

Research Article

Data-Driven Computer Choreography Based on Kinect and 3D Technology

Muyuan Ma,¹ Shan Sun ,¹ and Yang Gao²

¹Liaocheng University, Music and Dance Academy, Shandong Province, Liaocheng City 252000, China

²Shandong Youth University of Political Science Dance Academy, Shandong Province, Jinan City 250000, China

Correspondence should be addressed to Shan Sun; mamuyuan@lcu.edu.cn

Received 22 November 2021; Revised 13 December 2021; Accepted 16 December 2021; Published 19 January 2022

Academic Editor: Tongguang Ni

Copyright © 2022 Muyuan Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a form of artistic expression, dance accompanied by music enriches the cultural life of human beings and stimulates the creative enthusiasm of the public. Choreography is usually done by professional choreographers. It is highly professional and time-consuming. The development of technology is changing the way of artistic creation. The development of motion capture technology and artificial intelligence makes computer-based automatic choreography possible. This paper proposes a method of music choreography based on deep learning. First, we use Kinect to extract and filter actions and get actions with high authenticity and continuity. Then, based on the constant Q transformation, the overall note density and beats per minute (BPM) of the target music are extracted, and preliminary matching is performed with features such as action speed and spatiality, and then, the local features of the music and action segments based on rhythm and intensity are matched. The experimental results show that the method proposed in this paper can effectively synthesize dance movements. The speed and other characteristics of each movement segment in the synthesis result are very uniform, and the overall choreography is more aesthetic.

1. Introduction

Movement is the soul of dance. The first problem that needs to be solved in computer choreography is how to digitize dance movements. Kinect is a somatosensory device released by Microsoft in 2010. Its working principle is to automatically recognize the joint motion data of the human skeleton through infrared rays and capture the skeleton of human motion. Kinect is known as the epoch-making work of the third generation of human-computer interaction technology. In recent years, the use of Kinect for action extraction and then realization of computer dance art has become a research hotspot. In the display of dance, the traditional method is that the choreographer verbally explains or tells the performer by drawing a picture, and then, the dancer shows the specific movement, and subsequent modifications need to be carried out on the actual effect. This requires many repetitive performances by the actors, and the workload is heavy. The use of computer for choreography can show the choreography dance through 3D virtual reality

technology so that the choreographer can create, arrange, modify, and preview the effect in the computer in advance. Based on the above technology, it is possible to make simple choreography using a computer.

Many existing motion generation technologies based on machine learning have been applied to dance research, including dimension reduction technology [1], Gaussian process [2], and hidden Markov model [3], so as to capture the potential correlation between music and dance motion characteristics. Dimension reduction technology can map the high-dimensional features of motion to low-dimensional space, so as to capture the potential correlation behind joint rotation in motion capture data [4]. However, the algorithm needs preprocessing steps, such as sequence alignment and fixed data length, and cannot directly model the timeliness of motion data, which limits its application in real dance motion data. Gaussian process late variable models can effectively summarize the changes of human cloud force, but they are not suitable for real-time generation because they require a lot of computing and memory resources [5]. HMM

model overcomes the limitations of the two types of models mentioned above, but its ability to capture data changes is limited [6]. In order to solve the defects of computer choreography based on machine learning, this paper will introduce deep learning method to improve the novelty and coherence of generated actions.

2. Methods

2.1. Overview. First of all, the problem that needs to be solved is motion capture based on Kinect. When Kinect captures human body motion data, it captures 20 joint nodes of the human body through infrared rays. As long as you face Kinect and do dance movements, you can record the corresponding raw data. This avoids the need to place sensors on the capturer during traditional motion capture, which hinders the fluency of dance movements. Convert the captured raw data into the BVH file format to save the dance moves. The motion data file in BVH format is one of the standard file formats in the motion capture industry and is widely supported by mainstream animation production software. When using Kinect to collect dance movements, it is conducive to the optimal display and file storage of virtual dances based on the “Principle of Optimal Movements” and “Principle of Minimal Data Simplification” [7].

Secondly, it is necessary to edit and integrate dance movements. Using the 3D software MAYA as the platform, the basic dance moves stored in the BVH format are imported into MAYA for editing. The choreographer can make modifications and combinations based on the existing basic dance moves, such as modifying the angle of the limbs, the speed of a certain movement, and the number of repetitions [8]; at the same time, they can also arrange and integrate the sequence of specific dance moves to express different dance themes. In addition, the choreography of dance emphasizes innovation. Using Kinect equipment, you can capture newly designed dance moves at any time. After converting it into BVH format, it becomes the new basic action element.

