DANCEREPRODUCER: AN AUTOMATIC MASHUP MUSIC VIDEO GENERATION SYSTEM BY REUSING DANCE VIDEO CLIPS ON THE WEB

Tomoyasu Nakano[†1] **Sora Murofushi**[‡3] **Masataka Goto**[†2] **Shigeo Morishima** [‡3]

[†] National Institute of Advanced Industrial Science and Technology (AIST), Japan

[‡] Waseda University, Japan

[1] `t.nakano[at]aist.go.jp` [2] `m.goto[at]aist.go.jp` [3] `shigeo[at]waseda.jp`

ABSTRACT

We propose a dance video authoring system, *DanceReProducer*, that can automatically generate a dance video clip appropriate to a given piece of music by segmenting and concatenating existing dance video clips. In this paper, we focus on the *reuse* of ever-increasing user-generated dance video clips on a video sharing web service. In a video clip consisting of music (audio signals) and image sequences (video frames), the image sequences are often synchronized with or related to the music. Such relationships are diverse in different video clips, but were not dealt with by previous methods for automatic music video generation. Our system employs machine learning and beat tracking techniques to model these relationships. To generate new music video clips, short image sequences that have been previously extracted from other music clips are stretched and concatenated so that the emerging image sequence matches the rhythmic structure of the target song. Besides automatically generating music videos, DanceReProducer offers a user interface in which a user can interactively change image sequences just by choosing different candidates. This way people with little knowledge or experience in MAD movie generation can interactively create personalized video clips.

1. INTRODUCTION

User-generated video clips called *MAD movies*[1] or *mashup videos*, each of which is a derivative (mixture or combination) of some original video clips, are gaining popularity on the web and a lot of them have been uploaded and are available on video sharing web services. In this paper, we focus on music video clips of dance scenes (dance video clips) in the form of MAD movies or mashup videos. Such a MAD music video clip consists of a musical piece (audio signals) and image sequences (video frames) taken from other original video clips. The original video clips are called *1st generation (primary or original) content*, and the MAD video clips generated by users can be considered *2nd generation (secondary or derivative) content* (Figure 1). In a MAD video clip, good music-to-image synchronization with respect to rhythm, impression, and context is important.

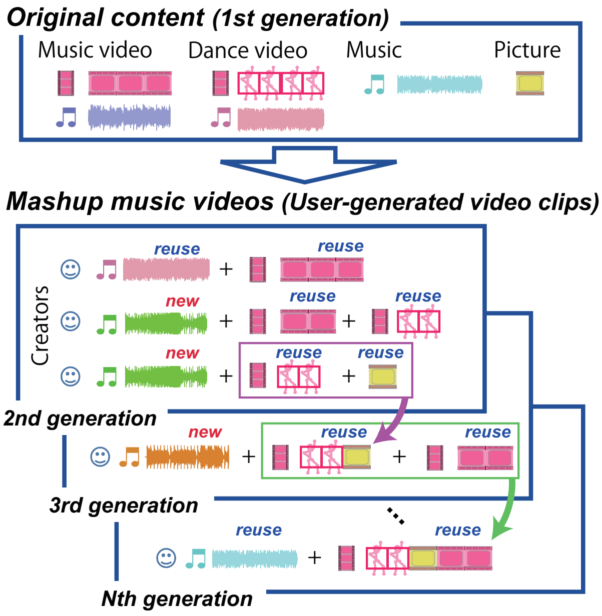

Figure 1. Generation of mashup music videos (user-generated music video clips) by reusing existing original content.

Although it is easy to enjoy watching MAD movies, it is not easy to generate them because a creator needs to (1) search, in existing video clips, for image sequences that give impressions appropriate to a given target musical piece, (2) segment and concatenate image sequences to fit the target piece, and (3) time-stretch the sequences to match the tempo of the target piece because existing video clips usually have tempi different from the tempo of the target piece. Moreover, for better music-to-image synchronization, the music structure and context of a musical piece and image sequences should be taken into account, but it requires considerable time and effort.

