Authoring System for Choreography Using Dance Motion Retrieval and Synthesis

Ryo Kakitsuka*[1] Kosetsu Tsukuda[2] Satoru Fukayama[2]
Naoya Iwamoto[1] Masataka Goto[2] Shigeo Morishima[1]

[1]Waseda University
[2]National Institute of Advanced Industrial Science and Technology (AIST)

Abstract

Dance animations of a three-dimensional (3D) character rendered with computer graphics (CG) have been created by using motion capture systems or 3DCG animation editing tools. Since these methods require skills and a large amount of work from the creator, automated choreography synthesis system has been developed to make dance motions much easier. However, those systems are not designed to reflect artists' preferred context of choreography into the synthesized result. Therefore, supporting the user's preference is still important. In this paper, we propose a novel dance creation system that supports the user to create choreography with their preferences. A user firstly selects a preferred motion from the database, and then assigns it to an arbitrary section of the music. With a few mouse clicks, our system helps a user to search for alternative dance motion that reflects his or her preference by using relevance feedback. The system automatically synthesizes a continuous choreography with the constraint condition that the motions specified by the user are fixed. We conduct user studies and show that users can create new dance motions easily with their preferences by using the system.

Keywords: relevance feedback, character animation, motion capture, dance

*Corresponding author:kakitsuka.99821@ruri.waseda.jp

1 Introduction

Owing to the recent development of 3DCG tools, 3D character dance animation has been widely used in the entertainment field; for example, in films and computer games, and even for live performances in the form of "virtual popstar", a 3DCG character animation that projected on stage. Although there is a great necessity for choreographic authoring tools for character dance animation, using 3DCG animation editing tools still requires a large amount of work, skill, and dance knowledge.

In light of the above, the automatic synthesis of dance motion has been the method used to construct a 3D character dance animation with ease. However, fully automated synthesis [1, 2] does not allow the animators to express their preferences. There are methods for editing timing and speed of exercise in a motion sequence [3]. Although these methods are useful to reflect a user's preference to choreography or to edit the dance motion more specifically, it is still insufficient when a user wants to replace motion in a choreography with another one in a specific part of the music. In this case, motion retrieval technique enables the animators to choose the alternative motion which matches the preference from the database. However, existing motion retrieval methods were insufficient for the purpose, since they often place a large burden on animators by requiring input motions as queries by means of figure drawing [4, 5] or motion capture devices such as Microsoft Kinect sensors[1] [6, 7].

The goal of this study is to provide an authoring system for creating character dance animation with users' preferences that does not require any knowledge of choreographic composition and variations, or skills for constructing CG animations. Our approach consists of the following two steps: 1) the user assigns a preferred motion obtained through our motion search algorithm to an arbitrary section of the given music. 2) our system automatically assigns motion segments from the database to the remaining sections of the music so that the sequence of motions matches the music input considering the connectivity of the motion segments and the climax level of music.

To implement the system, we address two technical issues. The first one is how to enable a user to easily search for the preferred motion from the motion database. The second is how to automatically synthesize a dance sequence under the condition where the motions selected by the user are fixed. We solve the first problem by visually presenting motion sequences that can be changed based on user's relevance feedback [8] and the diversification framework [9]. It is difficult to judge whether a dance motion is good or not without previewing a dance in conjunction with a corresponding musical piece. In our system, the user can view candidate sequences on the screen and find a preferred alternative motion with only a few mouse clicks. We address the second issue filling the time between fixed parts with reasonable dance motions automatically selected from the database. We select the motions by minimizing the sum of the transition cost function using Dijkstra's algorithm.

The main contributions in this paper are as follows.

1. We introduce a relevance feedback framework in dance motion retrieval from the dance motion capture database. Our method enables a user to easily search for the preferred dance motion.

2. Our method can automatically synthesize a dance sequence considering the connectivity of the motion segments and how well a candidate motion segment matches a corresponding piece of music.

3. We propose a new system for constructing character dance animation considering the animator's preference. Results of a user study with the system showed that participants found the system to be suitable for easily creating new dance motions with their preferences.

[1] www.xbox.com/en-US/kinect

2 Related Work

This section describes previous research relating to the two technical issues referred to above, namely, motion retrieval methods and study of automatic dance motion synthesis.

