Wenjun Zeng
University of Missouri-Columbia

Visual Rhythm Detection and Its Applications in Interactive Multimedia

Trista P. Chen, Ching-Wei Chen, Phillip Popp, and Bob Coover
Gracenote

Although music is typically considered a strictly auditory art form, human interaction with music consists of a much richer combination of auditory, visual, and physical experiences and responses. From traditional folk dance to classical ballet, or from the latest music videos to just getting down on the dance floor, physical movement and other visual stimuli have always been closely tied to the musical art form. The link between all of this is rhythm. Rhythm has been extensively studied in the fields of audio signal processing and music information retrieval, and many systems now can automatically extract the rhythm and tempo from an audio signal. In this article we present the concept of visual rhythm, and describe two systems that use it.

In music, *rhythm* refers to the repetitive pattern of melodies, phrases, or percussive events that form a piece's structure. It emerges from temporal elements such as the onset of a drum, a note played on an instrument, or the silence between two notes. Underneath the rhythm is a regularly spaced grid that organizes when, and in what succession, these elements occur. In western musical terms, the grid is referred to as the *meter*, while the speed at which the meter is traversed is the *tempo*. Much research has been done in the area of musical tempo and rhythm detection to automatically extract these significant rhythmic events from audio signals.[1] In their most basic form, automated methods search for musical onsets, characterized by a rapid increase in energy in the audio signal and then attempt to fit them to a grid. The best-fit grid inherently contains several useful pieces of information, such as a time signature and a tempo measured in beats per minute (BPM). Additionally, by searching for prominent, regularly spaced rhythmic elements, one can find the individual beats.

When we listen to music that we like, one of the most natural responses is for us to move our bodies. Those gifted with great coordination can perform an enticing salsa or a death-defying break-dance, while the less coordinated of us get by with a slight shuffling of the feet, a nodding of the head, or at least an inconspicuous tapping of the foot. But no matter what the level of our dancing ability, whenever we move along with music, what we are doing is constantly analyzing a stream of audio, searching for repeated patterns, and then adjusting our movements to match those patterns. Flipping this paradigm on its head, when we watch a silent video of a person dancing, we can similarly perceive a related rhythm and tempo from their movements. Such observations lead us to propose the concept of visual rhythm.

These rhythmic events might come from a wide variety of visual cues: a person dancing in front of a fixed camera, a camera capturing a fixed scene with rhythmic zooming or panning movements, or a video capture of a scene with periodic lighting changes. In other words, rhythmic changes in human

Editor's Note

True multimedia experience involves multimodal media and their intrinsic interactions. This article identifies rhythm as the link between physical movement, other visual stimuli, and the musical art form. It describes two example interactive multimedia applications in which visual rhythm is extracted for synchronization of music and video to allow more intuitive interaction between a user and both audio and video content, leaving the door open for many potential innovative applications.

—Wenjun Zeng

1070-986X/11/$26.00 © 2011 IEEE

Published by the IEEE Computer Society

movements, camera movements, and environmental lighting can all result in the perception of visual rhythm. Similar to musical tempo, the concept of *visual tempo* can be used to characterize visual rhythm by describing the rate at which rhythmic events occur.

Applications

The ability to detect visual rhythm can enable many novel multimedia applications that allow more intuitive interaction between a user and both audio and video content. Examples of such systems include automated music video editing, video jockey tools, video games, and other human-computer interface applications. We believe that successful multimedia applications of the future will be driven by a deep integration of audio and video technologies, to deliver a true multimedia experience.

While many use-cases exist, the two described here highlight cardinal applications in audiovisual alignment. The first, Visual Rhythm-based Audio Playback System (VRAPS), extracts visual rhythm from a live video stream and uses visual tempo to control the playback rate of a song.[2] The second, Automatic Music Video, extracts visual rhythm from a prerecorded video file and matches the visual tempo to the auditory tempo by time-stretching and time-compressing the video.