To give a chance of enjoying such difficult MAD movie generation to everybody, we have developed a new system called *DanceReProducer* that can automatically gen-

[1] http://en.wikipedia.org/wiki/MAD_Movie

erate a dance video clip for any given piece of music by segmenting, concatenating, and stretching existing dance video clips (Figure 2). This system provides an interface in which a user not only listens to music but also enjoys music visually by directing (supervising) the (semi)automatic generation of dance video image sequences. If the automatically generated video clip is satisfactory, the user can just watch it, but if the user does not like generated image sequences for some musical sections (*e.g.*, A, B, and C in Figure 2), the user can easily choose another favorite image sequence from ranked candidates for each musical section. These candidates are also automatically proposed by the system and would also match a given musical section of the input piece according to our mapping model. This mapping model was trained through an analysis of a large amount of user-generated dance video clips available on a video sharing web service. In particular, we focus on the reuse of video clips of the 2nd, 3rd, and Nth generation content (Figure 1) as well as the 1st generation content. In other words, our system enables a user to generate a new mashup video clip by reusing existing mashup video clips on the web.

2. RELATED WORK

Previous works generated visual patterns based on some musical aspects, such as visualizing music chords by color [1], visualizing musical mood [2], and controlling a computer-graphics dancer under musical beats [3, 4]. There were also previous works automatically generating music-synchronized video by reusing media content: for example, some reused images and photographs from the web [5, 6], and others reused home videos [7, 8] under audio changes [7] or repetitive visual and aural patterns [8]. Previous works, however, did not reuse dance video clips on the web to generate a new mashup video clip.

3. SYSTEM DESIGN

To develop DanceReProducer, we first considered the criteria that people use in judging “what is an appropriate image sequence for a particular piece of music”, as described below. We then describe functions of the system interface.

3.1 Criteria of natural/skillful relationships between an image sequence and music

To design the system, we considered the criteria from two aspects – local relationships and context (global) relationships explained below – taking into account previous work [7, 8] and the comments offered by human creators of MAD movies[2] .

Local relationships : criteria for impression synchronization between the music and image sequences.

- *Rhythm*: Visual rhythms such as dance motion, camera work, and cut (*e.g.*, dissolve) are synchronized with beat and musical accent.

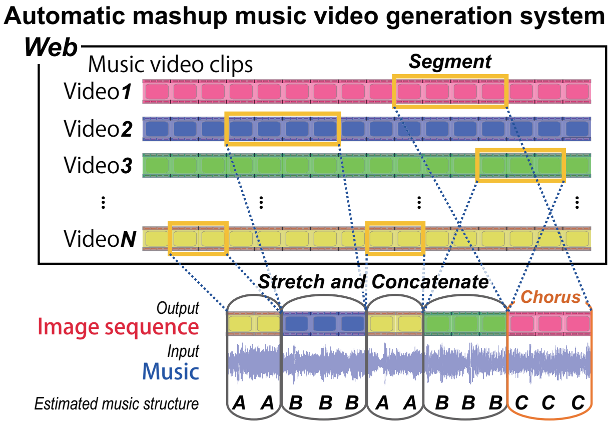

Figure 2. An automatic music video generation system *DanceReProducer* by reusing existing music video clips.

- *Impression*: Visual impressions such as dance motion, color, brightness, and lighting are synchronized with the musical impression.

Context relationships : criteria for context synchronization between music and image sequences.

- *Music structure*: Visual impression (temporal) changes are synchronized with the music structure. (*e.g.*, verse A, chorus).
- *Temporal continuity*: Image sequence has temporal continuity, but visual impression can be changed easily on a music structure boundary.

The above criteria are not all satisfied at any given time, and are not mutually independent. However, they provide a useful foundation for generating an image sequence appropriate to a particular piece of music.

3.2 Image sequence generation

The mashup video generation done manually is difficult and time-consuming. To enable more efficient generation, our system first automatically generates an image sequence appropriate to the music. However, the generated sequence may not be to the user’s taste. In such cases, other sequence candidates are shown on a screen so that the user can simply choose a preferred one. Even though it would be difficult for a user to manually find another candidate from among a huge number of candidates, it is easy to interactively choose a preferred candidate.

We provide an overview of the interface’s image sequence generation and functions below.

3.2.1 *Automatic image sequence generation*

To reuse existing content, we first gather dance video clips on a video sharing web service and the system estimates the tempo and bar line of the music (audio signals) in those video clips. We assume the music and its dance motions within each video clip are synchronized while dealing with the local relationships, and use each bar (measure) of the

[2] Some creators disclosed their creative processes on the web.

