

Visual Rhythm and Beat

ABE DAVIS, Stanford University

MANEESH AGRAWALA, Stanford University

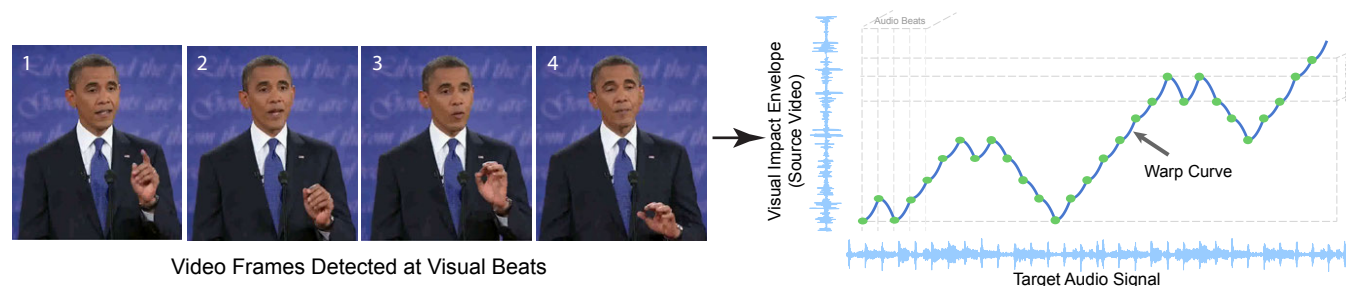


Fig. 1. **Accidental Dance and Unfolding** - we analyze visual rhythm in a collection of video to search for segments of unintentionally-rhythmic motion. Then, we warp those segments into precise temporal alignment with a musical target to synthesize dance performances. On the left we see video frames corresponding to the moments of four consecutive visual beats detected in a 2012 presidential debate video [WSJDigitalNetwork 2012]. These visual beats lie at the high and low points of a repetitive up-and-down hand gesture. On the right is a warp curve showing the process of unfolding, which synthesizes dance video corresponding to a random walk through the visual beats of a source segment.

We present a visual analogue for musical rhythm derived from an analysis of motion in video, and show that alignment of visual rhythm with its musical counterpart results in the appearance of dance. Central to our work is the concept of visual beats — patterns of motion that can be shifted in time to control visual rhythm. By warping visual beats into alignment with musical beats, we can create or manipulate the appearance of dance in video. Using this approach we demonstrate a variety of retargeting applications that control musical synchronization of audio and video: we can change what song performers are dancing to, warp irregular motion into alignment with music so that it appears to be dancing, or search collections of video for moments of accidentally dance-like motion that can be used to synthesize musical performances.

CCS Concepts: • **Computing methodologies** → *Computational photography; Image processing; Image-based rendering*; • **Human-centered computing** → Graphics input devices; Sound-based input / output;

Additional Key Words and Phrases: music, dance, video editing

ACM Reference Format:

Abe Davis and Maneesh Agrawala. 2018. Visual Rhythm and Beat. *ACM Trans. Graph.* 37, 4, Article 122 (August 2018), 11 pages. <https://doi.org/10.1145/3197517.3201371>

Authors' addresses: Abe Davis, Stanford University, abedavis@stanford.edu; Maneesh Agrawala, Stanford University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.

0730-0301/2018/8-ART122 \$15.00

<https://doi.org/10.1145/3197517.3201371>

1 INTRODUCTION

Music and dance are closely related through the concept of rhythm, which describes how events—e.g., the sound of an instrument or the movement of a body—are distributed in time. Rhythm is in some sense a very intuitive concept: infants can recognize and follow basic rhythms at as early as six months of age [Cirelli et al. 2016; Repp and Su 2013], and even some animals—certain parrots and elephants, for example—are known to move in time with simple music [Patel and Demorest 2013; Patel et al. [n. d.]]. However, the task of quantifying rhythm is not trivial, and has been the topic of extensive research in the context of both music [Böck and Gerhard Widmer 2013; Dixon 2006; Ellis 2007; Goto 2002; Grosche et al. 2010; Hu et al. 2017; Lerch 2012] and dance [Brick and Boker 2011; Dyaberi et al. 2006]. Our work builds on that research to explore a visual analogue for rhythm—which we call *visual rhythm*—in video. Just as musical rhythm captures the temporal arrangement of sounds, visual rhythm captures the temporal arrangement of visible motion. We focus on analyzing that motion to identify structure related to dance.

Our central hypothesis is that music and dance are characterized by complementary rhythmic structure in audible and visible signals. Our exploration of that structure builds on the concept of *visual beats*—visual events that, when temporally aligned with musical beats, create the appearance of dance. The relationship between visual and musical beats provides a starting point from which we derive visual analogues for other rhythmic concepts, including onset strength and tempo. Visual beats also give us a recipe for manipulating rhythmic structure in video: we first identify visual beats, then time-warp those beats into alignment with a specified target. Provided we are able to identify the necessary beats, we show that it is possible to warp video into dance-like alignment with any song of our choice.

