DJ-MVP : An Automatic Music Video Producer

Jianyu Fan, William Li, Jim Bizzocchi, Justine Bizzocchi, Philippe Pasquier
School of Interactive Art and Technology, Simon Fraser University
Vancouver, Canada
jianyuf, dla135, jimbiz, justine, pasquier@sfu.ca

ABSTRACT

A music video (MV) is a videotaped performance of a recorded popular song, usually accompanied by dancing and visual images. In this paper, we outline the design of a generative music video system, which automatically generates an audio-video mashup for a given target audio track. The system performs segmentation for the given target song based on beat detection. Next, according to audio similarity analysis and color heuristic selection methods, we obtain generated video segments. Then, these video segments are truncated to match the length of audio segments and are concatenated as the final music video. An evaluation of our system has shown that users are receptive to this novel presentation of music videos and are interested in future developments.

INTRODUCTION

Music videos provide rich visual representation to songs, and are commonly used by artists to market hit songs [1]. They can be created manually using readily available software such as Adobe Premiere. For music videos, images should align to the audio to enhance its acoustical aesthetic. However, manually creating a large number of videos is costly and time-consuming. Computational models can be used to generate music videos [1-5]. Automatic generation increases efficiency in the production of such media.

Our automatic music video producer (DJ-MVP) uses concatenative synthesis method to generate audio-visual mashups for the style of music video built upon existing music videos. Our system allows users control over the video corpus selection, providing a personalized experience. For example, users can select a Western or Eastern music video corpus, or both corpora to generate music video for a target song. The system automatically segments songs based on beat detection. Then, the corresponding videos are segmented using the same cutting points. When given an input target song, the system will segment the target in the same way by doing beat detection. The system acts as a video editor, arranging video segments based on audio similarity ranking, video similarity ranking and heuristic rules instantiated in the system. The target audio track is presented along with the new visuals. Studies show that synchronizing video and audio events enhances audiences' perception of both, and improves the effectiveness of a film clip [6]. The combination of selected video segments and target audio track ensures that the resulting audio-visual output has a sense of synchronization and contrast. Demonstration videos are online.[1]

Figure 1 illustrates the workflow of the DJ-MVP system, which consists of six major steps, including preliminary audio-video segmentation for the corpus, audio segmentation for targets, audio similarity analysis, heuristic selection methods and concatenation. The rest of the paper is organized as follows. First, we present related works. Second, the design of DJ-MVP is described. Third, we illustrate our evaluation of the system. Finally, we give the conclusion and discuss the future work.

RELATED WORKS

Previous approaches to generating music videos can be generally organized by two tasks. The first task is to analyze and process amateur home videos to mix with a target song. The second task is mixing up the target music with dance animation to generate a video remix. Here, we introduce 5 projects that are similar to ours, but differ in sources and outputs

Foote et al. designed a semi-automatic system that performs selection and alignment of home video segments to target music in order to generate a complete music video [1]. Given a target song, their system achieves automatic audio segmentation by applying audio self-similarity analysis and measuring a novelty score, namely the level of audio change. Source home videos are analyzed by a suitability function, which is defined based on camera motion and exposure. Therefore, only high-quality regions in the source video will be selected in the final edit. Their system uses a moving average window along the timeline of the source home video to generate unsuitability scores and identifies video segment boundaries as peaks in the moving average window. The number of segments is adjustable when

ACE2016, November 09-12, 2016, Osaka, Japan
© 2016 ACM. ISBN 978-1-4503-4773-0/16/11$15.00
DOI: http://dx.doi.org/10.1145/3001773.3001782

[1] https://vimeo.com/channels/djmvp

controlling the moving average window size. To align the video and audio tracks, the system truncates, combines, and discards video segments, so that the final set of clip boundaries align exactly with significant audio changes. They conducted a subjective study. Informal judgments showed the system produces reasonably convincing music videos.

Hua et al. created an automatic music video generation system [2]. The system detects repetitive patterns in a given song based on the audio similarity analysis. A raw home video is used as a source video. First, low-quality shots are filtered out based on camera motion speed and color entropy. Then the source video is segmented according to a shot similarity curve, which is computed based on content similarity of different shots within the source video. Lastly, they matched the tempos of the music repetitive patterns with the level of motion intensities in the corresponding video scenes to connect the music and the video. To maintain diversity and repetition at the same time, the system uses different portions of the same shot to match different occurrence of the same music pattern. The duration of video portions is selected based on the duration of the music segment to achieve alignment. Hua et al. compared their system with two related systems by subjective evaluation, one is the system of Foote et al. and the other one is a video summarization system [1, 7]. The system of Hua et al. performed better than the system of Foote et al. in the subjective evaluation.