VRAPS

VRAPS is an interactive multimedia application that lets a user control the playback speed of an audio file by making rhythmic motions in front of a camera (see Figure 1). The faster the user moves or dances, the faster the music plays. To achieve this effect, VRAPS draws upon rhythmic information from both the audio and video signals. First, it extracts the audio tempo of the chosen song using well-known audio tempo extraction techniques.[3] Then, as the song begins playing back, VRAPS analyzes the user's movements from a video camera in real time, applying a visual rhythm-detection algorithm to determine a visual tempo. The audio signal is then stretched or compressed in time to match the audio tempo to the visual tempo. The time-stretching factor is adjusted continuously on the basis of the instantaneous visual tempo detected from the user's movements. Therefore, the user is able to interactively control the audio playback speed by changing the pace of his or her movements. VRAPS can ensure that every user is on beat and in time by continuously following their every move, despite any attempts to dance poorly.

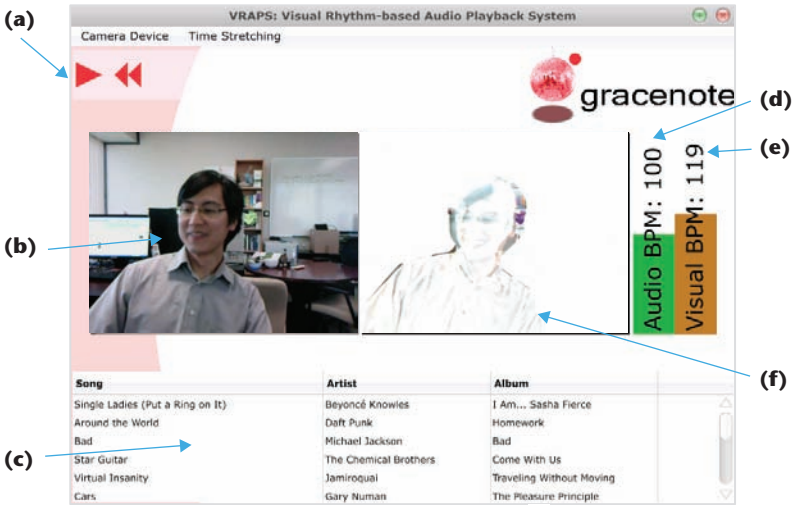

Figure 1. Visual Rhythm-based Audio Playback System user interface: (a) playback controls, (b) video signal from camera, (c) song list, (d) original audio tempo, (e) detected visual tempo, and (f) 2D visual feature.

Figure 2 (next page) outlines the process flow of the VRAPS system.

- The user selects a music file. Automatic rhythmic analysis techniques are applied to extract the audio tempo from the chosen music file. The system initially plays back the music at its original speed.
- The user then dances or makes rhythmic movements in front of a video camera. The system uses a visual tempo-extraction algorithm to detect the visual tempo present in the user's motion.
- This visual tempo is compared with the detected audio tempo, and the audio signal is either sped up or slowed down, using audio time-stretching techniques, to match the audio to the video tempo.
- The time-stretching factor is continuously updated on the basis of the instantaneous visual tempo detected from the user's movements. In this way, the users can interactively control the playback speed of the music by changing the speed of their rhythmic motion.

VRAPS works with standard PC hardware and a single webcam. The system doesn't require powerful hardware or stereo or 3D cameras, or sensor-based motion controllers like those used in Nintendo Wii, Microsoft Kinect,