Finally, it is necessary to match the skeleton and the character and realize the three-dimensional display. After the preliminary editing of a certain dance movement is completed, the 3D characters can be driven to dance. MAYA provides the function of binding the skeleton to the model so that the 3D model follows the movement of the skeleton. Modify the bound model by adjusting the weight of the model, the size of the skeleton, and other factors so that it can express the dance effect as realistically as possible. For the choreography of large-scale dances, after the choreography of a certain model is completed in advance, the skinned models can be copied and arranged directly to create the ideal formation and stage effect.

2.2. Motion Capture

2.2.1. Two-Dimensional Key Point Recognition Network. The key points of the two-dimensional human body are identified using the Stacked Hourglass Network architecture. This network is constructed by loop nesting of the Hourglass subnetwork. The Hourglass subnetwork is

composed of a basic structure called residual module. The residual module structure is shown in Figure 1(a). The input value will pass through the upper and lower paths. This design is inspired by the residual network. The above path is a convolution path, which contains three convolution layers with different sizes of convolution kernels, which are represented by white rectangles in the figure. The three lines of text in the rectangle represent the number of input channels (N_{In}), the size of the convolution kernel (K), and the number of output channels (N_{Out}) from top to bottom. Among the three convolutional layers, there are also batch regularization layers and nonlinear activation layers (RELU), all of which are represented by gray rectangles. The lower path is a jumper path. This path contains only a convolutional layer with a convolution kernel size of 1×1 , which will change the number of channels of the input value, and its output value will be directly added to the output value of the upper path.

The structure of the Hourglass subnetwork is shown in Figure 1(b). Each white rectangle in the figure represents one of the above residual modules. In the upper half, feature extraction is performed continuously, and in the lower half, downsampling is performed through the maximum pooling operation first, and after several residual modules, the nearest neighbor interpolation method is used for upsampling. The two gray rectangles in the figure represent the downsampling and upsampling processes, respectively. It can be seen that there is a dashed line frame in the Hourglass subnet structure diagram. The dashed line frame is replaced by an Hourglass subnet, which is called a second-order Hourglass network. In this study, the Hourglass subnetwork is looped and nested four times for two-dimensional key point detection, namely, the fourth-order Hourglass network.

2.2.2. Three-Dimensional Keypoint Regression Neural Network. After the two-dimensional human body key points are recognized, a set of two-dimensional coordinate values of the human body key points will be output. Then, use this set of two-dimensional coordinates $x \in R^{2n}$ and use the neural network to return to the corresponding set of three-dimensional coordinate values $y \in R^{3n}$. The mapping relationship of the three-dimensional coordinates regressed from the two-dimensional coordinates is $f: R^{2n} \rightarrow R^{3n}$. Optimize the neural network by minimizing the following prediction errors, namely,

$$f^* = \min \frac{1}{N} \sum_{i=1}^N L(f(x_i) - y_i). \quad (1)$$

Among them, L represents the Euclidean distance between vectors as the loss function, N represents the total number of key points to be identified, and f^* represents a regression neural network. The structure of the regression neural network model is shown in Figure 2. First, the dimension is converted to 1024 dimensions through a fully connected layer. After that, it passes through the batch regularization layer, the RELUs layer, and the Dropout layer in turn and contains a jump-through path. The structure in the

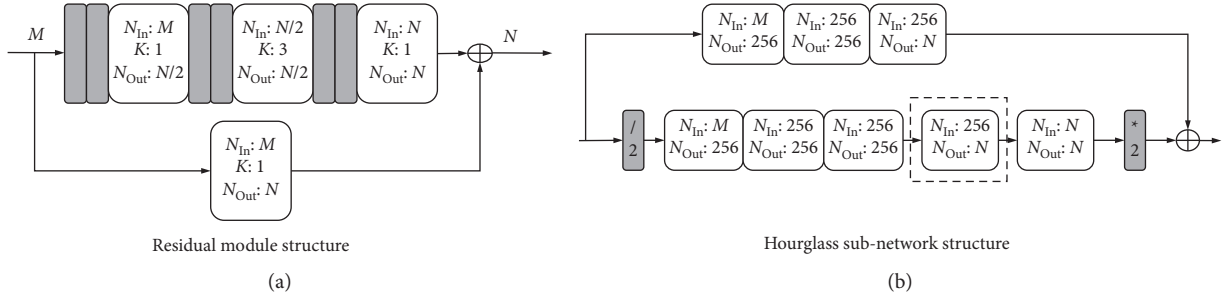


FIGURE 1: Two-dimensional key point recognition network. (a) Residual module structure. (b) Hourglass subnetwork structure.