A third project was performed by Yoon et al, who proposed a music video generation system based on multi-level feature-based segmentation [3]. Like Foote et al and Hua et al., their goal is to process home videos to mix with a target song. After using contour shape features to measure similarities between shapes, Yoon et al. then segmented the video based on extreme changes of shape features between frames. The music segmentation is done by using the novelty scoring method introduced by Footed et al. [1] Video segments and music segments are represented by three-dimensional feature vectors containing the velocity, the brightness, and the extreme boundary features, which are defined separately for music and videos. For the video, Yoon et al. defined velocity as a displacement over time derived from the camera or object movement, and brightness is a measure of the visual impact of luminance in each frame. They believe the onset of the audio track is usually set by percussion instruments, which dominate the amplitude of the signal. Therefore, for music, their system estimates the velocity from the root mean square of the amplitude of the signal. Next, Yoon et al. extracted the brightness feature using the spectral centroid [3]. Their system assembles a synchronized music video by matching segments based on these feature vectors. Finally, the durations of video and music are compared to avoid the need for excessive time-warping. A subjective experiment was conducted comparing with the results of Foote et al., which indicated that participants like their system better. [1]

Cai et al. built an automatic music video generator using semantic information from the lyrics to matching the images from the Internet and generate a slide show music video [4]. Based on semantic information from the lyric, their system sends different queries to image search services, such as Google Image and Flickr. To improve the coherence between image and music, the system ranks images based on color features and content features, music affect features and rhythm features, as well as music meta-data features such as song titles and artists. These images are aligned based on onset positions extracted from the music.

Nakano et al. developed the DanceReProducer, which automatically generates a Japanese animation video clip for target dance music [5]. Examples can be found on the website[2]. The segmentation of music and video is done by onset detection. They use both frame-level feature vectors and bar level-feature vectors to represent music and video. The audio frame level features include filter bank output (4 dims.), spectral flux (1 dim.), zero-crossing rate (1 dim.) and 12th order MFCCs. The video features are optical flow, hue, saturation, and brightness. The bar-level feature is an integration of the frame features in each bar using resampled data points for the time axis, and applying Discrete Cosine Transform for each dimension. To find the relationship between music and video, Nakano et al. first apply k-means clustering to feature vectors, which includes bar-level audio features and bar-level visual features in the database. After obtaining multiple clusters of audio segments, for each cluster, they trained a linear regression model to predict bar-level visual features based on bar-level audio features. Among previous works, this system is the only one that took advantage of existing music videos for synchronization.

The works mentioned above extract and map between video features and audio features. Foote et al., Hua et al., and Yoon et al. used home videos as the source video to generate music video. The system of Nakano et al. generates video for dance animation. On the other hand, DJ-MVP is designed for generating video remixes by recombining target music and existing music video segments. Our goal is to automatically generate music videos that look like professionally-made music videos. Moreover, most of the systems did not take context information such as color contrast and coherence between video segments into consideration. DJ-MVP uses color features of adjacent video segments for video sequencing. In addition, DJ-MVP not only generates a video for a target, but also generate an audio-video mash up for a given target song. We will illustrate these in the following section.

SYSTEM DESIGN

Figure 1 shows the design of DJ-MVP. For building the corpus, audio tracks and their corresponding videos are

[2] https://staff.aist.go.jp/t.nakano/DanceReProducer/

segmented based on the beat detection. Given a target song, the system segments it in the same way as above. It then conducts an audio similarity analysis between the target audio segment and segments in a corpus and returns a list of candidate audio segments. This maintains a strong connection between audio and video because they originally came from the same music videos. We take advantage of the color features of video segments and build video sequences that have color contrast or color coherence. We provide heuristic selection methods to enable users to control the number of repetition a particular segment appears so that the system can keep the consistency of the visual content toward the same audio segment and maintain the diversity of video segments. This is explained in the section of "Heuristic Selection Module". The duration of video segments and the duration of the target audio segment are compared in order to truncate the longer segments. After truncation and concatenation of video segments, the system outputs the generated music video.