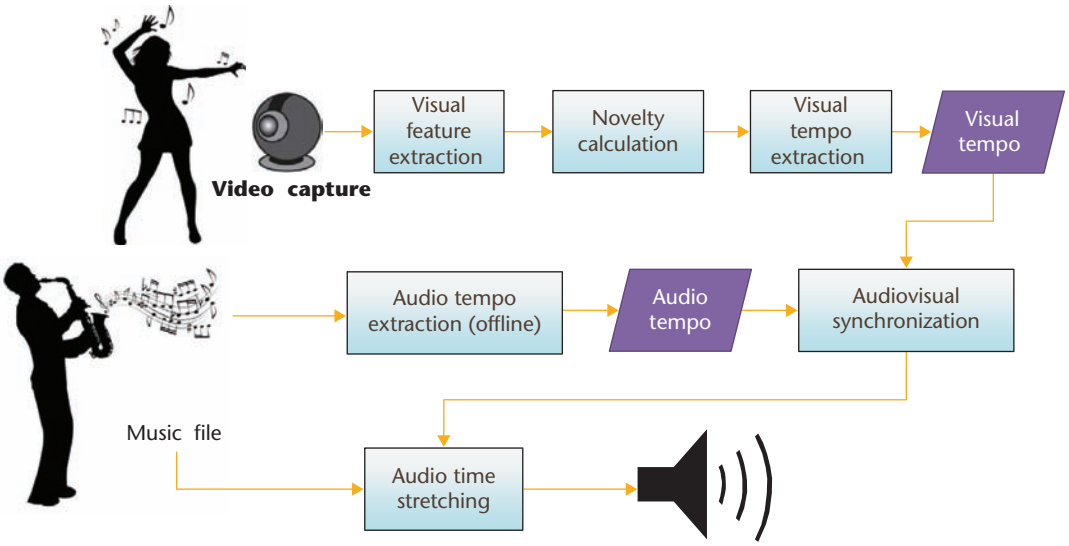

Figure 2. Visual Rhythm-based Audio Playback System flow chart.

or Sony Move. Nor does it require computationally intensive algorithms for gesture recognition.

Automatic Music Video

In a second demonstration, we use visual tempo to make a music video. The popularity of home-brewed music videos exploded in recent years, from the infamous *Auto-Tune the News* Web series to the music video mash-ups created from disparate video clips set to a beat. The concept of creating new music videos with different video data is not new. Beauregard et al. proposed intercutting user-supplied visual data with preexisting music videos.[5] Visual rhythm further empowers automated music-video creation by offering a common terminology, namely rhythm, through which the audio and video elements can communicate.

To create a rhythmically synchronized music video, we begin by extracting visual tempo and audio tempo from user-selected video and audio files. We then continuously adjust the video playback rate so that the local visual tempo matches the audio tempo. For example, a user-generated video of a bear bounding across a river displays a clear visual rhythm as the bear's body moves up and down in the water. The user might want to add a particular song as a soundtrack to this video before sharing it with friends. By estimating the visual tempo of the video and the audio tempo of the song, the system will determine a visual stretching factor by comparing the two tempos. We then speed-adjust the video to make the tempos equal by repeating or removing a given number of video frames on the basis of the visual stretching factor. Finally, we align the first auditory beat of the song to an early prominent visual beat to synchronize the two sources (see Figure 3). This process produces a tight coupling between auditory and visual cues, melding them into single perceptual events. This approach can obviously be applied to generate music videos from a variety of different video sources, which may include all types of rhythmic elements.

Visual tempo extraction

Visual tempo, the driving force behind these applications, is extracted by examining visual cues that correspond to motion and other perceptually significant changes in the video signal. We begin by describing two video features, absolute frame difference and optical flow, which capture many of the visual cues needed to extract visual tempo. They pick up on local cues, such as a dancing body or a waving hand, as well as global cues such as video cuts and lighting changes. To extract visual tempo from these features, we build off of previous work in music information retrieval and auditory tempo extraction. We analyze the 2D video features to derive a 1D novelty feature, then use traditional tempo-extraction methods, such as autocorrelation, to find the visual tempo.

Visual feature extraction

Two kinds of visual features are implemented in our demonstrations: the absolute frame difference (see Figure 4a) of two

