Multimed Tools Appl (2013) 62:895–912
DOI 10.1007/s11042-012-1288-5

Music similarity-based approach to generating dance motion sequence

Minho Lee · Kyogu Lee · Jaeheung Park

Published online: 28 November 2012

Abstract In this paper, we propose a novel approach to generating a sequence of dance motions using music similarity as a criterion to find the appropriate motions given a new musical input. Based on the observation that dance motions used in similar musical pieces can be a good reference in choreographing a new dance, we first construct a music-motion database that comprises a number of segment-wise music-motion pairs. When a new musical input is given, it is divided into short segments and for each segment our system suggests the dance motion candidates by finding from the database the music cluster that is most similar to the input. After a user selects the best motion segment, we perform music-dance synchronization by means of cross-correlation between the two music segments using the novelty functions as an input. We evaluate our system's performance using a user study, and the results show that the dance motion sequence generated by our system achieves significantly higher ratings than the one generated randomly.

Keywords Choreography · Dance motion generation · Music similarity · Music-motion database · Motion capture · Motion synthesis

Kyogu Lee and Jaeheung Park are the corresponding authors.

M. Lee · K. Lee (✉) · J. Park (✉)
Graduate School of Convergence Science & Technology,
Advanced Institutes of Convergence Technology, Seoul National University,
Seoul, Republic of Korea
e-mail: kglee@snu.ac.kr
e-mail: park73@snu.ac.kr

M. Lee
e-mail: setiem@snu.ac.kr

1 Introduction

Choreography is the art of arranging dance, with the objective of creating an aligned sequence of motions that reflects the accompanying music. Dance motion, which represents the accompanying music appropriately, has the ability to communicate more expressively with the audience than music or dance is presented by itself. Therefore, the audience may better appreciate music if the music is presented with a good dance performance. It indicates that music and dance are in a mutually supportive relationship, and we can see the trend that dance music is playing an important role in the music industry.

Importance of choreography is not limited to popular dance music. In various fields of sports or art such as rhythmic gymnastics, figure skating and ballet, music plays a critical role and creating a good dance for given music is a quintessential element for the players or actors/actresses to achieve a good performance. As choreography takes an important role in various fields as described above, there have been a great deal of efforts to compose better dance performances in various ways.

Choreographing a dance for a new musical work is a difficult task that requires a great amount of time and experience. Many choreographers often start by recollecting the dance movements they or others have used before, and alter them to create a new dance routine that better fits with a given musical piece. In this process it is not only natural but also critical to use the dance movements that were used in other musical works *similar* to what a choreographer is currently working with. Therefore, if we could build such a database that contains a number of dance movements used in real music, and design an algorithm that suggests a sequence of movements when a new musical input is given, they could be used as a reference that might help reduce the amount of time and labor for choreographers to create a new dance routine. Also, this dance sequence generation method is believed to have applications such as characters' dance in animations and games, or humanoid robot performances.

In this paper, we propose a method of generating a sequence of dance movements that we hope can be used as a reference in creating a novel dance for a given musical piece. We aim to realize this by taking the following steps. First, using a motion capture device we record the dance movements, which were choreographed by professionals, for various kinds of music. Second, we divide each musical work into short but musically meaningful segments, and repeat the same process for the dance motions using the same time boundaries used to divide music. This will result in a great number of music-motion segments in pairs.

The next step is clustering; i.e., grouping *musically* similar segments into one *cluster*. After this process all the segments will be grouped into a certain number of clusters, where the segments within a cluster will be musically similar to each other while they will be musically distinct from those of the other clusters. These clustered music-motion segments in pairs constitute the database of the proposed system.

Finally, when a new, unseen musical input is given, we divide it into a number of short segments and each segment is compared with the clusters in the database to select the one that is musically closest. The selected cluster will have several motion candidates and the appropriate one is chosen to accompany a given music segment. However, this does not guarantee that the selected motion is in sync with the music segment. Therefore, we undergo a music-motion alignment process by means of cross-correlation between the two music segments—i.e., the input music segment and