1.1 Applications

The quantification of visual rhythm enables many applications. We focus primarily on those related to video retargeting, which combines analysis and synthesis of dance. In addition to motivating our work, these applications serve to test our basic assumptions about visual rhythm and dance.

Dance Retargeting: By time-warping the visual beats of existing dance footage into alignment with new music, we can change the song that a performer is dancing to. This is a special case of retargeting where we can assume that visual beats are already aligned with musical beats in the source video, allowing us to find them with simple audio beat tracking. We leverage this to test our central hypothesis about visual beats and dance separately from any computer vision algorithms.

Dancification: Our visual beat hypothesis allows for the existence of visual beats in non-dance video, but implies such visual beats should not be distributed according to any discernible tempo. If we can find such beats through purely visual means, we can use them to transform non-dance video into dance video. We call this *dancification*. We can also use this strategy to improve bad or off-tempo dancing, providing a kind of "auto-tune" for dance.

Accidental Dance: We can adapt our strategy for identifying visual beats into a search criteria, which we can use to find segments of dance-like or near dance-like motion in large collections of video. If only short segments of such video can be found, we generate random walks through the visual beats of those segments to synthesize an arbitrary length of output dance video.

Visual Instrument: Visual beats provide temporal control points that can also be used for more general manipulation of video. For example, by warping visual beats into alignment with the notes of a musical instrument (e.g., recorded MIDI or a transcribed performance) we can use that instrument as a musical interface for editing video.

1.2 Beat Saliency

We begin by factoring the perception of beat — both for music and dance — into different types of saliency, drawing on observations from literature on the arts [Bolton 1894; Chion et al. 1994; Cowell and Nicholls 1996; McPherson 2006; Vernallis 2004] as well as heuristics used by related work on audio beat tracking [Ellis 2007; Goto 2002; Grosche et al. 2010; Hu et al. 2017; Lerch 2012; McFee et al. 2015] and the computational analysis of dance [Brick and Boker 2011; chul Lee and kwon Lee 2005; Dyaberi et al. 2006; Kim et al. 2003; Liao et al. 2015; P. Chen et al. 2011]. The saliency metrics described here guide our design of heuristics for visual beat tracking in Section 4 and a dance-specific strategy for time-warping video in Section 5.

Musical beats are often defined as moments where a listener would clap or tap their feet in accompaniment with music. This definition relies on an implied measure of saliency, with different sounds affecting the perception of beats in different ways. Most work on rhythmic analysis approximates this saliency implicitly through the use of a heuristic objective for finding beats in audio. Typically

that objective is expressed as a combination of two functions: one temporally local function that measures musical *onset strength* (indicating the start of musical notes), and another function that measures adherence to a particular *tempo*, as indicated by periodic patterns in the distribution of onset strength over time.

Our definition of visual beats implies a related type of saliency, rooted in the perception of dance. We assume this saliency can also be factored into local and rhythmic components, from which we will derive visual complements for onset strength and tempo. Note that the local component of visual beat saliency is different from classic image saliency [Judd et al. 2009; Liu et al. 2009; Pritch et al. 2008] in that it is a function of visible motion, and should reflect some measure of our ability to localize events in time.

We refer to the rhythmic components of visual and musical beat saliency as *rhythmic saliency* and the local components as *local saliency*.

1.3 Synchro-Saliency

The perception of dance is greatly influenced by musical accompaniment. This is why a dance can appear synchronized with one piece of music, and out of place with another. We discuss this synchronization in terms of what we call *synchro-saliency*, which measures the perceived strength of relationships between visible and audible events.

We describe any two functions $h_a(t_a)$ over audible events and $h_v(t_v)$ over visible events as *synchro-salient complements* if their product approximates synchro-saliency h_s :

$$h_a(t_a)h_v(t_v) \approx h_s(t_a, t_v) \quad (1)$$

In other words, synchro-salient complements are corresponding functions over audio and video that indicate high synchro-saliency when large values are aligned in time.

In Section 4 we design heuristics for the local and rhythmic saliency of video to be synchro-salient complements of corresponding heuristics used in audio beat detection. This lets us express dancification as the alignment of rhythmic saliency with a target.

2 RELATED WORK

Computational editing and video manipulation are popular topics in computer graphics and vision, with a strong history of work that draws inspiration from the arts. Some has focused on automating established editing tasks based on objectives derived from cinematography [Berthouzoz et al. 2012; Leake et al. 2017; Wang et al. 2008], while others have used computation to transform video into new types of visualizations, digests, or summaries [Bai et al. 2012; Burg and Beck 2012; Chuang et al. 2005; Pritch et al. 2008; Schödl et al. 2000]. Like us, Wang et al. [2014] and Bazin et al. [2016] explore non-uniform time-warping of video according to an objective function, in their case for the task of temporally aligning different videos. Witkin and Popovic [1995], White et al. [2006], and Wang et al. [2006] explore motion filters that change the visual style of animations and video, employing a strategy that is similar in spirit to the time-warping we introduce in Section 5.

