Rhythm of Motion Extraction and Rhythm-Based Cross-Media Alignment for Dance Videos

Wei-Ta Chu, *Member, IEEE*, and Shang-Yin Tsai

Abstract—**We present how to extract rhythm information in dance videos and music, and accordingly correlate them based on rhythmic representation. From dancer's movement, we construct motion trajectories, detect turnings and stops of trajectories, and then estimate rhythm of motion (ROM). For music, beats are detected to describe rhythm of music. Two modalities are therefore represented as sequences of rhythm information to facilitate finding cross-media correspondence. Two applications, i.e. background music replacement and music video generation, are developed to demonstrate the practicality of cross-media correspondence. We evaluate performance of ROM extraction, and conduct subjective/objective evaluation to show that rich browsing experience can be provided by the proposed applications.**

Index Terms—**Rhythm of motion, motion trajectory, music beat, background music replacement, music video generation.**

I. Introduction

When listening to music, people spontaneously tap their fingers or feet according to the music's periodic structure. Dancing with music is a human nature to express meaning of music or to show people's emotion. In recent years, hip-hop culture drives the development of street dance, and learning to dance has deeply attracted young people. Due to popularity of street dance and ease of video capturing, many dancers record their dances and share them on the web. However, quality of these videos, especially the audio tracks accompanying with the videos, is generally low. Moreover, to promote dance competitions or TV shows, music videos are elaborately produced by experts who have ample knowledge on choreography, music rhythm, and video editing. It is never an easy task for amateur dancers who want to share or preserve their performances for entertainment or education purposes.

In this paper, we investigate how rhythm information can be found and utilized in street dance videos. From the visual track, *periodic motion changes* of dancer's movement are extracted, which constitute "rhythm of motion" (ROM). From music, rhythm is constructed based on periodic properties of music beats. After extracting rhythm information from two modalities, cross-media correspondence is determined to facilitate replacing background music of a dance video by a high-quality music piece. In addition, music videos can be generated by concatenating multiple dance video clips with similar ROMs.

The concept "rhythm" describes patterns of changes in various disciplines. In music, *beat* refers to a perceived pulse marking off equal durational units [7], and is the basis with which we compare or measure rhythmic durations [9]. *Tempo* refers to the rate at which beats strike, and *meter* describes accent structure on beats. These parameters jointly determine how we perceive music rhythm. In contrast to the long history of music cognition study, analyzing rhythm of motion in videos is just at its infant stage. We focus on extracting *motion beats* from videos, which play an essential role in constituting ROM. To simplify description, we interchangeably use "rhythm" and "beats" in this paper.

Contributions of this work are summarized in Figure 1 and are described as follows.

- ROM extraction: By tracking distinctive feature points on human body, motion trajectories are constructed and transformed into time-varied signals, which are then analyzed to extract ROM. ROM represents periodic motion changes, such as "turning" and "stop" of trajectories.
- Music beat detection and segmentation: By integrating energy dynamics in different frequency bands, music beats are detected. Periodically evolved beats are then used to describe rhythm of music.
- Rhythm-based cross-media alignment: Two rhythm sequences are compared, and an appropriate correspondence between them is determined.
- Applications: Based on rhythm-based cross-media alignment, background music replacement and music video generation are developed, which demonstrate practicality of rhythm-based multimodal analysis.

The rest of this paper is organized as follows. Section II provides a survey on rhythm analysis in music and video, and introduces a related area derived from musicology. ROM extraction is described in Section III. Section IV first shows how we find rhythm of music, and then rhythm-based correspondence is determined to conduct background music replacement. Automatic music video generation is described in Section V. Section VI reports evaluation results and discussions, followed by concluding remarks in Section VII.

Wei-Ta Chu and Shang-Yin Tsai are with National Chung Cheng University, Chiayi, Taiwan (e-mail: wtchu@cs.ccu.edu.tw, shouyinz@hotmail.com).

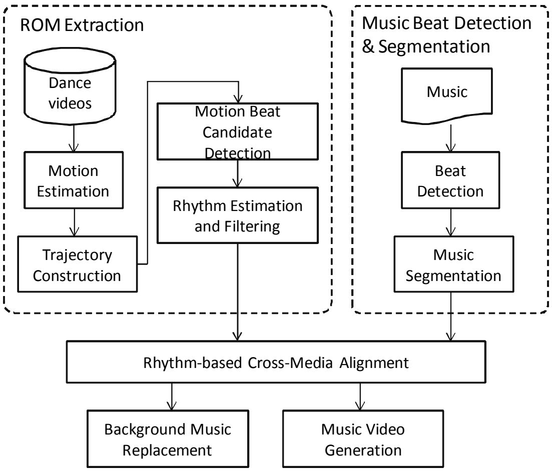
ROM Extraction
Dance videos
Motion Estimation
Trajectory Construction
Motion Beat Candidate Detection
Rhythm Estimation and Filtering
Music Beat Detection & Segmentation
Music
Beat Detection
Music Segmentation
Rhythm-based Cross-Media Alignment
Background Music Replacement
Music Video Generation

Figure 1. Framework of the proposed system.

II. Related Works

A. Motion Analysis in Videos

Research about motion analysis mainly focuses on three factors: motion magnitude (or moving speed, motion activity), motion direction, and motion trajectory. Shiratori et al. [28] detect changes of moving speed in traditional Japanese dances, and then segment dance videos into a series of basic patterns. Deman et al. [5] detect temporal discontinuities by extracting local minimums of motion magnitudes. Motion analysis is conducted for the same object part in neighboring frames. Based on motion trajectories, Su et al. [29] develop a framework for motion flow (i.e. motion trajectory in our work) construction, which is then adopted to conduct video retrieval. This work only extracts a single motion flow to represent video content. With feature point detection and motion prediction, the work proposed in [21] constructs multiple trajectories in dance videos based on a “color-optical flow” method, which jointly considers motion and color information to facilitate motion tracking. Based on the extracted dance patterns, this system segments dance videos automatically.

Despite rich studies on motion analysis, information to constitute ROM is not only motion magnitude or absolution/relative moving direction, but also the periodicity of substantial motion changes. To extract implicit rhythm derived from human body movement, we need finer motion analysis for body parts with complex dancing steps. For example, for a specific music rhythm, a dancer may move his left hand up and right hand down, followed by jumping at the instant of a music beat strikes. For the same music rhythm, a different dancer may squat, followed by twisting his body at the instant of the music beat strikes. They have different moving patterns, but we can easily sense that they move according to the same music rhythm.

We have to emphasize that ROM is not only derived from “periodic motion,” but also “periodic changes of motion.” According to [4], motion of a point is periodic if it repeats itself with a constant period, e.g. an object like a pendulum goes back and forward periodically or an object cyclically moves around a circle. However, ROM in dance mainly comes from “periodic changes of motion”, such as periodic characteristics of turning, twisting, jumping, or stopping. Dancer’s movement does not necessarily repeat, but we still perceive he/she follows an implicit periodicity to make movement changes.

Relatively fewer works have been done for periodic motion analysis. Deman et al. [5] explore the use of object-based motion to detect specific events in observational Psychology. Specific moving patterns are detected, but rhythm information from motion is not specially studied. Based on videos captured in light-controlled environments, Guedes calculates luminance changes of pixels in consecutive frames [15], which indicate motion magnitude between frames. Evolution of motion magnitude is then transformed into the frequency domain, and the dominant frequency component is detected by a pitch tracking technique. Our system detects periodic changes of motion by a method similar to Guedes’s. However, in our case, dance videos were captured in uncontrolled environments and varied luminance changes hurt Guedes’s approach. Cutler and Davis [4] compute object’s self-similarity as it evolves in time, and then apply time-frequency analysis to detect periodic motion. Laptev et al. [18] view periodic motion subsequences as the same sequence captured by multiple cameras. Periodic motion is thus detected and segmented by approximate sequence matching algorithms. Both [4] and [18] assume that orientation and size of objects do not change significantly, and they analyze how objects repeat themselves. However, in dance videos, ROM is not necessarily from motion repetition, and different body parts are not guaranteed to have consistent moving orientation and object size.

Kim et al. [17] provide us a hint to extract ROM from motion data. They detect rapid directional change on joints, and then transform this information as motion signals. Power spectrum density of signals is then analyzed to estimate the dominant period. This systematic approach is suitable for our case. However, motion data in [17] were explicitly captured from sensors. We focus on ROM from real dance videos. Estimating periodicity from noisy motion data is more challenging.

