MUSIC CONTENT DRIVEN AUTOMATED CHOREOGRAPHY WITH BEAT-WISE MOTION CONNECTIVITY CONSTRAINTS

Satoru Fukayama, Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{s.fukayama,m.goto}@aist.go.jp

ABSTRACT

We propose a novel method for generating choreographies driven by music content analysis. Although a considerable amount of research has been conducted in this field, a way to leverage various music features or music content in automated choreography has not been proposed. Previous methods suffer from a limitation in which they often generate motions giving the impression of randomness and lacking context. In this research, we first discuss what types of music content information can be used in automated choreography and then argue that creating choreography that reflects this music content requires novel beat-wise motion connectivity constraints. Finally, we propose a probabilistic framework for generating choreography that satisfies both music content and motion connectivity constraints. The evaluation indicates that the choreographies generated by our proposed method were chosen as having more realistic dance motion than those generated without the constraints.

1. INTRODUCTION

Motion capture systems are widely used to create choreographies for dancing robots or computer animated characters. However, this methodology does not provide flexibility in creating choreographies with various types of music since the choreography needs to be manually created from scratch for every change in the accompanying music. Motion capture systems are often unavailable to those who create dance motion video clips and upload them to video sharing services on the Internet. They usually design choreographies by setting each pose on key frames, which requires a considerable amount of time. We aim to achieve automated choreography to generate dance motions of computer animated characters accompanied by an arbitrary music.

We define automated choreography as a task to automatically generate choreography by leveraging the music content. Previous approaches to generating choreography tried to find dance motion that mostly match the music segment from the viewpoint of various music features. Music features such as tempo [1, 2], beats [3–5], combinations of acoustic features [6], music structures [7], pitches [8], and melodic contours [9] have been used to analyze the relationships between music and dance.

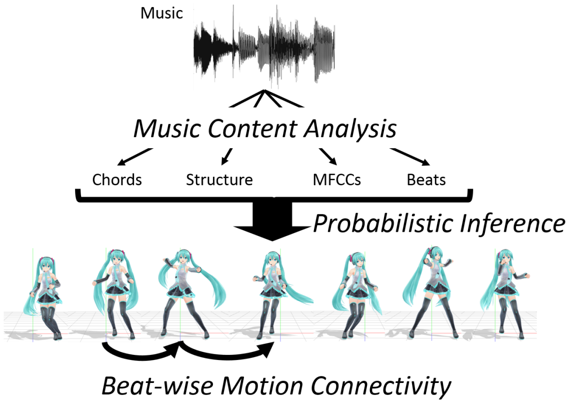

Figure 1. Overview of this research. Various music features are leveraged to generate choreography by concatenating the dance motions. To maintain quality of choreography, motion connectivity constraints are introduced. Generating process is implemented in probabilistic framework. Generated sample of choreographies can be found at https://staff.aist.go.jp/s.fukayama/SMC2015/.

However, the following three issues have not yet been addressed. First, which music features are useful in generating choreography has not been investigated. Second, the connectivity constraints of dance motions have not been considered when the length of the motions are short to reflect the constraints based on the music. Finally, the way to combine music constraints and motion connectivity constraints by using the limited amount of data is not clear.

We propose a novel framework for solving these problems. First, we investigate which music content gives the most useful constraints for choreography based on a data driven approach. Second, we propose a novel method for considering the physical constraints of choreography, such as avoiding unnatural motions and encouraging repetitions. Third, we discuss a probabilistic framework that can simultaneously consider both music constraints and motion connectivity constraints even when there is a limited amount of motion data.

In our probabilistic framework, there are two technical novelties. First, training the probabilistic model that represents the relationship between the music content and dance motion often suffers from data sparseness. We solve this by using the linear combination of probabilistic models where every model holds information about the relationship between each musical feature and the dance motion. Second, calculating the probability of concatenating the dance motion is difficult, since most of the dance motions only appears once in the data and it is impossible to observe various transitions from the same dance motion. We therefore perform the interpolation of probabilistic values by leveraging the distance between dance motions and calculating the transition probabilities.

2. MUSICAL CONSTRAINTS

What kind of musical features are useful for choreography? Our research aims to give a tentative answer in a data driven approach.

Our method is leveraged by various music analysis techniques. Although there has been previous research in automated choreography that is leveraged by acoustic analysis [6] and structural analysis [7] of pieces of music, we are not aware of research that tried to utilize musical features such as chord labels or up/down-beats, which is one of our research contributions.

