

$$Loss_{Style}\,(I) = \alpha \left\| Gram(\Phi(s)) - Gram(\Phi(\Gamma(I))) \right\| \quad (11)$$

where s is the style of the input action, and α is a small constant, set to 0.01.

Content Constraints: To ensure that the output of the motion contains the input content, the constraint cost function can be written as follows:

$$Loss_{Content}\,(I) = \beta \left\| \Phi(I) - \Phi(\Gamma(I)) \right\| \quad (12)$$

where I is the input action content, and β is a small constant, set to 0.1 here.

Then a gradient descent was used to adjust the space of the hidden unit until the total constraint converges to a threshold:

$$I^{'} = \arg_I^{\min} Loss_{Style}\,(I) + Loss_{Content}\,(I) \quad (13)$$

We then minimized both constraints of human motion data used in the hidden unit in 3.2 and motion constraints in this section as follows:

$$S = \arg^{\min} I^{'} + H^{'} \quad (14)$$

This section used some of the motion data sets in [25], such as walking, go, jogging, and running, as training sets, and the training was carried out in 100 stages, which took about six hours. After the network was trained, a realistic sequence of actions was obtained.

Figure 5 shows the 100-stage error after the training of the network. The error comparison was carried out with the method of [25]. The MSE was used to measure the change of the error. The left of Figure 5 is the error trend of the two

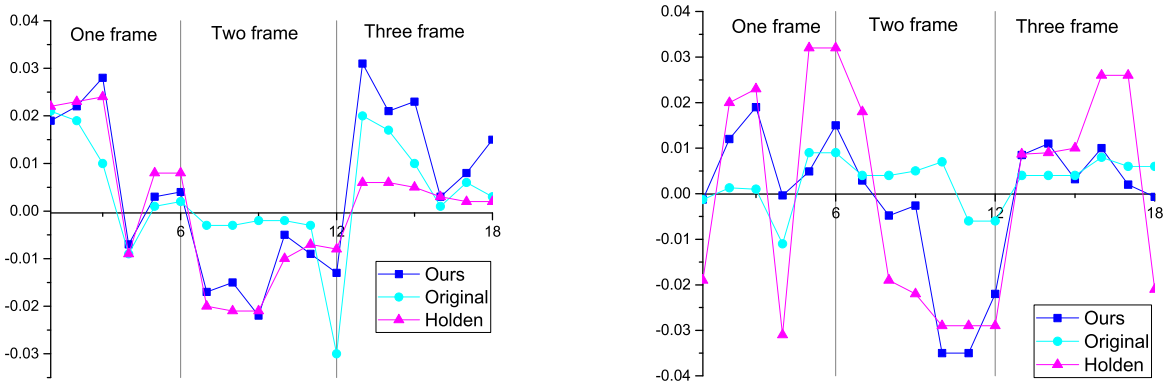


FIGURE 6. Comparison of joint motion.

methods, and the right is a partial enlarged view of iteration within 60 to 100. It can be seen that with the increase of the number of iterations, the value of errors gradually converges to stability and the value approaches to zero. When the model is carried out for 100 generations, the error value of the method is minimized.

In Figure 6, the left picture shows the result of the fitting curve of the synthetic joint of the old man running action, and the right is the result of the synthetic joint fitting curve of the monkey running action. In the two figures, the abscissa indicates the offset and rotation angle of the elbow joint data in the three directions of X, Y, and Z, which are recorded as Xpositon, Ypositon, Zpositon, Xrotation, Yrotation, Zrotation. A frame action is represented by the above six parameters in sequence. These discrete data change with time. In order to better reflect the continuity of the elbow joint point trajectory, the discrete data was fitted. In the figure, the movements of the two elbow joints with large changes in angle are selected, and each frame selects three frames of motion data. It can be seen from the fluctuation trend of the curve in the figure that the joint motion of the method is closer to the original elbow joint.

IV. VERIFICATION AND ANALYSIS OF EXPERIMENTAL RESULTS

In this section, the experimental environment was firstly introduced, and then the animations based on the synthetic motion data were evaluated.