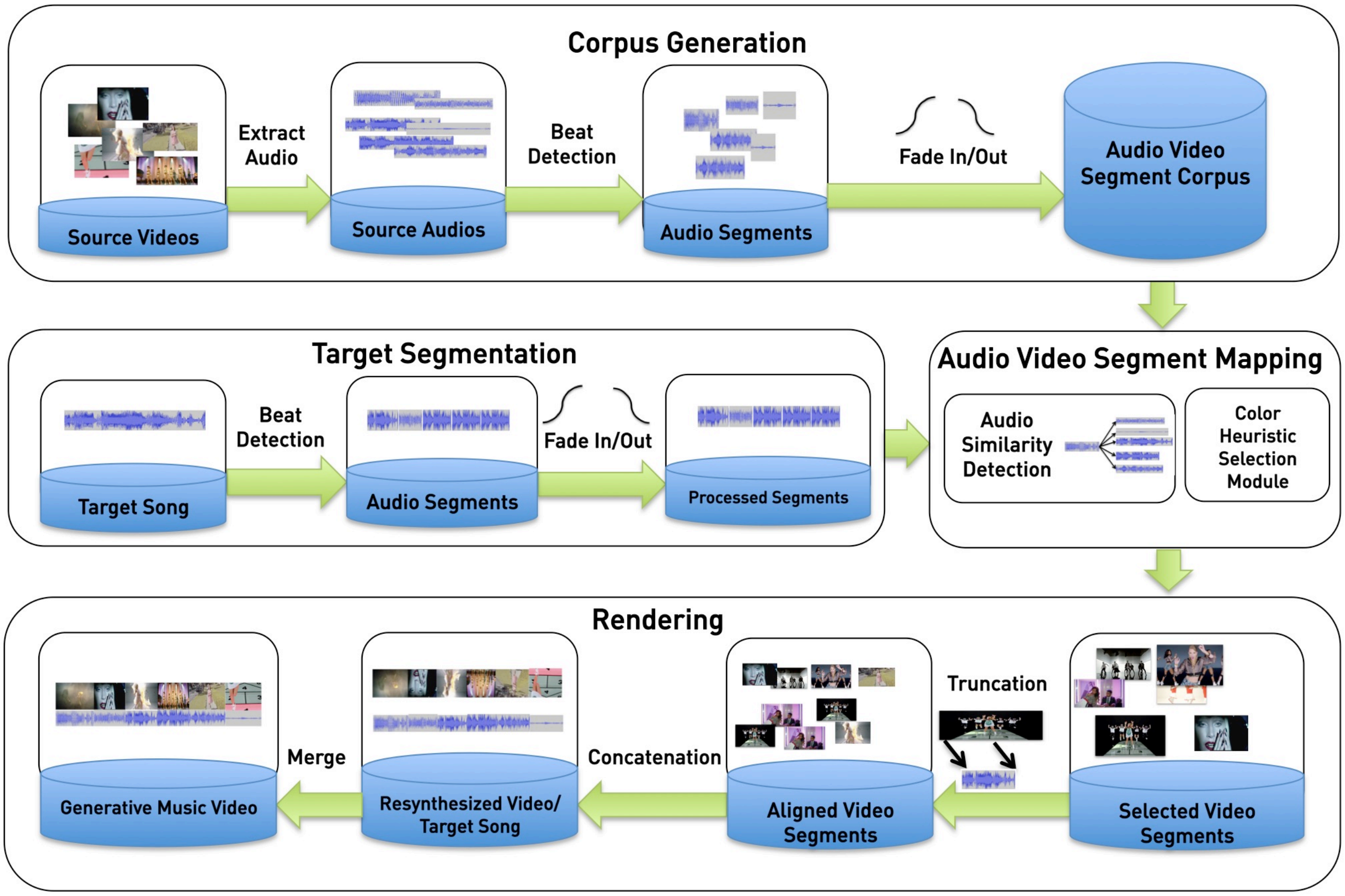

Figure 1. Overview of DJ-MVP.

Source

We have collected dance music videos of popular bands for curating the source MV corpus[3]. We curated three corpora: an Eastern music video corpus, a Western music video corpus and a corpus of select target songs.

The Eastern corpus has 35 songs and 4187 audio/video segments in total. We mainly selected songs from China, Korea, and India. The Western corpus consists of 72 songs containing 9103 audio/video segments. Songs we selected are from U.S, England, and Canada. We chose nine songs as the corpus of select target songs. Three of them are from Asia, three of them are from North America, and the other three are techno, which is a form of electronic dance music.