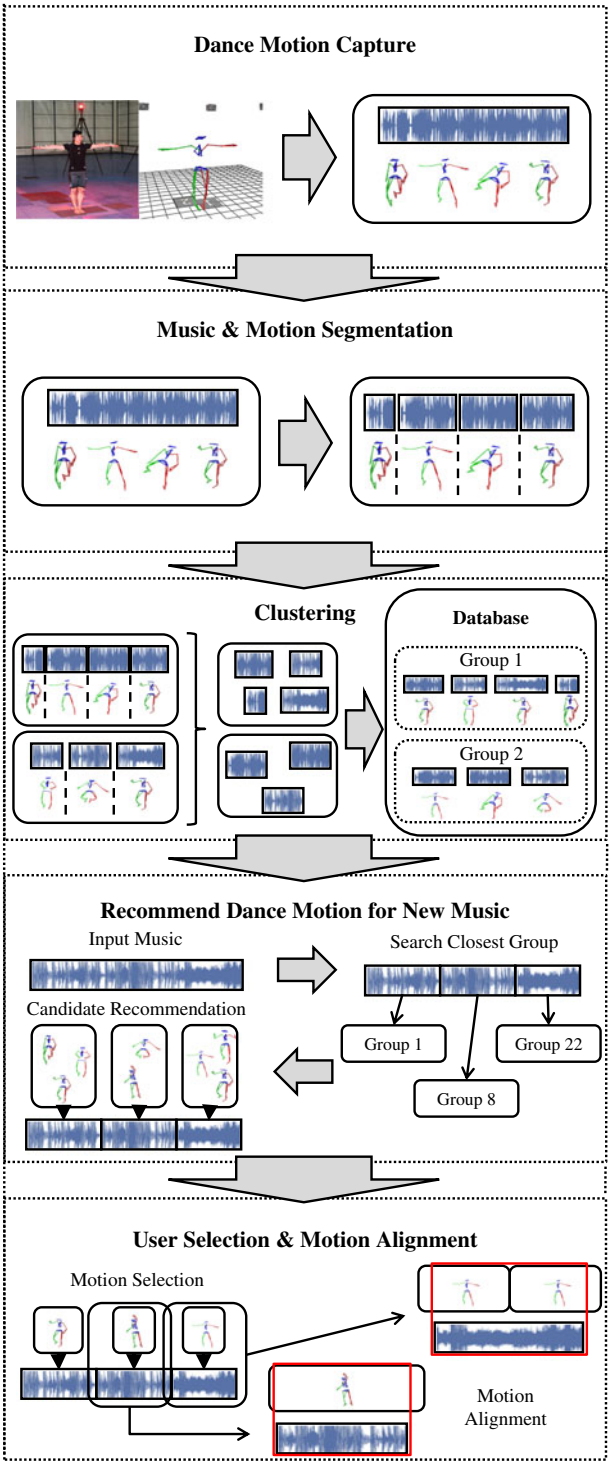

Fig. 1 System overview

the one that is associated with the selected motion segment. This process is repeated for every segment to generate a sequence of dance motions for the whole piece. The overall process is illustrated in Fig. 1.

There are several advantages to this approach. First of all, it is based on how human choreographers create a dance routine given a musical work; i.e., when new music is given, they retrieve dance motions from their memories or from the written records that were accumulated over the years of experiences. In this process, we believe that music similarity plays a critical role in retrieving appropriate motions for current music. A clear example would be reusing the exact same dance routine for the chorus parts—most repetitive sections in music—in most popular dance music.

Secondly, we do not need to go through a complex process of analyzing motion data to extract motion features, which are used to synchronize with musical features—this is the case with most dance generation systems.

Another advantage is that computing music similarity from audio has been extensively studied for the past years, and there are several algorithms to efficiently calculate music similarity from audio.

This paper continues with a review of related work in Section 2. In Section 3, we describe the proposed system in more detail. In Section 4, we describe our evaluation method and present experimental results with discussions in Section 5. We draw conclusions in Section 6, followed by directions for future work.

2 Related work

This section presents related research involving dance motion and music. These studies can be grouped into three categories: (1) matching music and dance motion; (2) synthesizing dance motions given music; and (3) synchronizing dance motion to music. These techniques have been widely studied in the domain of computer graphics or robotics. The research focus in the first category is on how to compare music segment and dance motion to find the best matching sequence of motion using various types of comparison methods and features. The work in the second category is about generating new motions given music, which basically are not from any motion database. The third group attempts to synchronize the key frames of motion to music and then to compose the complete motion trajectories that connect the key frames.

Among these studies those in the first category are most relevant to ours in that they try to find the most suitable motions by analyzing given music. Shiratori et al. proposed a dance generation method that matches motion intensity features with music rhythm features [7, 18]. They extracted the features such as rhythm, speed and mood from music, and the intensity from motion, and utilized these information to figure out suitable dance motion for given music. They also applied the generated motions to a humanoid robot by extracting key-pose from the motion sequence. Kang and Kim considered hand speed and distance as motion features, and used onset times and volume data from music for matching motions with music [8]. Alankus et al. used beat information from music to synthesize dance motions [1]. They analyzed music to extract rhythmic features and used this information to tune the existing motion, and generated new dance motion that was not described in the library. Recently, Ofli et al. proposed a music-driven choreography synthesis, where

they organized a music-motion paired database to train the HMMs & 2-gram model for dance sequence generation [15].

Kim et al. used more various features from music and motion to find the best motion match for given music [10]. They extracted 30 musical and 37 motion features, respectively, and calculated the correlation between music and motion by constructing the music-motion similarity matrix. Using the dance motions generated in this method, they performed statistical analysis on the user opinion results to verify their approach.

Researchers in the second category focus more on motion generation. Strictly speaking, the studies in the first category are also classified as motion generation. However, the studies in the second category *synthesize* motions from scratch and thus do not require any motion capture database. Loi and Li generated the upper body movements using musical features such as pitch, rhythm, beat, and intensity [12]. From these features extracted from music, they determined the shape qualities, weight effort, and time effort of the motion sequence. Sauer and Yang generated Celtic dance by mapping the beat information from music to motion's dynamics [17]. For example, as musical beat gets faster, the motion's dynamic level also goes up to the jumping motion. These works were also used for character animation. They also share in common that they both used musical beats as a core feature to correlate with motions.

The last category includes the studies that synchronize a choreographed dance motion sequence to music. Nakahara et al. manually designed the dance motion key-pose database, and controlled motor speed by predicting the tempo of music [13]. Grunberg et al. took the similar approach; they created 30 dance moves for HUBO, and by predicting the tempo of the musical audio, they synchronized the robot's movement with music [6]. Nakaoka et al. developed a software for creating motion trajectories between input key-pose motions. In this case, they did not consider to analyze music but generated motion when a choreographer designed key-point motions for the robot [14].

Our proposed approach is similar to the above-mentioned works—those in the first category in particular—in that the results of a music analysis are used to retrieve the most appropriate motions from a motion database. However, there is a significant difference in our approach: while most of other work try to find the best matching music-motion pair by maximizing the correlation between music and motion using some music/motion features, we do not analyze motion data nor extract any motion features at all. Instead, we only compute the music-to-music similarity to find the most appropriate motions based on the assumption that the motions used in some music will also match well with similar music. In the following section, we describe in detail our system to accomplish our goal.

