MUSIC PASTE: CONCATENATING MUSIC CLIPS BASED ON CHROMA AND RHYTHM FEATURES

Heng-Yi Lin† Yin-Tzu Lin‡ Ming-Chun Tien‡ Ja-Ling Wu†‡
National Taiwan University
†Department of Computer Science and Information Engineering
‡Graduate Institute of Networking and Multimedia
{waquey,known,trimy,wjl}@cmlab.csie.ntu.edu.tw

ABSTRACT

In this paper, we provide a tool for automatically choosing appropriate music clips from a given audio collection and properly combining the chosen clips. To seamlessly concatenate two different music clips without causing any audible defect is really a hard nut to crack. Borrowing the idea from the musical dice game and the DJ's strategy and considering psychoacoustics, we employ the currently available audio analysis and editing techniques to paste music sounded as pleasant as possible. Besides, we conduct subjective evaluations on the correlation between pasting methods and the auditory quality of combined clips. The experimental results show that the automatically generated music pastes are acceptable to most of the evaluators. The proposed system can be used to generate lengthened or shortened background music and dancing suite, which is useful for some audio-assisted multimedia applications.

1. INTRODUCTION AND MOTIVATION

Nowadays, more and more music lovers prefer to create their own music from the existing music audio collections, for the purpose of generating background music or dancing suite with specific length or composing a new song with all the favorite parts from different songs. However, they often confront difficulties in reaching a desirable result. The main problem lies in how to choose appropriate music clips from a large database and find out proper connecting-positions among these chosen clips. To our big surprise, studies on the relationship between the "hearing quality" and the "connecting-positions" in music combining has been long ignored. Conventionally, professional users would rely on their music sense and a few music theories to choose the clips and the connecting-positions, but the editing process is still try-and-error. As the amount of tasks increases, the process becomes time-consuming and labor-intensive. Therefore, the goal of this paper is two-fold: **(i)** providing a tool for automatically choosing appropriate music clips from given audio collections and combining the chosen clips as euphonious as possible, and **(ii)** conducting several experiments and investigating the relationship between pasting methods and the corresponding auditory quality. The ultimately combined music is named as "music paste" because it is just like the concept of pasting. The terminology "euphonious" is defined as follows: **(i)** listeners do not notice the transitions in the music paste, or **(ii)** listeners do notice the transition but they do not perceive the exact connecting-positions or the transitions sound pleasant to them.

2. RELATED WORK

2.1 Combine Music in Symbolic Domain

Combining two music clips in symbolic domain has been early studied. In the European classical era, preeminent composers developed a kind of musical dice game called Musikalische Würfelspiele [1] . Composers composed numbers of music clips for each measure in advance. While playing the game, players throw a dice to select the predetermined music clips. The action is performed for every measure. The generated music piece would not be strange because the music clip candidates for the same measure usually consist of the same chord or similar dominant tones. Based on this idea, Cope [2] conducted numerous experiments and developed a music-generating system. In the system, music clips of master composers have been analyzed and recombined to generate a new master style music piece.

The advantage of combining music clips in symbolic domain is that it causes less artifacts in auditory aspect. It is simple to transpose the midi clips to the same scale and directly combine two midi clips without causing artifacts while artifacts are usually inevitable in combined audio clips. However, the approaches in symbolic domain are not easy to be applied in audio domain due to the complication of polyphonic audio files. Moreover, current state of the art music transcription and separation techniques are not accurate enough to extract all the musical notes from polyphonic clips. Thus, the most commonly applicable editing operations in audio domain are only tempo change, remix, and concatenation.

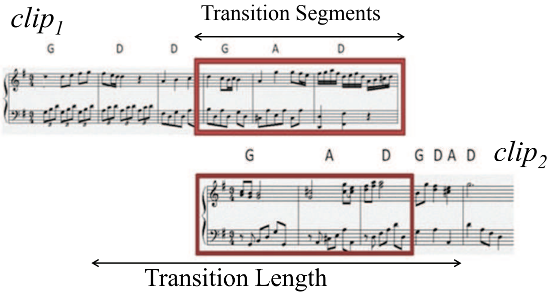

Figure 1. An example of pasting at the measures with the same chords.