[3] https://www.dropbox.com/sh/3b3ljqd6u6a7dv4/AACINN44Xr0lGMBwHg4nWWAUa?dl=0

For each video we selected, we assume that the dance motions and the music are synchronized. The source audio tracks are extracted from the corresponding source videos. Our audio tracks are in WAV format at a sample rate of 44100 Hz.

Beat Detection

Segmentation of songs is a crucial step in automatic music video generation. If the segmentation process is unreliable, the perceived quality of the final music video will suffer.

Comparing the segmentation based on beat detection–with the novelty score method used in previous systems, we believe that segmenting based on beat detection will provide better synchronization between target music and resynthesized video in general [1-3]. It detects a beat only when the instant sound energy is superior to the average local sound energy. The set of beats should reflect a locally constant inter-beat-interval. We employ the beat detection algorithm proposed by Partin [8].

Shorter segments and faster cuttings will cause stronger visual impact, whereas longer shots and slower cuttings lead to better synchronization of dance movements and music. To test the influence of duration of each segment on final rendered music videos, we set up two scales for the segmentation. For the first scale, the segmentation is done on each beat position, which resulted in 128 segments for a one-minute song that is 128 bpm. For the second scale, we provide a threshold, which limits the shortest duration of each segment, so that when the duration between two onsets is smaller than the threshold, the subsequent onset will be ignored. For example, when we set the threshold to be 2 seconds, for a song that is 128 bpm, each segment can contain 4 beats. For a one-minute song, which is 128 bpm, there will be around 30 segments[4].

The algorithm for beat detection is described below. Suppose there are two channels, a and b, the instant energy is shown in Eqn (1)

$$e_{stereo} = \sum_{k=i_0}^{i_0+1024} a[k]^2 + b[k]^2 \qquad (1)$$

For every 1024 samples, we computed the instant energy using Eqn(1). The sound energy history buffer B correspond to approximately 1 second of music, which contains 43 energy values, namely 44032 samples (groups of 1024). $B[0]$ contains the newest energy computed on the newest 1024 samples. $B[42]$ is the oldest energy computed on the oldest 1024 samples. The average local energy E and the variance of the energies in B are shown in Eqn (2) and Eqn (3).

$$E = \frac{1}{43} * \sum_{i=0}^{43} B[i] \qquad (2)$$

$$V = \frac{1}{43} \times \sum_{i=0}^{43} (B[i] - E)^2 \qquad (3)$$

Using the variance (V) of the energies, we computed the sensitivity of the beat detection. The threshold of a “beat” is $C \times E$. Eqn (4) shows the representation of C.

$$C = (-0.005714 * V) + \ 1.51425871 \qquad (4)$$

After shifting the sound energy history buffer B of 1 index to the right. We san save the new energy value and remove the oldest. Then we compared the newest instant energy e_{stereo} with the threshold. Figure 2 is the result of using the onset detection algorithm for the two songs. We added 0.02-second fade-in and 0.02-second fade-out to the audio segments after segmenting the source audio based on the beat detection.

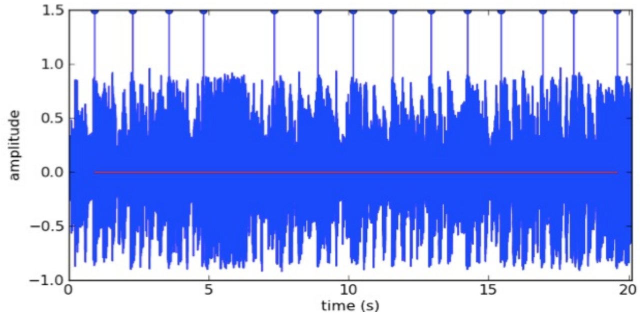

Figure 2. Beat Detection of part of the song “Fighter”

Audio Similarity Detection Module

For audio similarity analysis, we chose the Musly library written in C++, which is built on the work of Mandel et al. [9]. It uses the Kullback Leibler divergence (KL divergence) to compute the distance between audio excerpts. All of their features are based on Mel-frequency Cepstral Coefficients (MFCCs). MFCCs are common features in speech recognition systems, recognizing people from their voices [10]. They have also been used in timbre recognition [11]. Mel-frequency is based upon the human auditory system, which does not have a linear perception of sound and maps different frequencies to perceived pitches. Mandel et al.’s model only considers timbral features, which do not contain any temporal aspect of the music, only its short-time spectral characteristics. Mandel et al. used the bag-of-frame approach to model the song, which considers that frames representing a signal have possibly different values, and the aggregation of the frames provides a more effective representation than a singular frame [15].