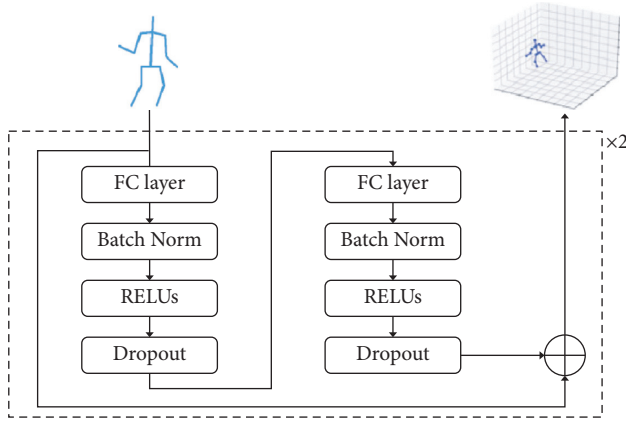


FIGURE 2: Three-dimensional key point regression neural network.

dashed frame in Figure 2 will be repeated once, and finally, a 3n-dimensional output vector will be formed through a fully connected layer. So far, the three-dimensional coordinates of the key points of the presenter are obtained.

2.3. Action Recurrence. After obtaining the coordinates of the key points of the presenter, they need to be mapped to the robot. The control of the robot takes the angle of each joint as the control command, and the process of using the coordinate value of each key point to calculate the joint angle is called inverse kinematics calculation of the robot. The names of each joint and the actual range of degrees of freedom are shown in Table 1.

In this paper, for each joint of the robot, its angle value is represented by θ_m , and m ($m = 1, 2, \dots, 16$) is the number of the joint. The n ($n = 1, 2, \dots, 15$)th key point captured by point p_n , and the vector formed by points p_{n_1} and p_{n_2} is $p_{n_1}p_{n_2}$. The key point numbers detected are shown in Figure 3.

In the angle calculation, there are mainly three situations: the angle between the vector and the vector, the angle between the vector and the plane, and the angle between the plane and the plane [9]. The following examples illustrate the calculation formulas for the three cases.

2.3.1. Calculation of the Angle between the Vector and the Vector. Here we take θ_3 as an example to introduce how to

TABLE 1: List of joints.

Num.	Joint name	Freedom range (°)
1	RShoulderPitch	0~180
2	RShoulderRoll	0~180
3	RElbow	0~180
4	LShoulderPitch	0~180
5	LShoulderRoll	0~180
6	LElbow	0~180
7	RHipRoll	0~180
8	RHipPitch	30~180
9	RKnee	0~150
10	RAnklePitch	90~180
11	RAnkleRoll	70~180
12	LHipRoll	0~180
13	LHipPitch	0~150
14	LKnee	30~180
15	LAnklePitch	0~180
16	LAnkleRoll	0~110

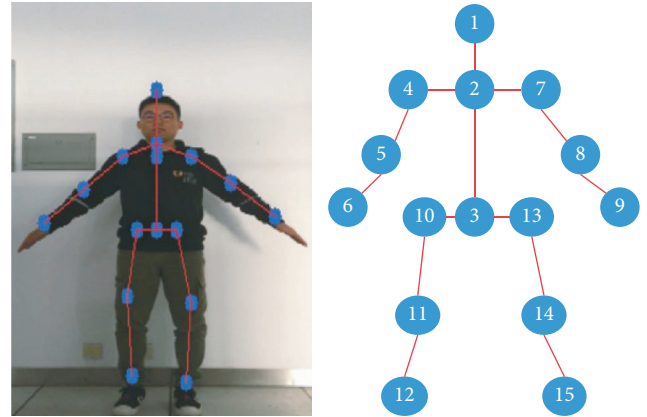


FIGURE 3: Correspondence of feature points.

calculate Relbow. The value of θ_3 is the angle between the vector p_4p_5 and the vector p_5p_6 . The calculation formula is

$$\theta_3 = (p_4p_5, p_5p_6) = \arccos\left(\frac{p_4p_5 \cdot p_5p_6}{|p_4p_5||p_5p_6|}\right). \quad (2)$$

2.3.2. Calculation of the Angle between the Vector and the Plane. For example, the angle θ_8 of the joint RHipPitch is the angle between vector $p_{10}p_{11}$ and the plane formed by vectors p_3p_{10} and p_3p_2 . To solve θ_8 , first use the cross

product of p_3p_{10} and p_3p_2 to obtain the normal vector $n_{2,3,10}$ of the plane, then calculate the angle of the normal vector $p_{10}p_{11}$, and then calculate the remaining angles to obtain θ_8 :

$$\theta_8 = \frac{\pi}{2} - (n_{2,3,10}, p_{10}p_{11}) = \frac{\pi}{2} - \arccos\left(\frac{n_{2,3,10} \cdot p_{10}p_{11}}{|n_{2,3,10}| |p_{10}p_{11}|}\right). \quad (3)$$