2.2 Combine Music in Waveform Domain

Conventionally, what a DJ does can be treated as a human version of music-combining system. DJs often have talents for combining music clips appropriately. They can obtain hidden messages from the music clips by hearing even without the score information. In addition to choosing proper clips and connecting-positions, they also change the tempi around the connecting-positions of music clips to make the pasted music clips be pleasant for hearing. Based on this concept, Tristan [3, 4] proposed an automated DJ system by extracting auditory features, connecting the clips at rhythm-similar segments and aligning the beats of clips. However, the concatenated result may be discordant if these rhythm-similar segments are not pitch-similar. Thus, in this work, we adopt the chroma-based similarity measurement to solve this problem. Moreover, several useful schemes for filtering out dissimilar music clips are presented as well.

3. SYSTEM OVERVIEW

The key idea of the proposed system is as follows:

People usually anticipate the succeeding notes while listening to music [1]. Figure 1 shows an example of 2 input clips: $clip_1$, $clip_2$. Originally, each input clip fits people's expectation. To continue the expectation between the clips, we choose the most similar segments (pitch-similar and rhythm-similar, inspired by the musical dice game and automated DJ) between them as the connecting-position. Then, we can ensure that in the pasted music, from the beginning through the connecting-position to the end will all conform to people's anticipation. In this example, the last three chords of $clip_1$ are the same as the first three chords of $clip_2$. So, we connect these 2 clips by superimposing the beginning of $clip_2$ onto $clip_1$ at the position of the third last chord. We define the overlapping parts (the marked chords) as "transition segments." It will be determined by finding the most alike segments of the two combined clips. Besides, we use another term "transition length" to represent the length (in beat) we need for gradually adjusting the tempo from $clip_1$ to that of $clip_2$ if there is a discrepancy of tempi in these two clips.

The proposed system framework is illustrated in Figure 2. First, we extract all the features we need from music clips such as loudness, chroma, rhythm, and tempo. Then, we filter out dissimilar music clips by pair-wise comparisons. Next, we construct a distance matrix by chroma and rhythm features. With this matrix, we determine the transition segments and decide an appropriate pasting ordering. After that, we determine the transition lengths and adjust the tempi within them. Then, we rearrange the volume within the transition segments and synthesize all the processed music clips. For better understanding, we will firstly describe how to paste two music clips in section 4. And the music ordering and filtering schemes for dealing with more than two clips will be illustrated in section 5.

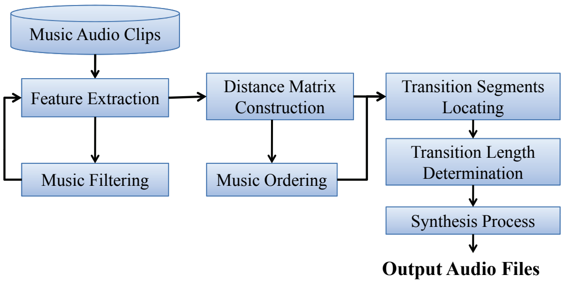

Figure 2. The block diagram of the proposed system.

4. CONCATENATION OF TWO CLIPS

In this section, we describe the process of pasting two music clips. We name these two clips as $clip_1$ and $clip_2$.