2.1 Motion Retrieval

In the past decade, motion retrieval technologies have been advancing owing to the increasing availability of large-scale motion databases[2,3]. There are several ways to retrieve a desired motion sequence from a large-scale human motion database, the simplest being the use of textual description key-words as queries; for example, "walking", "jumping", or more detailed sentences such as "a kick of the left foot followed by a punch" [10]. While this approach is intuitive and efficient, it requires a large amount of manual effort to annotate all the motion sequences in the database, and it is not sufficient for retrieving certain desired dance motion sequences, because some dance motions do not have appropriate names. To overcome these limitations, many researchers have studied the content-based human motion retrieval methods [6, 7, 11]. With those approaches, the desired motion sequences are retrieved by submitting a similar motion sequence as a query using simple and easy motion capture devices such as Microsoft Kinect sensors or the Perception Neuron[4]. However, it is still difficult to acquire the appropriate motion sequences as a query, especially when a user wishes to retrieve a dance motion sequence, because these often include complex or difficult motions and it is necessary to

[2] http://mocap.cs.cmu.edu
[3] http://smile.uta.edu/hmd/
[4] https://neuronmocap.com

be able to perform such movements. To acquire the appropriate motion sequences as a query, a puppet interface [12] and sketch-based interfaces [4, 5, 13] have been proposed. However, submitting a motion sequence by using the puppet interface or sketching a person's movement is not easy for a general user.

Moreover, these approaches are premised on the clarity of user's information desire. They cannot accommodate unclear requests of a user, such as searching for a dance motion sequence by matching an expected image to a piece of music. Exploratory search [14] is effective for accommodating such unclear requests. The exploratory search system [15, 16] needs to be designed so that it is tailored to a purpose and targeted users. For example, Bernard et al. [15] presented an exploratory search system in close collaboration with the targeted users, who were researchers working on human motion synthesis and analysis, including a summative field study. However, an exploratory dance motion search system for constructing character dance animation matching input music has not yet been established. Our dance motion search system enables the user to search a wide variety of motion sequences without in-depth knowledge of dance.

2.2 Dance Motion Synthesis

The mainstream of generating dance motion from input music is uncovering the relationships between existing dances and their accompanying music. Musical contents such as tempo [17], acoustic beat features [1, 18], pitch and chord [19], and melodic contour [20], have been used to analyze these relationships, in addition to combinations of acoustic features [21] and structural similarity [22]. These approaches construct a dance sequence by concatenating segments of existing dances. A method to generate dance choreography automatically using machine learning is introduced [2] and it enables various dance motions that are not included in existing motion databases to be generated.

However, in our proposed system, these methods cannot automatically synthesize a sequence under the constraint condition that the motions chosen by the user are fixed. The dance synthesis method proposed by Xu et al. [23] has the potential to overcome the problem described

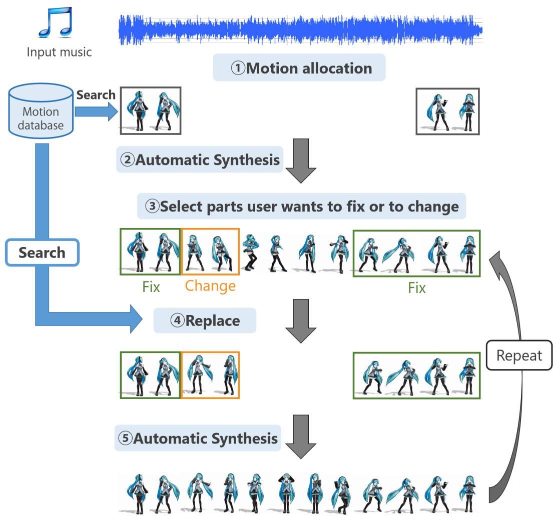

Figure 1. Overview of the system

above. The authors applied *Motion Graphs* [24] to motion synthesis for synchronization with streaming music. *Motion Graphs* is a method for creating realistic, controllable motions using the structural expression of relations among small bits of motion. The dance synthesis method introduced *Synchronization Capacity* to ensure that the duration of the generated motion matches with the duration of the input music. However, this constraint often causes the same movement to be generated repeatedly, because of a transitable loop of nodes permanently being passed. We define the transition cost function between motion segments, considering the connectivity of motion segments and how well a candidate motion segment matches a corresponding piece of music. Then, the filling dance motions are selected by minimizing the transition cost function using Dijkstra's algorithm. In this way, our method automatically synthesizes a dance sequence without a loop of nodes being made, under the constraint condition that the motions selected by the user are fixed.