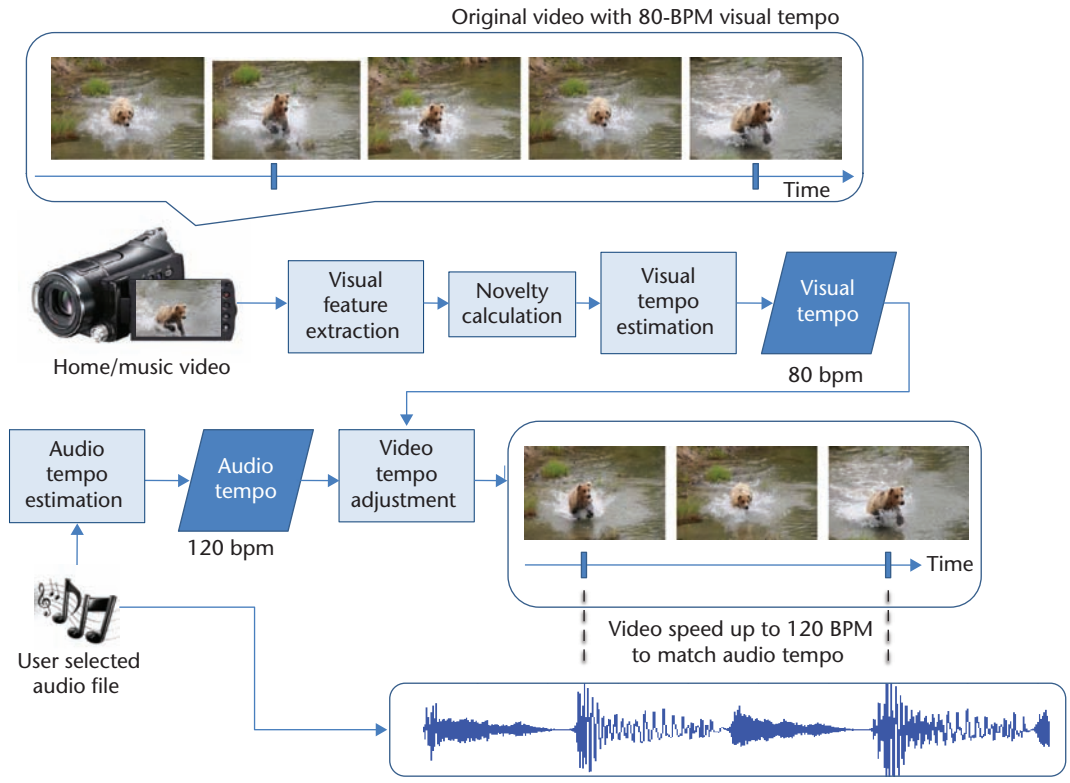

Figure 3. Automatic Music Video flow chart.

(a) **(b)** **(c)**

Figure 4. (a) Frame difference between frame t *and frame* (t + 1)*, (b) frame* t*, and (c) frame* (t + 1)*.*

consecutive frames (see Figures 4b and 4c), and 2D angle-magnitude histogram (see Figure 5a, next page) of optical flows (see Figure 5b). Both features characterize the dynamics of the overall picture without performing expensive, yet unreliable, body tracking in real time. Thus we avoid the computationally exorbitant blob and limb detection and tracking, as well as joint-angle estimation. Additionally, these features can accommodate multiple moving objects such as a group dance or rhythmic lighting effects.

The absolute frame difference of two consecutive frames is a fast and reliable way to highlight the transient characteristics of the overall picture. The motion of a bobbing head, as shown in Figure 4's frames (see Figures 4b and 4c), can easily be seen by the absolute pixel difference in the head contours in Figure 4a. Unfortunately, the absolute frame difference lacks stability when it comes to small disturbances in the environment. A more stable indicator can be derived from the optical flow feature. Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene. Given two consecutive video frames of an object in motion, optical flow analysis detects pixels that have changed position between the two frames. The optical flow feature consists of the 2D coordinates of the moving pixel in each of the successive

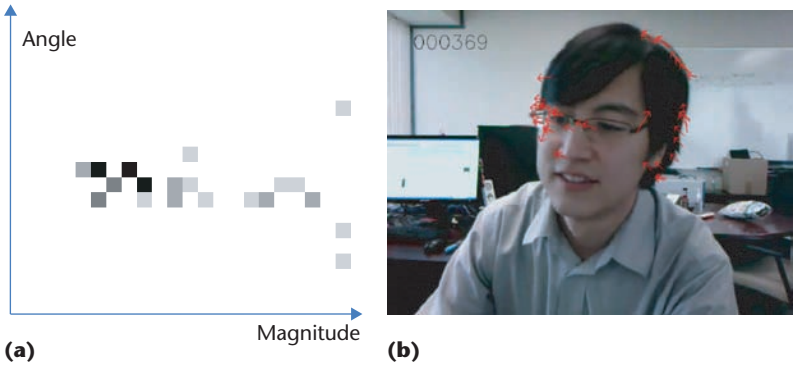

Figure 5. (a) 2D angle-magnitude histogram of optical flows and (b) a video frame with its corresponding optical flows.

frames. From these coordinates, we can compute an optical flow vector consisting of the angle and magnitude of the detected motion. There are several different methods for computing optical flow. In our system, we used the pyramidal implementation of the Lucas Kanade optical flow estimation algorithm.[4]

Having estimated a set of optical flow vectors for a two-frame sequence, we then derive statistics to describe the distribution of the overall motion between the frames. Ignoring the absolute locations of the moving pixels, we create a 2D histogram of the optical flow vectors, where the horizontal and vertical axes correspond to the magnitude and angle of the vector, respectively. The intensity in each bin in the histogram is computed by counting the number of optical flow vectors within the bounding angle and magnitude values for that bin (see Figure 5b). The darker the pixel, the more optical flow vectors are in the bin. The histogram shown in Figure 5b has angles ranging from $-\pi$ to π (top to bottom), and magnitudes ranging from three to 10 pixels (left to right).