4.1 Transition Segments Locating

The common-used similarity/distance matrix [5] method is applied to measure the similarities between $clip_1$ and $clip_2$. We extract chroma-based [6] and rhythm features [7] per beat and then calculate their Euclidian distance. Thus, the smaller the values are, the more similar the segments are. Let $D_{c_{12}}(i,j)$ and $D_{r_{12}}(i,j)$ represent the chroma and rhythm distance values between $clip_1$'s i^{th} beat and $clip_2$'s j^{th} beat, respectively. That is,

$$D_{c_{12}}(i,j) = ||\vec{C}_{1i} - \vec{C}_{2j}||_2 \tag{1}$$

$$D_{r_{12}}(i,j) = ||\vec{R}_{1i} - \vec{R}_{2j}||_2 \tag{2}$$

where $\vec{C}_{1i}$ and $\vec{C}_{2i}$ denote $clip_1$'s i^{th} and $clip_2$'s j^{th} chroma vectors, respectively. And similarly, $\vec{R}_{1i}$ and $\vec{R}_{2i}$ represent the rhythm feature vectors. The two matrices $D_{c_{12}}(i,j)$ and $D_{r_{12}}(i,j)$ are linearly combined into a new matrix $D_{cr_{12}}$ (as shown in Eqn. (3)), which is the distance matrix we used for finding transition segments:

$$D_{cr_{12}}(i,j) = \alpha D_{c_{12}}(i,j) + (1-\alpha) D_{r_{12}}(i,j) \tag{3}$$

where $\alpha \in [0,1]$. We set α = 0.5 as default to equally consider the two features. Figure 3(a) depicts a distance matrix ($D_{cr_{12}}$) of 2 clips chosen from Chinese pop songs: "real man" ($clip_1$), "Let's move it" ($clip_2$). The darker the color is, the more similar the segments are. Since the transition segment of $clip_1$ and the transition segment of $clip_2$ should be similar beat by beat, we trace the values diagonally by applying overlapping window with L_{min} to L_{max} beats long and compute the average value within each window. We pick the windows with the minimum average

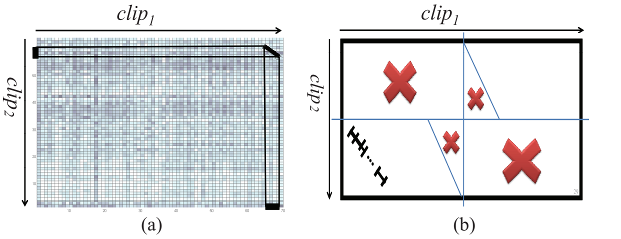

Figure 3. (a) The distance matrix of the two clips : "Real man" and "Let's move it." (b) The ignored areas.

value as the transition segments. Moreover, for the purpose of reducing the computational load and avoiding promptly switching clips, we consider only the last half of $clip_1$ and the first half of $clip_2$. Figure 3(a) shows the most similar segment we found. Figure 3(b) shows the ignored areas marked with thick crosses. The process is described by

$$[i^*, j^*, L^*] = \arg\min_{i,j,L} \frac{1}{L+1} \sum_{l=0}^{L} D_{cr_{12}}(i+l, j+l) \qquad (4)$$

where $L \in [L_{min}, L_{max}]$, $i \geq \frac{N}{2}$, $j \leq \frac{M}{2}$, N and M are the total beat number of $clip_1$ and $clip_2$, respectively.

4.2 Transition Length Determination

In musical theory, tempo is defined as the speed of a given piece [8] , usually measured by the number of beats per minute (BPM). The tempo value at the i^{th} beat ($T(i)$) can be calculated as follows:

$$T(i) = \frac{60}{beat_{i+1} - beat_i} \qquad (5)$$

where $beat_i$ and $beat_{i+1}$ are the time indices (in seconds) of the i^{th} and the $(i+1)^{th}$ beats of a clip extracted from the state-of-the-art tempo tracker: beatroot [9]. In order to gradually adjust the tempi from $clip_1$ to $clip_2$, the "transition length" should be long enough to let the difference between the adjacent $T(i)$ within the transition length small enough. An example is shown in Figure 4. To change the tempi from Tempo1 to Tempo2, the changing ratio (r_c) of the adjacent tempi within K beats is