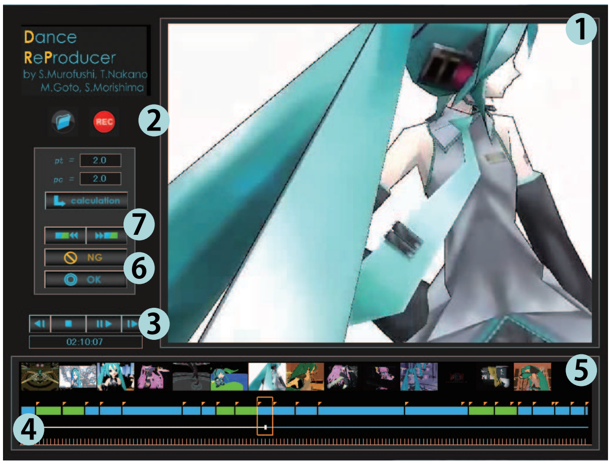

Figure 3. Example of the DanceReProducer screen.

music as the minimum unit for segmenting and concatenating image sequences. Hereafter, we denote an image sequence (series of video frames) for the bar-level minimum unit as a *visual unit*.

Second, the system searches for a visual unit appropriate to each bar for the input target musical piece. The units are time-stretched under the tempo of the input music, and then are concatenated to generate an image sequence. In this regard, to deal with the context relationships, the system selects visual units which take into account music structure and temporal continuity.

To satisfy each criterion described in 3.1, we implement the following processes.

Rhythmic synchronization: A musical bar is used as the minimum unit for segmenting and concatenating. A visual unit is stretched under input music tempo.

Impression synchronization: By modeling the mapping between the extracted audio and visual features for impression, the system automatically selects an appropriate visual unit to input music impression in each bar.

Music structure and *Temporal continuity*: By introducing costs representing the temporal continuity and music structure of the generated sequence, the system automatically selects an image sequence considering the context relationships.

3.2.2 *Interface*

Screenshots of the implemented DanceReProducer interface are shown in Figure 3 and 4. There are basic functions for viewing, such as a window showing the generated image sequence (Figure 3, ①), functions to load input music and save the generated video (②), to play and stop/pause the generated video (③), and a playback-position "slider" and the music structure estimated automatically [9] (④). The green rectangular markers in the music structure represent chorus sections, and the blue markers represent other sections. In addition, the total duration of the input music is equally divided into 15 sections (⑤).

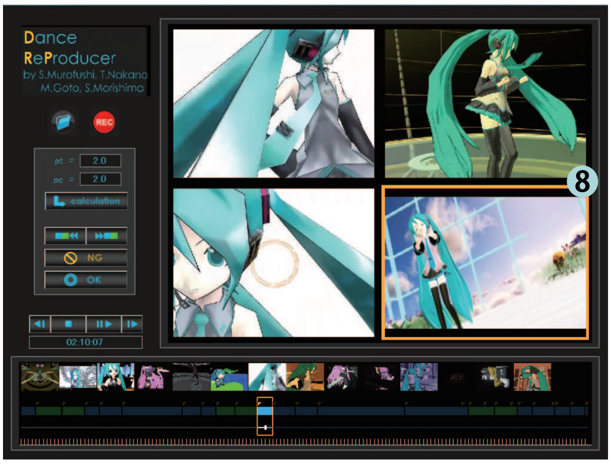

Figure 4. Example of interactive sequence selection. Four different image sequence candidates are previewed and the lower-right candidate is chosen by a user.

This interface also provides the following functions to reflect the user's preferences.

Interactive re-selection of a generated image sequence: By clicking the NG button (Figure 3, ⑥), the user can see other sequence candidates on a screen and simply choose the preferred one (Figure 4, ⑧). The user can see and compare different candidates during playback and can choose his/her favorite sequence. Since this interactive re-selection function works on each section of the music structure (*e.g.*, A, B, and C in Figure 2), the user can use this function to easily consider the music structure and context.

Jumping to the beginning of sections: By clicking the jump button (Figure 3, ⑦) or visualized sections (④), a user can directly jump to and view the previous or the next section of a song.

4. INTERNAL MECHANISM OF DANCEREPRODUCER

To develop DanceReProducer, we modeled the relationships between music and video, and then generated image sequences appropriate to input music by considering the local and context relationships. In general, it is difficult to model such relationships, but we solved this problem through training using a huge quantity mashup video clips posted to the web. Since the content videos were made by humans, there were various types of mutual relationship between the music and the image sequences. This suggests that such videos can be used to learn the relationships through a machine-learning technique.