3 System

Our system is designed to consist of two main steps of process, including (1) database organization and (2) dance motion generation. To construct a music-motion database, the following processes are required: motion capturing, music segmentation, audio feature extraction, and clustering. To generate a sequence of dance motions for an arbitrary musical input, we also need music segmentation, audio feature extraction

and nearest-neighbor search based on music similarity. Finally, after a user selects from the candidates the best motion that accompanies the input music segment, music-dance synchronization is also employed. We explain these steps in more detail in the following sections.

3.1 Database

The first step is organizing a database of music and dance motion data. Our system is designed to learn from data music-dance relationship to generate a new dance for an unseen musical work. Therefore, our motion data need to be obtained with the accompanying music playing simultaneously. Dance motion is recorded by a motion capture system to allow the possibility of analyzing motion and applying to character animation or humanoid robots.

Constructing the music-motion database is the most important component of our research. Details will be expounded in the discussion section (Section 5), but the diversity of the database would also affect the diversity of the motion generated by our system. As our system does not modify motion trajectories in the database, the generated dance motions will directly show the very same motions in our database.

Our raw database constructed so far contains musical works and corresponding dances. In order to make them usable we need further processing, which includes segmentation and clustering. Segmentation is a process of dividing music-dance into short but musically meaningful pieces. Segmenting music-dance data into much shorter snippets and having thousands of them in the database will yield a far more novel dance because we can now have an access to zillions of different music-dance combinations.

Clustering is important for two reasons. First, comparing the input music segment with a huge number of segments in the database is computationally very expensive. And this needs to be done for every segment in the input. Clustering the segments in the database into a manageable number of groups will be more efficient. The second reason for clustering is that it provides a user with more choices by suggesting many motion candidates for a single musical input.

3.1.1 Music database

Our proposed system stands on the basis of music analysis and music similarity, in particular. By the comparison of extracted musical features from the input and from the database, the system determines the dance motion candidates for the current input. Therefore, while we can have raw dance motions in the database, we must perform music analysis to extract compact and yet musically meaningful features because the raw audio samples are noisy and redundant and thus it is nearly impossible to use them to find similar music.

To compose a music database, acoustic features are used to represent music in a compact and yet musically meaningful manner, and music sources are segmented using these acoustic features. The details are explained in the following sections.

Feature extraction As described before, the raw audio samples are almost meaningless in computing music similarity. Therefore we need to extract more compact and representative acoustic features from the raw audio sources. Because music is multi-dimensional in nature, we carefully select the acoustic features that can represent

various attributes in music; namely, tonal (or melodic and harmonic), spectral, temporal, and timbral characteristics. These features include chromagram, spectral flux, spectral centroid/spread, MFCCs (Mel-Frequency Cepstral Coefficients), spectral rolloff, and zero-crossing rate (ZCR). They represent the timbral (MFCCs, spectral flux, spectral centroid/spread, and spectral rolloff), tonal (chromagram) and temporal (ZCR) characteristics in audio/music, and have been widely used in different kinds of musical applications [2, 5]. More information about these features is given in Table 1.

These features are extracted from all the music segments in the database. We use the frame size of 62.5 ms with no overlap. For each analysis frame, we concatenate the seven audio features to form a multivariate feature vector of dimension 30, and then average over the frames within each segment. Before concatenation, we normalized the features so that all the features have zero mean and unit variance. In this manner, we could compose 30 numerical values of audio features for each segment. We now have each music segment represented by a single 30-D feature vector. This information will later be used for clustering and for computing the music similarity, as well as in music segmentation described in the following section.

Music segmentation The second step for preparing a music database is to cut music sources into short segments. Segment boundaries are determined according to the points where there are significant changes in music. We hereby mean "significant changes" by the ones that make one segment distinct from others from the perspective of musical attributes, including tonal (melodic and harmonic), rhythmic, and timbral characteristics. And these characteristics are represented in the acoustic feature we use, as mentioned in the previous paragraph. Therefore, using these

Table 1 Description of acoustic features

Name	Type	Description	Dimension
Chromagram	Tonal	Harmonic pitch class profile. Shows the distribution of energy along the 12 pitch classes.	12
Spectral flux	Spectral/Temporal	Distance between the spectrum of each frames. Temporal position is indicated by the peaks in the distance curve.	1
Spectral centroid	Spectral	The first central moment (mean) of the spectrum.	1
Spectral spread	Spectral	The second central moment (variance) of the spectrum.	1
MFCC	Spectral/Timbral	Spectral shape of the sound. Frequency bands are positions on the Mel scale, which approximates the human auditory system.	13
Spectral rolloff	Spectral	The frequency such that a certain fraction of the total energy is contained below. The amount ratio is set to .85.	1
Zero-crossing rate	Temporal	Indicates the number of counts of signal crossing the X-axis (In other words, changes of signs.)	1

features, we can detect significant changes in music by means of audio novelty score or novelty function, which is calculated using the following steps proposed by Foote [3, 4].

First, we extract acoustic features from every analysis frame, as explained in the previous section, and compute the frame-to-frame distances for the whole music. This gives us an $N \times N$ *self-similarity* matrix, or S, where N is the total number of frames for a given musical piece. The measure of similarity between frame i and j is computed by a cosine distance between the feature vectors v_i and v_j; i.e.,

$$S(i, j) \equiv \frac{v_i \cdot v_j}{||v_i||||v_j||} \tag{1}$$

Figure 2 illustrates an example of a self-similarity matrix, where the similar frames are displayed in light colors while the dissimilar ones are represented in dark colors.