2.1 Acoustic feature (MFCCs)

Mel-frequency cepstral coefficients (MFCCs) and their first- and second-order frame-to-frame differences (delta MFCCs and delta-delta MFCCs, respectively) are used to change the choreography through the auditory differences of the music. MFCCs and the deltas are widely used in acoustical analysis of music as they are said to approximate the human auditory system's response. We chose 16 as the dimension of the coefficients, which led us to calculate vectors with 48 dimensions consisting of 16 dimensions for each MFCCs, delta MFCCs, and delta-delta MFCCs vector.

As the supply of choreographies and music training samples are limited, and to avoid overfitting between choreography and the musical features, we do not use the values of MFCCs themselves but instead use an index of a feature cluster. The feature clusters are obtained by conducting k-means clustering with a fixed number of clusters (500). The clustering is done after dimension reduction by performing principal component analysis (PCA) on the data to avoid insufficient clustering caused by the high dimensionality of the data. It reduced the number of dimensions from 48 to 16.

2.2 Musical structure

Analysis results of music structural segmentation are used to create structure and highlight segments in choreography. Structural segmentation detects and labels similar segments in a piece of music. It can be used in choreography to generate similar dances among segments with the same label. Segments labeled as chorus sections, which are the most highlighted segments, can be used to generate relatively active motions compared to the other parts.

Structural segmentation can be conducted by analyzing the self-similarity matrix (SSM) of frame-by-frame acoustic features such as MFCCs or chroma vectors. We used an SSM-based approach, analyzed the hierarchical structure of the music, and simultaneously detected the chorus section.

The results of the hierarchical structural segmentation were encoded into vectors, for example as $[1, 0, 0, 1, 0]$, containing binary values that each indicated whether the segment belonged to the n^{th} hierarchical structure. The dimension of this encoded vector was set to the maximum number of hierarchical structures observed in the music we used. When the segment was detected as a chorus section, the first component of the vector was replace with 2 as $[2, 0, 0, 1, 0]$.

2.3 Beat locations and measure boundaries

The information regarding beat locations and measure boundaries is useful for aligning choreography to music. Since choreography is usually described for every beat or "count", it is natural to consider beats when creating choreography. Furthermore, the up-beat and down-beat information that can be obtained from the measure boundaries and the beat locations is useful in differentiating the moves depending on the strength of each beat.

The beat locations and measure boundaries were first analyzed with the beat detection module based on the calculation of the beat salience function. The analysis results obtained from the module were manually corrected afterwards. All the beats were labeled with integers indicating the beat order in a measure and the number of beats in a measure, such as "1/4", "2/4", "3/4", and "4/4" for a measure with 4 beats.

2.4 Chord sequence

The chord sequences bring us similarity information for the beats while the hierarchical structure gives us more global similarity information for the sections. Although the chords, especially the chord labels, do not seem to be helpful in creating dances, their information can be used as supplementary queues for creating local structure in choreography. Even though we can not uniquely determine what kind of choreography should be aligned to a specific chord label, we can generate similar motions for segments with the same chord labels.

Chord sequences were analyzed with the automatic chord recognition module based on chroma features and Hidden Markov Models (HMM). The information described in a chord label included the root note, chord type, and base note if the root note was not the base note.

3. MOTION CONNECTIVITY CONSTRAINTS

When we try concatenating the fragments of dance motions (motion fragments) to generate choreography, concatenating fragments with long lengths seems to be a reasonable strategy to ensure the quality. This is because the generated results contain more motions which match those in the choreography database. Setting a shorter segment length increases the risk of generating motion that seems to be random and lacking context.

However, because of the limited size of the choreography database, there is a trade-off between finding longer segments and satisfying more musical constraints. A longer segment contains more beats than shorter segments, so the number of beat-wise musical constraints increase, and this make it difficult to find a long segment that satisfies those constraints.

Thus, we concatenate the fragments with a short length that can satisfy the beat-wise musical constraints. The motion connectivity constraints are simultaneously considered to avoid randomness in the choreography.

3.1 Smoothness of fragment transition

To avoid generating discrete moves when concatenating the fragments, the smoothness between two adjacent fragments should be considered. Previous research into choreography with concatenation approach has tended to check the smoothness by calculating only the similarity between the end of the first fragment and the beginning of the following fragment [3]. As this approach does not take smoothness between the connection points into account, the degree of success largely depends on what kind of interpolation (linear, spline and so forth) is used. Therefore, we calculate the distance between two fragments by summing up the distances among all the points between the connection points.