3 Choreographic Authoring System

Fig. 1 provides an overview of the system. Firstly, the user of the system assigns a preferred motion to any part of the input music timeline (Fig. 1, ①). Dance motions that correspond to other parts of the music are then synthesized automatically, so that the movements are connected naturally (Fig. 1, ②). Secondly, the user selects the parts he or she wishes to fix or change in a generated sequence of dance (Fig. 1, ③). The part the user wishes to change is then replaced with other preferred motions from the database (Fig. 1, ④). Dance motions that correspond to other parts in the music are updated automatically (Fig. 1, ⑤). Finally, these steps are repeated in order to compose a new dance motion considering the user's preference.

There are two technical challenges to realizing the proposed system, as described in Section 1. We describe the solution in Sections 4 and 5. Motion segments in the database are obtained by separating existing dance motions[5] at every four beats. Although automatic dance motion segmentation methods based on choreographic primitives are proposed [25], we do not use them owing to insufficient accuracy of them.

4 Dance Search System Using Relevance Feedback

Our approach described in Sections 4 and 5 is constructed under the following two hypotheses built by Shiratori et al. [1]: 1) the rhythm of dance motions is synchronized to that of music. 2) the intensity of dance motions is synchronized to that of music.

In this section, we describe how our system enables users to easily search for preferred motions from the motion database. For most users, it is difficult to judge whether a dance motion is good or not without previewing it with a corresponding musical piece. In our system, the user can view candidate sequences on a screen and simply choose the preferred one. Our system enables the user to re-retrieve motion data using relevance feedback [8]. During the early phases of feedback, our system expands the variety of candidates, and gradually converges as the number of re-retrievals increases. We utilize the diversification framework proposed by Dou et al. [9] to select candidates with simultaneous consideration of a candidate's relevance to the corresponding piece of music and motions already selected. The $n+1$-th motion segment m_{n+1} is given by

$$\begin{aligned} m_{n+1} = &\operatorname*{argmax}_{m \in U \setminus S_n} \{\rho \cdot \mathrm{rel}(q,m) \\ &+ (1-\rho) \cdot \Phi(m, S_n, L_l)\}. \end{aligned} \tag{1}$$

S_n is the set of motion segments already recommended by the system and L_l is the set of motion segments the user likes. U is the universal set of motion segments in the database. ρ $(0 \leq \rho \leq 1)$ is the parameter that controls the tradeoff between $\mathrm{rel}(q,m)$ and $\Phi(m, S_n, L_l)$. $\mathrm{rel}(q,m)$ represents the relation between the candidate motion segment m and the corresponding piece of music q. We use the root mean square (RMS) mean of each musical bar as the music feature and the motion intensity (as defined by Eq. (2)) mean of each motion segment as the motion feature. We define the motion intensity W as the linear sum of the approximated instantaneous speed calculated from the position of the joints:

$$W(f) = \sum_i \alpha_i \cdot \|\dot{\mathbf{x}}_i\| \tag{2}$$

where α_i is the regularization factor for the i-th joint. $\|\dot{\mathbf{x}}_i\|$ is the Euclidean norm of the approximated instantaneous velocity of the i-th joint. These regularization parameters depend on which parts we recognize as being important for dance expression. $\Phi(m, S_n, L_l)$ controls the variety of candidates. We define $\Phi(m, S_n, L_l)$ as follows:

$$\begin{aligned} \Phi(m, S_n, L_l) = &\tau \cdot \min\{D(m, m_i) | m_i \in S_n\} \\ &+ (1-\tau) \cdot \max\{\mathrm{Sim}(m, m_j) | m_j \in L_l\} \end{aligned} \tag{3}$$

where $D(m, m_i)$ and $\mathrm{Sim}(m, m_i)$ represent the dissimilarity and the similarity between m and m_i, respectively. $\mathrm{Sim}(m, m_i)$ is calculated using the method proposed by Wang et al. [7]. $D(m, m_i)$ is given by $D(m, m_i) = 1 - \mathrm{Sim}(m, m_i)$. τ is the parameter that controls the diversity of candidates and is given by

$$\begin{cases} \tau_0 = 1 \\ \tau_k = \frac{19}{20}\tau_{k-1} \ (k = 1, 2, \cdots) \end{cases} \tag{4}$$

where k is the number of feedback instances. As the number of re-retrievals increases, τ decreases, thereby contracting the variety of candidates.