A. EXPERIMENTAL ENVIRONMENT

In this paper, under Ubuntu 16.04 system, python 2.7 was chosen as the development platform. In the Theano framework based on deep learning, the data of locomotion and misc databases were synthesized respectively. All motion data were in BVH format. All experiments were carried out on a server which has two GeForce GTX 1080ti, one Intel i7-8700 processor and 32GB memories.

B. EXPERIMENTAL RESULTS AND ANALYSIS

This section mainly shows the animation effect of six groups of test data. Each group input two kinds of character motion at the same time. By using this algorithm model, the animation of composite motion was output. In the display of the renderings in this section, for each set of actions, they were intercepted respectively in four different time states. In a certain moment, the rectangle (red box) in the left (green) character indicates the main action content used for the synthesis. The rectangle (red box) in the middle (yellow) character indicates the path information to be extracted in the composition action. The right (white) character is the final composite image that contains both the left character action content and the middle character path information.

It can be seen from the synthetic animation diagrams of Figure 7 to Figure 12, the present method can better synthesize new actions. When the motion state of the green character is the same as that of the yellow character, the direction of the synthesized motion state is the same as the direction of the yellow character. When the motion state of the green character is different from that of the yellow character, the direction of motion and the state of motion in the combined motion are the same as those of the yellow. Therefore, regardless of the content of the action and the transformation of the path of the action, in the animation effect of the two, the content of the motion is the same as the action of the green character, and the state and direction of the motion are always consistent with those of the yellow character, and the synthesized animation looks smooth and natural.

C. EXPERIMENTAL COMPARISON

In order to verify the advanced nature of the proposed algorithm, we compared it with the method of Holden [25]. The comparison results are shown as below. In the figure, five combinations of animation effects were randomly selected. Each group shows the animation effect when the left (green) inputs action 1, the middle (yellow) input action 2 and the right (two white) input actions 1 and 2 at the same time. The right 1 represents the synthesis effect of this method, and the right 2 the result of Holden's method synthesis. The rectangle box (blue) in the Holden method represents the difference from the right 1 effect.

From the above animation, we can see that when the same set of actions are input, the method in this paper can better

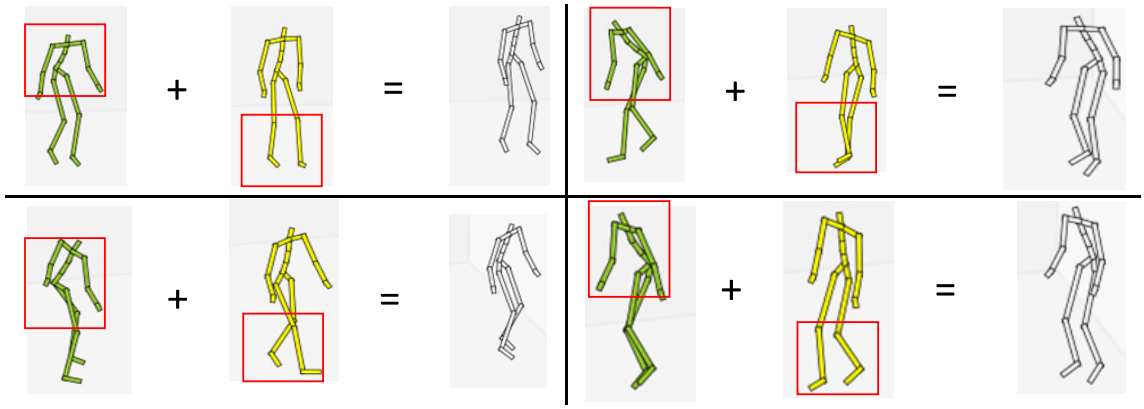


FIGURE 7. Effect diagram of the two movements of the old man walking (green) and walking (yellow).When the old man walks, the waist is curved, so the waist of the new character will be slightly curved. The direction of the walking movement determines the direction of the synthesized character, so the direction of the synthesized character is consistent with the direction of the walker.