B. Audio to Video Matching

Associating videos with music has been viewed as a good way to enrich presentation. Foote et al. [8] propose one of the earliest works on automatically generating music videos. Audio clips are segmented based on significant audio changes, and videos are segmented based on camera motion and exposure. Video clips are then adjusted to align with audio to generate a music video. Also for home videos, Hua et al. [16] discover repetitive patterns of music and estimate attention values from video shots, and then combine two media to generate music videos. Wang et al. [31] extend this idea to generate music video for sports videos. Events in sports videos are first detected, and two schemes (video-centric and music-centric) can be used to integrate two media. Yoon et al. [33] transform video and music into feature curves, and then apply the dynamic programming strategy to match these two modalities. To tackle with length difference between music and video, they adopt a music graph to elaborately scale music such that video-music synchronization can be guaranteed. Recently, Yoon et al. [32] align music with arbitrary videos by using features in a multi-level way.

Generally, these works first segment videos and music into segments, extract features from segments, and then match two sequences of segments to generate final results. Videos are first segmented based on color [16], events [31], camera motion and brightness [8], or shape [32]. These features characterize global information in video frames, and object-based information, e.g. object motion, may be overlooked. Works in [33] consider object motion and construct feature curves for videos. However, few discussions were made about integrating local motion from multiple parts, and the idea of periodic motion or periodic changes of motion was not mentioned.

Finding association between video and audio (music) is a crucial step for audiovisual applications. Recently, Feng et al. [35] propose a probabilistic framework to model correlation between video and audio, and automatically generate background music for home videos. Lee's group investigates association between music and animation [36] , or between music and video [37]. A directed graph is constructed and traversed to generate background music fit to the targeted animation. In fact, exploiting multimodal association to generate background music has been studied for a long time. An earlier idea can be found in [38].

C. *Embodied Music Cognition*

Most computer scientists separately detect rhythm information from music and video, and then synchronize them to generate audiovisual presentation. In fact, a branch of musicology, embodied music cognition [19], that investigates the role of human body in relation to music activities has been studied for years. Human body can be seen as a mediator that transfers physical energy to represent musical intentions, meanings, or signification. People move when listening to music, and through movement, people give meaning to music. This is exactly what dancers do in their performance. We provide a brief survey on this field in the following.

Leman's book [19] provides a great introduction to embodied music cognition, and provides a framework for engineers, psychologists, brain scientists, and musicologists to contribute to this field. More specifically, the EyesWeb project focuses on understanding affective and expressive content of human's gesture [3]. The developed system analyzes body movement and gesture to facilitate controling sound, music, and visual media. Similarly, Godøy [10] investigates relationships between musical imagery and gesture imagery. As it is an ongoing research field, Godøy describes ideas, needs, and research challenges to link music cognition with body movement. Currently, researchers in this field start to use signal processing techniques to demonstrate that different parts of the body often synchronize music at different metrical levels [30]. The latest results suggest that metric structure of music is encoded in body movements. For computer scientists, the studies mentioned above open another window to discover rhythmic relationship between music and motion.

III. Rhythm of Motion

A. *Overview of ROM*

Objects may move forward and backward periodically, move in the same trajectory periodically, or stop/turn according to some implicit tempo. In dance videos, ROM is a clue about how a dancer interprets a music piece. Figure 2 shows an example of rhythm of motion. The dancer stands up with hand moving down from frames 0 to 10, squats down with hand moving up from frames 10 to 20, and repeats the same action (almost) periodically. Note that the human body gives rise to non-rigid motion, with different parts moving toward different directions of different magnitudes. However, we can still realize that the dancer has periodic changes of motion. The implicit period thus forms rhythm of motion.

Different dancers may have different interpretations for the same music, and they may not completely move with rhythm of music. Fortunately, most dancers have common consensus about how and when to move their bodies. Therefore, dance videos with same background music may consist of similar but not completely the same ROM. Dancers usually divide the music into segments of "eight beats", and then design dancing steps for each segment [39]. Although different dancers have varied styles on poses or body movement, they make emphasized stop or turning at boundaries of eight-beat segments. This characteristic makes us capable to estimate the dominant period of emphasized motion stop/turning.

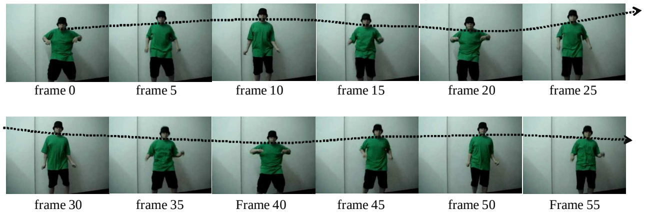

Figure 2. An example of rhythm of motion.

B. *Motion Trajectory*

To extract motion trajectories, we only consider motion on feature points rather than all pixels in video frames. Motion predicted from feature points effectively represents video content and decreases interference from background noise. Although our work is not limited to any specific feature detection method, we adopt the Shi-Tomasi (ST) corner detector [27], because it is shown to be robust under affine transformation and can be implemented easily. We apply the Pyramid Lucas-Kanade (PLK) optical flow detection method [2] to predict motion in various scales.

The moving direction of a feature point s_i from frame t to frame $t+1$ is estimated by:

$$s_i'(x', y') = PLK(s_i(x, y)), \qquad (1)$$

where $s_i(x, y)$ denotes position of the feature point s_i at frame t, $s_i'(x', y')$ denotes the estimated position of the feature point s_i at frame $t+1$, and $PLK(\cdot)$ denotes the estimation function.

To construct trajectories, we need to appropriately connect feature points in temporally adjacent frames. Motion and color information of feature points in neighboring frames are checked.

For the feature point s_i at frame t, we find the most appropriate feature point s_{j^*} at frame $t+1$ by

$$j^* = \arg\min_{s_j \in \mathcal{N}(s_i')} d(s_i, s_j), \tag{2}$$

where $\mathcal{N}(s_i')$ denotes neighborhood of the estimated location $s_i'(x', y')$ The neighborhood region is defined as the set of pixels in the circle centered by $s_i'(x', y')$, with radius r.

The distance $d(s_i, s_j)$ is defined as

$$d(s_i, s_j) = \sum_{m=0}^{M-1} |h_i(m) - h_j(m)|, \tag{3}$$

where h_i and h_j are HSV color histograms of the 9×9 image patches centered by s_i and s_j, respectively. The values of hue, saturation and volume are quantized into 8 bins, respectively.

By this process, we construct feature-based trajectories. If a feature point s_j at frame $t+1$ is able to be connected by multiple feature points $\{s_i\}$ at frame t, only the feature point s_{i^*} having the minimum distance to s_j is selected, i.e. $i^* = \arg\min_i d(s_i, s_j)$. In addition, to filter out short trajectory segments caused by noisy feature points, we eliminate motion trajectories shorter than a predefined threshold.

Figure 3 shows examples of motion trajectories in the same video sequence but constructed based on different feature points. Figures 3(a), 3(b), and 3(c) are correctly extracted motion trajectories, and Figure 3(d) is a falsely extracted motion trajectory. We roughly can see periodic properties of trajectories in Figures 3(a), 3(b), and 3(c).

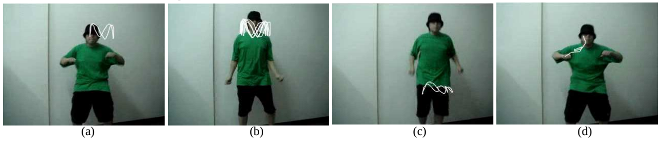

Figure 3. Examples of constructed motion trajectories based on different feature points.

C. Motion Beat Candidate Detection

Based on the extracted trajectories, we detect candidates of motion beats for ROM extraction. A motion trajectory is denoted by $J = \{s, (x_0, y_0), (x_1, y_1), ..., (x_m, y_m)\}$, where s denotes the frame number at which J starts, and (x_i, y_i) is the xy-coordinate of the feature point at the frame $s+i$. We detect stops and turns of motion trajectories as motion beat candidates, which can be described by substantial changes of motion magnitude and moving direction.

To alleviate the influence of trajectory extraction noise, motion estimation errors are assumed to be Gaussian distributed [1], and we conduct low-pass filtering by convolving motion trajectories with a Gaussian kernel function:

$$G(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}}, \tag{4}$$

where σ is the standard deviation controlling smoothness, and t denotes the difference (in terms of frame number) from an arbitrary frame to the frame centered by the Gaussian.

The horizontal movement data $J_x = \{x_0, x_1, ..., x_m\}$ is filtered as

$$\hat{J}_x(i) = \sum_{u=0}^{m} J_x(u) \cdot G(i-u), \tag{5}$$

where $\hat{J}_x(i)$ denotes the filtered horizontal displacement at frame $s+i$. The vertical displacement $J_y = \{y_0, y_1, ..., y_m\}$ is filtered in the same way. After filtering, the motion trajectory $\hat{J} = \{s, \hat{J}_x, \hat{J}_y\}$ is smoother, and then we are able to detect stops and turns more precisely.