3.2 Repetitions

Repetitive moves are often observed in choreography. We created a hypothesis for preferred and not-preferred types of repetitions and imposed constraints on the concatenation of motion fragments.

Too much repetition affects the naturalness of the dance especially when the repetitions are within a few beats. To avoid this, we set a constraint to prohibit using fragments that appeared in the past 4 beats.

On the other hand, repetition of segments of 4 beats or 8 beats is popular. Thus, we impose constraints on the motion to encourage this kind of repetition. The way to constrain the motion in this manner is described in the next section.

3.3 Phrasing of dances

Without proper constraints, concatenation of motion fragments tends to generate motions without phrasing. Here, the phrasing is the segmented structure of continuous movements, not having a sudden halt in the middle of a segment.

Therefore, we monitor the “activeness” value of each motion fragment, which is calculated by taking the squared sum of the frame-to-frame differential of the body movements. Constraints are imposed on the sequence of fragments to prevent a drastic change in activeness between adjacent fragments.

3.4 Parallel shift

Even though we impose constraints to ensure a smooth change between fragments as described above, smooth movements, such as the parallel shift of the dancer, look strange and are hard to recognize as human movements. This is because the legs are usually used to move a body horizontally; however, some concatenation that considers only the smoothness between the fragments may generate motions of shifting horizontally without moving the dancers legs.

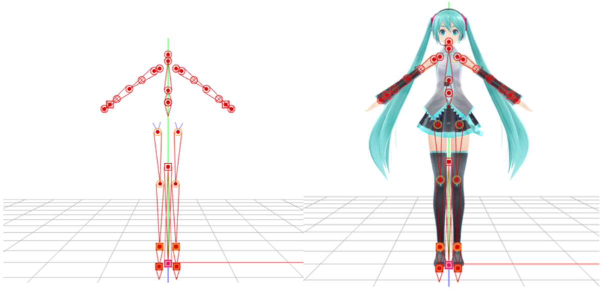

Figure 2. 28 bones (on the left) are used in our formulation to simulate the movements of a dancer (on the right). 5 bones are inverse kinematic bones (IK bones), which jointly move other bones, and movements of each bone are described with 7 values (3 values for position, and 4 values for rotation). Movements of other 21 bones are described with 4 values for rotation.

From our observation, these strange moves often occur when there is a position change of the body center without changes in the rotation angles of the legs. We impose constraints to avoid these kinds of movement.

4. MATHEMATICAL FORMULATION OF AUTOMATED CHOREOGRAPHY

4.1 Data structure for poses

Choreography can be represented with frame-by-frame sequential values of positions and rotations of “bones”. Here, bones are the structures embedded in the 3D model of a dancer that approximately correspond to the real bones in a human. Each bone is connected to the other bone to construct a human body. We chose 26 bones to simulate the dancer’s movements. The chosen bones are shown in Fig. 2.

The chosen bones consist of 5 inverse kinematic (IK) bones and 21 ordinary bones. The IK bones jointly move the other bones that are connected to the IK bones to avoid unrealistic gestures such as disjointed toes. The IK bones consist of “body center”, “left toe”, “right toe”, “left leg”, and “right leg”. The moves of these IK bones are represented with position and rotation from the original position, which is shown in Fig. 2. The bones are described with 3 values for the three dimensional position of a dancer on the stage and 4 values for the rotation represented with a quaternion. To summarize, 7 values represent the state of an IK bone. The other ordinary bones are represented with only rotation, which requires 4 values since the position of these bones are calculated from the position of the IK bones. In total, 119 values describe a pose at each frame, although our proposed framework can cope with different settings of bones and values.

4.2 Concatenation approach

We aim to use the relationship between the musical constraints and the choreography to generate dance motion from music. We chose the concatenation approach that firstly extracts motion fragments from the dance motion database, then analyzes the relationships between the fragments and the corresponding music constraints, and finally concatenates them to generate a new choreography.

Since the musical constraints can change at every beat, motion fragments should include the pose at (or closest to) the time of the beat onset. The fragment also needs to include the motion behind the current beat and the one towards the next beat since the connectivity between the fragments should be analyzed.