[4] https://vimeo.com/181667680

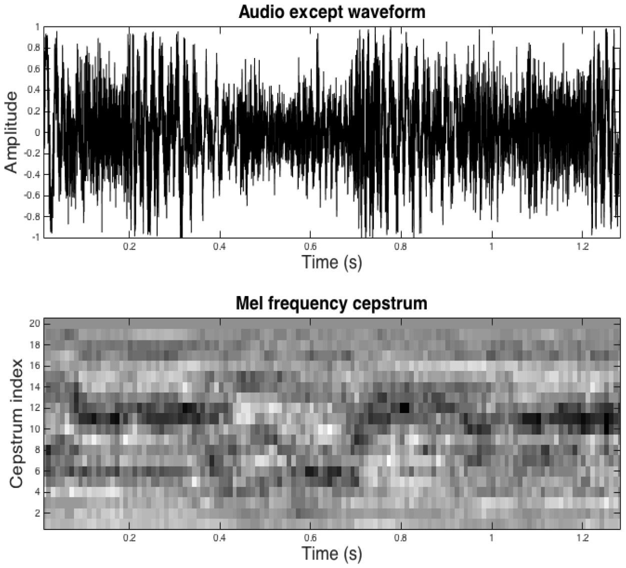

Figure 3. MFCC of the segment 126 of the song "Danger"

Mandel et al. extract 20 coefficient MFCCs per frame and use the mean and covariance of the MFCCs over the duration of each song to describe the Gaussian distribution with the maximum likelihood of generating those points under the bag-of-frame approach. After obtaining a single Gaussian model of each song, they measure the distance between songs using the KL divergence. For two distributions, $a(x)$ and $b(x)$, the KL divergence is defined:

$$KL(a||b) = \int a(x) log \frac{a(x)}{b(x)} dx \qquad (4)$$

For each target audio excerpt, our system selects 50 most similar audio excerpts in our corpora as candidates.

Heuristic Selection Module

To maintain a certain level of diversity in the video content and provide more controls to users, we added three heuristic methods:

- Segment Diversity Heuristic Method: A segment is blacklisted after 4-appearances.
- Song Diversity Heuristic Method: A song is blacklisted after 12 appearances.
- Color Heuristic Method: The next segment is the closest (color coherence) or the furthest (color contrast) in HSV color space to the previous ones among the 50 closest segments to the given target audio segment.

The number of appearances and the number of closest/furthest segments can be set by users. The distance between two video segments in the HSV color space is computed through a video similarity detection model, which adopts the histogram intersection algorithm [13]. Let a and b represent two color histograms. The intersection of histograms a and b is given by:

$$d(a,b) = \frac{\sum_H \sum_S \sum_V \min\left(a(h,s,v), b(h,s,v)\right)}{\min\left(|a|,|b|\right)} \qquad (8)$$

We use 32 bins for each dimension: hue, saturation, and value. The value of each bin is normalized to be between 0 and 1. Higher histogram intersection indicates higher similarity between video segments in the HSV color space. Figure 4 shows the hue, saturation, and brightness histogram of two video segments. These two segments are selected to be concatenated because of color coherence.

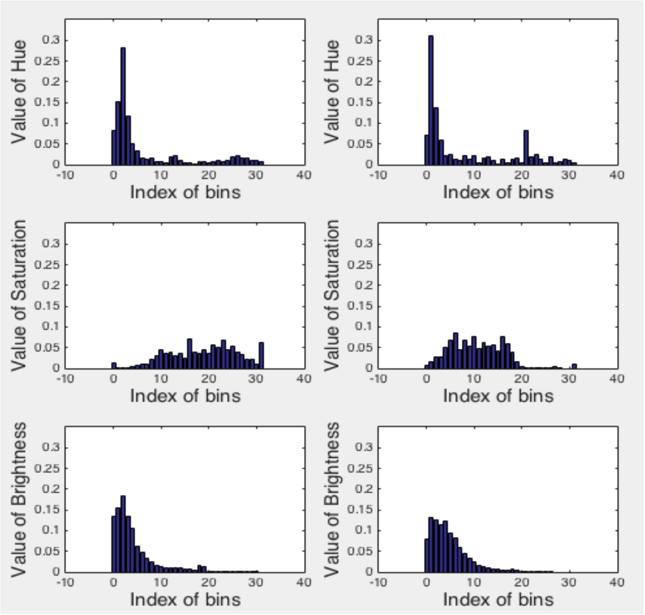

Figure 4. Color Histograms of segment 47 of the Song "1234" (three Figures on the left side), and segment 111 of the Song "DJ Got Us Falling In Love" (three Figures on the right side). The histogram intersection value between these two video segments is 0.56.