Novelty calculation

Given either the absolute frame difference or the 2D histogram of the optical flows, we need to reduce the data to a 1D novelty feature that can be used with more traditional tempo-detection approaches from the music information retrieval domain. The 1D novelty feature derived from the absolute frame difference can be calculated as the energy of the difference frame. The 1D novelty feature derived from the 2D histogram of the optical flows can be calculated as the moment of the 2D histogram.

$$M = \sum_x \sum_y (x - x_0)(y - y_o)h(x, y)$$

where x and y correspond to coordinates in angle and magnitude, $h(x, y)$ is the bin count at the (x, y) coordinate, and (x_0, y_o) is the center coordinate of the 2D histogram. An example 1D novelty function is shown in Figure 6.

It should be noted that taking the energy or the moment are not the only ways to transform 2D features to 1D novelty features. Readers are encouraged to experiment.

Visual tempo calculation

Given the 1D novelty feature, we are ready to calculate the visual tempo. We measure the tempo in BPM and compute it by taking the autocorrelation of the 1D novelty function (among other more advanced methods[3]). While most music-tempo calculations assume stationary or nearly stationary tempos, VRAPS must handle a constantly varying visual tempo. Fitting our implementation to real-time applications such as VRAPS, a sliding window is used to compute the autocorrelation of the 1D novelty feature.

An example autocorrelation of the 1D novelty feature is shown in Figure 7. Given a video capturing frame rate at 24 frames per second and the first peak of the autocorrelation function at 29 frames, the visual tempo is then: 29 frames * 1/24 seconds/frames * 60 minutes/second = 72.5 BPM.

Synchronizing audio tempo to visual tempo

In the VRAPS application, given the original music BPM and the instantaneous visual BPM, the music needs to be stretched in time by a factor of BPM_{visual}/BPM_{audio}, where BPM_{visual} is the instantaneous tempo of the video signal, and BPM_{audio} is the overall tempo of the original audio signal. As the instantaneous tempo of the video signal changes, the playback speed of the audio signal will be adjusted by the calculated time-shifting factor. In a variant of this mode, the tempo of the audio signal is estimated on a continuous basis using a sliding analysis window.

If the ratio is larger than one, the music is sped up, or compressed; if the ratio is smaller than one, the music is slowed down, or expanded. This ratio is fed into a time compression and expansion algorithm to increase or decrease the audio playback rate. In particular, the synchronous over-lap add (SOLA) algorithm was chosen for its ability to compress or expand in real time without altering the pitch of the original audio.

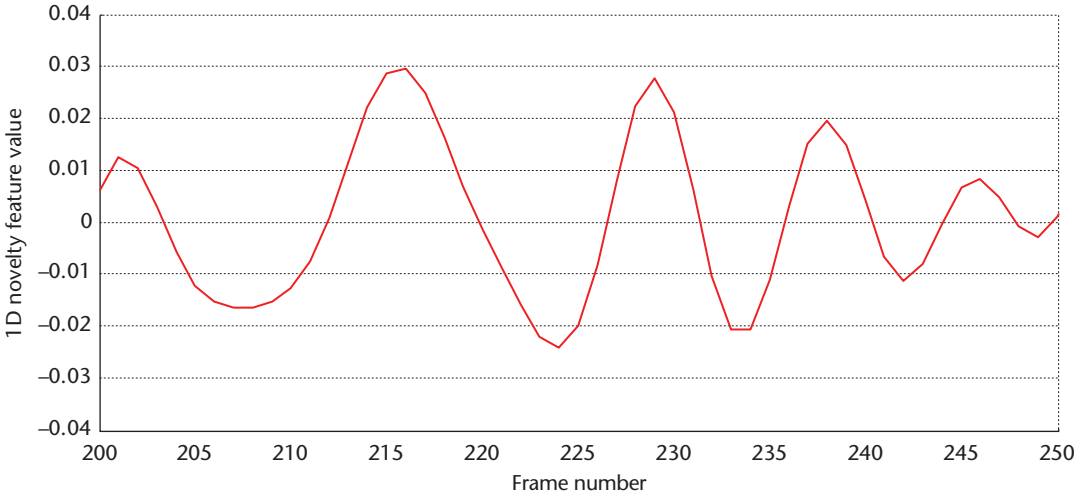

Figure 6. An example 1D novelty feature.