The fragments are cut out in lengths of 2 beats, locating the beat onset in the center of each fragment. Let b_i be the frame index of the i^{th} beat onset. Let $\mathbf{x}[n]$ be the pose vector consisting of 119 values for the positions and rotations of the bones at frame index n. The motion fragment extracted from the neighborhood of b_i is:

$$\mathbf{X} = \{\mathbf{x}[b_{i-1}], \cdots, \mathbf{x}[b_i], \cdots, \mathbf{x}[b_{i+1}]\} \tag{1}$$

where $\mathbf{X}$ denotes the set of pose vectors of the fragment.

We can concatenate the adjacent motion fragments through linear interpolation. For instance, the concatenation of $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ is

$$\mathbf{x}[n] = \begin{cases} \mathbf{x}^{(i)}[n] & b_{i-1} \leq n \leq b_i \\ \frac{b_j - n}{b_j - b_i}\mathbf{x}^{(i)}[n] + \frac{n - b_i}{b_j - b_i}\mathbf{x}^{(j)}[n] & b_i \leq n \leq b_j. \\ \mathbf{x}^{(j)}[n] & b_j \leq n \leq b_{j+1} \end{cases} \tag{2}$$

Following the discussion in Section 3.1, the smoothness $\mathscr{S}$ for concatenation of $\mathbf{X}^{(i)}$ and $\mathbf{X}^{(j)}$ can be defined with the distance between portions of two motion fragments where these two are interpolated:

$$\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right) = \sum_{n=b_i}^{b_j} \left|\mathbf{x}^{(i)}[n] - \mathbf{x}^{(j)}[n]\right|^2 \tag{3}$$

4.3 Probabilistic models for choreography

To generate choreography that is as human-like as possible, we use the tendencies of how the motion fragments appear corresponding to the musical constraints in the motion database. In our method, we capture these tendencies by using probabilistic modeling.

Let A_k, S_k, B_k, and C_k be the labels of musical constraints (acoustic feature, musical structure, beat, and chord, respectively) described in Section 2, which are aligned to the k^{th} motion fragment in the database. The probability for observing $\mathbf{X}^{(i)}$ at the k^{th} frame is the conditional probability, which is represented as $P\left(\mathbf{X}^{(i)}|A_k, S_k, B_k, C_k\right)$.

When we try to train this model, it is difficult to exhaustively observe all the combinations of the musical constraints. Therefore we factorize the probability into sub-models holding information for each musical constraint as:

$$\begin{aligned} &P\left(\mathbf{X}^{(i)}|A_k, S_k, B_k, C_k\right) \\ &= \lambda_0 P\left(\mathbf{X}^{(i)}\right) + \lambda_1 P\left(\mathbf{X}^{(i)}|A_k\right) + \lambda_2 P\left(\mathbf{X}^{(i)}|S_k\right) \\ &\quad + \lambda_3 P\left(\mathbf{X}^{(i)}|B_k\right) + \lambda_4 P\left(\mathbf{X}^{(i)}|C_k\right) + \lambda_5 U \end{aligned} \tag{4}$$

where λ_m $(m = 0, \ldots, 5)$ are the interpolation coefficients satisfying $\sum_m \lambda_m = 1$ and ${}^\forall m, \lambda_m > 0$, and U is the uniform distribution to conduct smoothing. These coefficients are tuned by splitting the training data into two portions, training the sub-models with the first portion, and then maximizing the log-likelihood of the second portion with respect to λ_m.

Since the frequency of the appearance of fragment $\mathbf{X}$ given the condition $Y \in \{A, S, B, C\}$ is sparse, we revise the frequencies using a kernel function and then calculate the conditional probabilities using the revised frequencies. This method introduce kernel functions $\phi_m(\mathbf{X})$ $(m = 1, \cdots, M)$, which returns the similarity between an arbitrary $\mathbf{X}$ and $\mathbf{X}^{(m)}$ in the training data, where M is the number of fragments extracted from the database. Let $c(\mathbf{X}, Y)$ be the frequency of the appearance of fragment $\mathbf{X}$ when the condition value is Y, and let $\hat{c}(\mathbf{X}, Y)$ be the revised frequency. The revised frequency and the conditional probability can be obtained by

$$\hat{c}(\mathbf{X}, Y) = \sum_{m=1}^{M} \phi_m(\mathbf{X})\, c\left(\mathbf{X}^{(m)}, Y\right), \tag{5}$$

$$P\left(\mathbf{X}^{(i)}|Y\right) = \frac{\hat{c}\left(\mathbf{X}^{(i)}, Y\right)}{\sum_{m=1}^{M} \hat{c}\left(\mathbf{X}^{(m)}, Y\right)}. \tag{6}$$

P$\left(\mathbf{X}^{(i)}\right)$ can also be inferred in this manner. We set the kernel function to be the Gaussian distribution as $\phi_m(\mathbf{X}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\mathscr{D}\left(\mathbf{X}, \mathbf{X}^{(m)}\right)\right)$ where $\mathscr{D}\left(\mathbf{X}, \mathbf{X}^{(m)}\right) = \sum_n \left|\mathbf{x}[n] - \mathbf{x}^{(m)}[n]\right|^2$.