[5] http://www.perfume-global.com/

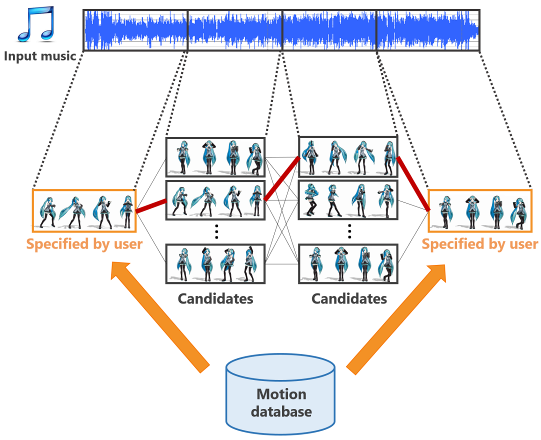

Figure 2. Automatic synthesis of a dance sequence using Dijkstra's algorithm. The red bold lines show the route minimizing the sum of the cost functions.

5 Automatic Synthesis of a Dance Sequence

Our system automatically synthesizes a dance sequence matched to input music, with the constraint condition that the motions selected by the user are fixed. Therefore, the user is able to focus on the parts that he or she has a strong preference for of the input music timeline.

First, the user assigns a preferred motion obtained through our dance search system described in Section 4 to the parts with strong preference. Subsequently the system assigns motions from the motion database to other parts of the music by considering the connectivity of motion segments and how well a candidate motion segment matches a given piece of music. The goal of this is to acquire the most appropriate motion segment sequence. The rhythm of the candidate dance motion segments is synchronized to that of the input music by resizing the motion segments. The motion segment sequence that minimizes the sum of the transition cost function C is obtained by using Dijkstra's algorithm (Fig. 2). The transition cost function from the motion segments A to B is given by

$$C(A,B) = \beta \cdot C_{\text{connect}}(A,B) + (1-\beta)\cdot C_{\text{music}}(q,B). \quad (5)$$

β ($0 \leq \beta \leq 1$) is the parameter that controls the tradeoff between $C_{\text{connect}}(A,B)$ and $C_{\text{music}}(q,B)$. $C_{\text{connect}}(A,B)$ analyzes the connectivity from motion segments A to B, while $C_{\text{music}}(q,B)$ analyzes how well the intensity of the motion segment B matches that of the corresponding piece of music q. These evaluation functions are described in detail in Sections 5.1 and 5.2.

5.1 *Evaluating the Climax Level of Music*

In the evaluation function $C_{\text{music}}(q,B)$, to analyze how well the intensity of the motion segment B matches that of the corresponding piece of music q, we use the climax level analysis of music proposed by Sato et al. [26]. The authors calculate the temporal climax transition of a piece of music $A_s(t)$ using music features such as RMS energy, ΔRMS energy, roughness, and spectral flux. They select music features from MIR toolbox Ver. 1.3 [27].

We use the temporal climax transition, normalized to values ranging from 0 to 1.

$$A'_s(t) = \frac{A_s(t)}{\max A_s(t)} \quad (6)$$

Then, we calculate the means of $A'_s(t)$ in each of the music bars, and $C_{\text{music}}(q,B)$ is given by

$$C_{\text{music}}(q,B) = \left| A'_s(t,q) - A'_s(t,q^B) \right|. \quad (7)$$

$A'_s(t,q)$ represents the mean of $A'_s(t)$ in the music bar q. $A'_s(t,q^B)$ represents the mean of $A'_s(t)$ in the original music bar q^B corresponding to the motion segment B.