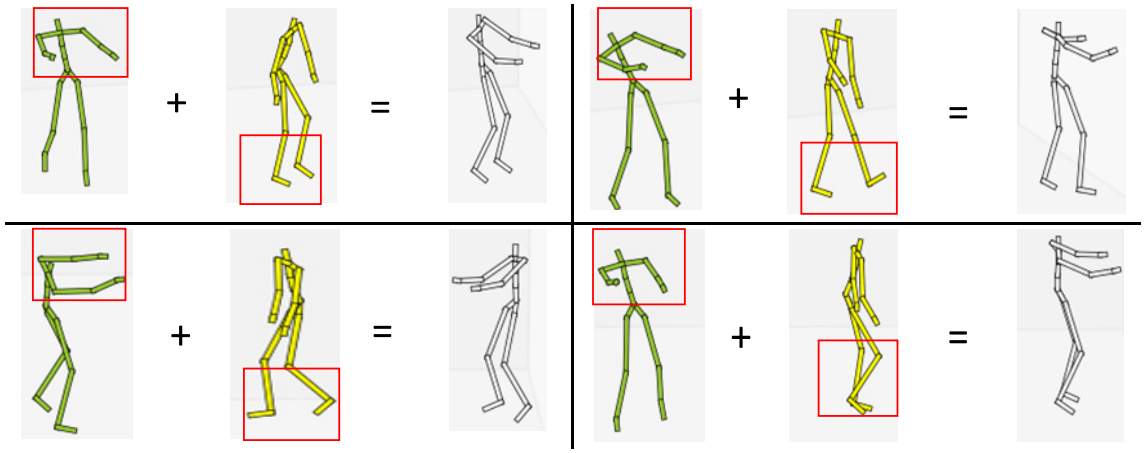


FIGURE 8. Effect diagram of the combination of zombie walking (green) and walking (yellow).When the zombie is walking, its two arm joints are standing up, so the joints of the synthetic characters should also be the same as the actions of the zombies. The walking direction of the walking character is the direction of the synthetic movement. Therefore, when the character walks to the right, the synthetic character faces the right side, and when the walking character moves to the left, the synthetic character also has a tendency to go to the left.

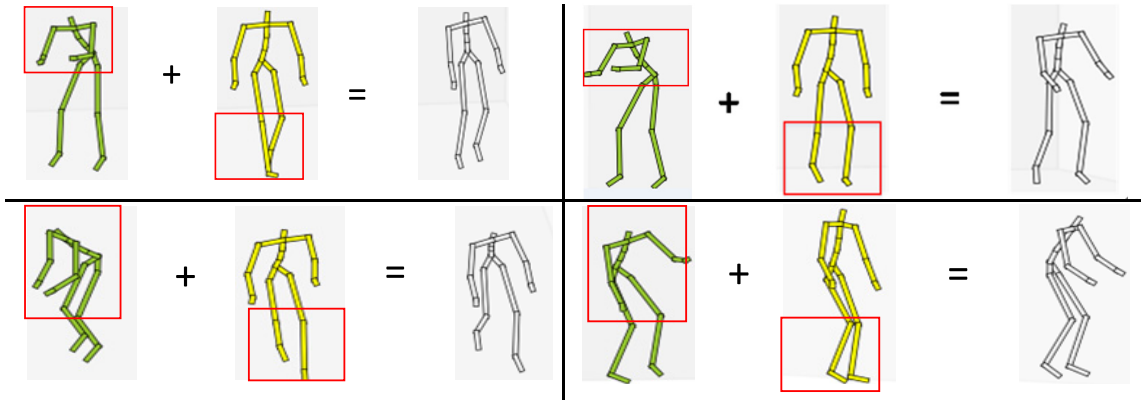


FIGURE 9. Effect diagram of the two movements of the old man walking (green) and running (yellow). Before synthesis, the state of one movement is walking and that of the other is running. In the synthetic action, it includes not only the walking action of the elderly, but also the running, generating a new running action of the old man.

learn the content and path of the action. Holden's method can also learn the path of the action, but in the action of synthesized characters, the movement of the characters is distorted. For example, the blue rectangular box in Figure 13 shows a difference from the original motion in action 1, which is not well synthesized, and the hand joints are not well learned.