$$r_c = \sqrt[K]{\frac{Tempo2}{Tempo1}} \qquad (6)$$

By choosing a proper value of r_c, we can determine the minimum value of K. We adopt the concept of just noticeable difference [10] (JND) in the domain of psychoacoustics to determine r_c. JND is defined as the minimum difference of stimuli that people can perceive. These stimuli include loudness, tempo and pitch. According to *Weber's law*, the JND can be computed with the *Weber's Constant*. However, the *Weber's Constant* of tempo varies with changes in the environment. Thus, inspired by Thomas [11], we conduct experiments to find the JND of tempo on our music clip datasets. For quick (i.e. fast tempo) clips, we found out that the ratio of the tempi from 0.95 to 1.03 will not be perceived. For slow clips, the JND range is from 0.96 to 1.04. Nevertheless, real world music clips may contain more than one tempo, e.g. the pieces with accelerando or ritardando. Therefore, we developed Algorithm 1 to find the transition length and the target tempi T_{t1}, T_{t2}. The procedure is also illustrated in Figure 5. Then, we use phase vocoder [12] to adjust the tempi from T_1, T_2 to T_{t1}, T_{t2} , respectively. Figure 6 shows the corresponding results of two song clips: "Let's move it" and "Real man." The ratio of change appears like a linear decay because the ratios are usually very close to 1.

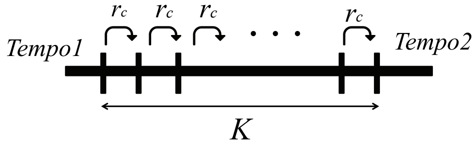

Figure 4. Diagram for changing tempi within a length K.

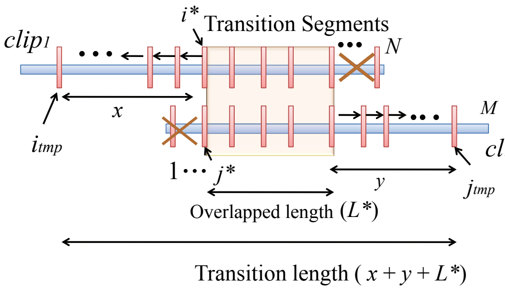

Figure 5. The sketch map of finding the transition length.

Algorithm 1

```
Input: the tempi of $clip_1$ and $clip_2$: $T_1(i)$, $T_2(j)$, for $i = 1 \ldots N, j = 1 \ldots M$
for $x = 0$ to $i^*$, $y = 0$ to $(M - L^* - j^*)$ do
    $i_{tmp} \Leftarrow (i^* - x)$
    $j_{tmp} \Leftarrow (j^* + L^* + y)$
    $r_c \Leftarrow \sqrt[x+y+L^*]{\frac{T_2(j_{tmp})}{T_1(i_{tmp})}}$
    if $r_c$ is within JND then
        break
    end if
end for
$T_{t1}(i) \Leftarrow \begin{cases} T_1(i), & \text{for } i \leq i_{tmp} \\ T_1(i_{tmp}) \times r_c^{(i-i_{tmp})}, & \text{otherwise.} \end{cases}$
$T_{t2}(j) \Leftarrow \begin{cases} T_2(j), & \text{for } i \geq j_{tmp} \\ T_2(j_{tmp}) \times r_c^{-(j_{tmp}-j)}, & \text{otherwise.} \end{cases}$
Output: the target tempi $T_{t1}, T_{t2}$
```

4.3 Synthesis Process

After changing the tempi, we align $clip_2$ at the start position of $clip_1$'s transition segment. Then, we apply cross-fading on the transition segments. The effect of cross-fading is achieved by using the log-transform method [3] because it better fits the actual human auditory system.

5. MUSIC FILTERING AND ORDERING

In this section, we describe the extra steps for dealing with more than two clips: music filtering and ordering.