5.2 *Connectivity Analysis of Motion Segments*

In the evaluation function C_{connect}, we utilize the connectivity analysis of motion segments [1]. We consider both the posture similarity S_{pose} and the movement similarity S_{move}. The posture similarity S_{pose} between the i^A-th frame of the motion segment A and the j^B-th frame of the motion segment B is defined as the angular similarity of the link direction vectors $\mathbf{v}$:

$$S_{pose}(i^A, j^B) = \sum_l \gamma_l \cdot (h(\mathbf{v}_l(i^A)) \cdot h(\mathbf{v}_l(j^B))) \quad (8)$$

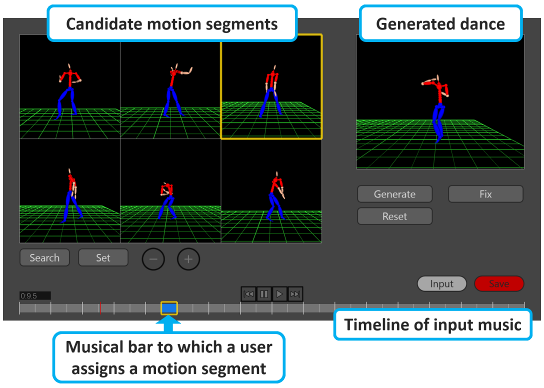

Figure 3. Example of the choreographic authoring system screen

where γ_l is the regularization factor for the l-th link. Through h, an input vector $\mathbf{a}$ is converted to the unit vector $\mathbf{a}/|\mathbf{a}|$. The movement similarity S_{move} is calculated as follows:

$$\begin{aligned} &S_{move}(i^A, j^B) \\ &= \sum_l \gamma_l \cdot g[h(\mathbf{v}_l(j^B) - \mathbf{v}_l(i^A)) \cdot h(\dot{\mathbf{v}}_l(i^A))] \\ &\quad \cdot g[h(\mathbf{v}_l(j^B) - \mathbf{v}_l(i^A)) \cdot h(\dot{\mathbf{v}}_l(j^B))] \end{aligned} \tag{9}$$

where $g[x] = x$ if $x > 0$, and $g[x] = 0$ otherwise. Here, $\dot{\mathbf{v}}$ is calculated from the candidate motion segment. Thus, connectivity from motion segments A to B is given by $C_{\text{connect}}(A, B) = 1/(S_{pose} + S_{move})^2$.

5.3 Transition Motion Generation

The resulting motion sequence is obtained by connecting the motion segment sequence using cubic interpolation for the posture of a character. For the position of the character, we pay attention to the position relative to the ground, to avoid effects such as sliding or floating in the air.

6 Interface

The screenshot of the implemented user interface of our system is shown in Fig. 3. The interface includes basic viewing functions, such as a window showing the dance sequence, functions to load the input music, save, play and stop/pause the generated video, and the input music timeline. A musical bar is used as the unit to which the user assigns a preferred motion segment.

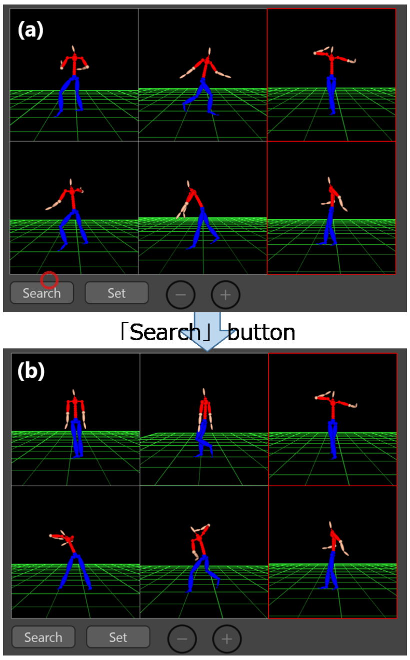

Figure 4. (a) Example of interactive sequence selection. Six different dance candidate sequences are previewed. The upper-right and lower-right candidates are selected by the user. (b) By clicking the search button, the dance sequence candidates are updated using relevance feedback.

The interface also provides the following functions for constructing a character dance animation reflecting the user's preferences.