Weight Control

Users can control the weight of audio similarity ranking and video similarity ranking. Figure 5 explains this process. A target audio segment is on the left side. The system first selects a list of 50 candidates based on audio similarity detection, which is shown in the middle column of Figure 5. The rankings in parentheses represent the rankings of the level of audio similarity. On the right side, there is a list of video segments. Each video segment corresponds to the audio segment on its left.

If the user puts more weight on the audio similarity ranking, the system selects the top ranked audio segment without considering the similarity rankings of the corresponding video segments. On the other hand, if the user puts more weight on the color-coherence or color-contrast, the system selects the top ranked video segment within these 50 segments, whose corresponding audio is also among the top 50 in similarity to the target audio.

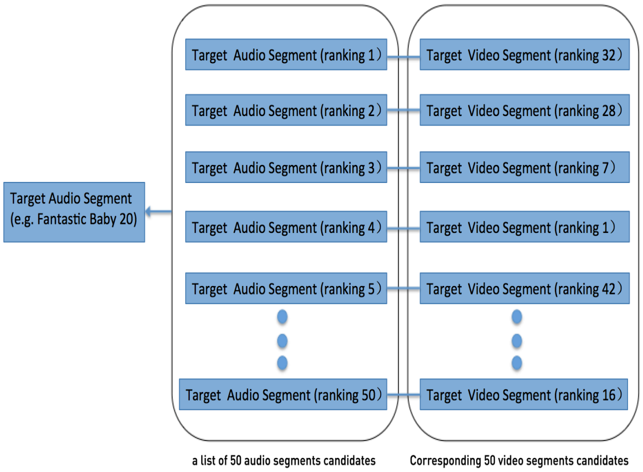
Target Audio Segment
(e.g. Fantastic Baby 20)
Target Audio Segment (ranking 1)
Target Audio Segment (ranking 2)
Target Audio Segment (ranking 3)
Target Audio Segment (ranking 4)
Target Audio Segment (ranking 5)
Target Audio Segment (ranking 50)
Target Video Segment (ranking 32)
Target Video Segment (ranking 28)
Target Video Segment (ranking 7)
Target Video Segment (ranking 1)
Target Video Segment (ranking 42)
Target Video Segment (ranking 16)
a list of 50 audio segments candidates
Corresponding 50 video segments candidates

Figure 5. Weight Control between Audio Similarity and Color Heuristics

Truncation

For each target segment, we only use video segments that are longer than the target. We apply truncation to each video segment so that its length is the same as that of the corresponding given target audio segment. Figure 6 shows how to truncate the video segments and the concatenation. An example is shown at: https://vimeo.com/166312586.

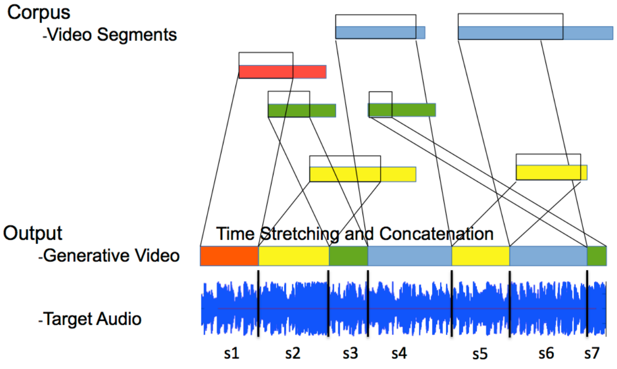

Figure 6. Mechanism of Selecting Video Segments from the Video Corpus to Generate the Target Video

Audio video Remix

We also added an audio-video remix function to the system. The function enables the system to generate new music based on input targets. Instead of merging users' input audio track, the system will concatenate selected audio segments and merge this rendered audio track with concatenated video segments. This method creates a type of experimental music and music video. An example can be found at: https://vimeo.com/166312522.