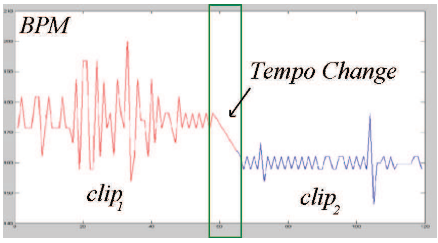

Figure 6. Tempi change in the transition length of the clips of “Let’s move it” and “Real man.”

5.1 Music Filtering

In order to reduce the probability of pasting quite distinct clips and the computational load in the ordering process (c.f. Section 5.2), we eliminate clips with extreme values by pair-wise comparison. A $clip_p$ is said to be extreme dissimilar and should be eliminated if there are more than half of the other clips ($clip_q$) in the database dissimilar to $clip_p$. The dissimilarity and similarity of any two clips are measured sequentially as follows.

5.1.1 Loudness Dissimilarity

The loudness dissimilarity is defined by the ratio $r_L(p,q)$ of the average loudness value of two clips $clip_p$ and $clip_q$, as shown in Eqn. (7)

$$r_L(p,q) = \frac{|Ld_p - Ld_q|}{Ld_p}, q = 1 \ldots W, q \neq p \qquad (7)$$

where Ld_p and Ld_q are the average loudness values of the p^{th} and the q^{th} clips in the datasets and W is the total number of clips. The loudness values are computed by accumulating log-energy (in db) in all the frequency bands. $clip_p$ and $clip_q$ are said to be loudness-dissimilar if $r_L(p,q)$ is greater than a certain threshold. By Weber’s law [10], the JND of loudness in db is 0.1, i.e. we will perceive the loudness change between $clip_p$ and $clip_q$ when the changing ratio ($r_L(p,q)$) is greater than 0.1. Since we have applied log-transform mechanism to smooth the change of volume in the sound effect module, we set the threshold value as 0.2 instead of the original strict standard.

5.1.2 Tempo Dissimilarity

Borrowing the concept in section 4.2, $clip_p$ and $clip_q$ are said to be tempo-dissimilar if there are not enough length for them to gradually adjusting the tempi from one to the other. The tempo dissimilarity is defined by $r_T(p,q)$:

$$r_T(p,q) = \left(\frac{T_q}{T_p}\right)^{\frac{1}{L_p+L_q}}, q = 1 \ldots W, q \neq p \qquad (8)$$

where T_p and L_p are the minimal tempo value of the last quarter in $clip_p$ and the corresponding length from the position of T_p to the end of $clip_p$. Similarly, T_q and L_q are the maximal tempo value of the first quarter in $clip_q$ and the corresponding length, as shown in Figure 7. If $r_T(p,q)$ does not lie in the range of JND mentioned in section 4.2, there will not be enough transition length for changing tempi from $clip_p$ to $clip_q$ and they should be regarded as tempo-dissimilar.

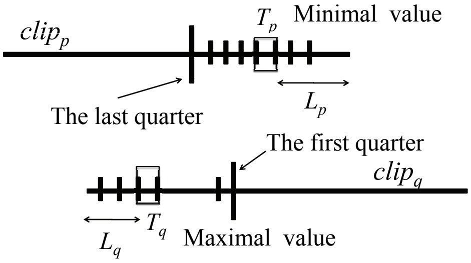

Figure 7. Finding tempo dissimilarity.

5.1.3 Chroma Histogram Similarity

In this module, we tend to avoid concatenating clips with different pitch distribution. The reason is as follows: the music paste will be unpleasant if we directly combine clips of different tonalities (e.g. C Major → e♭ minor) without modulation. Generally speaking, music clips with the same tonality contain similar pitch distributions. Thus, we construct a chroma histogram for each clip to represent its dominant pitch distribution and compare the clips by this histograms. For the 12 dimensional chroma vector ($\vec{C}_{pi}$) of the i^{th} beat in $clip_p$, we choose the index of its maximal value to represent the chroma dominant pitch (CM_{pi}) of this beat. That is,