Searching for a user preference dance motion segment: Fig. 4 shows how a user searches for a preferred motion segment from the motion database. Upon clicking a musical bar to which the user wishes to assign a motion segment, the system visually presents the user with candidate sequences selected on the basis of the sampling method described in Section 4. The number of candidate sequences can be changed from four to eight at an interval of two using the "+" and "−" buttons (the default number is six). Users can update candidate sequences reflect their preferences by clicking the search button after selecting one or more motion segments they like. When the users find a preferred motion segment, they can assign it to the musical

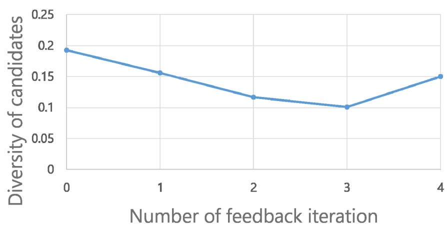

Figure 5. Diversity of the candidates

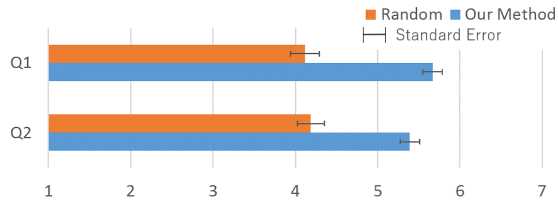

Figure 6. Results of (Q1) and (Q2) for each dance synthesis method.

bar by clicking the set button.

Generating a dance sequence matched to input music: When the generate button is clicked, the system automatically synthesizes a continuous choreography with the constraint condition that the motion segments are selected by the user. The user can view the generated dance sequence in the upper-right section of the interface.

7 Experimental Results

7.1 Diversity of Candidates

We defined the diversity of candidates as the mean of the dissimilarities between each pair of six sequence candidates ($\frac{1}{15}\sum_{i=1}^{5}\sum_{j=i+1}^{6} D(m_i, m_j)$). Fig. 5 portrays the relationship between the diversity of candidates and the number of feedback instances. As expected, the variety of candidates gradually converges as the number of feedback iterations is increased. However, the diversity of candidates begins to increase after the number of feedback iterations is 3. This is because the motion segments selected as sequence candidates previously are neglected, that is to say, $m \in U \backslash S_n$ in Eq. (1). If $m \in U \backslash S_n$ is replaced with $m \in U$, the user cannot update the dance sequence candidates displayed in the screen by clicking the search button after the combination of the minimum mean of the dissimilarities between each pair of sequence candidates is displayed.

7.2 Resulting Dance Animation

To check the effectiveness of our dance synthesis method in the system, we conducted a user study. Twenty-three computer science students participated in this study. The participants were shown six dance animations in total.

As input, we used three pieces of music (one piece of "Perfume_ globalsite_ sound" and two pieces of "2U Night Drive"). Each piece of music involved adding two types of generated dance. These dance animations were generated by using motion segments from the database, with the constraint condition that the motions specified by the user are fixed. Other than using our method, motion segments were randomly selected from the database. The synthesized motion sequence is obtained by connecting the motion segment sequence using cubic interpolation as with our method. To evaluate satisfaction and utility, after watching each of the video clips, the participants were asked the following questions,

(Q1) Did you feel satisfied with the movements of character body? (7-points Likert scale, 7: most satisfied; 1: least satisfied)

(Q2) Did you feel that the dance is matched to the music? (7-points Likert scale, 7: most satisfied; 1: least satisfied)

(Q3) When did you feel satisfied or dissatisfied when watching the video? (open ended)

Fig. 6 shows the results of the questionnaire. The results statistically proved that our method outperforms random selection in terms of the resulting movements quality of character body (p-value was 2.80×10^{-9} using the Wilcoxon signed-rank test) and how well a dance matches to input music (p-value was 3.17×10^{-7} using the Wilcoxon signed-rank test). In addition, the result showed that our dance synthesis method is effective in the system (the average score of Q1 and Q2 were 5.67 and 5.39 respectively).