2.3.3. Calculation of the Angle between the Plane and the Plane. For example, if the RShoulderPitch angle θ_1 of the joint is solved, the angle between the plane and the plane is solved. Specifically, first cross-multiply vectors p_5p_6 and p_5p_4 to obtain a plane normal vector $n_{4,5,6}$, then cross-multiply vectors p_2p_4 and p_2p_3 to obtain a normal vector $n_{2,3,4}$ of another plane, and then calculate the angle between the two normal vectors to obtain θ_1 :

$$\theta_1 = (n_{4,5,6}, n_{2,3,4}) = \arccos\left(\frac{n_{4,5,6} \cdot n_{2,3,4}}{|n_{4,5,6}| |n_{2,3,4}|}\right). \quad (4)$$

In the calculation process of each angle, due to the selection of the vector direction, it is necessary to find the complementary angle or the complementary angle according to the situation; considering the angle range of each joint, it is necessary to limit the calculated angle value to ensure that it is not exceeding the maximum allowable angle value.

The above calculation process can get all the angle values except the two degrees of freedom of the ankle [10]. For the calculation of the angle of the ankle joint, the relative position of the robot and the ground needs to be taken into consideration because the sole of the foot must be parallel to the ground. As shown in Figure 4, θ_{10} is the angle between the vector $p_{12}p_{11}$ and the plane $Zp_{10}Y$, and θ_{11} is the angle between the plane where the legs are located and the plane $Zp_{10}Y$, where the plane and the legs are located is the plane formed by the vectors $p_{12}p_{11}$ and $p_{11}p_{10}$. θ_{15} and θ_{16} can be obtained in the same way.

2.4. Action Selection Algorithm Based on Continuity. Just as a classification model cannot guarantee that the classification accuracy rate reaches 100%, the actions generated by the action generation model also guarantee that every frame generated is a real, coherent, and high-quality action. Therefore, in order to make the generated action suitable for music-based dance choreography, it is necessary to perform a coherence-based action screening first to remove mutation data, so as to ensure that the action data of each frame in an action segment is coherent and improve the quality of the generated action [11].

If an action sequence is coherent, then the actions of adjacent frames in the sequence should have sufficient similarity, which is reflected in the small distance between the corresponding joint points. The motion data used in this article are sampled at a frequency of 30 frames per second. The distance between the corresponding joint point positions in two adjacent frames can be approximated as the speed of the joint at that moment. Because the overall speed

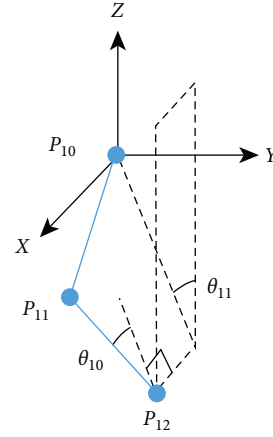


FIGURE 4: Schematic diagram of ankle joint angle calculation.

of different action sequences is different, it is not easy to judge whether there is a sudden change or a pure rapid movement based solely on the absolute position change [12]. This study believes that, for a coherent action sequence, regardless of the overall speed, the rate of change of the speed between adjacent frames should be small, that is to say, for a sequence with a faster action speed, the instantaneous speed of each frame is higher. For a sequence with a slower action speed, the instantaneous speed of each frame is smaller, but for a continuous action sequence, the speed difference between adjacent frames should be relatively stable.

Therefore, this paper conducts continuous screening of the action sequence based on the rate of speed change. First, calculate the sum of the absolute value of the first-order difference of the joint speeds of adjacent frames:

$$V(f) = \sum_{k=1}^c |v(f+1, k) - v(f, k)|, \quad (5)$$

$$v(f, k) = \left\| x_{f+1}^{(k)} - x_f^{(k)} \right\|.$$

Among them, f is the sequence number of the frame in the action segment N_i , x is the action vector, $x_f^{(k)}$ is the k th dimensional action data of the f th frame, and c is the vector dimension of the action per frame. $v(f, k)$ represents the speed of the k th dimensional data of the f th frame.

Set the maximum threshold ϵ_{\max} if the f th frame action satisfies $V(f) \leq \epsilon_{\max}$; the action at that moment meets the continuity requirement. According to the condition, the action sequence is divided, and the continuous action frame that meets the condition is divided into a new action sequence. In order to remove new sequences whose length is too short after segmentation, a minimum length threshold value needs to be set, and only new sequences whose length exceeds this threshold value can be saved in the generated action database.

2.5. Choreography and Composition. Through the feature extraction and matching algorithm of music and action in the previous section, multiple action segments that match the target music have been obtained, and the connectivity constraints between adjacent segments have been met. In

this section, on this basis, the adjacent action segments will be transitionally connected to solve the problem of sudden action changes. The action segments will be spliced into a complete sequence of actions to complete the final choreography. The sudden change of action referred to in this section refers to the fact that there is a certain distance between the action at the connection between the action segment and the adjacent action segment N_{i+1} , which causes the directly connected actions to be incoherent, which affects the visual effect of the dance.