The transition probability between fragments can be calculated by using the smoothness measure $\mathscr{S}$ defined in Equation (3). The probability for transitioning from $\mathbf{X}^{(i)}$ to $\mathbf{X}^{(j)}$ is calculated by

$$P\left(\mathbf{X}^{(j)}|\mathbf{X}^{(i)}\right) = \frac{\exp\left(-\frac{1}{2}\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right)\right)}{\sum_{m=1}^{M} \exp\left(-\frac{1}{2}\mathscr{S}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(m)}\right)\right)}. \tag{7}$$

Now we can define the automated choreography in a probabilistic formulation. Given the musical constraints on beats $(k = 1, \cdots, K)$, generating the concatenation of fragments is performed by maximizing the probability $P\left(\mathbf{X}_1 \cdots \mathbf{X}_K | \{A_k\}_{k=1}^K, \{S_k\}_{k=1}^K, \{B_k\}_{k=1}^K, \{C_k\}_{k=1}^K\right)$ with respect to $\mathbf{X}_1 \cdots \mathbf{X}_K$. By taking the logarithm of this probability with first-order Markov assumption, we can derive that this is equivalent to maximizing the objective function

$$\begin{aligned} J(\mathbf{X}_1 \cdots \mathbf{X}_K) &= \sum_{k=1}^{K} \ln P(\mathbf{X}_k | A_k, S_k, B_k, C_k) \\ &\quad + \sum_{k=1}^{K} \ln P(\mathbf{X}_k | \mathbf{X}_{k-1}) \end{aligned} \tag{8}$$

with respect to $\mathbf{X}_1 \cdots \mathbf{X}_K$. Note that we calculated $P(\mathbf{X}_1|\mathbf{X}_0)$ as $P(\mathbf{X}_1)$. The motion fragments that maximize J can be calculated by using dynamic programming. Since the search space for concatenating fragments is huge (M^K possibilities), we used pruning methods to limit the search space and to make the problem computationally feasible.

4.4 Applying motion connectivity constraints

To impose motion connectivity constraints (described in Section 3), the probability distributions and the search space for generating choreographies are revised. Note that the smoothness constraints are already considered in the transition probability of fragments as described in Section 4.3.

Generating repetition of fragments can be implemented by sharing the musical constraints and revising the probability. For instance, if we expect similar fragments at k and k', then probabilities $P(\mathbf{X}|A_k, S_k, B_k, C_k)$ and $P(\mathbf{X}|A_{k'}, S_{k'}, B_{k'}, C_{k'})$ are both renewed to the linear interpolation of these distributions *i.e.* in accordance with $\frac{1}{2}\{P(\mathbf{X}|A_k, S_k, B_k, C_k) + P(\mathbf{X}|A_{k'}, S_{k'}, B_{k'}, C_{k'})\}$.

Phrasings of choreography are generated by monitoring the "activeness" $\mathscr{E}(\mathbf{X})$, which is the squared sum of the frame-to-frame differential of the motion fragment. We can calculate this measure as $\mathscr{E}(\mathbf{X}) = \sum_n |\mathbf{x}[n] - \mathbf{x}[n-1]|^2$. In particular, we reject concatenating $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$ when $|\ln \mathscr{E}(\mathbf{X}_{k+1}) - \ln \mathscr{E}(\mathbf{X}_k)| > 2.0$.

Parallel shift of body center can be checked by monitoring the difference of the "center bone" position per beat and the "activeness measure" with respect to only the "leg bones". We prohibit parallel shift when the difference of the "center bone" position is large but the small "activeness measure" of the "leg bones" is small, which means the dancer is moving without using his/her legs.

5. EVALUATIONS

5.1 Effect of each musical constraint

We conducted an evaluation to verify which musical constraint (among acoustic feature, musical structure, beat, and chord) was "useful" in automated choreography. The verification was performed with an information theoretical method. That is, the "usefulness" of the music constraint Y in choreography was verified when the choreography became more predictable with the probabilistic model using Y than the model without using Y.