The work of Davis et al. [2017; 2015a; 2015b; 2014] is similar to ours in that it applies concepts from audio processing to motion

derived from video. However, their work focuses on the frequencies of extremely subtle motion, which they use to extract sound from silent high-speed video [Davis et al. 2014], estimate material properties [Davis et al. 2017, 2015a], and build interactive physical models of captured objects [Davis et al. 2015b].

Several works in computer graphics have focused on synchronizing audio and 3D animation. [chul Lee and kwon Lee 2005; Kim et al. 2003; Langlois and James 2014], with work on synthesizing dance in 3D character animation being especially relevant to our own. Kim et al. [2003] and Lee et al. [2005] identify temporal control points in semi-periodic human character animations, which they then synchronize with MIDI music to synthesize dance. We perform similar synchronization on a more general class of video and audio signals by drawing a more explicit connection between rhythmic structure in music and dance.

Most closely related to our work is that of Liao et al. [2015] and Chen et al. [2011]. Like us, both explore music as a way to drive the manipulation of video. However, their work is more application-driven, and focuses primarily on the pace or placement of clips relative to music. In contrast, we focus explicitly on characterizing the relationship between musical rhythm and dance, introducing visual beats as a way to drive more precise and dramatic time-warping. These differences lead to very different-looking output; Liao et al. produces well-paced montages, often with a time-lapse aesthetic; Chen et al. gradually speed up or slow down video to match music; while our output produces precisely synchronized dance video.

All of the related works that use music to drive character animation [chul Lee and kwon Lee 2005; Kim et al. 2003] or video [Liao et al. 2015; P. Chen et al. 2011] factor the saliency of dance into local and rhythmic heuristics. Kim et al. [2003] even use this heuristic to define something analogous to what we call visual beats. However, their heuristic for rhythmic saliency measures error relative to fixed reference beats of a constant tempo, limiting their ability to identify visual beats in source animations with very different or locally variable tempo. By contrast, our heuristic for rhythmic saliency extends the dynamic programming approach that Ellis [2007] developed for audio to incorporate a locally-adaptive measure of tempo, allowing us to find visual beats in input with very irregular visual rhythm. We also derive a dance-specific interpolation strategy for warping visual beats, which we show to amplify rhythmic saliency in video.

3 QUANTIFYING RHYTHM IN AUDIO

Here we re-derive common strategies for quantifying rhythm in audio, paying close attention to details that relate audio features to the visual analogues we derive in Section 4. Each of the algorithms described in this section can be found in the open-source python package LibROSA [McFee et al. 2017, 2015], with more of the underlying theory described in [Ellis 2007; Goto 2002; Grosche et al. 2010; Hu et al. 2017; Lerch 2012]. Visualizations of the audio features described here are shown next to their visual analogues in Figure 2.

Starting with a 1D audio signal, we re-derive a method for extracting discrete audio beats, represented as different points in time.

Our approach builds on the common assumption that beats and tempo are determined by the distribution of musical onsets in time.

3.1 Spectrograms and Spectral Flux

Onsets are generally indicated by a sudden increase in the volume of a signal. However, volume alone is not sufficient to detect all onsets. Consider an instrument like the theremin, capable of playing sustained continuous sound that changes in pitch over time. Onsets in this case are indicated by changes in pitch, rather than volume. One way to measure both changes in volume and pitch is to use a **spectrogram**, computed as the time-windowed FFT of an audio signal $x(t)$:

$$S(n, k) = \sum_{q=-\frac{N}{2}}^{\frac{N}{2}-1} x(hn + q) w(q) e^{-\frac{2j\pi qk}{N}} \quad (2)$$

which yields the complex-valued matrix S , with $S(n, k)$ representing the k th frequency bin at time corresponding to frame n . Here, w is a window function (e.g., a Hamming window), and h is a hop size (the separation between successive windows), which also determines the relationship between n and t . The amplitude of each $S(n, k)$ approximates the volume of x at time n and frequency k .

Spectrograms offer **spectral flux**, which measures the change in amplitude of different frequencies over time, as an alternative to volume for finding onsets [Böck and Gerhard Widmer 2013; Dixon 2006]:

$$S_F(n, k) = |S(n, k)| - |S(n - 1, k)| \quad (3)$$

3.2 Onset Envelopes

Onset envelopes (also sometimes called **novelty curves**) are an approximate measure of how likely an onset has occurred at each point in time. Each onsets generally coincides with an increase in spectral flux at certain frequencies. One algorithm for computing an onset envelope is to sum positive spectral flux over the frequencies of a spectrogram, yielding a non-negative 1D time-signal $u_a(n)$ [Böck and Gerhard Widmer 2013; Dixon 2006]:

$$u_a(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{S_F(n, k) + |S_F(n, k)|}{2} \quad (4)$$

In Section 3.5 we use the onset strength measured in our onset envelope as our heuristic for local saliency in audio beat detection.