This section uses the intermediate frame interpolation algorithm to interpolate between the end k frames of the action segment M and the intermediate action of the action segment N according to the interpolation weight to obtain the final interpolation action. The interpolation action obtained by using this algorithm can realize the natural connection of two action segments and, at the same time, retain the characteristics of the ending action of the previous action segment to a certain extent [13]. The interpolation action ensures the continuity of the action, but there may be unreal actions such as footsteps. In order to avoid the long duration of the interpolation action and affect the perception, the value of k should not be too large. In this study, $k = 14$ is used in the experiment.

Suppose the length of the action segment M is m , the last k frame action is recorded as $M_{m-k+1} = \{f_{m-k+1}^M, f_{m-k+2}^M, \dots, f_m^M\}$, and the first k frame intermediate action of the action segment N is recorded as $N_{1,\dots,k} = \{f_1^N, f_2^N, \dots, f_k^N\}$. First, the starting position of the intermediate action $N_{1,\dots,k}$ is translated to the same as M_{m-k+1} , and then, the interpolation action $P_{1,\dots,k}$ is synthesized. Among them, $P_{1,\dots,k} = \{f_1^P, f_2^P, \dots, f_k^P\}$ performs linear interpolation on the node displacement:

$$p_i^{P,s} = \alpha(i)f_{m-k+i}^{M,s} + (1 - \alpha(i))f_i^{N,s}, \quad 1 \leq i \leq k, \quad (6)$$

$$\alpha(i) = 2 \cdot \left(\frac{i}{k}\right)^3 - 3 \cdot \left(\frac{i}{k}\right)^2 + 1, \quad 0 \leq i \leq k - 1.$$

Among them, $p_i^{P,s}$ represents the coordinates of the s th joint point of the i th frame of the P action segment, and $\alpha(i)$ is the interpolation weight.

3. Experiments and Results

3.1. Dataset. In recent years, the research on dance movement synthesis is mainly based on motion capture data. At present, the main public motion capture motion datasets are as follows. (1) SBU dataset includes 8 types of actions, a total of 230 sequences of 6614 frames, but the actions in the dataset are all nondance movements such as handshake and punching. (2) HDM05 data set contains about 100 action types, with a total of 2337 sequences of 1840046 frames, but basically all kinds of nondance actions such as walking and kicking. (3) UCY dataset (University of Cyprus), containing a total of 161 sequences 147509 frames, providing dance moves in Greek, Cypriot dance, and other styles, but only 8 of them are relatively complete movements accompanied by music, a total of 28892 frames, and the rest of the sequences

are single movement fragments with a short time, which is not conducive to complete dance [14]. (4) The CMU dataset contains a total of 2235 sequences of 98,7341 frames. This is the largest motion capture dataset published so far, covering a wide range of motion types, of which only 64,300 frames are for pure dance movements, with no accompanying music. To sum up, although there are some public exercise datasets, most of them are not dance moves, and there are very few dance moves' data accompanied by music [15]. For the study of the relationship between music and dance movements based on deep learning, movement and music data play a key role in the training of the model.

Therefore, this paper additionally constructs a music-action dataset composed of a complete music choreography sequence. Compared with using professional dancers, it is more economical and convenient to obtain enough dance data through motion capture equipment and download the motion data in VMD format corresponding to different music on the Internet. This paper uses the VMD action files obtained from the Internet and the accompanying WAV music files to construct a music-dance action dataset, with a total of 192 segments, 1057344 frames, and about 587 minutes. Each segment is an independent dance.

The music and dance styles contained in the dataset constructed in this study are not exactly the same, and the speed is also fast and slow. Therefore, it is necessary to classify the actions before performing network training. Based on manual experience, combined with the overall characteristics of the 192 songs and dances in the dataset, the overall style of music and dance can be divided into three categories: modern dance, street dance, and house dance. The overall speed is divided into two categories, fast and slow, for a total of six categories. Since the overall speed of the movement in the same song and dance is not static, for example, the movement is more relaxed at the beginning and end of the song and the movement is more intense at the climax of the song, so the overall dance speed of the entire song is not enough. To classify actions, based on the overall speed of the dance, this study continues to divide movements into fast and slow based on the frame granularity. Based on the above principles, this study manually annotates the constructed dataset. Table 2 shows the number of frames and duration of various actions after classification.