The predictability can be compared with the values of cross-entropy between various combinations of musical constraints. In our situation the cross-entropy can be obtained with

$$H(\mathbf{X}|Y) = -\frac{1}{K}\sum_{k=1}^{K} \log_2 P(\mathbf{X}_k|Y_k), \qquad (9)$$

where $k = 1, \ldots, K$ are the indices of motion fragments in the database. Y_k is the musical constraint at the kth fragment. The predictability is high when the value of cross-entropy is low.

Evaluator	1	2	3	4	5
Accuracy	8 /10	10/10	10/10	10 /10	9 /10

Table 1. Results of subjective evaluation. Five evaluators were asked to choose more natural choreography out of two choreographies: one generated with motion connectivity constraints and other generated without them accompanied by 10 different pieces of music. The number of chosen choreographies which were generated with the proposed method is shown above (Accuracy). Mean accuracy for choosing the motion-connectivity constrained choreography was 0.94 with 95% confidence interval ± 0.11 (Student's t-test).

In our experiment, we prepared 20 different combinations of musical constraints. For every combination of constraints, first we split the motion database into three portions and then trained the five sub-models $P(\mathbf{X}), P(\mathbf{X}|A), P(\mathbf{X}|B), P(\mathbf{X}|C), P(\mathbf{X}|S)$ by using the first portion of the database. Second, we optimized the combination weights λ_m in Eq. (4) using the second portion of the database. The optimized λ_ms are the contribution ratio of musical constraints in predicting the motion fragments. Finally, we calculated the cross-entropy by using the third portion of the database. The motion database consisted of $24,527$ motion fragments accompanied with music. $22,527$ motion fragments were used to train the sub-models, $1,000$ fragments were used to optimize the combination weights, and $1,000$ fragments were used to calculate the cross-entropy.

To obtain the music constraints, the music tracks were first automatically analyzed by using our web service called Songle (http://songle.jp) [10] and then corrected manually using the Songle's error correction interface. Songle leverages various music content analysis techniques to automatically analyze songs publicly available on the web and is open to the public.

The result of the evaluation is shown in Figure 3. We confirmed that the cross-entropy decreased when several musical constraints were taken into account ($H(\mathbf{X}) = 7.643 > H(\mathbf{X}|A, B, C, S) = 7.383$). This indicated that the predictability of dance motion had been increased by using the combination of several music constraints. Optimized results of λ_m $(m = 0, \ldots 4)$ are represented as stacked bars in Figure 3. The length of each color in a bar is calculated by $H \times \frac{\lambda_m}{\sum_{m=0}^{4} \lambda_m}$. Note that we did not use λ_5 for calculating the ratio, since the uniform distribution did not hold information from the motion dataset or the musical constraints. The optimized λ_m indicated that the structure label was the most valuable information for predicting a motion fragment.

5.2 Effect of motion connectivity constraints

We conducted a subjective evaluation to confirm that the motion connectivity constraints were effective in maintaining the naturalness of the choreography. The excerpts of choreography we used in this experiment are uploaded at https://staff.aist.go.jp/s.fukayama/SMC2015/. The screen-