Modeling using the mashup clips suffers from two problems. One is that complex relationships exist, such as where "the same image sequences are used for different music" or "different image sequences are used for the same music" (Figure 1). Another problem is that the video quality varies strongly, and it is difficult to judge the possibility

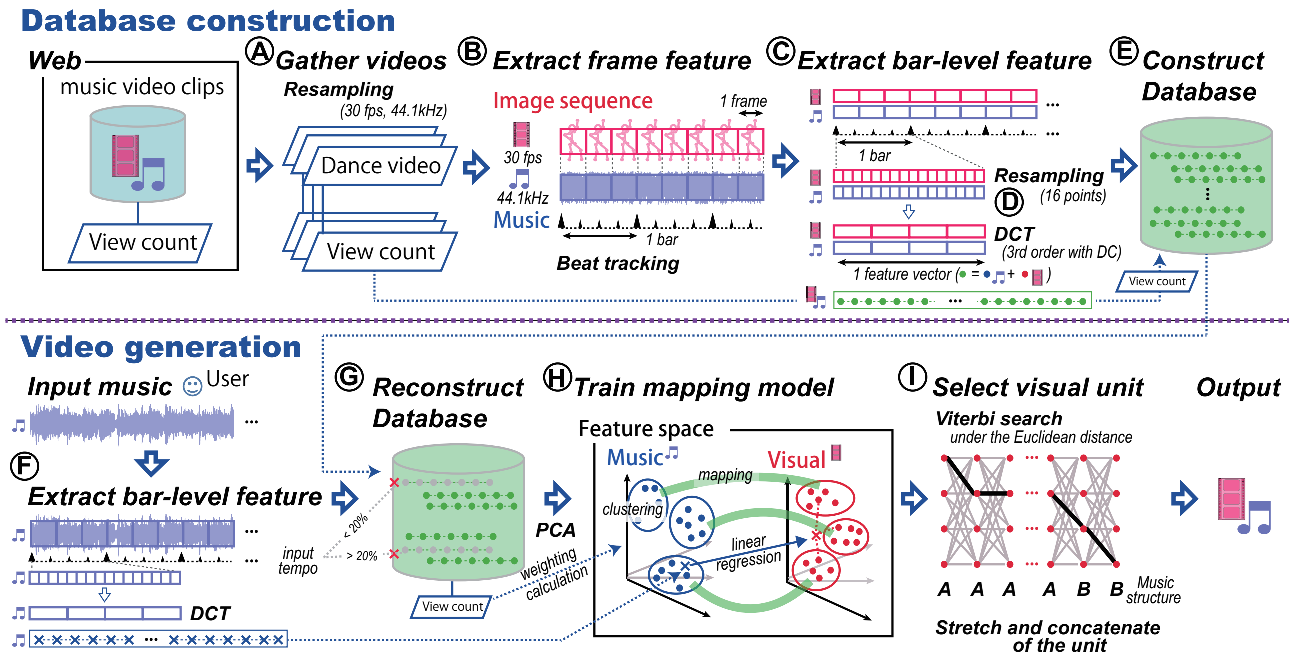

Figure 5. Overview of DanceReProducer, a dance video authoring system that can automatically generate a dance video clip appropriate to a given pice of music by segmenting, concatenating, and stretching existing dance video clips.

of its reuse. These obstacles make it difficult to model the relationships, and were not dealt with previous works.

Figure 5 gives an overview of the DanceReProducer system. The system consists of two procedures: database construction and video generation. In this section, we describe the details of the system and explain how we solve the above two problems in modeling using the mashup clips.

4.1 Database construction

In the database construction, database videos are gathered via the web and then audio and visual features are extracted from the videos through the following steps.

Step 1) Gather dance music videos via web, and resample the sampling frequency of the music to 44.1 kHz, and the frame-rate of the image sequence to 30 fps (Figure 5, Ⓐ).

Step 2) Estimate bar line of the videos by using beat tracking techniques (Ⓑ).

Step 3) Extract feature vectors to learn their relationship (Ⓑ–Ⓒ). Since the analysis frame matches the frame rate, the discrete time step (1 *frame-time*) is about 33 ms (about 1470 points). The extracted features in each frame-time are called *frame features*. The frame features are then integrated in each bar to obtain what are called *bar-level features*.

4.1.1 Beat tracking

Much work has been done on beat tracking [3], and we plan to focus on using such techniques in the future, but our current implementation is a simple one which was effective in our preliminary experiment.