3.2. Model Training and Prediction. In order to obtain good results in deep neural network training, it is necessary to provide sufficient data so that the neural network can fully tap the inner relationship between the data. For the three types of dances, house dance, street dance, and modern dance are included in the dataset constructed in this paper; according to the data volume of different speed movements, this paper trains different movement generation models. During training, 12 Gaussian distributions are used to form a mixed model ($m = 12$), the number of batches (batch size) is set to 100, the sequence length of one-time input is set to 120, the learning rate is 500, and the total training is 500 cycles (epoch), optimized using the RMS Prop optimizer and

TABLE 2: Detailed information of various types of data in the dataset.

Dance style	Overall speed	Number of fragments	Frame rate	Number of frames	Duration (min)
House dance	Fast	54	Fast	285393	158.6
			Slow	26428	14.7
	Slow	35	Fast	157175	87.3
			Slow	54034	30.0
Street dance	Fast	67	Fast	357287	197.4
			Slow	18048	10.0
	Slow	6	Fast	25504	13.7
			Slow	9063	4.9
Modern dance	Fast	5	Fast	21679	12.6
			Slow	249	0.1
	Slow	25	Fast	41516	22.6
			Slow	56410	31.9

the learning rate is set to 0.01. Table 3 shows the training results at different times.

Figure 5 shows the model loss of the training set and validation set during model training. It is worth noting that the error function E^q needs to be minimized during the model training process. Unlike the commonly used loss functions in other networks (such as cross entropy), E^q does not meet the condition of constant greater than zero. Therefore, when the model loss is less than zero, the training process will be terminated in advance, as shown in Figure 5(a). The loss of the validation set and the loss of the training set are inconsistent, and there is even no obvious downward trend. This is because dance movement generation tasks are different from other tasks such as target classification. The choreography and expression of dance moves are not unique. This is where the diversity of dance moves lies. The training process of the movement generation model seeks regularity in it. The diversity and regularity of data are a pair of contradictory standards. Although the actions in the dataset have been preliminarily classified, in the limited dataset, it is difficult to ensure that the verification set selected at random each time has the same law as the training set. Therefore, in the verification set during the first 350 cycles of training, the loss has not changed much. As the model continues to be trained, overfitting occurs, resulting in a sharp increase in the loss of the validation set. You can refer to the loss of the validation set to judge the training situation of the model.



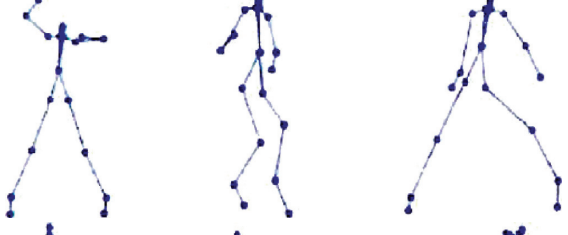

3.3. Qualitative Results. According to the visual effect of the synthesized dance, the choreography result is evaluated, and the effect of the algorithm in this paper is evaluated. First, extract the overall characteristics of “Tokyo Teddy Bear;” after calculation, the BPM value is 126.05, the duration of the change note is 1.93, and the rapid house dance motion generation model is selected to generate the candidate motion database, which is consistent with the user’s intuitive hearing. Observing the final dance effect, you can feel that the rhythm and intensity of the dance and the target music match to a certain extent, and the movements are smooth and coherent. Figure 6 shows a posture snapshot of the synthesized dance. Only from the intuitive visual effects, the

choreography algorithm in this paper can be considered to be effective.

3.4. Quantitative Results. Experimental purpose: use user ratings to analyze the synthesizing effects of three styles of dance. First, analyze the style of the music according to the overall characteristics of the target music (BPM and the duration of changing notes) and generate the corresponding choreography. This experiment analyzes a number of target music, selects three target music suitable for generating hip-hop, house dance, and modern dance, and choreographs them. The detailed information and overall characteristic values of the target music are shown in Table 4. The test users, respectively, judged the dance styles of the three segments and evaluated the matching degree of music and dance and the continuity and authenticity of dance movements, and the results are shown in Table 5.

From the results, it can be found that the average value of the coherence, authenticity, and matching with music of the three styles of dance are all above 4 points, indicating that users are satisfied with the results of the synthesized dance styles. Therefore, the music choreography algorithm proposed in this paper is Effective. The three indicators of street dance have the highest scores, followed by modern dance, and finally house dance. This study interviewed these users after they completed the test. The users said that the rhythm of hip-hop style movements is obvious and the range of movements is larger, while the range of house dance movements is generally small. Sometimes, it is difficult for users to distinguish between small movements and jitter data. After analysis, the reason why the synthesis result of house dance is worse than that of street dance is that, in the action dataset constructed in this study, the diversity of house dance actions is more abundant, and the concentration of actions is worse than that of street dance, so it is not conducive to the training and learning of the action generation model. Among the 33 users who participated in the score, only 1 user made a wrong judgment on the style of hip-hop and home dance, and the judgments of other users were accurate. After follow-up interviews, the user stated that it was because she did not know the specific concepts of home dance and hip-hop and unable to judge.