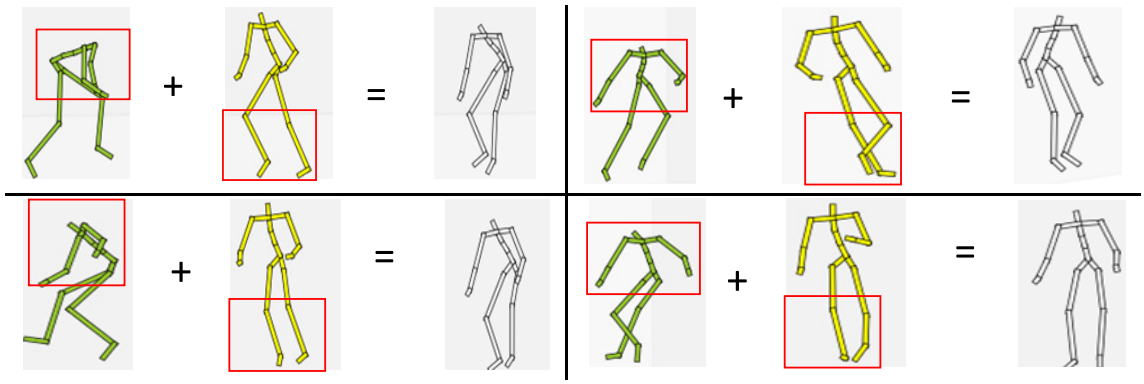


FIGURE 10. Effect diagram of the combination of monkey walking (green) and running (yellow). The two movement states of this figure are also different, and the final synthetic action should be the action of the monkey running.

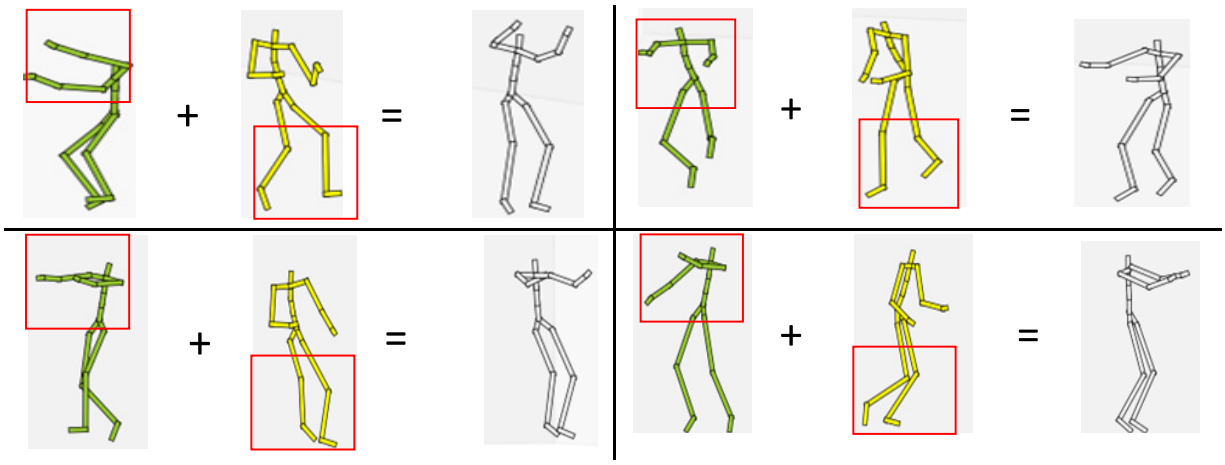


FIGURE 11. Effect diagram of the combination of zombie walking (green) and running (yellow). Similarly, the action in this composition should be a zombie running, and the content of the action is determined by the green character.