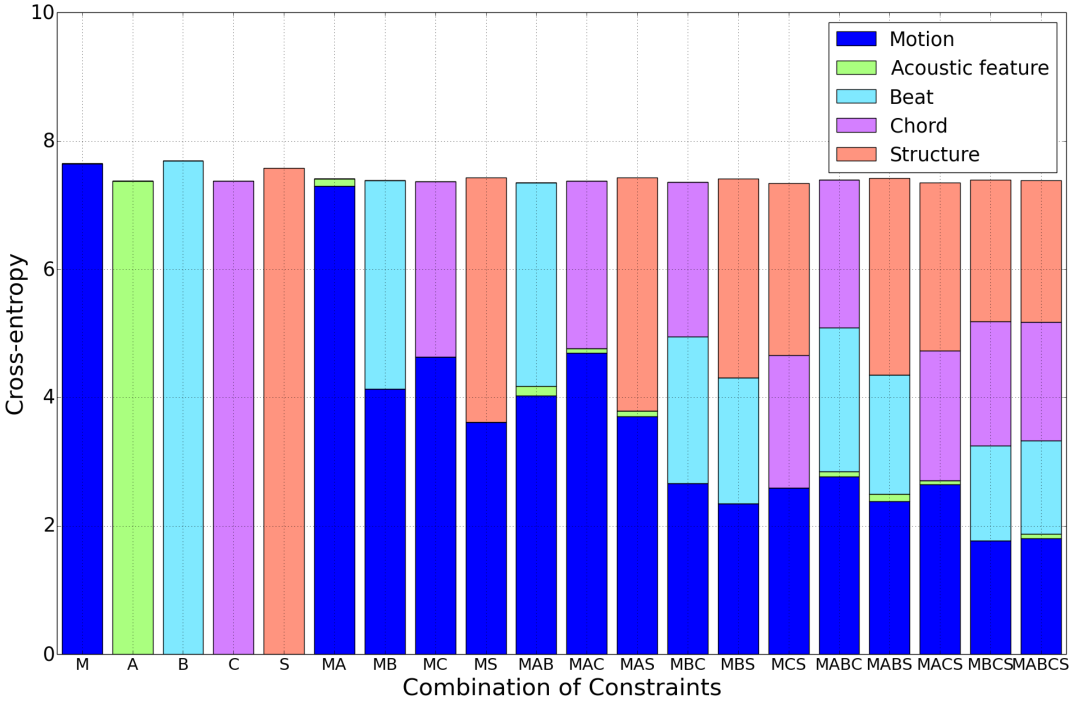

Figure 3. Cross-entropies of motion dataset calculated with probabilistic models with different combinations of music constraints. Height of each bar represents value of cross-entropy, and height of each stacked colored bar indicates ratio of contribution in predicting motion from the music content. Each M, A, B, C and S in the horizontal axis represents motion fragment, acoustic feature, beat, chord and structure, respectively. The cross-entropy decreased by combining several musical constraints ($H\left(\mathbf{X}\right) = 7.643 > H\left(\mathbf{X}|A, B, C, S\right) = 7.383$), which means the predictability of motion fragments was increased. The contribution of structure label tends to be larger than other musical constraints.

shots of the generated choreographies are shown in Figure. 4.

Ten music excerpts were used in the experiment. The music excerpts were sampled from a song (RWC-MDB-P-2001 No. 07) in the RWC Music Database [11]. For every music excerpt, two different 20-second choreographies were shown to the evaluators. They were asked to choose the one which they felt was more natural. One choreography was generated with motion connectivity constraints, which we proposed in this paper, and the other one was generated without them. The order of showing the two different choreographies was randomized per piece of music.

Five evaluators participated in the experiment. The evaluators did not have any particular knowledge of rules that affect the quality of dancing motions. Therefore, we asked them to intuitively choose a more natural choreography from each pair.

The numbers of chosen choreographies which were generated with the proposed method are shown in Table 1. The evaluators found more than 8 choreographies with motion connectivity constraints to be more natural than the other. The mean accuracy for choosing the motion-connectivity constrained choreography was 0.94 ± 0.11 where 0.11 is the 95% confidence interval by the Student's t-test.

6. DISCUSSION

The objective evaluation in Section 5.1 indicates that combining various musical constraints are useful in automated choreography. The structure label is the most valuable information for predicting the dance motion. Features, such as beat labels and chord labels, also contribute in generating choreography. The subjective evaluation in Section 5.2 indicates that the motion connectivity constraints are effective in maintaining the naturalness of the choreography.

We confirmed that the probabilistic modeling is useful in combining several different constraints driven by different music content analysis modules. It can also generate choreographies by maximizing the probability of the concatenated motion fragments.

To improve the quality of the choreography, we are planning to consider various types of audio features such as spectral flux and chroma vectors. These features can be used to reflect detailed information of music content in the choreography. For instance, spectral flux holds information of acoustic events, such as note onsets, and can be used to make the choreography aligned to the melody notes. Chroma vectors might be useful especially when using delta-chromas to capture the chord changes and when reflecting those changes in the choreography.

Another promising direction for improving the quality is to increase the amount of dance motion data. As our approach is data driven, we expect more variety in generating