We then use a Gaussian kernel, with the kernel size of four seconds, which emphasizes the similar frames while de-emphasizing the dissimilar ones, and slide it along the diagonal of the self-similarity matrix to obtain the novelty function NF as follows:

$$NF(i) = \sum_{m=-L/2}^{L/2} \sum_{n=-L/2}^{L/2} C(m, n) S(i + m, i + n) \tag{2}$$

where C is the kernel matrix, L is the kernel size of 64 frames, and S is the self-similarity matrix. This yields a one-dimensional *novelty function* NF over time, where noticeable peaks indicate significant changes in music. To account for dynamics in music, we applied an adaptive thresholding technique, and the peaks above the thresholding curve T_{NF} are used as segment boundaries.

$$T_{NF}(i) = \delta + \lambda \text{ median}\left[NF(i_m)\right], \quad i_m \in \left[i - \frac{H}{2}, i + \frac{H}{2}\right] \tag{3}$$

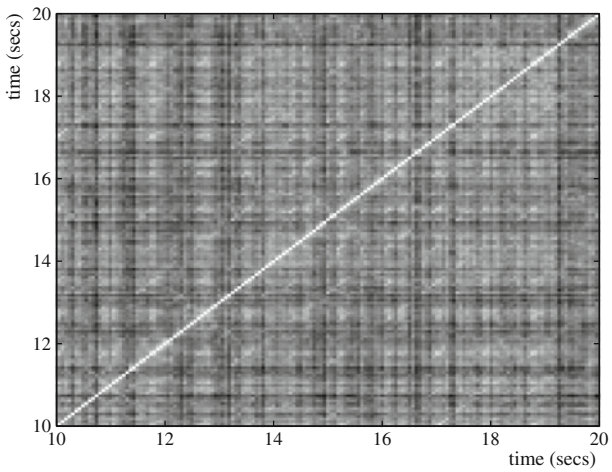

Fig. 2 Example of self-similarity matrix for a musical excerpt

where δ is a constant value for an offset, λ is the weighting factor, and H is the window size in frames. Figure 3 shows the novelty function calculated from the self-similarity matrix shown in Fig. 2. Interested readers may refer to Foote [3, 4] for more details on music segmentation using self-similarity and a novelty function.

We applied this segmentation algorithm not only for constructing the database but also for segmenting the input music signal. In this manner, time information of each music segment is marked, and imported to the database with its music source file (mono, 8 bit, 11025 Hz sampling rate, and WAVE format). During the study, we found that the length of the most segments tends to be 3–5 s. For example, if music is three minutes long, it can be expected to be divided into 36–60 segments. We used 22 songs of Korean popular dance music, where 11 of them are performed by male artist and the other 11 by female artist. Using the above-mentioned segmentation process, these result in 1,370 music-motion segments in total.

3.1.2 Motion database

The motion database consists of dance motion segments, which are coupled with music segments. That is, the boundaries of the music segments coincide with those of the motion segments. Experiments show that the motion pattern also shows noticeable changes along the boundaries determined from music. This is not surprising because the dance motions are usually choreographed along with the musical characteristics. Therefore, we can argue that if the music shows changes in pattern, dance motion is also likely to change in general and vice versa.

The dance motions are recorded using the motion capture system at AICT (Advanced Institutes of Convergence Technology) in Korea. Its studio has 12 Vicon T-160 cameras, and Vicon nexus software. The 12 infrared light cameras around a dancer record dance motions at a frame rate of 100 Hz. 35 markers were attached to a dancer. The dancer performed while a music plays so that the motion and music are synchronized. The captured dance motion data is segmented according to the segmented music data and then linked with the corresponding music segment. The scene of recording dance motion at the motion capture studio is shown in Fig. 4.

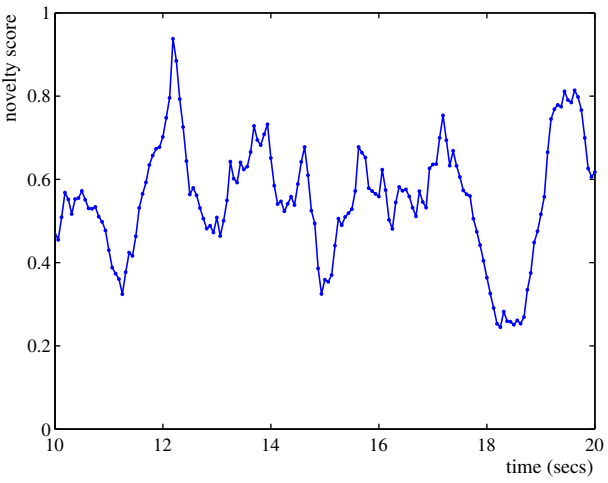

Fig. 3 Novelty function computed from Fig. 2

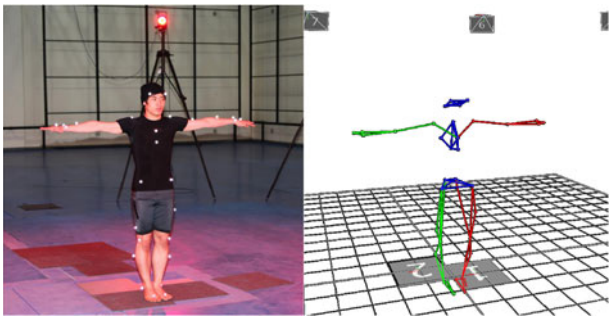

Fig. 4 Recording dance motion by motion capture

The 3D data from the motion capture system enables a user or a choreographer to view the motion from different view points. Also, this captured data could be used for analyzing motion features, which could bring better performance to our system. Currently, only music similarity is considered to generate dance performances, thus the captured motion is used only for displaying.