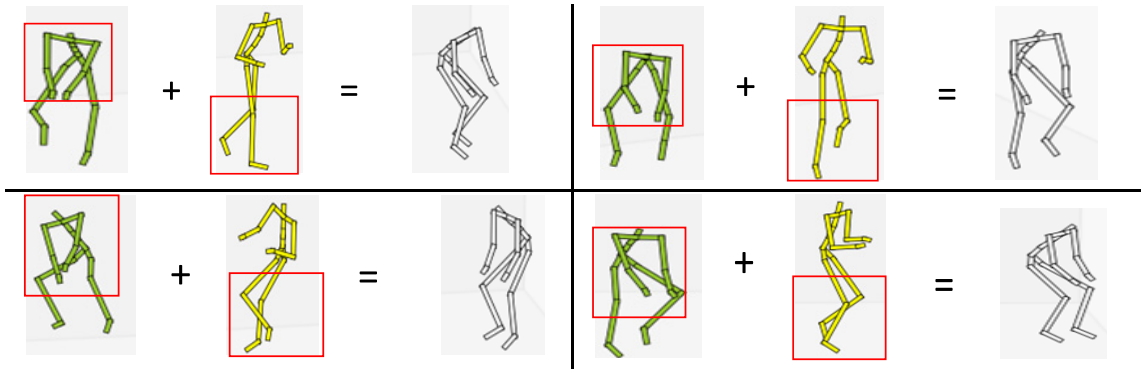


FIGURE 12. Effect diagram of the combination of gorilla running (green) and running (yellow). The movement of the gorilla is more complicated than the previous movement. In the synthetic movement, the yellow character determines the direction in which the gorilla runs.

As for the method in this paper, although there is no complete match between the motion and motion 1, compared with Holden's method, our network structure is relatively simpler, and the synthetic effect looks more natural and realistic. When the motion path changes, the joint coordinates of the characters may also change accordingly, this makes the motion consistent with the real character's inertia requirements.

We performed experiments on the six sets using the Holden's method and our method. In the actions of generating

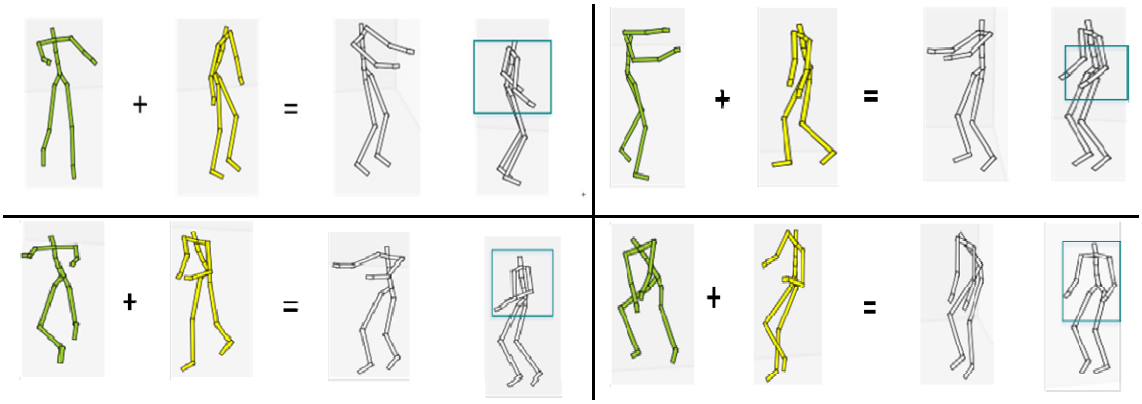


FIGURE 13. Comparison of synthetic animation effects. It can be seen from the above animation effect diagram that when inputting the same set of actions, the method of this paper can better learn the content and path of the action, and the Holden method can also learn the path of the action well. In the action of the synthesized character, however, the action is distorted compared to the method of the present invention.

TABLE 1. Description of important parameters in the motion manifold network.