On the other hand, a participant who has experience of dancing provided a negative com-

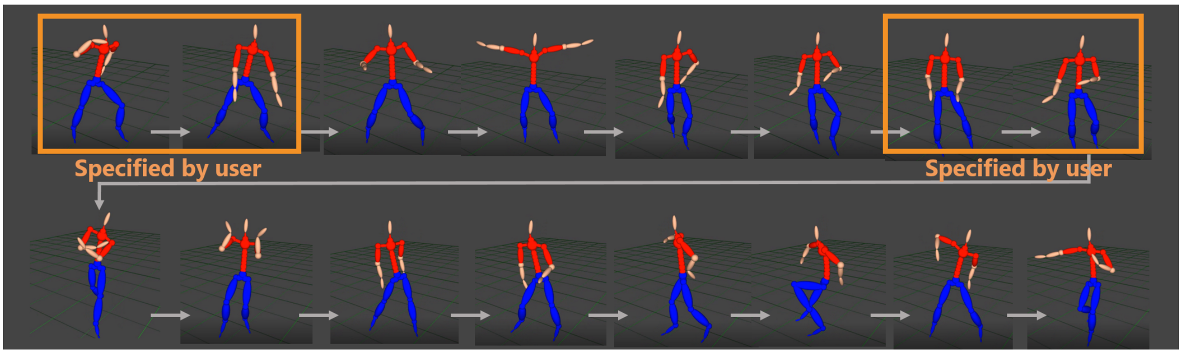

Figure 7. An example result of generated dance motion. The motions that a user selects are within orange frames. The system automatically synthesizes a continuous choreography with the constraint condition that the motions selected by the user are fixed. The duration of the resulting dance is 32 beats (approximately 14.77 sec) and the screenshots are taken at two beat (approximately 0.92 sec) intervals.

ment about the dance synthesized by our method as follows.

> “ I felt slightly dissatisfied with the quality of the dance owing to lack of choreographic coherence on the same music structure.”

To improve the quality of a synthesized dance, it is effective to consider how well a choreographic structure matches to a music structure. The dance animations generated by our system are shown in Fig. 7.

7.3 User Feedback about the System

In this section, we discuss the early reactions of our targeted users who are interested in seeing a dance without any knowledge of choreography and any experience of making 3D animation. The number of them was about 20 people or more. We asked them to use our user interface or to observe authoring choreography after sufficiently describing the purpose of our system and how to use our interface.

Most of them felt that the computing time of generating a dance sequence is short enough to operate without causing stress to the user. On the other hand, some others presented improvements to our interface. They felt that the timeline without visualizing the music structure and the state of being unable to browse a history of past searches are inconvenient. Visualizing the input music structure estimated automatically and implementation of the search history function are our future work. In addition, another user presented new functions to import an arbitrary character model data, stage and background, since he felt that the suitable choreography varies by character or scene.

Many of them agree with our motivation to develop this program. Moreover, they had an interest in the system and told us they want to use it. Some of them felt that it is magnificent to enable people who cannot dance to create a choreography, since only someone who has an enough skill to dance is permitted to create the choreography, previously.

8 Conclusion and Future Work

In this paper, we have proposed a novel system for choreographic authoring for character dance animation. To consider in implementing the system, we have provided the dance search system using relevance feedback and automatic synthesis of a dance sequence considering the connectivity of the dance motion segments and how well a candidate motion segment matches a corresponding piece of music. With this system, we can create new dance performances that consider the user’s preference in the virtual space.

However, since the variety of the choreography of the synthesized dance depends on the dance motion database, it is important to expand the database. In addition, to improve the quality of a resulting dance, it is effective to consider how well a choreographic structure matches to a music structure. Furthermore, it is important to improve our interface in ways such as the visualization of the music structure that is estimated automatically [28] and the implementation of the search history function as future works.

This paper should be regarded as being the new initiative addressing a challenging problem in the animation community. It can enhance the quality of dance creation for the people in need and facilitate creative activities.

Acknowledgements

Hatsune Miku is copyrighted by Cripton Future Media INC., and we used it in our research under PIAPRO license (`www.piapro.net`). 3D model of *Hatsune Miku* used in our research is created by Lat. This work was supported in part by ACCEL, JST.

References

[1] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, pages 449–458. Wiley Online Library, 2006.

[2] Satoru Fukayama and Masataka Goto. Automated choreography synthesis using a gaussian process leveraging consumer-generated dance motions. In *Proc. of Advances in Computer Entertainment Technology*, page 23. ACM, 2014.

[3] Tomohiko Mukai and Shigeru Kuriyama. Pose-timeline for propagating motion edits. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 113–122. ACM, 2009.

[4] Myung Geol Choi, Kyungyong Yang, Takeo Igarashi, Jun Mitani, and Jehee Lee. Retrieval and visualization of human motion data via stick figures. In *Computer Graphics Forum*, volume 31, pages 2057–2065. Wiley Online Library, 2012.