TABLE 3: Action generation results at different times (street dance actions).

Epoch	Time	Generate sample action poses	Describe
1	2.5 min		Unable to generate qualified actions, most joint positions are random, and the outline of the human body can be vaguely seen
50	~2 h		Understand the joint positions and be able to generate basic movements
200	~9 h		The relative relationship of joints is more stable, which can generate richer and more stable movements
500	~22 h		The generated action has unstable continuity and may be overfitting

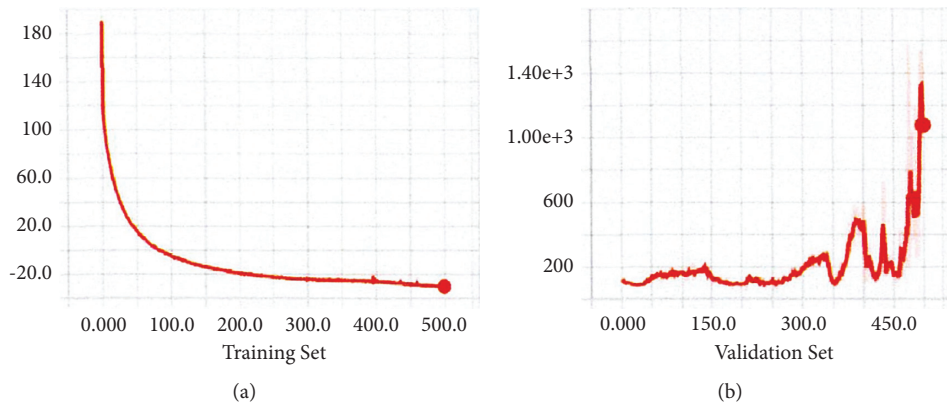


FIGURE 5: Model loss. (a) Training set. (b) Validation set.

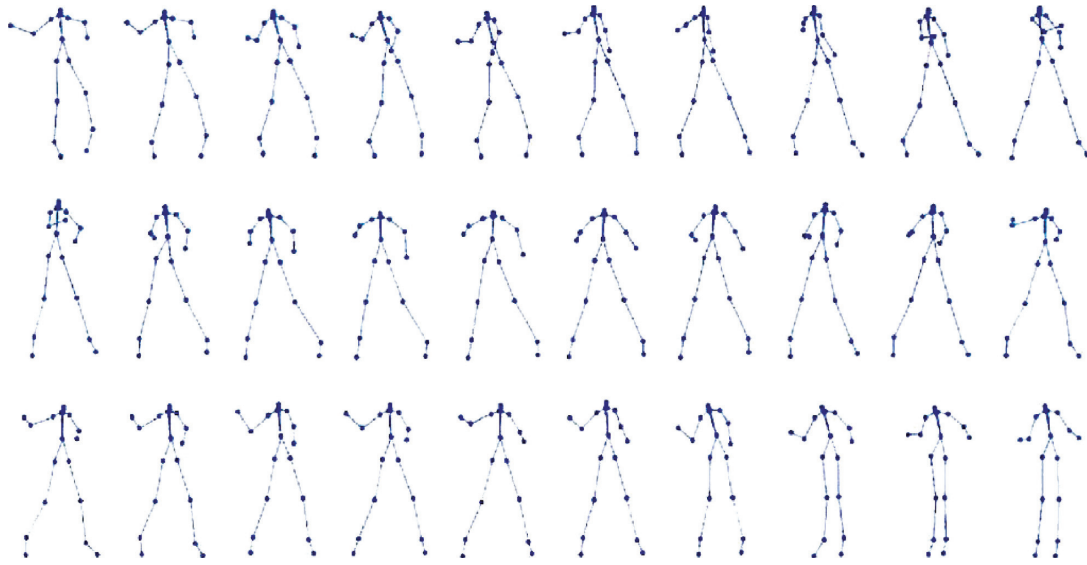


FIGURE 6: “Tokyo teddy bear” synthetic dance pose snapshot.

TABLE 4: Details of the three target music.

Music name	Duration	Music BPM	Varying note duration	Dance style
Bad apple	00:05:10	138.6	1.57	Street dance
Chocolate	00:03:46	129.2	1.66	House dance
From Y to Y	00:05:30	96.4	2.54	Modern dance

TABLE 5: Manual scoring of three styles of dance.