Data Moshing Effect

In video compression, three types of frames are stored in a video file to contain enough information about itself. I-Frames, known as keyframes, are frames that store complete images. If there is a drastic change in the video, there must be a keyframe stored. In DJ-MVP, keyframes are usually created in the transition between video segments. P-frames, which are called predicted frames, contain changes in the image from the previous frame. B-frames are bidirectional predicted frames. We used a data mashing method, which destructs the original video file by destructing I-Frames. I-Frames. When we remove I-Frame, it results in the previous video clip stay on top of the motion of the next video clip. Figure 7 shows the effect of data moshing.

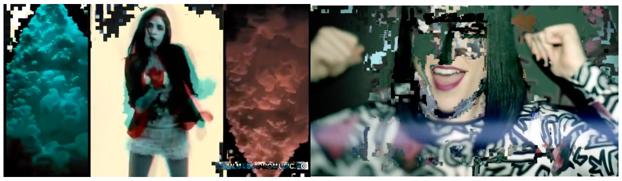

Figure 7. After Adding Data Moshing Effect to the Generative Music Video

Implementation

We used FFMpeg, a command line utility, to extract source audio from source videos, to do time stretching of segments and to concatenate audio and video segments [14]. We used the Musly software package to do audio similarity detection. The Beads library is used for adding fades on segments [16]. HSV feature extraction is done by Matlab. Other parts were implemented in Java.

EVALUATION

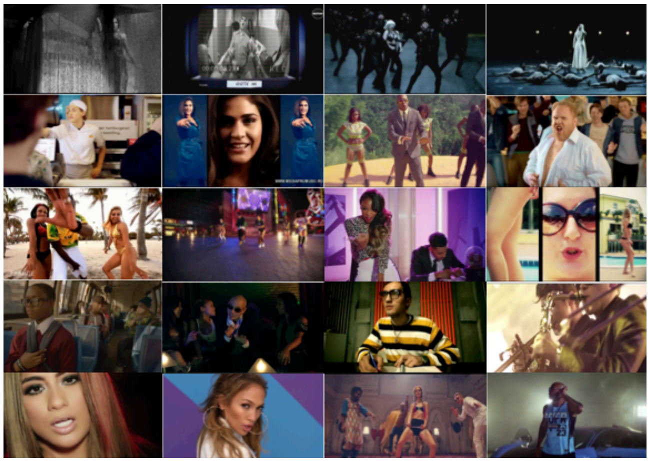

Figure 8. An example of key frames extracted from several automatically selected video segments for generating the music video for the target song "Fantastic Baby".

To better evaluate the system, we conducted a qualitative study and interviewed 10 users to have their feedbacks. Users are graduate students at Simon Fraser University. Because our system differs in sources and outputs, it is not possible to directly compare the performance of our system to the previous state of the art systems.

Ten users watched three of our demonstrations, including "Tiao Jin Lai", "Fantastic Baby", and "Style Machine", which are shown at https://vimeo.com/channels/djmvp. When we asked users to describe their general feelings about watching these generative music videos and audio video remixes, all of the users said they believe the audio tracks and generative music videos sync very well, and the style of videos match the style of the music. Three of the users pointed out that the third music video, the audio-video remix, has better synchronization between music and video; but the first two, which are generative music videos, have better flow and coherency in both music and video. Three users claimed the generative music video for "Fantastic Baby" was amazing, and they would even believe it to be an actual professionally-edited music video. What is surprising is that two users mentioned that the audio-video remix reminded them of a novel video editing method that is popular among youth China called "Gui Xu Video", which contains high-level synchronization between video and music and uses repetitions of videos and audios to emphasize a perspective that is usually ironic or absurd. Since we did not know of this specific video editing method, it was a surprise to us and also gave us a new potential direction to further develop the system. Although users enjoyed our generative music videos, they pointed out several aspects in which DJ-MVP can be improved. Six users mentioned it was distracting that the shape of actors' mouth does not match the lyrics. Four users noticed that the level of synchronization dropped when the rhythm was slower or when there was no obvious beat. This is not surprising since our plan to achieve synchronization is based on beat detection. Two users felt that it was likely to have one music video contain multiple singers, which is not the case in professional music videos. Moreover, generative music video does not guarantee the qualities of narratives.

Regarding the difference they feel when watching generated versus professional music videos, five users think that music videos they have seen before often have much longer shots. These generated ones are more dynamic and provide heavier visual impact, which gives different experience. We summarize the users' positive and negative feedback in Table 1.