3.1.3 Clustering

The segmented music & motion data pairs are classified into clusters based on the music features only. The 1,370 music segments, as described in Section 3.1.1, are classified into 137 clusters using the K-means clustering algorithm [9, 11]. There is no rule of thumb in determining the number of clusters for grouping music segments. Therefore, in our experiment, we set the number of motion candidates for each music cluster 10 because this number would give us sufficient degree of freedom from which we could choose the best dance motion. The dance motions in each cluster will be motion candidates for a new music segment that is similar to the music segments in the cluster.

To verify our hypothesis that there is a correlation between music and motion similarities, we computed and compared the inter- and intra-cluster distances for music and motion segments. To calculate the distances between the motion segments, we applied a motion feature extraction method proposed by Kim et al. [10]. The effort components of Laban Movement Analysis (LMA) are defined as follows:

$$Motion\_Velocity = \sum_{j=1}^{N}\sum_{i=1}^{M} ||x_i(j+1) - x_i(j)|| \tag{4}$$

$$Motion\_Acceleration = \sum_{j=1}^{N}\sum_{i=1}^{M} ||v_i(j+1) - v_i(j)|| \tag{5}$$

$$Directional\_Change\_of\_Motion = \sum_{j=1}^{N}\sum_{i=1}^{M} \cos^{-1}\left(\frac{x_i(j+1)\cdot x_i(j)}{||x_i(j+1)||||x_i(j)||}\right) \tag{6}$$

Here, $x_i(j)$ represents the position vector of marker i at frame j, M is the number of markers and N is the number of frames, and $v_i(j)$ is the velocity of marker i in frame j. We performed a statistical analysis to prove our hypothesis, which is shown in Table 2.

Table 2 shows the inter-cluster distance (off-diagonal elements) as well as the intra-cluster distance of the segments in each cluster (diagonal elements) for music and motion, respectively. The inter-cluster distance is computed by the Euclidean distance between the cluster centers, and the intra-cluster distance by the standard deviation of all the segments in a cluster. The song we used for this analysis has 62 segments in total, which are grouped into 18 clusters, giving 3–4 segments per cluster in average. The clusters with just one segment are excluded, resulting in just 7 clusters.

As shown in Table 2(a), it is obvious that the inter-cluster distance is far greater than the intra-cluster distance of the music segments that belong to each cluster. This in turn means that the clustering algorithm successfully groups musically similar segments into a single cluster. The mean of the inter-cluster distances and the intra-cluster distances is 5.07 and 0.43, respectively.

More importantly, however, it can be seen in Table 2(b) that the inter-cluster distance of the motion segments is also greater, although not always and not as drastic as in the case of music, than the intra-cluster distance. This clearly supports our hypothesis that the motion segments clustered based on music similarities are similar to each other and are different from those in other clusters. The mean of the inter-cluster distances and the intra-cluster distances is 0.93 and 0.36, respectively.

3.2 Dance motion generation

The procedure of the dance motion sequence generation for a new music is the following. The block diagram of this procedure is shown in Fig. 5.

STEP 1 A new piece of music is segmented automatically according to the significant changes denoted by an audio novelty function.

STEP 2 For each segmented music, the system will choose the cluster of music segments that is most similar to the input music segment.

Table 2 Inter-cluster distance & standard deviation: (a) Music features (b) Motion features

(a)							
1	0.661						
2	5.898	0.415					
3	7.572	5.249	0.278				
4	5.979	3.987	5.484	0.446			
5	5.842	5.301	4.281	4.792	0.271		
6	6.147	4.413	3.115	4.333	2.764	0.396	
7	7.329	3.357	4.590	6.034	5.814	4.235	0.545
Cluster no.	1	2	3	4	5	6	7
(b)							
1	0.275						
2	1.044	0.379					
3	0.631	0.601	0.224				
4	1.311	0.269	0.821	0.354			
5	0.985	0.071	0.576	0.336	0.454		
6	1.311	0.432	1.004	0.424	0.433	0.633	
7	2.341	1.367	1.967	1.164	1.403	1.031	0.228
Cluster no.	1	2	3	4	5	6	7