The system first calculates the power of the input audio signal, and then calculates its autocorrelation values and estimates their peak time. Since it represents the periodicity of the power, we use the time as tempo (one beat time). In this regard, to avoid octave error (*e.g.*, half-/double-tempo error), the estimation is limited to tempo within a range of $60 - 120$ bpm (beat per minute).

Second, the system calculates cross-correlation between the power and the pulse signal generated under the estimated tempo. Since the peak time of the cross-correlation represents the first beat time, the system regards the time as the beginning time of the first bar. In addition, we assume that the dataset videos have a length of 4 beats (one measure in 4/4 time), and then the system decides all bar lines mechanically.

4.1.2 Frame feature extraction (Music)

The frame features of music are defined with the help of previous work on relationships between audio and visual [10, 11] and musical genre classification [12]. These features represent musical accents and impressions.

As the frame features for accents, to represent temporal change in the power of the audio signal, we extract the filter bank output (4 dims.) and spectral flux (1 dim.). As the frame features for impressions, to represent timbre, we extract the zero-crossing rate (1 dim.) and 12th order MFCCs (mel-frequency cepstral coefficients) with a DC component (13 dims.).

4.1.3 Frame feature extraction (Image sequence)

The frame features of an image sequence are defined with the help of previous work on relationships between audio and visual [10, 11]. These features represent visual accents and impressions. To extract the features, the image resolution is resampled to 128×96.

As the frame features for accents, to represent camera work and dance motion and related temporal changes, we extract the mean values of the temporal derivative of the well-known optical flow and brightness (2 dims.). We use a block-matching algorithm to detect the optical flow from image sequences; we use a 64×48 block which is shifted by 1 (maximum range is 4). The frame features for impressions are the mean values and standard deviations of the hue, saturation, and brightness values (6 dims.). In addition, 2-dimensional DCT (discrete cosine transform) coefficients are extracted (4 dims. for vertical and 3 dims. for horizontal).

4.1.4 Bar-level feature extraction

We propose a *bar-level feature* which is an integration of the frame features in each bar. To extract features from one piece of music or one video clip, in most previous work (*e.g.*, musical genre classification) integration was done using the time average and its standard deviation [12]. However, such integration drops temporal information of the audio/visual features.

In this paper, we integrate these frame features to bar-level features via using DCT (Figure 5, Ⓓ). In each bar, frame features are resampled to 16 points for the time axis, the system computes DCT for each dimension, and then the 3rd order DCT coefficients with a DC component used as the bar-level features. Therefore, the number of dimensions of the bar-level features is four times the number from the frame features.

4.2 Video generation

In the video generation, to select visual units for each frame from the database, the system process consists of the following steps.

Step 1) Extract the bar-level features of a given musical piece (Figure 5, Ⓕ).

Step 2) Reconstruct the database (Ⓖ). To avoid generating a video with unnaturally fast/slow tempo, visual units with tempi 20% above or below the input tempo are not used for the following steps.

Step 3) Apply PCA (principal component analysis) for all bar-level features of all bars, and store low N-dimensional features. The N-dimension is decided based on the cumulative contribution ratio ($\leq 95\%$). For our investigations, the dimensions of audio and visual features described above were reduced from 76 to 62 and from 80 to 68, respectively[3] .

Step 4) Model relationship between music and image sequence from the database (Ⓗ). This step is explained in more detail below (section **4.2.1**).

Step 5) Select visual units under the criteria of the relationships described in 3.1 (Ⓘ).

4.2.1 Linear regression models for multiple clusters

In this paper, a local cost is calculated by a linear regression model, which is used to learn the relationships between the audio and visual bar-level features. However, to model complex relationships, such as "the same visual units are used for different music" or "different visual units are used for the same music" (Figure 1), one regression model is insufficient.

Therefore, we propose a linear regression, where the system uses linear regression models for multiple clusters. The multiple clusters are obtained by applying k-means clustering to feature vectors, where a feature vector is defined as a concatenation of a bar-level audio feature (of music) and a bar-level visual feature (of image sequences) in the database. Note that this feature vector is used just for the clustering. For each cluster, a linear regression model is trained so that bar-level visual features can be predicted by bar-level audio (music) features (Figure 5, Ⓗ).