A stop action is often a joint of movements. A dancer may move his hand toward some direction, stops when a music beat strikes, and later moves reversely. The stop action in dance videos represents that the movement has completely ended, or just a temporary stop which serves a start of another movement. To detect stops of a motion trajectory, we examine evolution of the motion magnitude, $H_g = \{g_0, g_1, ..., g_{m-1}\}$, where g_i denotes the magnitude of the motion from frame i to frame $i+1$, i.e. $g_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$. We use $H_g[i]$ to represent g_i in the following description. Magnitude decreases when movement decelerates, and a local minimum occurs at the moment of a stop. In this work, we detect local minimums of the magnitude history based on a modified hill climbing algorithm [11].

There may be many stop points in a motion trajectory. To detect every local minimum, we modify the hill climbing algorithm as in Algorithm 1. If the magnitude of the $nIdx$-th frame in the neighborhood of the current frame (indexed by $cIdx$) is smaller than $H_g[cIdx]$, we replace $cIdx$ by $nIdx$. This procedure repeats until $H_g[cIdx]$ is the smallest within the neighborhood. Neighborhood of the index $cIdx$ is defined as $\mathcal{N}(cIdx) = \{cIdx+1, cIdx+2, ..., cIdx+\Delta\}$. The value Δ is set as 7 in our work, i.e. only the seven temporally adjacent frames following the $cIdx$-th frame are checked. After the local minimum is found, we again adopt hill climbing to find the local maximum, which serves the start for finding the next local minimum. This process repeats until the whole magnitude history is checked. Finally, the set of local minimums are viewed as motion beat candidates.

To find trajectory turning, we analyze evolution of motion orientation. The orientation history is denoted as $H_o = \{o_0, o_1, ..., o_{m-1}\}$, where o_i is the motion vector from frame i to frame $i+1$, and is represented in a united vector form, i.e. $o_i = \frac{1}{g_i}(x_{i+1} - x_i, y_{i+1} - y_i)$. Based on this information, we design a method shown in Algorithm 2 to find turnings in a trajectory. When the trajectory keeps moving at the same direction at frames i and $i+1$, the inner product of o_i and o_{i+1} (denoted as $\langle H_o[i], H_o[i+1]\rangle$) would be close to 1. On the other hand, when the trajectory turns, the value of inner product decreases or even reverses. Therefore, we accumulate inner products between motion vectors in a sequence of frames, and then find the turning points by checking the average value of accumulated inner products (line 7 to line 11 in Algorithm 2). If the average value is less than a threshold ϵ, we find the instant at which the average value of the accumulated inner product changes the most (line 12). This instant is stored, and is then updated as the next $start$ point. This process repeats until the whole orientation history is checked. The set of turning points is also viewed as motion beat candidates.

```
Algorithm 1: Finding stop points of a trajectory
Input: magnitude history $H_g$
Output: a set of local minimums $L$ in $H_g$
$L \leftarrow \varnothing$
$cIdx \leftarrow 0$
$decreaseFlag \leftarrow True$
while $cIdx \leq m-1$
    if $decreaseFlag$
        $nIdx \leftarrow \arg\min_i H_g[i],\ i \in \mathcal{N}(cIdx)$
        if $H_g[nIdx] \leq H_g[cIdx]$
            $cIdx \leftarrow nIdx$
        else
            $L = L \bigcup \{cIdx\}$
            $decreaseFlag \leftarrow False$
    else
        $nIdx \leftarrow \arg\max_i H_g[i],\ i \in \mathcal{N}(cIdx)$
        if $H_g[cIdx] \leq H_g[nIdx]$
            $cIdx \leftarrow nIdx$
        else
            $decreaseFlag \leftarrow True$
end while
```

```
Algorithm 2: Finding turning points of a trajectory
Input: orientation history $H_o$
Output: a set of turnings $U$
$U \leftarrow \varnothing$
$start \leftarrow 0$
while $start \leq m-1$
    $history \leftarrow 0$
    $diff \leftarrow \varnothing$
    $avg \leftarrow \varnothing$
    for $j = start+1$ to $m-1$
        $history \leftarrow history + \langle H_o[start], H_o[j] \rangle$
        $avg[j] \leftarrow \frac{history}{j-start}$
        $diff[j] \leftarrow avg[j] - avg[j-1]$
        if $avg[j] \leq \epsilon$
            $i^* = \arg\max_{start<i<j} diff[i]$
            $U \leftarrow U \bigcup \{i^*\}$
            $start = j$
            break
end while
```

D. *Rhythm Estimation and Filtering*

In this section, we use the scheme proposed in [17] for motion beat refinement and dominant period estimation. Note that not every detected turning point or stop point is truly a motion beat. Therefore, the scheme first finds the dominant period from motion beat candidates, and accordingly estimates the reference beats. Guided by reference beats, we estimate actual motion beats by finding the candidate beats that have small temporal differences to reference beats.

Single trajectory:

To predict the dominant period from motion beat candidates, we estimate pulse repetition interval (PRI) from a signal generated based on the time instants of beats striking [24]. This method is computationally tractable and is robust to trajectory extraction errors. From a motion trajectory, a motion beat sequence is denoted as $S = \{b_0, b_1, \ldots, b_n\}$, where b_i is the timestamp (in terms of frame number) of the ith motion beat candidate. We can model generation of these motion beats as

$$b_i = \phi + k_i T + \eta_i, \tag{6}$$

where T is the unknown period, ϕ is a shift ranging in the interval $[0, T)$, η_i is noise caused by the dancer or the beat detection module and is set as in the interval $[-T/2, T/2)$, and k_i is a positive number indicating the index of beat. The reference motion beats can be modeled as $r_j = \phi + jT$, which represents periodic appearance of actual motion beats. With this model, we would find T and ϕ for reference beat estimation.

Figure 4 shows how we estimate reference beats based on motion beat candidates. First, we transform the sequence of motion beat candidates into a continuous-time signal as

$$y_k(t) = \begin{cases} \cos\left(2\pi\left(\frac{t-b_{k-1}}{b_k - b_{k-1}}\right)\right), & \text{if } b_{k-1} < t < b_k, \\ 1, & \text{if } t = b_k, \end{cases} \tag{7}$$

where $k = 2, 3, \ldots, n$. This signal is maximized when a motion beat candidate appears, i.e. $y_k(t) = 1$ when $t = b_k$. When t is located between two motion beat candidates, the value of $y_k(t)$ is determined by a cosine function. For each beat candidate b_i, a cosine centered at b_i is applied, and all sinusoids generated from beat candidates are accumulated to generate a signal $y(t)$, as shown in the second row of Figure 4.

Based on $y(t)$, we estimate the dominant period by calculating maximum of power spectrum density (PSD) [23]. This process calculates energy of the accumulated sinusoid in different frequency bands. According to the Nyquist sampling theorem, the maximum frequency able to be detected is half of sampling rate. Fortunately, we can reasonably assume that the frequency of motion beats is lower than half of frame rate (30 fps), because the human body hardly moves so fast. We calculate PSD by

$$PSD_y(f) = \left| \sum_{j=1}^{M} y(t) e^{i2\pi f t_j} \right|^2, \tag{8}$$

where M is the length of the accumulated sinusoid, and f is the index of a frequency band. The dominant frequency is the frequency that gives the maximal $PSD_y(f)$:

$$f_d = \arg\max_f PSD_y(f). \tag{9}$$

The dominant period $T = 1/f_d$ implies that most motion beats periodically appear at multiples of T.

We then estimate ϕ by finding the shift that causes the maximal sum of periodic positive peaks:

$$\tilde{\phi} = \arg\max_\phi \sum_{j=1}^{M-1} y(jT + \phi), \tag{10}$$

where ϕ is in the interval $[0, T)$.

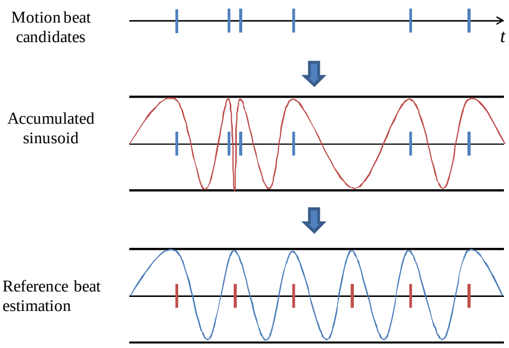

Figure 4. Reference beat estimation based on motion beat candidates.