3.3 Onset Detection

A large spike in $u_a(n)$ is typically sufficient but not necessary indication of an onset, while smoothness in $u_a(n)$ generally indicates greater uncertainty in estimated onsets. Discrete onset detection can therefore be formulated as peak-picking in the onset envelope. One simple strategy for this is to look for local maxima that are some threshold above their local mean.

3.4 Tempo and Tempograms

Tempo can be estimated by looking for spikes in the autocorrelation of an onset envelope. Such spikes indicate self-similarity at a

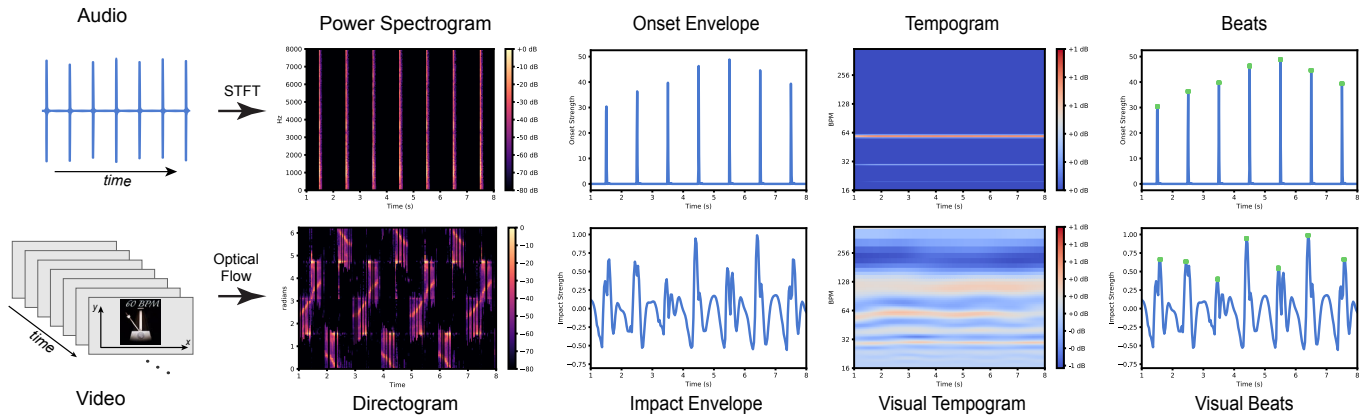


Fig. 2. **Rhythmic Features in Audio and Video** – The top row shows features used to quantify metric structure in audio. The bottom row shows the synchro-salient complements that we use to quantify metric structure in video. The visualizations here correspond to the audio and video from footage of a simple metronome [LumBeat 2013]. As we would expect from footage of a metronome, the detected visual metric structure is aligned with its synchro-salient complement in audio.

particular time offset. The tempo is typically the largest spike corresponding to the period of a countable frequency (e.g., 20–300bpm).

Time-varying tempo can be measured with a *tempogram* [Grosche et al. 2010], which is derived from an onset envelope in a similar manner to a spectrogram, but with the windowed FFT replaced by unbiased local autocorrelation, giving us the equation:

$$\mathcal{T}_a(n, k) = \sum_{q=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{u_a(hn + q) u_a(hn + q + k) w(q)}{2N + 1 - k} \quad (5)$$

where $w(q)$ is a 5-second window. The columns of \mathcal{T}_a are typically normalized by their maximum values to factor out the effect of local volume, leaving $\mathcal{T}_a(n, k)$ a heat map of likely tempos over time. The tempo $\tau(n)$ around frame n can be estimated in beats per minute (bpm) as:

$$\tau(n) = \frac{60}{Pk^*(n)} \quad (6)$$

$$k^*(n) = \operatorname{argmax}_{k \in K} \mathcal{T}_a(n, k) \quad (7)$$

where P is the period of a frame, and K is the set of values corresponding to countable tempi (e.g., 20–300 bpm). Limiting the search for k_m to K has the added benefit of discarding false peaks at small offsets, which can be the result of local self-similarity in smooth signals, and at large offsets, where the denominator in Equation 5 becomes small, boosting signal noise.

Note that by the Wiener-Khinchin theorem, \mathcal{T}_a provides similar information to the power of a spectrogram. However, energy exhibited at harmonics in power spectra tends to show up in subharmonics with autocorrelation. We discuss the advantage of favoring subharmonics later in Section 4.5.

Rhythmic structure in audio can be seen as energy distributed in horizontal lines (constant bpm over time) at a tempo and possibly its harmonics and subharmonics, as seen in Figures 2, 3 and 7.