$$CM_{pi} = \arg\max_u C_{pi}(u), u = 1 \ldots 12 \qquad (9)$$

The chroma histogram of $clip_p$ (CH_p) is constructed from CM_{pi}’s. Inspired by the commonly used color histogram intersection method [14] in the computer vision field, we define the chroma histogram similarity between $clip_p$ and $clip_q$ by

$$S_H(p,q) = \frac{\sum_{u=1}^{12} \min(CH_p(u), CH_q(u))}{\sum_{u=1}^{12} CH_p(u)} \qquad (10)$$

where $q = 1 \ldots W$, $p \neq q$. Analogous to the two previous subsections, $clip_p$ and $clip_q$ are viewed as dissimilar if $S_H(p,q)$ is less than 0.5.

5.2 Music Ordering

In the music ordering process, we tend to find an appropriate order to minimize the average distance values between each clip pair. For example, if the transition segments between $clip_1$ and $clip_3$ is not similar enough, maybe $clip_2$ can be the bridge of them. Besides, the transition segments from $clip_1$ to $clip_2$ may be less similar as compared with the transition segments from $clip_2$ to $clip_1$. Therefore, the ordering problem can be formulated as finding a path which goes through all clips in the datasets with minimum cost in the ordering matrix (D_o) defined as follows:

$$D_o(p,q) = \min_{i,j,L} \frac{1}{L+1} \sum_{l=0}^{L} D_{cr_{pq}}(i+l, j+l) \qquad (11)$$

where $L \in [L_{min}, L_{max}]$. To reduce the computation, we use a method analogous to the greedy algorithm but the path found cannot be guaranteed to reach the global optimum. The procedure is as follows:

	clip₁	***clip₂***	***clip₃***	***clip₄***
clip₁	0	0.3486	0.329	0.342
clip₂	0.3936	0	0.4704	0.4577
clip₃	0.2609	0.537	0	0.4806
clip₄	0.2898	0.4826	0.3732	0

Figure 8. An example of the ordering matrix for 4 clips.

1. Find the minimum value in the ordering matrix and set the corresponding two clips as the initial clips.
2. Find the minimum value in the row that corresponding to the last clip in the order found previously (each clip can only be visited once) and then add the corresponding clips to the order.
3. Repeat step 2 until all the values in the target row are larger than a predefined threshold or all clips have been visited.

Figure 8 shows an example of an ordering matrix constructed by four clips. First, we look for the minimum value in the matrix: 0.2609. We set the order as 3 → 1. Then, we check the values of first row: {0, 0.3486, 0.3290, 0.3420}. Since the first entry (0) represents $clip_1$ goes to $clip_1$ itself and the third entry (0.3290) means $clip_1$ goes to $clip_3$ again, we would not consider these two values. We find the minimum value of the rest: {0.3486, 0.3420} is 0.3420. Thus, the order becomes 3 → 1 → 4. Next, we check the fourth row and find 0.4826 is the only left value, so we compare it with the predefined threshold. If it is smaller than the threshold, the order would become 3 → 1 → 4 → 2. Otherwise, we would not concatenate $clip_2$ and the order would be just 3 → 1 → 4. Currently, the threshold is 0.5.

6. EXPERIMENTS AND USER EVALUATIONS

The experiments are conducted on the basis of user evaluations. In order to reduce the impact of the prejudices, evaluators will not be informed the methods used in the testing sequences.

6.1 Overlap Length Discussion

Assuming that the smoothness of the results depends on the overlap length, we let 15 evaluators judge the music pastes with different overlap lengths. 8 sets of clips (≈ 40 secs/clip) from different types of Chinese pop songs are used. We generate music pastes with 2 overlap lengths (force L^* = 4, 12 beats) and give each of them three different α values (c.f. Eqn. (3)) in the transition segments finding process. Figure 9 describes the overall results. The vertical axis represents the percentages of how many people prefer each method. We found that results with longer overlap length aren't really more acceptable than the shorter ones. The reason is probably that the similarity of transition segments decreases as the overlap length grows. Another observation is that the evaluator's acceptance varies with the types of the music clips. For instance, the accepted overlap length between two rap clips may be shorter than those of two lyric clips. Besides, over 60% of the evaluators preferred 4 beats as the overlapping length. Hence, we set the default overlap length to 4 beats long in the next section to compare the influence of different similarity measurements.