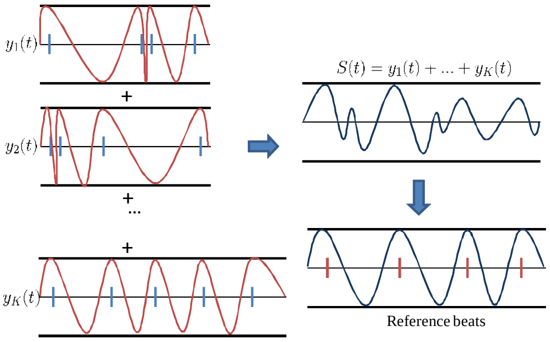

Figure 5. Estimation of reference beats with multiple motion trajectories.

Multiple trajectories:

The aforementioned process is applied to a motion beat sequence derived from a single motion trajectory. To jointly consider multiple beat sequences derived from multiple motion trajectories, we extend the process as illustrated in Figure 5. The idea of this process is similar to extracting fundamental frequency or pitch detection from a signal that is a superposition of sinusoids. This process is often adopted in pitch detection for speech [20] or music. In our case, because different parts of the dancer's body acts according to the same rhythm of music, sinusoids generated from different body parts are nearly harmonically related. Although motion trajectories may have different durations, this process is able to resist variations of different sequences and robustly finds the dominant period.

Based on this idea, we construct an accumulated sinusoid $y_i(t)$ for each trajectory separately, and then superpose K sinusoidal signals $y_1(t)$, $y_2(t)$, …, $y_K(t)$ as a superposed signal, assuming K different motion trajectories. The PSD of the signal $S(t) = y_1(t) + y_2(t) + ... + y_K(t)$ is computed as

$$PSD_S(f) = \left| \sum_{j=1}^{M} S(t) e^{i2\pi f t_j} \right|^2, \quad (11)$$

where M is the length of the superposed signal, and f is the index of a frequency band. The dominant period and the phase can be estimated by the way same as in single trajectory.

After estimating reference motion beats, we detect actual motion beats and filter out outliers. The actual motion beats may appear close to reference beats. A beat candidate b_i is claimed as in the neighborhood (as an inlier) of a reference beat r_j if

$$r_j - \alpha\frac{T}{2} \leq b_i \leq r_j + \alpha\frac{T}{2}. \quad (12)$$

The value α is a parameter controlling the range of neighborhood. If α is too large, outliers may be included in the final process. If α is too small, we may filter out actual motion beats. We will test this parameter in the evaluation section.

After removing outliers, we detect actual motion beats by

$$B = \left\{ b_j^* \middle| b_j^* = \arg\min_{b \in \mathcal{N}(r_j)} |b - r_j|, j = 1, ..., N \right\}, \quad (13)$$

where b_j^* is the detected actual motion beat corresponding to reference beat r_j, and $\mathcal{N}(r_j)$ is the neighborhood of r_j defined in eqn. (12). The candidate beat that is in the neighborhood of r_j and is closest to r_j is detected as an actual motion beat. If a reference beat has no neighboring candidate beat, no corresponding actual motion beat exists at this moment.

IV. Background Music Replacement

We would like to replace the original audio track of a dance video, which is captured in an uncontrolled environment and is deteriorated by noises, by a higher-quality music piece, which conveys similar pulse as the original audio track but is from a CD recording or a high-quality mp3 file. We conduct background music replacement based on ROM in dance videos and music beats in the selected music piece.

A. *Music Beat Detection*

Music beat detection and tracking has been studied in the last decade. Scheirer [25] divides spectrum into several frequency bands, analyzes energy dynamics in each, and then fuses information from different bands to detect beats. Dixon [6] develops another classical work to automatically extract tempo and beat from music performance. More recently, Oliveira et al. [22] improve Dixon's approach to achieve real-time performance. Beat tracking becomes more challenging for non-percussive music with soft onsets and time-varying tempo. Grosche and Muller [13] propose a mid-level representation to derive musically meaningful tempo and beat information. They also propose a framework to evaluate consistencies of beat tracking results over multiple performances of the same music piece [14]. Covering a wide range of music, Eyben et al. [34] propose one of the state-of-the-art onset detection approaches based on neural networks. Readers who are interested in relationship between rhythm and mathematical models are referred to [26]. A complete review for rhythm description systems can be found in [12].

Although a more recent approach such as [34] can be applied to analyze music beats, music accompanied with street dance often has strong beats, and the typical Scheirer's method [25] is used to detect music beats in our work. Energy evolution in each frequency band is extracted, followed by envelope smoothing with a half-Hanning window. We again conduct hill climbing for peaks finding in each envelope, and then integrate results in different frequency bands to estimate music beats. Because there are many detection noises, we refine the result by the process described in Section III.D. A sinusoidal function is constructed based on the detected music beats, and the dominant period and time shift of the sinusoid are estimated to determine reference music beats. The actual music beats are detected by finding the ones that are closest to reference beats.

B. *Rhythm-Based Cross-Media Alignment*

Based on rhythm information, we would like to determine appropriate alignment between two modalities. Motion beats and music beats are respectively represented by a binary vector, denoted by $B_{mt} = \{b_0, b_1, ..., b_{M-1}\}$ and $B_{mu} = \{b_0, b_1, ..., b_{N-1}\}$, where $b_i = \{0, 1\}$ and $b_i = 1$ indicates a beat at the ith millisecond of the video (music).

Basically, this is an approximate sequence matching problem, which can be solved by widely-known algorithms such as dynamic time warping (DTW). However, given two binary sequences, e.g. $B_{mu} = 101010101011$ and $B_{mt} = 1001001$, the DTW algorithm equally treats characters 0 and 1 and finds the longest common subsequence between B_{mu} and B_{mt}. In dance videos, dancers only interpret parts of music beats, and the priorities of 0 and 1 should be different. Although we can design a variant of DTW to handle this problem, we found that the following simply alternative already achieves satisfactory performance.

To simplify description, we assume duration of the higher-quality music is longer than that of video without loss of generality. We also note that motion beats only correspond to parts of music beats. With these characteristics, we would like to find a music segment that is appropriately to be aligned with the video. The original background music of the video is then replaced by the newly-aligned music segment.

We try different time shifts for the music beat sequence to find the best match between two sequences. To measure degree of matching, we define the temporal distance between the ith motion beat $B_{mt}[i]$ and its closest music beat in the sequence with the shift Δ by

$$d(\Delta, i) = \min_{0 \le j \le N-1} |i - j|, \forall B_{mu}[j + \Delta] = 1, \quad (14)$$

where N is the length of the music beat sequence, and $B_{mu}[j + \Delta]$ denotes the value of the jth sample in the sequence with the shift Δ.

Degree of match between two sequences with the shift Δ is defined as the ratio of coherence to distance. The coherence value $C(\Delta)$ is defined as

$$C(\Delta) = \frac{1}{M} \sum_{i=0}^{M-1} \frac{B_{mt}[i]}{d(\Delta,i)+1}, \quad (15)$$

which is larger if temporal distances between motion beats and their closest music beats are smaller.

The difference value $D(\Delta)$ is calculated as

$$D(\Delta) = \frac{1}{M} \sum_{i=0}^{M-1} B_{mt}[i] \cdot d(\Delta, i). \quad (16)$$

These two factors are integrated as the final degree of matching:

$$DOM(\Delta) = \frac{C(\Delta)}{D(\Delta)}. \quad (17)$$

Finally, we determine the most appropriate shift Δ^* by

$$\Delta^* = \arg\max_{0 \le k \le N-M} DOM(k). \quad (18)$$

After finding the best shift Δ^* , the music segment $\{b_{\Delta^*}, b_{\Delta^*+1}, ..., b_{\Delta^*+M-1}\}$ from B_{mu} is used to replace the original background music. For example, if the best shift Δ^* is 3.8 seconds, and the video clip's length is 28.1 seconds, then the music segment from 3.8 to 31.9 seconds of the selected music piece is used to replace the original background music.

According to eqn. (18), we have at most $N - M + 1$ possible shifts. Given a shift Δ, the complexity for calculating degree of matching (eqn. (17)) is $O(M^2N + M^2N) = O(M^2N)$ because M instructions are respectively needed to calculate $C(\Delta)$ and $D(\Delta)$, and MN comparisons are needed to calculate $d(\Delta, i)$ in the worst case. Because both sequences B_{mt} and B_{mu} are temporally sorted, to find the closest music beat to the ith motion beat $B_{mt}[i]$, we just need to search neighborhood of the $(i + \Delta)$-th point in the sequence B_{mu}. Therefore, the number of comparison for determining $d(\Delta, i)$ is much less than MN.