3.5 Beat Tracking

As we discussed in Section 1.2, beat tracking is often performed by optimizing over a heuristic approximation of beat saliency. The local component of that approximation favors placing beats on musical onsets, while the rhythmic component favors distributing beats according to a constant tempo. This results in an objective function C_a :

$$C_a(\{n_i\}) = \sum_{i=1}^{|\{n_i\}|} u_a(n_i) + \gamma \sum_{i=2}^{|\{n_i\}|} v(n_i - n_{i-1}, \tau) \quad (8)$$

$$v(\Delta n, \tau) = - \left(\log \frac{P \Delta n}{\tau} \right)^2 \quad (9)$$

where u_a is the onset envelope, and v is a pairwise objective penalizing deviation from the tempo τ . The optimal sequence of beat times can then be found through dynamic programming using the recursive relationship:

$$C_a^*(n) = u_a(n) + \max_{m=0 \dots n} \{C_a^*(m) + \gamma v(n - m, \tau)\} \quad (10)$$

where α controls the relative weight of the two terms. Each $C_a^*(n)$ can be traced to extract a sequence of beats. The sequence corresponding to the maximum score is then returned as a final result.

4 QUANTIFYING RHYTHM IN VIDEO

In this section we derive visual analogues for each of the audio features described in Section 3. Figure 2 visualizes each analogue next to its musical counterpart.

We begin by choosing a heuristic for local saliency in video. In general, we want this heuristic to be a synchro-salient complement of whatever we use for audio. In Section 3 that was onset strength. Onset strength is maximized at impulses and the onsets of impulse responses. In the physical world, these tend to coincide with the impact of a moving object and a resonating surface—often resulting in sudden visible deceleration of the moving object. We use this

sudden visible deceleration as the basis of our heuristic for local saliency in video, which we call **visible impact**.

Our input is regular video, from which we first compute the optical flow $F_{t+1}(x, y)$ from each frame t to its neighbor $t + 1$ using the method of Bouquet [yves Bouquet 2000].

4.1 Directograms

In Section 3.1 we used the spectrogram S to factor volume changes into different frequencies. Here we calculate a 2D matrix $\mathcal{D}(t, \theta)$, which we call a **directogram**, to factor motion into different angles. Each column of \mathcal{D} is computed as the weighted histogram of angles for the optical flow field F_t of an input frame t :

$$\mathcal{D}(t, \theta) = \sum_{x, y} |F_t(x, y)| \mathbb{1}_\theta(\angle F_t(x, y)) \quad (11)$$

$$\mathbb{1}_\theta(\phi) := \begin{cases} 1 & \text{if } |\theta - \phi| \leq \frac{2\pi}{N_{bins}} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

Here $\mathbb{1}_\theta(\phi)$ is an indicator function used to separate flow vectors into N_{bins} different angular bins (i.e. calculate a weighted histogram).

Certain video codecs introduce repeated frames into videos, creating blank columns in \mathcal{D} . We address this by applying a small, 3×3 median filter to \mathcal{D} , noting that both dimensions of the kernel are necessary to account for curved motion.

Our directogram now looks very similar to the amplitudes of our spectrogram from Section 3.1: a scalar-valued matrix with x -axis corresponding to time. From this, we can calculate per-direction deceleration as an analogue for spectral flux using a formula nearly identical to Equation 3:

$$\mathcal{D}_F(n, k) = \mathcal{D}(n, k) - \mathcal{D}(n - 1, k) \quad (13)$$

4.2 Impact Envelopes

Equation 13 gives us a matrix \mathcal{D}_F with the same form as spectral flux S_F . We next compute our visual analogue for an onset envelope, which we call an **impact envelope**, by summing over positive entries in the columns of \mathcal{D}_F just as we did with S_F in Equation 4:

$$u_v(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\mathcal{D}_F(n, k) + |\mathcal{D}_F(n, k)|}{2} \quad (14)$$

This gives us an impact envelope u_v with precisely the same form as an onset envelope. To account for large outlying spikes that sometimes happen at shot boundaries (i.e., cuts), we clip the 99th percentile of values in u_v to the 98th percentile. We then normalize u_v by its maximum to make our calculations more consistent across video resolutions.

4.3 Impact Detection

To detect discrete visible impacts, we first calculate the local mean u_v using a 0.1-second window, and local maxima using a 0.15-second window. We then define impacts as local maxima that are above their local mean by at least 10% of the envelope's global maximum.

4.4 Visual Tempograms

Our visual analogue for a tempogram, which we call a **visual tempogram**, is computed by simply replacing u_a in Equation 5 with our impact envelope u_v :

$$\mathcal{T}_v(n, k) = \sum_{q=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{u_v(hn + q) u_v(hn + q + k) w(q)}{2N + 1 - k} \quad (15)$$

As with audio tempograms, we can see rhythmic structure in visual tempograms as horizontal lines at a visual tempo and optionally is subharmonics or harmonics. Figures 2, 3 and 7 show that dance video is characterized by aligned structure in video and audio tempograms—indicating that, at least under certain conditions, tempograms and visual tempograms can be treated as synchro-salient complements.