Parameter	Sign	Value
Window size	n	240
The number of hidden units	m	256
The time filter width	W_4	25
The number of epoch	Epoch	100
Activation function	r	0.25
Regularization	l	0.25

TABLE 2. Time comparison for motion synthesis.

Method	Style/ms	Content/ms	Total/ms
Holden	20.384	8.783	29.167
Ours	14.022	2.714	16.736

six sets of experiments, the total time consumed by the two methods is as shown in table 1. It can be seen that the actions synthesized by the proposed method consumed less time, a decrease about 42.6%.

V. CONCLUSIONS

The presented method mainly introduces the algorithm of human motion synthesis based on convolution auto-encoder. The algorithm is mainly based on a three-layer feed-forward convolutional neural network and the constraints of motion style. In the paper, the re-input of the convolutional self-encoder is used to solve the artifacts and noise problems of the three-dimensional human motion data itself. The experimental results show that the human body motion synthesized by the presented algorithm is natural. And, the human motion trajectory is smooth. Compared with the existing literature, the method performs better in visual effect, and the whole time required for the synthetic action is less.

In future work, we will continue to extend the method to synthesize more types of motion data, especially to synthesize the detailed flexible human motion data, such as hand motion or face motion. In addition, the data mentioned here is limited to human motion data. In the future, we will extend the model to suitable for processing other types of motion data with a different topology than the human skeleton.

SUPPORTING VIDEO

The experimental video can be downloaded from Google Drive:

https://drive.google.com/open?id=12NZoNQJHlCKgHE-yGU1 pcixn2gLJ8WHg

Or it can be downloaded from Baidu Pan:

https://pan.baidu.com/s/1KTMuDIauWesx3tpmX2yE5A

REFERENCES

[1] Y. Liu, L. Feng, S. Liu, and M. Sun, "Sensor network oriented human motion segmentation with motion change measurement," *IEEE Access*, vol. 6, pp. 9281–9291, 2018.

[2] S. Qiu, Z. Wang, H. Zhao, K. Qin, Z. Li, and H. Hu, "Inertial/magnetic sensors based pedestrian dead reckoning by means of multi-sensor fusion," *Inf. Fusion*, vol. 39, pp. 108–119, Jan. 2018.

[3] T. Kwon and S. Y. Shin, "Motion modeling for on-line locomotion synthesis," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, Jul. 2005, pp. 29–38.

[4] S. Xia, C. Wang, J. Chai, and J. Hodgins, "Realtime style transfer for unlabeled heterogeneous human motion," *ACM Trans. Graph.*, vol. 34, no. 4, p. 119, Aug. 2015.

[5] M. E. Yumer and N. J. Mitra, "Spectral style transfer for human motion between independent actions," *ACM Trans. Graph.*, vol. 35, no. 4, p. 137, Jul. 2016.

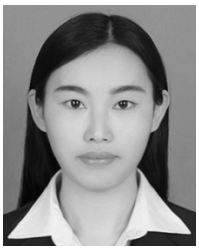
[6] J. Min, H. Liu, and J. Chai, "Synthesis and editing of personalized stylistic human motion," in *Proc. ACM SIGGRAPH Symp. Interact. Graph.*, Feb. 2010, pp. 39–46.

[7] D. Holden, I. Habibie, I. Kusajima, and T. Komura, "Fast neural style transfer for motion data," *IEEE Comput. Graph. Appl.*, vol. 37, no. 4, pp. 42–49, Aug. 2017.

[8] Y. Feng *et al.*, "Mining spatial-temporal patterns and structural sparsity for human motion data denoising," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2693–2706, Dec. 2015.