V. Music Video Generation

A. Music Segmentation

To generate music videos, we first segment music and then select suitable video clips for each music segment. By comparing audio frames, a self-similarity matrix is constructed to describe autocorrelation, and the entries in the main diagonal with local maximum novelty values indicate boundaries between music segments. To calculate novelty values, we convolute the self-similarity matrix with a radially-symmetric Gaussian taper [8]. Theoretically, if the size of a music segment is L, the most appropriate size of the checkerboard kernel is $2L + 1$. Although we do not know the size of music segments, we know that a reasonable music segment often ends at the end of eight beats. With the dominant period T determined by the method in Section III.C, we set the size of the checkerboard kernel as $16T + 1$. The novelty values of the ith audio frame is then calculated as

$$Novelty(i) = \sum_{m=-M/2}^{M/2} \sum_{n=-M/2}^{M/2} K(m,n) \cdot s(i+m, i+n), \quad (19)$$

where $M = 16T + 1$ and K denotes the checkerboard kernel.

We adopt the hill climbing algorithm again for detecting peaks from the sequence of novelty values. These peaks are denoted as $P = (p_0, p_1, ..., p_N)$, which is sorted in descending order according to the corresponding novelty value, and p_i denotes the timestamp of the ith peak. To keep representative peaks in P and avoid too short music segments, we design Algorithm 3. To define the threshold δ_s, we observe music videos produced by professional editors, and set it as twice of eight beats. The length of an eight beats can be calculated as eight times of the dominant period.

B. Video Clip Selection

For every music segment, we select a video clip from the database that has the best degree of matching to it. Assume that a music segment of length ℓ is shorter than the video clip. Therefore, from the video clip we would like to find a video segment that best matches with the music segment. The method of finding the best shift Δ^* in Section IV.B is again adopted to find a video segment ranging from Δ^* to $\Delta^* + \ell$.

To generate a music video that includes video segments of similar rhythm but from different dancers' performances, we avoid that the same video segment is selected by more than one music segment. Algorithm 4 is designed to accomplish music video generation. Assume that there are N_m music segments and there are N_v videos in the database. For every music segment, we calculate DOMs (eqn. (17)) between it and every videos. The value $DOM[i][j]$ denotes the DOM between the ith music segment and its most appropriate video segment deriving from the jth video. We use a boolean vector V_s to record whether the videos have been selected by a music segment, and a boolean vector M_s to record whether the music segments have selected videos. Algorithm 4 is designed based on the greedy strategy that maximizes the sum of DOMs for all music segments.

```
Algorithm 3: Boundary finding based on novelty
Input: novelty peaks $P = \{p_0, p_1, ..., p_N\}$
Output: a set of boundaries $B_s$
$B_s \leftarrow \varnothing$
$B_s = B_s \cup \{p_0\}$
for $i = 0$ to $N$
    $short \leftarrow False$
    for $j = 0$ to $sizeof(B_s) - 1$
        if $|p_i - B_s[j]| < \delta_s$
            $short \leftarrow True$
            break
    end for
    if $!short$
        $B_s = B_s \cup \{p_i\}$
end for
```

```
Algorithm 4: Music video generation
Input: DOMs between music segments and video segments
Output: a set of video segments $S$ that constitute the music video
$S \leftarrow \varnothing$
$V_s \leftarrow$ an array with each entry inserted by $False$
$M_s \leftarrow$ an array with each entry inserted by $False$
while $i < N_m$
    $domIdx \leftarrow \varnothing$
    for $j = 1$ to $N_m$
        if $M_s[j] = False$
            $domIdx[j] = \arg\max_{1 \le k \le N_v, k \notin \mathcal{A}} DOM[j][k]$,
                $\mathcal{A} = \{a | V_s[a] = True\}$
            $mIdx = \arg\max_{1 \le k \le N_s} DOM_k[domIdx[k]]$
    end for
    $vIdx = domIdx[mIdx]$
    $M_s[mIdx] \leftarrow True,\ V_s[vIdx] \leftarrow True$
    $S[mIdx] = vIdx$
    $i = i + 1$
end for
```

VI. Experiments

A. Evaluation Dataset

Table 1 lists information of the three datasets used in evaluation. The first dataset is captured from two people's dances according to six different music pieces, with a relatively simple background (c.f. Figure 6(a)). They just perform simple periodic movement to be the reference dataset for evaluating ROM extraction. Videos in the first two datasets were captured from dancers in the street dance club of our university. Each of them has taken at least two years of dancing training. The second dataset includes eleven different dancers' performances, and was captured in a much cluttered environment, as shown in Figure 6(b). According to five music pieces, these dancers perform in their preferable ways (hip-hop, popping, locking, or freestyle) and dance for 30 to 40 seconds. Numbers of different types of dances are listed in Table 2. Different from the first two datasets, the third dataset includes clips downloaded from the web and is much more challenging (Figure 6(c)). Multiple professional dancers dance in cluttered environments, and some of them dance for more than one minute. All videos in the evaluation datasets are coded as MPEG-4 videos, with 320×240 resolution. These datasets and experimental results described in the following are available on our website: http://www.cs.ccu.edu.tw/~wtchu/projects/ROM/index.html.

Extracting rhythm information from these videos is very challenging. We see apparent and time-varied shadows in Figure 6(a). In Figure 6(b), dancers may have different scales of motions, and motion may appear in anywhere on the screen. In the third dataset, not all dancers move accurately as music beats, and different dancers may have different dancing steps. Quality of videos in the third dataset is not as good as that in others. Moreover, sort of global motion caused by camera moving can be seen in both the second and the third datasets.

To verify the motivation of background music replacement, we exploit the package developed in [40] to assess quality of background music in the second dataset, in terms of the average perceptual similarity measure (PSM) [40]. The PSM value ranges from 0 to 1, and a higher value indicates larger correlation between the original one and the degraded version. From the experiments in [40], six audio signals used for evaluating low bit-rate audio codecs by ITU and MPEG have PSM values ranging from 0.88 to 1. In our case, the average PSM value of the background music is 0.68. By comparing these two cases, we see that quality of background music is significantly downgraded, and thus replacing it with higher-quality music would be valuable.

Table 1. Information of evaluation datasets.

	1st dataset	2nd dataset	3rd dataset
# video clips	30	50	13
Average length	11 sec	35 sec	1 min 23 sec
Multiple dancers	No	No	Yes
Cluttered background	No	Yes	Yes
Dancing types	Simple periodic movement	Hip-Hop, Popping, Locking, and Freestyle	Hip-Hop, Popping, Locking, and Freestyle
Dancers	Dancers from street dance club	Dancers from street dance club	Professional dancers
Source	Capturing	Capturing	Downloaded from the web

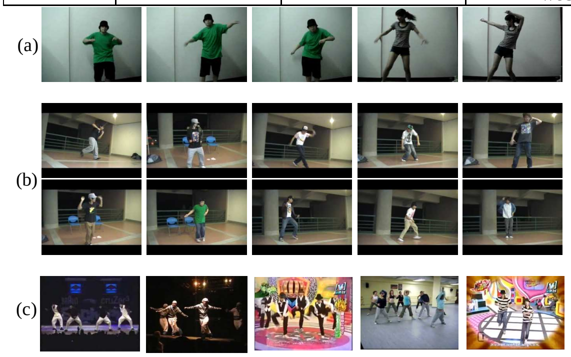

Figure 6. Snapshots of (a) the first, (b) the second, and (c) the third evaluation datasets.

B. Performance of ROM Extraction

A detected motion beat is claimed as correctly detected if the temporal distance between it and a truth beat is less than two video frames, i.e. $2/30$ seconds in 30-fps videos. Ground truths of motion beats were manually defined frame by frame, by the second author who had taken dancing training for years. We calculate average accuracy of motion beat detection for the 30 video clips in the first dataset, with various settings of the following parameters: 1) the definition of neighborhood $\mathcal{N}(s'_i)$ in eqn. (2); 2) the degree of smoothness controlled by σ in eqn. (4); 3) the threshold ϵ in Algorithm 2 for detecting turning points in trajectories; and 4) the parameter α in eqn. (12) for filtering out outliers in motion beat candidates.

Figure 7 shows performance in terms of precision, recall, and F-measure. From Figure 7(a), we see that the detection performance varies slightly when the radius of neighborhood is larger than three pixels. Similar effects can be observed from other sub-plots of Figure 7. This means the proposed method has stable performance once parameters in an appropriate range are set. In the following experiments, these four parameters are chosen as $r = 4$, $\sigma = 9$, $\epsilon = 0.6$, and $\alpha = 0.25$.

Generally, the proposed method has higher recall than precision. We estimate the fundamental period from the constructed sinusoid, and thus describe repeated characteristics of the signal. More truth beats can be detected if the reference beats are better estimated, and therefore the recall rate increases. In the developed applications, we prefer to detect motion beats as many as possible for providing finer ROM. If the music well matches strong motion beats, humans may be highly satisfied with the manipulated videos. That is why the average value 0.5 in F-measure is enough for the following applications.