4.2.2 Image sequence selection under the criteria for natural/skillful relationships

By introducing costs representing the local and context relationships, we can solve this video generation problem by minimizing the costs through a Viterbi search (Figure 5, Ⓘ). The model of the cluster having the centroid nearest to the input features is selected, and visual features appropriate to the input audio features are estimated by using the model. To calculate the costs of the local relationships, the system calculates the distance between the estimated features and the visual features of all units.

To represent the costs of the context relationships, a musical structure and chorus section are estimated using RefraiD [9]. The estimated beginning and ending times of all sections are used as the boundaries of a musical section. However, sections less than 4 bars in length are not used as a section for this purpose.

Let $d(n, k_m)$ be the Euclidean distance representing the local cost between the $n (1 \leq n \leq N)$th bar level feature of the input and the mth video's kth unit's features of the database. The calculated local costs and accumulated costs are defined as follows.

$$c_l(n, k_m) \quad = \quad \begin{cases} d(n, k_m) & \text{if } ch(n) = 1 \\ & \wedge ch(k_m) = 1 \; , \\ p_c \times d(n, k_m) & \text{otherwise} \end{cases} \tag{1}$$

$$\begin{aligned} & c_a(n, k_m) \\ & \quad = \min_{\tau, \mu} \begin{cases} c_l(n, k_m) & \text{if } (\mu = m \wedge \kappa = k - 1) \\ \quad + c_a(n-1, \kappa_\mu) & \vee st(n) \neq st(n-1) \\ p_t \times c_l(n, k_m) & \\ \quad + c_a(n-1, \kappa_\mu) & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where $ch(n)$ returns 1 if n is included in a chorus section, and $st(n)$ returns the number of musical sections. A higher p_c value means that the unit of chorus sections are more easily selected at a chorus section. A lower p_t value means that the selected unit has less time continuity. To minimize the accumulated cost, at the N measure, the system

[3] Since the database is reconstructed depending on the tempo of the input, the reduced dimension is not constant.

selects a unit which has minimum accumulated cost d_{min}, and then a image sequence is generated by back-tracing.

$$d_{min} = \underset{k,m}{\operatorname{argmin}} \quad c_a(N, k_m). \tag{3}$$

The interactive re-selection function is implemented so that the system chooses four different candidates for each section (Figure 4). These candidates are made from four different accumulated costs, and then four image sequences are generated by back-tracing. To expand the variety of generated image sequences, the chosen candidates are made from minimum, 1/3 minimum, 2/3 minimum, and maximum accumulated costs. This enables generation of a variational image sequence.

4.3 Model training weighted according to view counts

This paper focuses on the reuse of the MAD movies available on the web. Since there are many creators, the authoring quality of generated videos varies widely. In other words, each video will have a different level of reliability regarding the relationships between music and image. We assume that a video generated by a user having good MAD movie skills will have higher reliability and higher possibility of its reuse. Therefore, to model an appropriate image sequence to particular music, the system should introduce a weighting factor in the model training process where higher quality video will be given a greater weight.

To enable automatic judgment of the quality, we introduce the idea of using the view count of each video clip on the web as a weight since the view count reflects the video quality. Let ω be an integer weighting factor defined as follows, where V_c indicates the view count:

$$w \quad = \quad \max\left(\alpha \times \lfloor \log_{10}(V_c) + 0.5 \rfloor + \beta, 0\right). \tag{4}$$

In our current implementation, α and β are set to 2 and -7, respectively. This means, a view count of $10,000$ corresponds to $\omega = 1$, while a view count of $100,000$ corresponds to $\omega = 3$. To implement the weighted training, the number of bar-level audio/visual features (training samples) of a video clip is virtually increased by its ω (doubled by $\omega = 2$, for example) in training linear regression models.

5. IMPLEMENTATION OF DANCEREPRODUCER

In this section, we describe the dataset used and trial user comments regarding the system effectiveness.

5.1 Dataset

To generate a dance video by segmenting and concatenating from existing dance video, and to model the various relationships between music and an image sequence, the database should fulfill the following four conditions.

Condition 1) The main content of video clips is dance.

Condition 2) Video clips are similar types of MAD movies so that their mixture generated by our system can look like a consistent content.

Condition 3) Each video clip has the view count by users on the web.

Condition 4) The number of available video clips is large enough.

As content fulfilling all of the above conditions, we used mashup videos which are generated from Japanese dance simulation games full of dance scenes, “THE IDOLM@STER” and “THE IDOLM@STER LIVE FOR YOU!”[4]. In addition, we also used dance videos which are generated using *MikuMikuDance (MMD)*[5] that is a 3-dimentional human motion synthesizer for dance performance. Both videos can be found on a video sharing service *NicoNicoDouga*[6]. To construct a database, we gathered 100 of these mashup video clips and 100 of these MMD video clips, all of which had the view count of over 10,000 on the NicoNicoDouga.