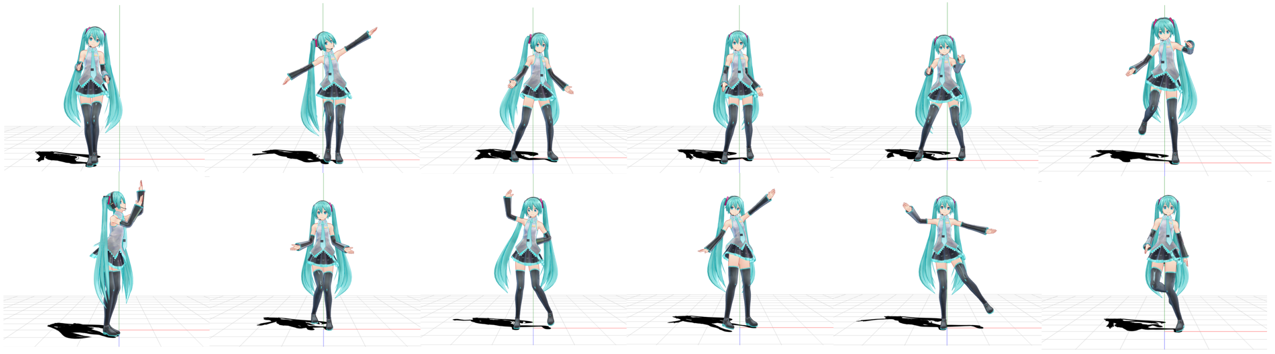

Figure 4. Example of generated choreography with the proposed method.

choreography leveraged by the various dance motions in a larger database. Furthermore, the subjective evaluation can provide more statistical evidence by using a larger dataset.

Finally, we plan to tune the parameters of the probabilistic models. For now, the variance of the Gaussian distribution used as a kernel function is fixed to 1.0, and this value can be tuned with the maximum likelihood framework by using the training data. The value of the variance corresponds to how the motion fragments are roughly categorized as motions with the same character. This may affect the quality of predicting the motion fragment from the music content and therefore needs to be investigated.

7. CONCLUSION

We investigated how to generate choreography automatically by leveraging music content. The proposed method used various music features, not only low-level features such as MFCCs but also features such as structure labels and chord labels to generate choreography. Furthermore, we proposed a set of motion connectivity constraints to ensure the naturalness of the dance motion. These two types of constraints, musical constraints and motion connectivity constraints, were taken into account in a novel probabilistic modeling framework that enabled generating natural music-content driven choreography. Our future work includes more improvements in automated choreography by leveraging more music content analysis techniques that have been considerably developed in the sound and music computing community. The generated sample of our automated choreographies can be found at https://staff.aist.go.jp/s.fukayama/SMC2015/.

Acknowledgement *Hatsune Miku* is copyrighted by Crypton Future Media Inc., and we used it in our research to render the computer graphics of the generated choreography under a PIAPRO license (www.piapro.net). The 3D model of *Hatsune Miku* used in our research was created by koron. This work was supported in part by OngaCREST, CREST, JST.

8. REFERENCES

[1] K. M. Chen, S. T. Shen, and S. D. Prior, “Using music and motion analysis to construct 3D animations and visualisations,” *Digital Creativity*, vol. 19, no. 2, 2008.

[2] C. Panagiotakis, A. Holzapfel, D. Michel, and A. A. Argyros, “Beat synchronous dance animation based on visual analysis of human motion and audio analysis of music tempo,” in *Proc. ISVC 2013*, 2013, pp. 118–127.

[3] T. Shiratori, A. Nakazawa, and K. Ikeuchi, “Synthesizing dance performance using musical and motion features,” in *Proc. ICRA 2006*, 2006, pp. 3654–3659.

[4] J. W. Kim, H. Fouad, J. L. Sibert, and J. K. Hahn, “Perceptually motivated automatic dance motion generation for music,” *Computer Animation and Virtual Worlds 2009*, vol. 20, pp. 375–384, 2009.

[5] G. Alankus, A. A. Bayazit, and O. B. Bayazit, *Computer Animation and Virtual Worlds*, no. 16, pp. 259–271, 2005.

[6] R. Fan, S. Xu, and W. Geng, “Example-based automatic music-driven conventional dance motion synthesis,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 3, 2012.

[7] M. Lee, L. Lee, and J. Park, “Music similarity-based approach to generating dance motion sequence,” *Multimedia Tools and Applications*, vol. 62, no. 3, pp. 895–912, 2013.

[8] F. Ofli, E. Erzin, Y. Yemez, and A. M. Tekalp, “Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, 2012.

[9] S. Oore and Y. Akiyama, “Learning to synthesize arm motion to music by example,” in *Proc. WSCG 2006*, 2006, pp. 201–208.

[10] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, “Songle: A web service for active music listening improved by user contributions,” in *Proc. ISMIR 2011*, 2011, pp. 311–316.

[11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical, and jazz music databases,” in *Proc. ISMIR 2002*, 2002, pp. 287–288.