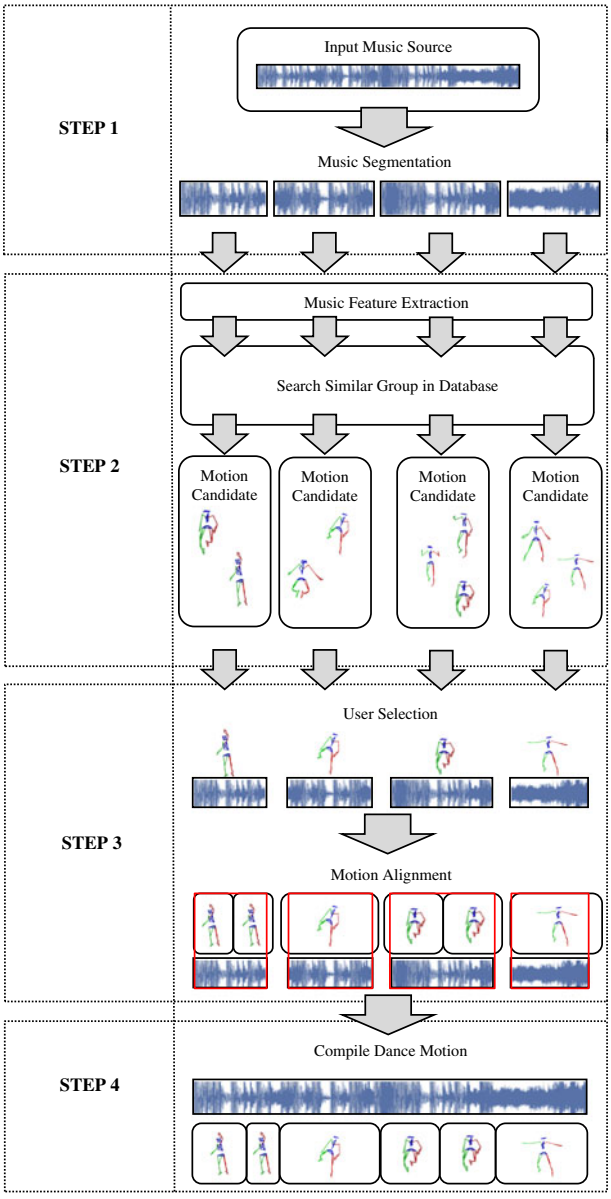

Fig. 5 Process of dance motion generation

STEP 3 The best motion candidate is selected by a user and is synchronized with the input music segment using cross-correlation between the input music segment and the one associated with the chosen motion segment.

STEP 4 The complete sequence of motions will be compiled and played for the user.

In STEP 2, we extract the acoustic features from an input music segment and compute the distance to the clusters. The distance to the center of each cluster is used for selecting the most similar music cluster.

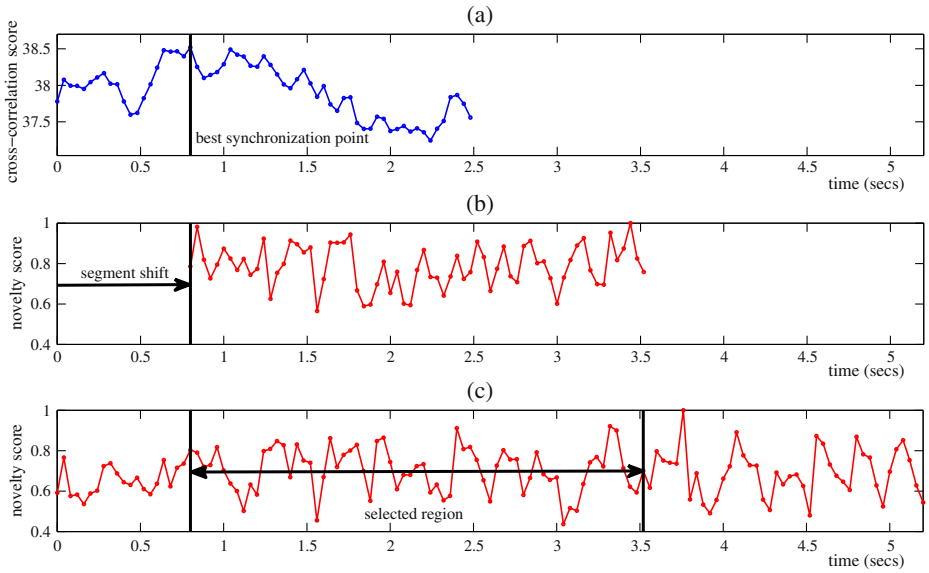

Fig. 6 Alignment procedure. **a** Cross-correlation between two novelty functions, **b** Shifted novelty function of input segment by the time lag that yields the maximum correlation score, **c** The novelty function of candidate segment & selected region after synchronization

In STEP 3, a user selects from the motion candidates the one that fits best with the input music segment. This procedure helps to reflect users' preferences in dance performance. After the selection, we perform a fine alignment between the input music segment and the selected motion segment by finding the time-lag that gives the maximum cross-correlation between the two segments. As an input to compute cross-correlation, we use the novelty function (NF) we computed in Section 3 because it indicates where an important musical event occurs and therefore is closely related with rhythm.

Figure 6 illustrates the alignment procedure when an input music segment is shorter than the candidate.[1] After synchronization between the two music segments with different lengths, the corresponding part to the input segment is cut from the candidate music segment to be used for compiling in STEP 4. In the case that the candidate is shorter than an input music segment, the candidate music segment is lengthened to be longer than the input segment by repeating the candidate music segment.

When compiling selected dance motions into a sequence in STEP 4, transitions between the consecutive motions are smoothed by following the spline-fill method, which is widely used in most motion editing software packages [16, 19]. This method refers the past and future trajectory data to fill approximate spline data for the missing marker values. The transition period is set to be 4 % of each segment length.

[1]In order to show the effect of music-motion synchronization, two video clips before and after synchronization are exemplified for comparison. The video clips can be found at http://plaza4.snu.ac.kr/~park73/download/DMGS.html.

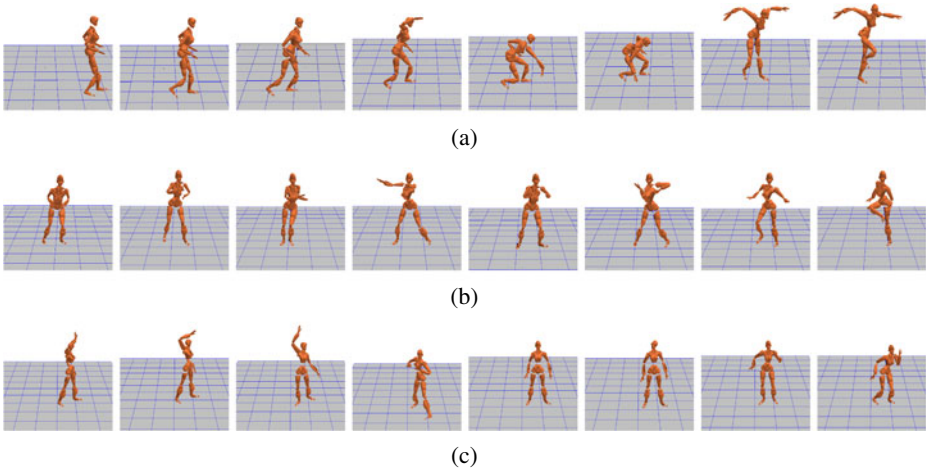

Fig. 7 Snapshots of dance motions. **a** Dance I: original. **b** Dance II: proposed. **c** Dance III: random