[5] Jun Xiao, Zhangpeng Tang, Yinfu Feng, and Zhidong Xiao. Sketch-based human motion retrieval via selected 2d geometric posture descriptor. *Signal Processing*, 113:1–8, 2015.

[6] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proc. of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 147–156. ACM, 2011.

[7] Pengjie Wang, Rynson WH Lau, Zhigeng Pan, Jiang Wang, and Haiyu Song. An eigen-based motion retrieval method for real-time animation. *Computers & Graphics*, 38:255–267, 2014.

[8] Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

[9] Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. Multi-dimensional search result diversification. In *Proc. of ACM international conference on Web search and data mining*, pages 475–484. ACM, 2011.

[10] Atsuo Yoshitaka and Tadao Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Trans. on Knowledge and Data Engineering*, 11(1):81–93, 1999.

[11] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. In *ACM Trans. on Graphics*, volume 24, pages 677–685. ACM, 2005.

[12] Naoki Numaguchi, Atsushi Nakazawa, Takaaki Shiratori, and Jessica K Hodgins. A puppet interface for retrieval of motion capture data. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 157–166. ACM, 2011.

[13] Min-Wen Chao, Chao-Hung Lin, Jackie Assa, and Tong-Yee Lee. Human motion retrieval from hand-drawn sketch. *IEEE Trans. on Visualization and Computer Graphics*, 18(5):729–740, 2012.

[14] Ryen W White and Resa A Roth. Exploratory search: beyond the query-response paradigm (synthesis lectures on information concepts, retrieval & services). *Morgan and Claypool Publishers*, 3, 2009.

[15] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Trans. on visualization and computer graphics*, 19(12):2257–2266, 2013.

[16] Songle Chen, Zhengxing Sun, Yan Zhang, and Qian Li. Relevance feedback for human motion retrieval using a boosting approach. *Multimedia Tools and Applications*, 75(2):787–817, 2016.

[17] Costas Panagiotakis, Andre Holzapfel, Damien Michel, and Antonis A Argyros. Beat synchronous dance animation based on visual analysis of human motion and audio analysis of music tempo. In *International Symposium on Visual Computing*, pages 118–127. Springer, 2013.

[18] Jae Woo Kim, Hesham Fouad, John L Sibert, and James K Hahn. Perceptually motivated automatic dance motion generation for music. *Computer Animation and Virtual Worlds*, 20(2-3):375–384, 2009.

[19] Ferda Ofli, Engin Erzin, Yücel Yemez, and A Murat Tekalp. Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis. *IEEE Trans. on Multimedia*, 14(3):747–759, 2012.

[20] Sageev Oore and Yasushi Akiyama. Learning to synthesize arm motion to music by example. 2006.

[21] Rukun Fan, Songhua Xu, and Weidong Geng. Example-based automatic music-driven conventional dance motion synthesis. *IEEE Trans. on visualization and computer graphics*, 18(3):501–515, 2012.

[22] Minho Lee, Kyogu Lee, and Jaeheung Park. Music similarity-based approach to generating dance motion sequence. *Multimedia tools and applications*, 62(3):895–912, 2013.

[23] Jianfeng Xu, Koichi Takagi, and Shigeyuki Sakazawa. Motion synthesis for synchronizing with streaming music by segment-based search on metadata motion graphs. In *Multimedia and Expo, IEEE International Conference on*, pages 1–6. IEEE, 2011.

[24] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM Trans. on graphics*, volume 21, pages 473–482. ACM, 2002.

[25] Narumi Okada, Naoya Iwamoto, Tsukasa Fukusato, and Shigeo Morishima. Dance motion segmentation method based on choreographic primitives. In *GRAPP*, pages 332–339, 2015.

[26] Haruki Sato, Tatsunori Hirai, Tomoyasu Nakano, Masataka Goto, and Shigeo Morishima. A soundtrack generation system to synchronize the climax of a video clip with music. In *Multimedia and Expo, IEEE International Conference on*, pages 1–6. IEEE, 2016.

[27] Olivier Lartillot and Petri Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International Conference on Digital Audio Effects*, pages 237–244, 2007.

[28] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(5):1783–1794, 2006.