Based on the second dataset, we compare motion beats detected by three different methods: (1) detection based on motion magnitude difference (baseline), (2) detection based on luminance difference [15], and (3) our approach – motion trajectory analysis. Figure 8 shows the best F-measure values achieved by the three methods are 0.13, 0.18, and 0.58, respectively. Guedes estimated motion magnitude by luminance changes between frames [15], and then estimated the dominant frequency from motion magnitude evolution. However, in the second dataset, dance videos were captured in uncontrolled environments and varied luminance changes hurt Guedes's approach. The proposed method analyzes motion trajectories and thus can more reliably capture motion beats.

Figures 9(c), (e), and (g) show frames right at the detected motion beats, and Figures 9(b), (d), (f), and (h) show frames in-between motion beats. We see that movements at detected motion beats are really stops of movements or ends of postures.

We further evaluate the proposed method for videos consisting of multiple dancers and lasting for more than one minute. Similar to the demand of stationary properties in digital signal processing, the proposed method only works well for video clips with stationary motion beats. Therefore, videos longer than one minute are appropriately segmented in advance, and in each segment motion beats are stationary. Figure 10 shows the average precision, recall, and F-measure for the third dataset, respectively. Our method has slightly higher precision, but performs significantly better in recall. For the videos in the third dataset, Guedes's approach does not have clear advantage over the baseline approach. By comparing Figure 8 with Figure 10, we confirm that extracting motion beats in videos with multiple dancers is much harder than that with single dancer.

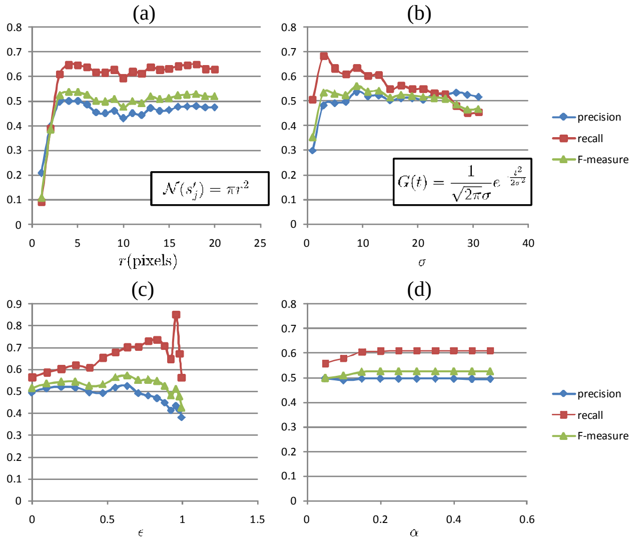

Figure 7. Performance of motion beat detection in terms of precision, recall, and F-measure, under different parameter settings.

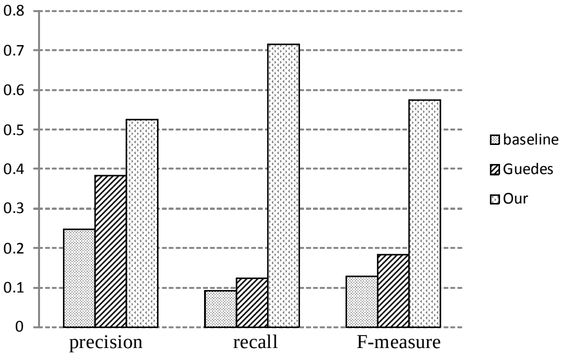

Figure 8. Performance comparison of ROM extraction for the second dataset.

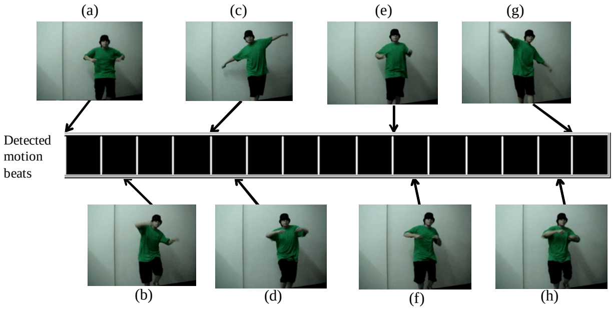

Figure 9. A sequence of video frames and the corresponding motion beats.

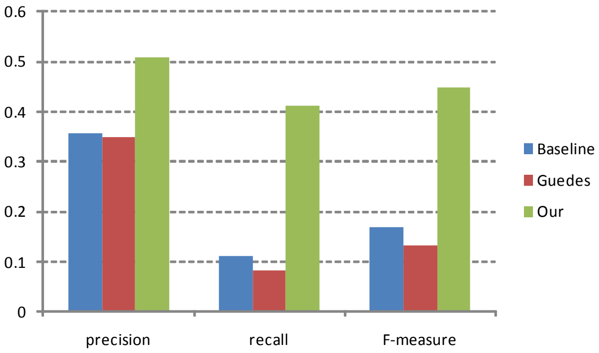

Figure 10. Performance comparison of ROM extraction for the third dataset.

C. Performance of Background Music Replacement

Performance of background music replacement is hard to be measured, because the judgement is subjective and the ground truth is hard to be formulated. Moreover, not every music beat is interpreted by dancers and different dancers may interpret differently, which make quantitative measurement infeasible. Therefore, we conduct subjective tests on the basis of replacement results for the second dataset.

Two sets of subjects were invited in the subjective evaluation: twenty ordinary users who had varied musical knowledge and were not familiar with street dance, and eleven dancers from the street dance club of our university who had taken dancing training for years. The former set of subjects was invited to verify whether the proposed method generally achieves satisfactory performance for ordinary users. Basic musical and choreographic knowledge was introduced to them before the test. The latter set of subjects was invited to examine finer rhythmic relationship between video and music. We separately describe two experiments as follows.

Ordinary users' evaluation:

The questionnaire for ordinary users is designed as:

Q1: Do you think the videos with background music replacement provide better viewing experience than the original videos? (Yes/No)

Q2: According to how the dancer moves with the rhythm of music (caused by drum, cymbal, etc.), evaluate how close the video with background music replacement is to the original video. The score ranges from one to five, and a higher score means "rhythmic properties between music and motion" is closer to the original video.

Q3: According to how the dancer moves with the emotion of music content (derived from melody, vocal, lyric, etc.), evaluate the degree of satisfaction of the video with background music replacement. The score ranges from one to five, and a higher score means higher satisfaction.

Q4: Rank videos generated by the three methods in Section VI.B. The value of rank ranges from one to three in integral, and a smaller value means higher preference.

We conduct background music replacement based on ROM extracted by motion magnitude difference (baseline), Guede's approach, and our approach, respectively. In subjective tests, we follow the DSIS (Double Stimulus Impairment Scale) scheme defined in ITU-R Recommendation BT.500-11. The original video was played first, followed by the result generated based on one of the three approaches.

For the first question, 87.5% of videos with background music replacement are thought to provide better viewing experience. This result confirms that it is worth to conduct background music replacement.

Table 2 shows the results of Q2 and Q3 for different dance styles. The standard deviations of scores are reported in parentheses. Videos in the second dataset can be divided into four sub-categories: hip-hop, popping, locking, and freestyle. Hip-hop is a dance style focusing on grooving and interpreting drums in music. Popping consists of pop, wave, and stopping poses, which is able to describe music beats well. Locking is about arm twisting, kick, point, and elastic movements. Locking is funky, and dancers often pay attention to moments of music beats appearance. Freestyle does not have major movements, but focuses on how to precisely interpret music emotion represented by melody, vocal, etc. Overall, our method jointly considers evolutions of motion magnitude and orientation, and more accurately extracts rhythm of motion to facilitate better background music replacement.

For hip-hop, our approach does not have clear superiority over other methods. In general, hip-hop movements not only interpret music beats, but also interpret progress between music beats. Our current method focuses on time instants of motion beats and music beats, and a further study about progress between beats is needed in the future. We achieve good performance for popping and locking. Dancers with such styles strike strong motion beats according to music beats caused by percussion instruments. We have much better performance for freestyle dances, which focus on artistic conception conveyed in music content. Generally, different dance styles affect ROM extraction and background music replacement.

Figure 11 shows results of Q4. We clearly see that our approach is the most preferable expect for hip-hop dances, which confirms the trend shown in Table 2.

Table 2. Subjective performance of BGM replacement evaluated by ordinary users.