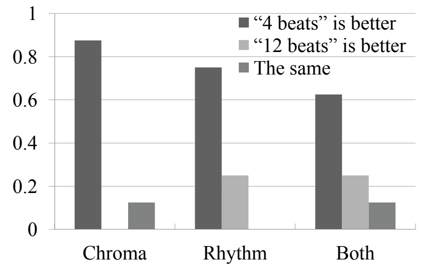

Figure 9. Overlap length comparison results.

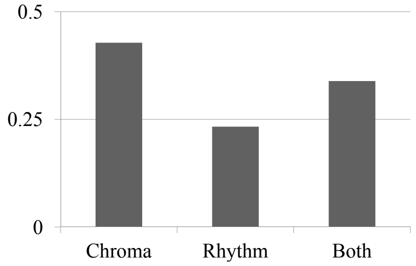

Figure 10. Comparison of 3 measurements.

6.2 Comparison of different similarity measurements

The similarity measurements we used for comparison are chroma only, rhythm only and both chroma and rhythm, i.e. α = 0, 1, 0.5. We utilized 8 sets of clips from songs in different languages. Fifteen evaluators gave scores from 1 to 10 to represent their satisfactions (higher score means better satisfaction) with respective to the feeling of intrusion. Figure 10 shows the percentages of how many people prefer each method. We found that chroma may be the most preferred measurement. Therefore, we choose the chroma measurements to conduct the following comparison with automated DJ.

6.3 Comparison with Automated DJ

In the automated DJ system [4], we can only use the music clips existing on the Amazon website and they should overlap at least 2 seconds. Thus, we selected 5 sets of pop songs available on Amazon and set the overlap length to 8 beats (≈ 2 secs). Each set consists of two music pastes, one is generated by automated DJ and the other is by our approach. Seventeen evaluators participate in choosing their preferable method. Similarly, Figure 11 shows the percentages of how many people prefer each method. Overall speaking, the music pastes generated by our approach are promising and preferred as compared to those generated by the automated DJ. The acceptances may vary with different sets. For instance, our method is superior to automated DJ

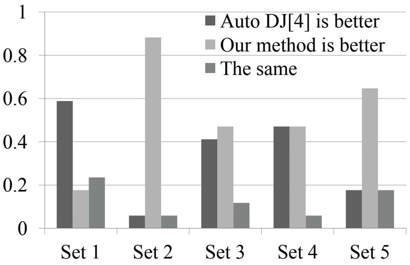

Figure 11. Comparisons with Automated DJ.

in a great level for set 2. The reason is that set 2 is a combination of a female voice and a male voice singing in quite different pitches. The results will be a little bit intrusive if we just concatenate these clips by rhythm-similar segments. Instead, we choose pitch-similar segments where the pitch of female voice downwards and the pitch of male voice upwards. The results would be more pleasant to hear.