4.5 Visual Beat Tracking

For simple dance video we can identify visual beats by applying the same algorithm described in Section 3.5 to an impact envelope. However, most of our motivating applications use visual beats as control points for some type of time-warping. In this case our criteria for selecting visual beats may be quite different, as the quality of that selection will be evaluated in some warped output. To account for this, we must consider the effect of warping on local and rhythmic saliency.

Recall that our local saliency metric is visible impact, which estimates discontinuous deceleration in video. We first want to ensure that time-warping does not create false visible impacts in our output, which can happen when a discontinuous rate of time-warping is applied to continuous motion in a source video. To avoid such false impacts, we first restrict the selection of visual beats to those local extrema of u_v identified as impacts in Section 4.3. We then enforce continuity on the rate of time-warping everywhere except at visual beats (see Section 5), ensuring that new visual impacts are not created at moments where there were none in the input.

With consideration limited to the detected impacts m_i , we now define an objective similar to Equation 8:

$$C_v(\{m_i\}) = \sum_{i=1}^{|\{m_i\}|} u_v(m_i) + \gamma \sum_{i=2}^{|\{m_i\}|} V(m_i, m_{i-1}) \quad (16)$$

Here we have several options for the pairwise objective V .

In the audio case, we used a pairwise objective to penalize variation from the dominant tempo of our signal. For retargeting applications, we often assume there is no such dominant tempo to begin with; our job, in a sense, is to create one. We can however penalize variation from the dominant tempo of a target signal as a way of favoring rates of time-warping in our output that are close to 1.

Another option is simply set $V = 0$, thereby turning all impacts into visual beats. This strategy works well when all movements in the video are large, and it ensures that all visible impacts map to beats. However, it makes results sensitive to the window and threshold parameters described in Section 4.3 when frequent, subtle motion is present.