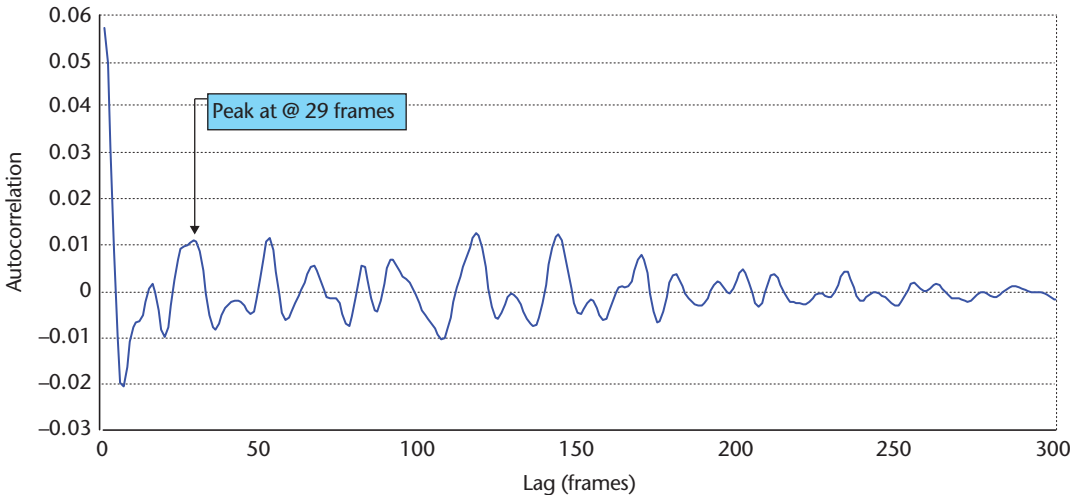

Figure 7. Autocorrelation of the 1D novelty feature.

SOLA achieves its pitch-preserving compression and expansion via a two-step process. First, it breaks up the original audio signal into sufficiently small segments. Then, it recreates the audio by piecing these segments back together. As it rebuilds, it will repeat segments of the audio when it needs to expand the audio, or skip segments of the audio when it needs to compress. As an example, if the ratio indicates 75 percent of the original tempo, then the algorithm will make the audio segments 1.0/0.75, or 133 percent of their original length. It does this by overlapping the segments by 33 percent and crossfading between them. To provide smooth transitions joining these segments, the algorithm performs a cross correlation between the overlapping portion of one segment and the next, searching for the location where they can be placed back together with the least amount of discontinuity.

The SOLA algorithm doesn't handle transients well, nor does it make any adjustments for the type of audio that it is trying to splice back together. To stretch audio containing a complete stereo mix including drums, two upgrades were made to the original SOLA algorithm. The first improvement changes the type of cross-fade used depending on the correlation between two segments. One segment is projected onto the next at the cross-fade point via a dot product to choose between an equal volume and an equal power cross fade window. This process removes most of the volume modulations that are typical artifacts of this type of algorithm. The second modification avoids repeating areas in the audio that have transients, for instance drum beats. If a portion of an audio block is marked as a transient, then that segment is only allowed to be repeated once in the case of stretching, and it is not

allowed to be skipped in the case of compressing. This avoids the echo, on drums and other transients, that is a typical artifact of the SOLA algorithm.

Other uses of visual rhythm

We believe that visual rhythm detection can enable many new interactive multimedia applications in addition to controlling audio playback and creating music videos. We have listed a few of these ideas here.

Music substitution in workout videos

Many people complain that watching the same workout video every day can become boring. If, instead, users could select songs from their own music collection rather than listen to the same old song on the DVD, they might be more motivated to keep up with their regimen. A rhythm-matching system could first analyze the desired workout video to detect the position of visual beats, as well as the visual tempo. Users could then direct the system to a collection of audio files whereupon traditional audio beat-detection and tempo-estimation techniques would be used to extract the rhythmic characteristics of each file. The system could then find candidate audio files that have the same audio tempo in BPM as the selected workout video. These audio files could then be substituted for the original soundtrack to provide a fresh workout video every day. To make sure the new piece of music and the video content blend naturally, the audio beats could be aligned with the visual beats.