[9] S. Y. Shin and C. Kim, "Human-like motion generation and control for humanoid's dual arm object manipulation," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2265–2276, Apr. 2015.
[10] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Bayesian nonparametric methods for learning Markov switching processes," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 43–54, Nov. 2010.
[11] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Dec. 2007.
[12] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2006, pp. 1441–1448.
[13] C. Murdock and F. D. L. Torre, "Additive component analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2491–2499.
[14] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4966–4975.
[15] D. Mehta *et al.*, "Monocular 3D human pose estimation in the wild using improved CNN supervision," in *Proc. Int. Conf. 3D Vis.*, Oct. 2017, pp. 506–516.
[16] Q. Tan, L. Gao, Y. K. Lai, and S. Xia, "Variational autoencoders for deforming 3D mesh models," Sep. 2017, *arXiv:1709.04307*. [Online]. Available: https://arxiv.org/abs/1709.04307
[17] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, and T. Komura, "A recurrent variational autoencoder for human motion synthesis," in *Proc. 28th Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., 2017.
[18] Q. Tan, L. Gao, Y. K. Lai, J. Yang, and S. Xia, "Mesh-based autoencoders for localized deformation component analysis," in *Proc. 30th AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 1–8.
[19] Z. Li, Y. Zhou, S. Xiao, C. He, Z. Huang, and H. Li, "Auto-conditioned recurrent networks for extended complex human motion synthesis," Jul. 2017, *arXiv:1707.05363*. [Online]. Available: https://arxiv.org/abs/1707.05363
[20] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2891–2900.
[21] J. Merel *et al.*, "Learning human behaviors from motion capture by adversarial imitation," Jul. 2017, *arXiv:1707.02201*. [Online]. Available: https://arxiv.org/abs/1707.02201
[22] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "MonoCap: Monocular human motion capture using a CNN coupled with a geometric prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 901–914, Apr. 2019.
[23] F. G. Harvey and C. Pal, "Semi-supervised learning with encoder-decoder recurrent neural networks: Experiments with motion capture sequences," *Comput. Sci.*, vol. 3, pp. 553–562, Nov. 2015.
[24] 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.
[25] 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.
[26] C. Mandery, Ö. Terlemez, M. Do, N. Vahrenkamp, and T. Asfour, "Unifying representations and large-scale whole-body motion databases for studying human motion," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 796–809, Aug. 2016.

DONGSHENG ZHOU was born in 1978. He received the Ph.D. degree from the Dalian University of Technology. He is currently a Distinguish Professor of Liaoning Province. His research interests include CG, intelligence computing, and human–robot interaction. He is a member of the ACM, CGS, and CCF.



XINZHU FENG was born in Xuzhou, China, in 1992. She is currently pursuing the master's degree with the School of Information Engineering, Dalian University, and the Key Laboratory of Advanced Design and Intelligent Computing. Her research interests include computer graphics and computer animation.

PENGFEI YI was born in 1983. He received the Ph.D. degree with the Dalian University of Technology, Dalian, China, where he is currently a Lecturer. His research interests include commuter graphics, artificial intelligence, and human–robot interaction.

XIN YANG received the B.S. degree in computer science from Jilin University, in 2007, and the Ph.D. degree, in 2012. From 2007 to 2012, he was a joint Ph.D. student with Zhejiang University and UC Davis for Graphics. He is currently an Associate Professor with the Department of Computer Science, Dalian University of Technology, China. His research interests include computer graphics and robotic vision.

QIANG ZHANG was born in Xian, China, in 1971. He received the M.Eng. degree in economic engineering and the Ph.D. degree in circuits and systems from Xidian University, Xian, China, in 1999 and 2002, respectively. He was a Lecturer with the Center of Advanced Design Technology, Dalian University, Dalian, China, in 2003, and was a Professor, in 2005. His research interest is bio-inspired computing and its applications. He has authored over 70 papers in the above fields. So far, he has served in the editorial board of seven international journals and has edited special issues in journals such as the *Neurocomputing* and the *International Journal of Computer Applications in Technology*.

XIAOPENG WEI was born in Dalian, China, in 1959. He received the Ph.D. degree from the Dalian University of Technology, in 1993, where he is currently a Professor. His research areas include computer animation, computer vision, robot, and intelligent CAD. So far, he has coauthored about 200 papers published.

DEYUN YANG, photograph and biography not available at the time of publication.