6.4 Discussion

According to the above experimental results, we discovered the following factors affecting people's feeling toward the music paste: **(i) Language and lyrics.** The music pastes with unfamiliar languages will be probably more acceptable. In our datasets, half of sets are with unfamiliar languages to the evaluators. And 75% of this kind of pastes are scored higher than the average scores. This is probably because the intrusiveness increases when the lyrics of clips in familiar language conflict with each other. In contrast, it is not easy for evaluators to perceive the transition in clips with unfamiliar languages. **(ii) The ending position of phrases in the clips.** The music pastes will be probably more acceptable if the clips are transformed at the ending position of phrases. We have gathered statistics on our experiment datasets. There are 50% of the sets fit the mentioned condition. The scores are all higher than the average scores of all sets. The reason is probably that people's anticipation for the ending phrases will smooth the intrusiveness at the transition. **(iii) Familiarity with the music clips.** The music pastes are probably less acceptable if the evaluators have heard the music clips before. 71% of the evaluators gave higher score to unfamiliar music pastes than familiar ones. **(iv) The influence of vision.** The transition in music paste would be less noticeable if vision information involves. We combined one of our music pastes with a photo slideshow and let the evaluators view and listen again. Over 90% of evaluators gave higher scores to it because they almost did not notice the transition.

7. CONCLUSION AND FUTURE WORKS

In this paper, we provide a tool for automatically choosing proper music clips from a given audio collection and combining the chosen clips as euphonious as possible. We employ common auditory music features and borrow the concept from distance matrix to determine the transition segment and choosing music clips. The transition length is determined by Weber's Law. Besides, we apply phase vocoder to adjust the audio files and use cross-fading in synthesis process. Moreover, we conduct subjective evaluations on the correlation between pasting methods and auditory quality of combined clips. The overall experiment results show that the generated music pastes are acceptable to humans.

There are rooms for improving the proposed system. First, the pasting method is restricted to clips with similar enough transition segments. Perhaps the clips can be connected by automatically generating appropriate intermezzo or bridge music. Second, the proposed work can be meliorated by the improvement of music analysis techniques. More similarity measurements closer to style-similarity (timbre, rhythm) would improve the filtering process. Furthermore, more representative auditory features and similarity measurements, techniques for music structure analysis and phrase boundary extraction would help the process of locating transition segments. Third, studies on the variant overlapped length range in the transition segments are still worth investigating while currently the whole transition segments are overlapped. In the future, we will continue our investigation in these directions.

8. REFERENCES

[1] G. Loy: *Musimathics*, pp. 295–296, 347–350, The MIT Press, 2006.

[2] D. Cope: *Experiments in Musical Intelligence*, Madison, WI: A-R Editions, 1996.

[3] T. Jehan: "Creating music by listening," PhD thesis, MIT Media Lab, Cambridge, MA, 2005.

[4] T. Jehan: "This is My Jam," visited at Feb. 18, 2008; `http://thisismyjam.com/`

[5] M. Cooper, and J. Foote: "Automatic Music Summarization via Similarity Analysis," *Proceedings of the International Symposium on Music Information Retrieval* (ISMIR '02), Paris, France, 2002.

[6] C. A. Harte and M. B. Sandler: "Automatic chord identification using a quantised chromagram," *Proceedings of the Audio Engineering Society*, Spain, 2005.

[7] M. Cicconet: "Rhythm features," visited at Dec. 13, 2008; `http://w3.impa.br/~cicconet/cursos/ae/spmirPresentation.html`

[8] "Virginia Tech Multimedia Music Dictionary," visited at July 31, 2009; `http://www.music.vt.edu/musicdictionary/`.

[9] S. Dixon: "Evaluation of the Audio Beat Tracking System BeatRoot," *Journal of New Music Research,* Vol. 36, No. 1, pp. 39–50, 2007;

[10] G.T. Fechner: *Elements of psychophysics 1,* Holt, Rinehart & Winston, New York, 1860.

[11] Kim Thomas: "Just Noticeable Difference and Tempo Change," *Journal of Scientific Psychology,* May 2007.

[12] M. Dolson: "The phase vocoder: a tutorial," *Computer Music Journal*, Vol. 10, No. 4, pp. 14–27, 1986.

[13] C.J. Plack and R.P. Carlyon: "Loudness perception and intensity coding," *Hand book of Perception and Recognition 6: Hearing*, pp. 123-160, Editorial B.C.J. Moore, Academic Press, London, 1995.

[14] M. J. Swain and D. H. Ballard: "Color indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, pp. 11–32, 1991.