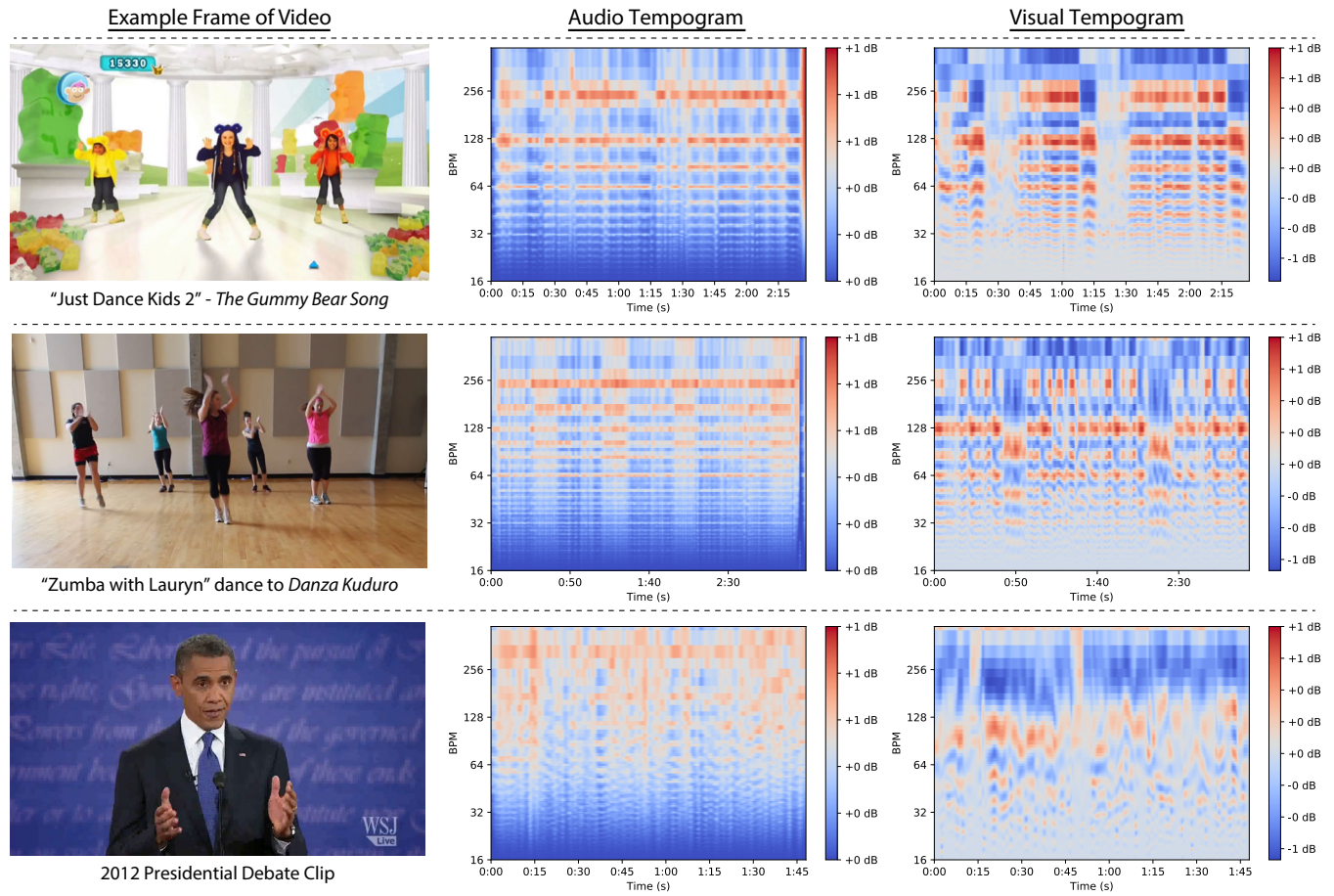


Fig. 3. **Comparison of Regular (Audio) and Visual Tempograms for Dance and Non-Dance Video** – Here we compare audio (middle column) and visual (right column) tempograms for three videos, with representative frames shown on the left. The top row visualizes dance video from a videogame made for children [Ubisoft 2013]. The middle row shows the visual tempogram for a zumba dance routine [Zumba with Layryn 2014] set to the song *Danza Kuduro* by Don Omar (Note: here we calculated the audio tempogram on an aligned version of the original track, as the audio recorded with the video was low quality). The bottom row shows tempograms for a clip from the first 2012 Presidential Debate [WSJDigitalNetwork 2012] (the same source video is shown in Figure 1 and featured in our supplemental video). In the dance examples (top two rows), we see high energy across matching harmonic tempos for both audio and video. In the non-dance video (bottom row), local tempo is more ambiguous and less consistent.

For results in this paper we opt for a solution that adapts Equation 9 to a locally-varying notion of tempo. Our aim in this case is to bias the selection of visual beats toward motion that is locally-rhythmic. Such motion is common in certain types of video; humans, for example, tend to take on momentary rhythms when using gestures to emphasize speech [Turk 2002]. Using our visual tempogram \mathcal{T}_V to measure the strength of local rhythm at different beat separations (i.e., tempo periods), we define the adaptive objective $V_{\mathcal{T}}$:

$$V_{\mathcal{T}}(m_i, m_{i-1}) = \mathcal{T}_V(m_i, \frac{|m_i - m_{i-1} - 1|}{P}) - 1 \quad (17)$$

Recall that the tempogram \mathcal{T}_V is normalized by the maximum of each column, so $V_{\mathcal{T}}$ takes the value 0 for impacts occurring at their local tempi, and a value < 0 for impacts that deviate from those tempi. We use a window size of 5 seconds to calculate \mathcal{T}_V , and consider \mathcal{T}_V for any impacts separated by more than this to be 0.

Taken together, Equations 16 and 17 provide motivation for using autocorrelation to measure tempo instead of a Fourier transform. Both transformations are associated with an equivalence class of periods, with signals of one period generally exhibiting high response across the corresponding equivalence class. In the Fourier transform, equivalence classes correspond to a base frequency and its harmonics. In autocorrelation, they correspond to a base frequency and its subharmonics (integer divisors). Excess energy at subharmonics is better for the objective described in Equation 17 because the corresponding periods are larger than that of the base frequency, making them less likely to be chosen over the base frequency in Equation 16 (as smaller beat separations result in more beats, and therefore additional positive unary terms in our objective). Note that by allowing impacts within an equivalence class to contribute energy to the selection of other impacts, Equation 17 favors placing