5.2 Trial usage and introspective comments

Many videos generated by DanceReProducer were synchronized regarding rhythm and impression between the music and image sequence. This suggests that the system can be effective and the modeling is appropriate.

Trial users of the system offered comments, especially regarding the effectiveness of the interactive re-selection function. A typical comment was that “the function was useful and effective”; however, in contrast, another user commented that “occasionally there was no appropriate candidate”.

Some comments were on ways to improve the system performance. One user, who had no experience in MAD movie generation, said it would be useful to have “more candidates for the image sequence”. Another comment, from a user who had MAD movie experience, was that the system needed an “adjustment function for the bar and boundary of the musical section ”.

6. CONCLUSION

DanceReProducer is a dance video authoring system that can automatically generate dance video appropriate to music by reusing existing dance video sequences. Trial usage of the system has shown that it is a useful tool for users with little knowledge or experience in MAD movie generation[7]. Although dance video content is currently supported in our implementation, our approach has capability to utilize for any other music video clips.

One benefit of DanceReProducer is that a user does not need to engage in time-consuming manual generation. Moreover, the “reuse” approach described in this paper is novel in that it allows the use of ever-increasing user-generated content on the web. We expect the expansion of mashup content (nth generation content), and its supporting systems, to create an opportunity for a new form of entertainment. Remaining issues, such as a quantitative

[4] http://www.bandainamcogames.co.jp/cs/list/idolmaster/

[5] http://www.geocities.jp/higuchuu4/index_e.htm

[6] http://www.nicovideo.jp/

[7] Demonstration video clips generated by our system are available at http://staff.aist.go.jp/t.nakano/DanceReProducer/

evaluation of this system, feature extraction for dance motion in detail (like the body motion detection[8]), and an interface that can adjust measure or section boundaries, will be topics covered in our future work.

Acknowledgments

We thank Yuki Hasegawa and Tatsunori Hirai for their help.

7. REFERENCES

[1] T. X. Fujisawa, M. Tani, N. Nagata, and H. Katayose, "Music mood visualization based on quantitative model of chord perception," in *Journal of Information Processing Society of Japan*, vol. 50, no. 3, 2009, pp. 1133–1138. (in Japanese)

[2] C. Laurier and P. Herrera, "Mood Cloud : A realtime music mood visualization tool," in *Proc. of the 2008 Computers in Music Modeling and Retrieval Conference*, 2008, pp. 163–167.

[3] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," in *Journal of New Music Research*, vol. 30, no. 2, 2001, pp. 159–171.

[4] T. Shiratori and K. Ikeuchi, "Synthesis of dance performance based on analyses of human motion and music," in *IPSJ Transactions on Computer Vision and Image Media*, vol. 1, no. 1, 2008, pp. 34–47.

[5] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatically Converting Photographic Series into Video," in *Proc. of the 12th annual ACM international conference on Multimedia*, 2004, pp. 708–715.

[6] R. Cai, L. Zhang, F. Jing, W. Lai, and W.-Y. Ma, "Automated Music Video Generation using WEB Image Resource," in *Proc. of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2007)*, 2007, pp. II–737–II740.

[7] J. Foote, M. Cooperand, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proc. of the tenth ACM international conference on Multimedia*, 2002, pp. 553–560.

[8] X.-S. Hua, L. Lu, and H.-J. Zhang, "Automatic music video generation based on temporal pattern analysis," in *Proc. of the 12th annual ACM international conference on Multimedia*, 2004, pp. 472–475.

[9] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music," in *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, 2006, pp. 1784–1794.

[10] O. Gillet, S. Essid, and G. Richard, "On the correlation of audio and visual segmentations of music videos," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 2, 2007, pp. 347–355.

[11] M. Nishiyama, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno, "A Computational Model of Congruency between Music and Video in Multimedia Content," in *IPSJ SIG Technical Reports 2007-MUS-069*, vol. 2007, no. 15, 2007, pp. 111–118. (in Japanese)

[12] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in *IEEE Trans. on Speech and Audio Processing*, vol. 17, no. 2, 2002, pp. 293–302.

[8] EyesWeb: http://www.infomus.org/EywMain.html