	Hip-Hop (12)	Popping (13)	Locking (13)	Freestyle (12)	Overall (50)
Q2 (base)	3.29 (0.43)	3.13 (0.58)	3.08 (0.63)	3.00 (0.84)	3.12
Q2 ([15])	3.37 (0.44)	3.05 (0.55)	3.05 (0.65)	2.89 (0.53)	3.08
Q2 (Our)	**3.37** (0.35)	**3.26** (0.46)	**3.42** (0.52)	**3.63** (0.48)	**3.42**
Q3 (base)	3.24 (0.54)	3.30 (0.53)	3.08 (0.57)	3.10 (0.74)	3.17
Q3 ([15])	3.32 (0.53)	3.19 (0.60)	2.98 (0.63)	3.01 (0.53)	3.11
Q3 (Our)	**3.33** (0.45)	**3.41** (0.59)	**3.40** (0.57)	**3.61** (0.52)	**3.44**

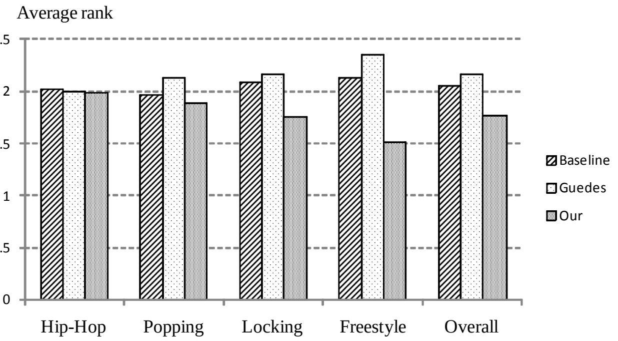

Figure 11. Ordinary users' preference on BGM replacement results for different dance styles.

Dancers' evaluation:

Because dancers have richer musical and choreographic knowledge, more detailed evaluation can be conducted. To observe detailed rhythm relationship between video and music, the second question Q2 was divided into two finer questions:

Q2-1: According to how the dancer moves with the dominant rhythm of music[1], evaluate how close the video with background music replacement is to the original video.

Q2-2: According to how the dancer moves with the characteristic rhythm of music[2], evaluate how close the video with background music replacement is to the original video.

The question Q1 does not need to be measured again, because this application is intuitive to dancers. Table 3 provides the evaluation results from dancers for Q2-1, Q2-2, and Q3. Our method also has promising performance based on dancer's evaluation. The performance for Q2-1 is better than that for Q2-2, which confirms that dominant rhythm is easier to be detected than characteristic rhythm. The results for Q3 are worse than Q2-1 and Q2-2. It is reasonable because Q3 is related to music emotion, which has not been considered currently.

Figure 12 shows dancer's preference on replacement results for different dance styles. These results are similar to that in ordinary user's evaluation. However, for popping our ranking result is worse than the baseline. Popping contains lots of static poses, which facilitate motion beat detection by the baseline approach. In Table 3, for Q2-1 the baseline method achieves better performance in popping, which corresponds to the ranking result in Figure 12. Overall, our method has better performance for all dance styles except for popping. The performance variation between ordinary users and dancers reveals their knowledge gaps on music and choreography.

Table 3. Subjective performance of BGM replacement evaluated by dancers.

	Hip-Hop	Popping	Locking	Freestyle	Overall
Q2-1 (base)	3.48 (0.06)	**3.30** (0.10)	3.18 (0.40)	2.70 (0.26)	3.16
Q2-1 ([15])	3.33 (0.40)	2.93 (0.23)	3.09 (0.10)	2.80 (0.10)	3.03
Q2-1 (Our)	**3.63** (0.17)	3.23 (0.12)	**3.36** (0.16)	**3.60** (0.26)	**3.45**
Q2-2 (base)	3.67 (0.29)	**3.50** (0.10)	3.39 (0.28)	2.83 (0.29)	3.34
Q2-2 ([15])	3.41 (0.45)	3.33 (0.31)	3.09 (0.24)	2.97 (0.06)	3.19
Q2-2 (Our)	**3.85** (0.28)	3.23 (0.61)	**3.61** (0.10)	**3.73** (0.31)	**3.60**
Q3 (base)	3.41 (0.17)	3.40 (0.26)	3.09 (0.24)	2.57 (0.31)	3.11
Q3 ([15])	3.26 (0.55)	3.17 (0.38)	3.09 (0.24)	2.63 (0.06)	3.03
Q3 (Our)	**3.44** (0.19)	**3.47** (0.31)	**3.48** (0.14)	**3.50** (0.35)	**3.48**

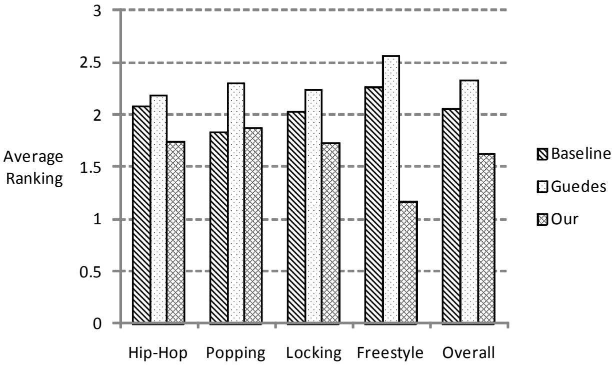

Figure 12. Dancer's preference on BGM replacement results for different dance styles.

D. Performance of Music Video Generation

Evaluating music segmentation is subjective, and the performance may differ from different music types and applications. In our work, we provide an evaluation guide as in Table 4 to reduce variations of subjective evaluation. If the difference between the best boundary and a detected boundary is smaller than twice of the dominant period, the detected boundary is claimed to be *close* to the best boundary. For the second dataset, the average score is 3.104, i.e. most boundaries are given scores over three and are located at music beats.

To verify that the proposed rhythm-based music video generation is attractive, we compare music videos generated by Algorithm 4 with that generated by randomly selecting a video segment to a music segment. Ten music videos are generated by two approaches, respectively. The observers were asked to evaluate whether the selected video segments are suitable for the background music, and give a score ranging from one to five (a higher score means higher satisfaction). Overall, our music videos obtain 3.42 on average, while the music videos generated by random selection obtain 2.46 on average. The score is especially high if ROM in the selected video segment is a multiple of that of the background music.

Table 4. The guideline for evaluating music segmentation.

Score	Description
5	The boundary is accurately located at the best boundary (music beat) between music segments.
4	The boundary is located at a music beat, which is not the best but is close to the best boundary.
3	The boundary is not located at a music beat but is close to the best boundary.
2	Although the boundary is located at a music beat, it is far from the best boundary.
1	The boundary is not located at a music beat, and it is far from the best boundary.

E. Discussion

We describe limitation of our current work in the following:

- In videos with substantial lighting changes, to our best knowledge there is no robust method to extract motion trajectories. Much more advancement should be made, and this issue cannot be addressed by our current paper.
- Noisy trajectories influence performance, and that is why we do not achieve perfect ROM extraction (Figure 8). Dancers often have violent and non-rigid movements, which makes significant challenges in trajectory extraction.
- In contrast to music rhythm that has been studied for a century, currently the extracted ROM is poorer. For example, different body parts may synchronize to different levels of music rhythm [30]. A dancer may move the main trunk with the base pulse, but arms or legs move more drastically at a finer metrical level. In this work we just extract one dominant period from various motion. Extracting motion at different metrical levels may be achieved if motion sensors

[1] The music beats produced by drum form the dominant rhythm of music. They are strong and repeat with a fixed period. If the speeds of two music pieces are the same, their dominant rhythms are identical.

[2] The music beats produced by cymbal and snare-drum form the characteristic rhythm of music. They are relatively weaker than the dominant rhythm. Two music pieces that have the same dominant rhythm may have different characteristic rhythm, depending on arrangement of music.

are attached to human body.

While the current work is limited by the aspects mentioned above, we also point out a few extensions:

- The proposed rhythm-based analysis can be extended to more applications. For example, as we have developed a way to transform videos and music into rhythm sequences, and have designed a metric to evaluate cross-media similarity, we are able to retrieve videos by giving a musical query or retrieve music by giving a video query. Rhythm-based cross-media retrieval would be a new way to retrieve media that have clear periodic or rhythmic content.
- Another plausible extension is surveillance video analysis. By analyzing periodic changes of motion from specific objects or humans, events such as person walking/running or car entering through a gate can be detected.

Rhythmic patterns can be found in various media, such as motion in videos, beats in music, and emphasized tones in speech. For a specific domain, rhythm information may be clear and can be explicitly extracted. However, for media that are disordered, the proposed techniques may make no sense. The former perspective shows the feasibility of the proposed idea, while the later perspective gives the limitation.

VII. Conclusion

We have presented associating rhythm of motion with rhythm of music to facilitate rhythm-based multimodal analysis. We devise a method to reliably extract rhythm of motion from motion trajectories. This approach well captures finer human motion, especially periodic motion changes in dance videos. Dance videos and music are respectively transformed into motion beat and music beat sequences, and are accordingly compared and aligned. We demonstrate effects of rhythm-based cross-media alignment with the applications of background music replacement and music video generation. The objective evaluation shows promising performance of rhythm of motion extraction. We also show that video with background music replacement really provides better viewing experience, while the impacts of different dance styles may be varied. Another subjective evaluation verifies that rhythm information provides useful clues to generate rhythmic musical videos.