4 Evaluation

The performance of the proposed approach is evaluated by a subjective user opinion study because there is no criteria to verify if the created dance motion is suitable or not for the music. In this study, the controlled experiments are conducted by asking the subjects to score three dances: (1) a dance that is originally performed for input music (Dance I hereafter), (2) a dance that is composed by the proposed method (Dance II hereafter), and (3) a randomly generated dance motion selected from our music-motion database (Dance III hereafter). The three dances are for the same music clip and presented in a random order. Figure 7 shows the snapshots of the dance motion sequence used in evaluation.[2] The questionnaire uses a five-level Likert scale for scoring each dance motion sequence. In addition to scoring, the subjects were asked for any comments or opinions on where and why the dance motion does not look suitable to them for each clip.

Thirty people voluntarily participated in the experiment: 30 male and 6 female participants. Their ages range from 23 to 41 years old and the average age is 28 years old. In this test, four music clips, selected from a disjoint music database, are used. Three dance motions (Dances I, II, III) for each music clip are presented to the subjects. These three dance motions are consecutively played but the order was different for different songs. All the generated motions were presented with the audio, for scoring how it looks to be matched with the music clip. Participants were asked to score after watching each of the clip, and the length of each video clip was about 1 min.

5 Results and discussion

Statistical analysis is performed on the collected user feedback data. The means and p-values for pairwise comparison using t-test of the scores are shown in Table 3. The

[2]The video clips can be found at http://plaza4.snu.ac.kr/~park73/download/DMGS.html.

Table 3 Statistical analysis of user opinion study results

	Method		
	Original (Dance I)	Proposed (Dance II)	Random (Dance III)
Music I			
Mean	3.53	3.03	1.97
p-value	Original:	0.086	<0.001*
		Proposed:	<0.001*
Music II			
Mean	4.19	2.94	2.42
p-value	Original:	<0.001*	<0.001*
		Proposed:	0.022*
Music III			
Mean	4.19	3.28	1.89
p-value	Original:	<0.001*	<0.001*
		Proposed:	<0.001*
Music IV			
Mean	4.64	2.47	1.83
p-value	Original:	<0.001*	<0.001*
		Proposed:	0.001*

Significance level: $*p < 0.05$

results show that the order of preference is Dance I, Dance II, and Dance III with statistically meaningful differences for the entire music clips. It verifies that the dance motion generated by the proposed approach appeals to the people significantly more than the randomly generated ones.

For Music I, the *p*-value between Dances I and II is 0.086. It means that the difference in the means of two dance clips is not statistically meaningful. Therefore, the dance motion by the proposed motion is quite equivalently as good as the original dance although the score is lower. For the other three Music clips, all the *p*-values from the pairwise t-test are lower than 0.05.

In this user study, we also asked people about the familiarity of the original dance. The users were very familiar with the dance for Music IV because it was a very popular music. It is believed that Dance I received relatively high score than the original dances for the other music clips due to this reason. This, we believe, negatively influenced on the score of Dance II in that Music clip.

In the questionnaire, many participants pointed out awkwardness of a generated dance when the motions are selected from the opposite sex's motion database; i.e., when music is by male singers but the dance is from the female's data or vice versa. For Dance II in Music II, the numbers of male and female dance motion segments were 14 and 6, respectively, while Music II was sung by a female artist. This may happen because our current approach does not consider the artist's gender. The user study result somewhat reflects this fact that the *p*-value between Dances II and III for Music II is relatively high −0.022 although it is still statistically meaningful. In other music cases, 60–82 % motion segments of the same gender are selected so, this effect seems to be minor. This factor could be included in our database so that the gender can be accounted for during the selection process of dance motions.

It is believed that the proposed approach produces better dance motions than random selections for all simulated cases. Although it is tested on only four musical

pieces, the user study shows that the proposed algorithm surely generates reliable results.

6 Conclusion

A novel approach is proposed for generating dance performance based on music similarity. The proposed algorithm in this paper is based on the idea that a dance motion used in certain music, will also go well with similar music. Therefore, the key to our algorithm is to learn from or refer to prior dance motions, and then to adopt these motions to similar musical inputs.

The database comprises dance motion and music segments in pairs, which are clustered into groups by means of music similarity. When a new musical input is given to the system, it is first segmented and for each segment dance motion candidates are presented to the user as a result of finding the closest music cluster in the database. The complete sequence of dance motion is determined by the user's selection among the candidates. After selection of motion segments, the system synchronizes each motion segment with its target music segment by cross-correlations, to provide a finer alignment between the two.

The proposed approach is verified by the statistical analysis of user opinion study. It shows that the dance motion sequence generated by our system achieves significantly higher ratings, for all four cases, than the one generated randomly, although it received lower ratings than the original dance in general. We believe that the proposed system will provide the users or choreographers with a good reference in composing a dance for new music.

We are currently working on developing the system further in various aspects. The suggested motion candidates can be graded using motion features. Generating better motion connectivity between motion sequences is one factor to be considered. We also plan to build the motion-music databases separately for different genders because the results of our user study suggest that it has an effect to some degree. Finally, we believe that there are many promising applications such as generating motion for characters in animations and humanoid robots.