Dance style	Music and dance match	Continuity	Authenticity	Style judgment accuracy (%)
Street dance	4.28 ± 0.10	4.07 ± 0.07	4.04 ± 0.10	96.9
House dance	4.13 ± 0.09	4.05 ± 0.08	4.02 ± 0.08	96.9
Modern dance	4.22 ± 0.11	4.08 ± 0.08	4.08 ± 0.11	100

4. Conclusion

This paper proposes a dance choreography algorithm based on the deep learning model, starting from improving the harmony of music and choreography. First, extract the overall characteristics of the music, including the average duration of BPM and notes, and perform preliminary matching of the action speed characteristics based on this. Then, complete segmentation of music and action sequences and feature extraction of rhythm intensity and combine the results of feature matching and connectability analysis to obtain action sequences that match the target music. Finally, the adjacent action fragments are interpolated to complete the computer music choreography. In this paper, qualitative and quantitative experiments are designed to evaluate the effectiveness of the algorithm. The experimental results show that the local bone movement speed feature extraction algorithm and the dance spatial feature extraction algorithm proposed in this paper can effectively reflect the corresponding characteristics of dance; the overall characteristics of music can accurately reflect the style of music, and the corresponding styles can be synthesized accordingly. Choreographed action: the hierarchical

feature matching algorithm is better than the feature matching algorithm based on local rhythm and intensity. The addition of the overall feature matching can synthesize a dance that is more matched to the target music. In the future, it is necessary to further improve the accuracy of the model through the optimization of the model structure to make the generated actions more realistic.

Data Availability

The dataset can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Shandong Provincial Social Science Planning Research Project-“Research on the Differences of Inheritance Modes of Traditional Dance between

China and South Korea (1948–2020)” (21WYJ14) and Shandong Provincial Art Science Key Research Project: “Research on Inheritance Strategy of Shandong Red Dance in the New Era” (L2021Z07080285).

References

- [1] C.-M. Hsieh and A. Luciani, “Generating dance verbs and assisting computer choreography,” in *Proceedings of the 13th Annual ACM international Conference on Multimedia*, Hilton Singapore, 6 November 2005.
- [2] F. Sagasti, “Information technology and the arts: the evolution of computer choreography during the last half century,” *Dance Chronicle*, vol. 42, no. 1, pp. 1–52, 2019.
- [3] T. Joseph Dube and G. İnce, “A novel interface for generating choreography based on augmented reality,” *International Journal of Human-Computer Studies*, vol. 132, pp. 12–24, 2019.
- [4] M. Autili, A. Di Salle, F. Gallo, C. Pompilio, and M. Tivoli, “CHOReVOLUTION: service choreography in practice,” *Science of Computer Programming*, vol. 197, Article ID 102498, 2020.
- [5] S. Jadhav, M. Joshi, and J. Pawar, “Art to SMart: an automated BharataNatyam dance choreography,” *Applied Artificial Intelligence*, vol. 29, no. 2, pp. 148–163, 2015.
- [6] F. Visi, E. Coorevits, R. Schramm, and E. R. Miranda, “Musical instruments, body movement, space, and motion data: music as an emergent multimodal choreography,” *Human Technology*, vol. 13, p. 1, 2017.
- [7] M. Autili, P. Inverardi, and M. Tivoli, “Choreography realizability enforcement through the automatic synthesis of distributed coordination delegates,” *Science of Computer Programming*, vol. 160, pp. 3–29, 2018.
- [8] Y. Zhou, “Application of automatic choreography software based on virtual technology in the gymnastics teaching,” *International Journal of Emerging Technologies in Learning*, vol. 11, p. 5, 2016.
- [9] S. Basu and T. Bultan, “Automated choreography repair,” in *Proceedings of the International Conference on Fundamental Approaches to Software Engineering*, Springer, Berlin, Heidelberg, 2 April 2016.
- [10] V. de Boer, J. Jansen, A.-L. Tjon-A-Pauw, and F. Nack, “Interactive dance choreography assistance,” in *Proceedings of the International Conference on Advances in Computer Entertainment*, Springer, London, UK, December 2017.
- [11] P. Valderas, V. Torres, and P. Vicente, “A microservice composition approach based on the choreography of BPMN fragments,” *Information and Software Technology*, vol. 127, Article ID 106370, 2020.
- [12] Z. H. O. U. Yingbao, “Application of automatic choreography software based on virtual technology in the gymnastics teaching,” *International Journal of Emerging Technologies in Learning (IJET)*, vol. 11, no. 5, pp. 39–44, 2016.
- [13] J. Jiang, “Choreography algorithm based on hybrid density network,” in *Proceedings of the International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy*, Springer, Shanghai, China, 6 November 2021.
- [14] S. Orlando, V. D. Pasquale, F. Barbanera, I. Lanese, and E. Tuosto, “Corinne, a tool for choreography automata,” in *Proceedings of the International Conference on Formal Aspects of Component Software*, Springer, Cham, 28 October 2021.
- [15] L. D. Introna, “The algorithmic choreography of the impressionable subject,” *Algorithmic Cultures*, pp. 38–63, Routledge, Oxfordshire, England, UK, 2016.