Acknowledgement

The authors would like to thank Yi-Sheng Chang for conducting parts of experiments. The authors would also like to thank anonymous reviewers for giving valuable comments. The work was partially supported by the National Science Council of Taiwan, Republic of China under research contract NSC 100-2221-E-194-061 and NSC 99-2221-E-194-036.

References

[1] T.-J. Borer, "Motion Vector Field Error Estimation," *U.S. Patent 6442202B1*, 2002.

[2] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm," *Intel Corporation Microprocessor Research Labs*, 2000.

[3] A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe, "EyesWeb: Toward Gesture and Affect Recognition in Interactive Dance and Music Systems," *Computer Music Journal*, 24(1), pp. 57-69, 2000.

[4] R. Cutler and L.S. Davis, "Robust Real-Time Periodic Motion Detection, Analysis, and Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp. 781-796, 2000.

[5] H. Denman, E. Doyle, A. Kokaram, D. Lennon, R. Dahyot, and R. Fuller, "Exploiting Temporal Discontinuities for Event Detection and Manipulation in Video Streams," In *Proceedings of ACM International Workshop on Multimedia Information Retrieval*, pp. 183-192, 2005.

[6] S. Dixon, "Automatic Extraction of Tempo and Beat from Expressive Performances," *Journal of New Music Research*, 30(1), pp. 39-58, 2001.

[7] W.J. Dowling and D.L. Harwood, *Music Cognition*. Academic Press, 1985.

[8] J. Foote, M. Cooper, and A. Girgensohn, "Creating Music Videos Using Automatic Media Analysis," In *Proceedings of ACM Multimedia*, pp. 553-560, 2002.

[9] R. Gauldin, *Harmonic Practice in Tonal Music*. W.W. Norton & Company, 2nd edition, 2004.

[10] R.I. Godoy, "Gestural Imagery in the Service of Musical Imagery," *Lecture Notes in Computer Science*, 2915, pp. 55-62, 2004.

[11] S.M. Goldfeld, R.E. Quandt, and H.F. Trotter, "Maximization by Quadratic Hill-Climbing," *Econometrica*, 34(3), pp. 541-551, 1966.

[12] F. Gouyon and S. Dixon, "A Review of Automatic Rhythm Description Systems," *Computer Music Journal*, 29(1), pp. 34-54, 2005.

[13] P. Grosche and M. Muller, "A Mid-Level Representation for Capturing Dominant Tempo and Pulse Information in Music Recordings," In *Proceedings of International Society for Music Information Retrieval*, pp. 189-194, 2009.

[14] P. Grosche, M. Muller, and C.S. Sapp, "What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas," In *Proceedings of International Society for Music Information Retrieval*, pp. 649-654, 2010.

[15] C. Guedes, "Extracting Musically-Relevant Rhythmic Information from Dance Movement by Applying Pitch Tracking Techniques to a Video Signal," In *Proceedings of Sound and Music Computing Conference*, pp. 25-33, 2006.

[16] X.-S. Hua, L. Lu, and H.-J. Zhang,"Automatic Music Video Generation Based on Temporal Pattern Analysis," In *Proceedings of ACM Multimedia*, pp. 472-475, 2004.

[17] T.-H. Kim, S.-I. Park, and S.Y. Shin, "Rhythmic-Motion Synthesis Based on Motion-Beat Analysis," *ACM Transactions on Graphics*, 22(3), pp. 392-401, 2003.

[18] I. Laptev, S.J. Belongie, P. Perez, and J. Wills, "Periodic Motion Detection and Segmentation via Approximate Sequence Alignment," In *Proceedings of International Conference on Computer Vision*, 2005.

[19] M. Leman, *Embodied Music Cognition and Mediation Technology*. MIT Press, 2007.

[20] J.S. Marques and L.B. Almeida, "Frequency-Varying Sinusoidal Modeling of Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(5), pp. 763-765, 1989.

[21] J. Min, R. Kasturi, and O. Camps, "Extraction and Temporal Segmentation of Multiple Motion Trajectories in Human Motion," In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 118–122, 2004.

[22] J.L. Oliveira, F. Gouyon, L.G. Martins, and L.P. Reis,"IBT: A Real-Time Tempo and Beat Tracking System," In *Proceedings of International Society for Music Information Retrieval*, 2010.

[23] J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Prentice Hall, 1999.

[24] B.M. Sadler and S.D. Casey, "On Periodic Pulse Interval Analysis with Outliers and Missing Observations," *IEEE Transactions on Signal Processing*, 46(11), pp. 2990-3002, 1986.

[25] E.D. Scheirer, "Tempo and Beat Analysis of Acoustic Musical Signals," *The Journal of the Acoustical Society of America*, 103(1), pp. 588-601, 1998.

[26] W.A. Sethares, *Rhythm and Transforms*. Springer, 2007.

[27] J. Shi and C. Tomasi, “Good Features to Track,” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[28] T. Shiratori, A. Nakazawa, and K. Ikeuchi, “Rhythmic Motion Analysis Using Motion Capture and Musical Information,” In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 89-92, 2003.

[29] C.-W. Su, H.-Y. M. Liao, H.-R. Tyan, C.-W. Lin, D.-Y. Chen, and K.-C. Fan, “Motion Flow-Based Video Retrieval,” *IEEE Transactions on Multimedia*, 9(6), pp. 1193-1201, 2007.

[30] P. Toiviainen, G. Luck, and M. Thompson, “Embodied Meter: Hierarchical Eigenmodes in Music-Induced Movement,” *Music Perception*, 28(1), pp. 59-70, 2010.

[31] J. Wang, C. Xu, E. Chng, L. Duan, K.-W. Wan, and Q. Tian, “Automatic Generation of Personalized Music Sports Video,” In *Proceedings of ACM Multimedia*, pp. 735-744, 2005.

[32] J.-C. Yoon, I.-K. Lee, and S. Byun, “Automated Music Video Generation Using Multi-Level Feature-Based Segmentation,” *Multimedia Tools and Applications*, 41(2), pp. 197-214, 2009.

[33] J.-C. Yoon, I.-K. Lee, and H.-C. Lee, “Feature-Based Synchronization of Video and Background Music,” *Lecture Notes in Computer Science*, 4153, pp. 205-214, 2006.

[34] F. Eyben, S. Bock, B. Schuller, and A. Graves, “Universal Onset Detection with Bidirectional Long Short-Term Memory Neural Networks,” In *Proceedings of International Society for Music Information Retrieval Conference*, pp. 589-594, 2010.

[35] J. Feng, B. Ni, and S. Yan, “Auto-Generation of Professional Background Music for Home-Made Videos,” In *Proceedings of International Conference on Internet Multimedia Computing and Service*, pp. 15-18, 2010.

[36] H.-C. Lee and I.-K. Lee, “Automatic Synchronization of Background Music and Motion in Computer Animation,” In *Proceedings of Eurographics*, pp. 353-362, 2005.

[37] J.-C. Yoon and I.-K. Lee, “Synchronized Background Music Generation for Video,” In *Proceedings of International Conference on Advances in Computer Entertainment Technology*, pp. 353-362, 2005.

[38] J.-I. Nakamura, T. Kaku, K. Hyun, T. Noma, and S. Yoshida, “Automatic Background Music Generation based on Actors’ Mood and Motions,” *The Journal of Visualization and Computer Animation*, 5, pp. 247-264, 1994.

[39] L.G. Ratner, “Eighteen-Century Theories of Musical Period Structure,” *The Musical Quarterly*, vol XLII, no. 4, pp. 439-454, 1956.

[40] R. Huber and B. Kollmeier, “PEMO-Q – A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1902-1911, 2006.

Wei-Ta Chu received the B.S. and M.S. degrees in Computer Science from National Chi Nan University, Taiwan, in 2000 and 2002, and received the Ph.D. degree in Computer Science from National Taiwan University, Taiwan, in 2006. Since 2007, he has been the Assistant Professor in the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan. His research interests include digital content analysis, multimedia indexing, digital signal process, and pattern recognition.

He won the Best Full Technical Paper Award in ACM Multimedia 2006. He was a visiting scholar at Digital Video & Multimedia Laboratory, Columbia University, from July to August 2008. He serves as an editorial board member for Journal of Signal and Information Processing, and guest editors for Advances in Multimedia and IEEE Transactions on Multimedia.

Shang-Yin Tsai received the B.S. and M.S degrees in Computer Science from National Chung Cheng University, Taiwan, in 2008 and 2010. His research interests include digital content analysis and multimedia systems.