Positive Feedback	Negative Feedback
Strong synchronization	Shape of actors' mouth does not match the lyrics
Good flow and coherency	When there is no obvious beat, synchronization is not good
Strong visual impact	Multiple singers show in one generative music video
Similar to "Gui Xu Video"	There is no narrative

Table 1. Summary of Users' Feedback

The feedback shows the promising potential of DJ-MVP. Based on those suggestions for improvement, we plan to update the current DJ-MVP in the future by utilizing high-level features of both music and video, to make outputs more similar to professional music videos.

CONCLUSION AND FUTURE WORK

In this paper, we have presented the design of DJ-MVP, a new generative music video system. Qualitative evaluations of DJ-MVP showed positive results, but also revealed some problems and challenges. For future work, we plan to implement music structure estimation on the given target song so that we can adopt different heuristic selection methods on the different part of the song. Then, we plan to use Audio Oracle, an algorithm for fast indexing of audio data, to reshuffle repeated sub-clips to produce variation from a music recording. Finally, we plan to extract harmony information from the audio segment as another heuristic selection method.

ACKNOWLEDGMENTS

We would like to acknowledge the Social Sciences and Humanities Research Council of Canada for their ongoing financial support. And we would like to thank the reviewers, who through their thoughtful comments have been assisting with this publication.

REFERENCES

1. Jonathan Foote, Matthew Cooper, and Andreas Girgensohn. 2002. Creating music videos using automatic media analysis. In *Proceedings of the ACM Multimedia* (MM'02), 553-560.

2. Xian-Sheng Hua and Hong Jiang Zhang. 2004. Automatic music video generation based on temporal pattern analysis. In *Proceedings of* the *ACM Multimedia*, (MM'04), 472-475.

3. Jong Chul Yoon, In-Kwon Lee, Siwoo Byun. 2009. Automated music video generation using multi-level feature-based segmentation. *Multimedia Tools and Applications.* 41, 2, (January 2009), 197-214.

4. Rui Cai, Lei Zhang, Feng Jing, Wei Lai, and Wei-Ying Ma. 2007. Automated music video generation using web image resource. In *Proceedings of* the *IEEE International Conference on Acoustics, Speech and Signal Processing*, (ICASSP'07), 737-740.

5. Tomoyasu Nakano, Sora Murofushi, Masataka Goto and Shigeo Morishima. 2011. Dancereproducer: An automatic mashup music video generation system by reusing dance video clips on the web. In *Proceedings of Sound and Music Computing Conference*, (SMC'11), 183-189.

6. Scott D. Lipscomb. 1997. Perceptual measures of visual and auditory cues in film music. *The Journal of the Acoustical Society of America*. 101, 5, (June 1997), 3190.

7. Yu Fei Ma, Lie Lu, Hong Jiang Zhang and Mingjing Li. 2002. An attention model for video summarization. In *Proceedings of ACM Multimedia*, (MM'02), 533-542.

8. Frederic Patin. 2003. Accessible Online Tutorial. Retrieved March 22, 2016, from

http://www.flipcode.com/misc/OnsetDetectionAlgorithms.pdf

9. Michael I. Mandel and Daniel P.W. Ellis. 2005. Song-level features and support vector machines for music classification. In P*roceedings of* I*nternational. Symposium. Music Information Retrieval*, (ISMIR'05), 594-599.

10. Alan V. Oppenheim. 1969. Speech analysis-synthesis system based on homomorphic filtering. *Journal of the Acoustical Society of America*, 45, 2, (February 1969), 458-465.

11. Beth Logan. 2000. Mel frequency cepstral coefficients for music modeling. In *Proceedings of* International Symposium on Music Information Retrieval, (ISMIR'00).

12. Shlomo Dubnov, G´erard Assayagm, and Arshia Cont. 2011. Audio oracle analysis of musical information rate. In *Proceedings of* IEEE International Conference on Semantic Computing, (ICSC'11), 567-571.

13. Sangoh Jeong. 2001. Histogram-Based Color Image Retrieval. Accessible Online Report, Retrieved March 22, 2016, from

https://ece.uwaterloo.ca/~nnikvand/Coderep/ColorHist/Histogram-Based%20Color%20Image%20Retrieval.pdf

14. ffmpeg. FFmpeg website, Retrieved July 27, 2015 from https://www.ffmpeg.org/

15. Jean, Julien Aucouturier and Boris, Defreville. 2007. Sounds like a park: a computational technique to recognize soundscapes holistically, without source identification. In the *Proceedings of International Congress on Acoustics,* (ICA'07).

16. Beads. Open source audio processing library. Retrieved August 27, 2015 from http://www.beadsproject.net/