Acknowledgements This study was supported by the grant (No. 2011-P3-15) of Advanced Institutes of Convergence Technology (AICT). Also, we greatly acknowledge Dr. Junghoon Kwon for his help on the use of the motion capture system.

References

1. Alankus G, Bayazit AA, Bayazit OB (2005) Automated motion synthesis for dancing characters. Comput Animat Virt W 16(3–4):259–271
2. Bartsch MA, Wakefield GH (2001) To catch a chorus: using chroma-based representations for audio thumbnailing. In: 2001 IEEE workshop on the applications of signal processing to audio and acoustics, pp 15–18
3. Foote J (1999) Visualizing music and audio using self-similarity. In: Proc. ACM multimedia, pp 70–80
4. Foote J (2000) Automatic audio segmentation using a measure of audio novelty. In: Proc. IEEE int. conf. multimedia and expo (ICME2000), vol 1, pp 452–455
5. Gray JM (1975) An exploration of musical timbre. PhD thesis, Dept. of Psychology, Stanford University, Stanford, CA, USA (1975)

6. Grunberg D, Ellenberg R, Kim Y, Oh P (2009) Creating an autonomous dancing robot. In: Proceedings of the international conference on hybrid information technology (ICHIT), pp 221–227
7. Ikeuchi K, Shiratori T, Nakazawa A (2006) Dancing-to-music character animation. Comput Graph Forum 25:449–458
8. Kang K-K, Kim D (2007) Synthesis of dancing character motion from beatboxing sounds, smart graphics. In: Lecture notes in computer science, vol 4569. Springer, Berlin, pp 216–219
9. Kanungo T, Netanyahu NS, Wu AY (2002) An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans Pattern Anal Mach Intell 24(7):881–892
10. Kim JW, Fouad H, Sibert JL, Hahn JK (2009) Perceptually motivated automatic dance motion generation for music. Comput Animat Virt W 20:375–384
11. Likas A, Vlassis N, Verbeek JJ (2003) The global k-means clustering algorithm. Pattern Recogn 36(2):451–461
12. Loi K-C, Li T-Y (2009) Automatic generation of character animations expressing music features. In: Proceedings of the APSIPA annual summit and conference, pp 216–221
13. Nakahara N, Miyazaki K, Sakamoto H, Fujisawa TX, Nagata N, Nakatsu R (2009) Dance motion control of a humanoid robot based on real-time tempo tracking from musical audio signals. In: Entertainment computing—ICEC 2009, lecture notes in computer science, vol 5709. Springer, Berlin, pp 36–47
14. Nakaoka S, Kajita S, Yokoi K (2010) Intuitive and flexible user interface for creating whole body motions of biped humanoid robots. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1675–1682
15. Ofli F, Erzin E, Yemez Y, Tekalp AM (2010) Multi-modal analysis of dance performances for music-driven choreography synthesis. In: Proc. IEEE international conference on acoustics speech and signal processing (ICASSP), pp 2466–2469
16. Sandholm A, Pronost N, Thalmann D (2009) MotionLab: a Matlab toolbox for extracting and processing experimental motion capture data for neuromuscular simulations. In: Modelling the physiological human—lecture notes in computer science, vol 5903, pp 110–124
17. Sauer D, Yang Y-H (2009) Music-driven character animation. ACM T Multim Comput 5(4):1–16
18. Shiratori T, Ikeuchi K (2008) Synthesis of dance performance based on analyses of human motion and music. Inf Process Soc JPN 1:80–93
19. Taylor GW, Hinton GE, Roweis ST (2007) Modeling human motion using binary latent variables. In: Advances in neural information processing systems, vol 19, pp 1345–1352

Minho Lee received the B.S. degree in electrical and electronic engineering from Pohang University of Science and Technology, Korea, in 2008, and the M.S. degree in Department of Intelligent Convergence Systems from Seoul National University, Korea, in 2010. He is now a PhD course student in Graduate School of Convergence Science & Technology from Seoul National University, Korea. His research interests lie in the areas of human movement analysis, and human motion imitation by humanoid robot.

Kyogu Lee received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1996, the M.M. degree in music technology from New York University, New York, in 2002, and the M.S. degree in electrical engineering and the PhD degree in computer-based music theory and acoustics from Stanford University, Stanford, CA, in 2007 and 2008, respectively. He worked as a senior researcher in the Media Technology Lab at Gracenote from 2007 to 2009. He is now an assistant professor in the Graduate School of Convergence Science & Technology at Seoul National University, Seoul, Korea and is leading the Music and Audio Research Group (MARG). His research focuses on signal processing and machine learning applied to music/audio. Some of his research interests include: Music Information Retrieval, Probabilistic Modeling of Audio/Music, Computational Music Perception/Cognition, Source Separation, Interactive Music, and New Music Interface.

Jaeheung Park received the B.S. and M.S. degrees in aerospace engineering from Seoul National University, Korea, in 1995 and 1999, respectively, and the PhD degree in aeronautics and astronautics from Stanford University, U.S. in 2006. From 2006 to 2009, He was a post-doctoral researcher and later a research associate at Stanford Artificial Intelligence Laboratory. From 2007 to 2008, he worked part-time at Hansen Medical Inc., a medical robotics company in U.S. Since 2009, he has been an assistant professor in the Graduate School of Convergence Science & Technology at Seoul National University, Korea. His research interests lie in the areas of robot-environment interaction, contact force control, robust haptic teleoperation, multicontact control, whole-body dynamic control, biomechanics, and medical robotics.