Video jockey tools

Video jockeys and disc jockeys already collaborate to create alluring audio/visual experiences at nightclubs and music festivals. Visual rhythm could empower video jockeys in the same way that automatic audio rhythm extraction has empowered disc jockeys. It would give them a way to organize, index, and search their videos, as well as provide a common element with which to select and match visual and audio content. It would allow them to synchronize with the music being played as well as choose videos appropriate for a given audio tempo.

Camera-based dancing video game

In the real world, a dance student learns how to dance by mimicking an instructor's moves while the instructor gives feedback on how well the student follows his or her moves. In a game setting, such as in the popular arcade game *Dance Dance Revolution* (DDR), a specialized floor pad and visual arrow indicators are used to judge how well the gamer matches both the sequence and timing of the designated moves of the dance. While learning how to dance with a live instructor is probably more natural, the game setting might be more entertaining. We can use our visual rhythm-detection technique to enable rhythmic dance-based game interactions that are both natural and fun.

A possible visual-rhythm-based game might work as follows: without supplying any music or annotated arrows as DDR does, a dance video is simply presented on a screen. The player must follow the video by dancing or moving in time with the rhythm of the on-screen dancer. The player's movements are captured by a camera mounted on the game console. Visual beats from the captured video are detected in real time, and are matched to the visual beats from the dance video. The more closely the two visual beats match, the higher the gamer scores.

Conclusions

The temporal nature of visual and auditory signals is indisputable, and the interdependence between these two domains has many applications in the fields of musical performance, multimedia content creation, and human–computer interaction. However, the study of rhythm and tempo has until now been largely confined to the audio domain. We hope that our exploration of visual rhythm detection will inspire further research into rhythm and tempo as meaningful video descriptors for the next generation of multimedia applications.

There are a few areas where future work on the visual rhythm system would be beneficial. The first would be to go beyond detecting a more high-level visual tempo, to detect the precise timing of visual onset events that make up the visual tempo. These can be associated with sudden changes in the direction of motion, shot changes, or lighting effects. Knowing the exact temporal location of visual onsets would give a more granular description of the visual rhythm in the signal for use in synchronization or interaction with audio signals.

In addition to this, while the visual rhythm detection we have described is robust at detecting

binary (back and forth) rhythmic motion, further development is required to make the algorithm more robust at detecting visual tempo from more complicated periodic movements. We can also envision being able to capture the rhythmic pattern of a video signal, which characterizes the repetitive pattern of different visual events that make up more complicated motion such as dance movements and gestures.

Another area of improvement involves using more advanced techniques for the detection of both visual and audio tempo to circumvent the problem of recognizing something that is either double or half the tempo of the actual tempo. Both the audio and video tempo could use a probability-based method to determine the most likely tempo from among a set of candidate tempos.

We look forward to seeing additional creative uses of the visual rhythm concept in the future. **MM**

References

1. J.P. Bello et al., ''A Tutorial on Onset Detection in Music Signals,'' *IEEE Trans. Speech and Audio Processing,* vol. 13, no. 5, 2005.
2. T.P. Chen et al., ''VRAPS: Visual Rhythm-Based Audio Playback System,'' *Proc. Int'l Conf. Multimedia and Expo* (ICME), IEEE Press, 2010.
3. F. Gouyon and S. Dixon, ''Computational Rhythm Description,'' *Proc. 7th Int'l Conf. Music Information Retrieval,* Int'l Soc. Music Information Retrieval, 2006.
4. J.-Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm,* OpenCV distribution package 2000, Microprocessor Research Labs, Intel Corp., 2000.
5. G.T. Beauregard, S.K. Subramanian, and P.R. Kellock, *Creating a New Music Video by Intercutting User-Supplied Visual Data with a Pre-Existing Music Video,* US patent 2008/0016114 A1, Patent and Trademark Office, 17 January 2008.

Contact author Trista P. Chen at trista.chen@gmail.com.

Contact editor Wenjun Zeng at zengw@missouri.edu.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.