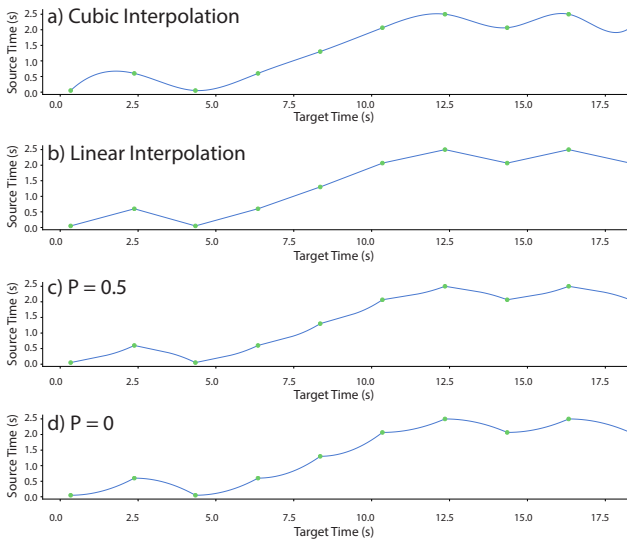


Fig. 4. **Warp Curves** - Warp curves plot time in our input as a function of time in our output. Green dots represent the matching of a visual beat (a horizontal in the graph) with an audio beat (a vertical). Here we see four different interpolation strategies applied to the same mapping of visual beats to audio beats. Cubic (a) and linear (b) interpolation tend to over-smooth in time, damping visual impact. We opt for a parameterized interpolation that accelerates time as it approaches visual beats. (c) shows this approach with the parameter p set to 0.5. (d) shows the same approach with $p = 0$. In our accidental dance results, we set p proportional to the impact strength of the beat being approached. This avoids exaggerating motion toward low-confidence visual beats.

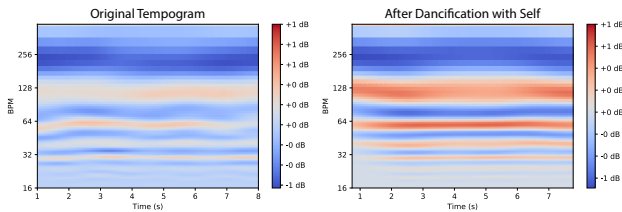


Fig. 5. **$p=0$ Time-Warping of a Video to Its Own Visual Beats** - Here we see the effect of $p = 0$ interpolation on the metronome video [LumBeat 2013] from Figure 2 when visual beats in the original video are warped to themselves. The timing of visual beats does not change in this case, but the acceleration and deceleration of our interpolation strategy emphasizes rhythmic structure already in the video. Note that linear and cubic interpolation would have no effect on the input in this case.

impacts in harmonic relation to beats even if those impacts are not selected themselves.

5 WARP CURVES

As mentioned in Section 4.5, ensuring that the rate of time-warping is continuous everywhere other than visual beats avoids introducing spurious visible impacts. We can visualize this continuity by plotting time in our output (target) against corresponding times from our input (source), as in Figure 4. The resulting **warp curves** have slopes equal to the instantaneous rate of time warping at each corresponding time in our output.

Many interpolation strategies guarantee continuity; linear and cubic interpolation are simple choices, and both work fine in practice. However, when time is being stretched (i.e., when the target is longer than the source), both linear and cubic interpolation tend to have small derivatives at beat times, which effectively dampens visible impact. This is problematic, as it reduces rhythmic saliency in our output.

We offer an alternative interpolation strategy that accelerates into visual beats, slowing the rate of time-warping before and after acceleration to maintain synchronization with control points. We parameterize this strategy by separating interpolation between beats into two segments. We then use linear interpolation for the first segment, and add an acceleration term during the second, maintaining continuity in the rate of warping throughout. Let $f(t)$ represent the map from target times to source times, normalized to the region between a neighboring pair of corresponding control points. Given the linear segment $[0, p]$ and the accelerating segment $(p, 1]$, we have:

$$f(t) = \begin{cases} \alpha t & \text{if } t \leq p \\ \alpha t + g(t - p) & \text{if } t > p \end{cases} \quad (18)$$

$$f(0) = 0 \quad (19)$$

$$f(1) = 1 \quad (20)$$

$$g(0) = 0 \quad (21)$$

$$g'(0) = 0 \quad (22)$$

setting our acceleration term to $g(x) = x^2$, we can solve for the relationship between α and p :

$$p = 1 - \sqrt{1 - \alpha} \quad (23)$$

$$\alpha = 1 - (1 - p)^2 \quad (24)$$

we can now use p to specify constraints on how much time should be spent accelerating (e.g., accelerate for one-fifth of a second before every beat), or α to specify constraints on motion at the start of each segment (e.g., slow to 1/3 the rate of linear interpolation at the start of each segment). Visual comparisons can be found in our supplemental material.

Figure 5 visualizes the effect of acceleration in our warping strategy on rhythmic saliency.

6 UNFOLDING

While non-dance video often contains segments of momentarily rhythmic motion, contiguous segments are rarely long enough to fill a complete song. We address this by introducing a technique called **unfolding** to extend short segments.

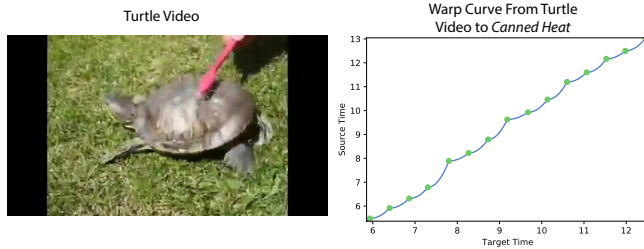


Fig. 6. **Turtle Video Warp Curve** — Left shows an example frame from the turtle video [YouTube:shubhgupta91 2015]. Right shows a warp curve corresponding to the green box regions of Figure 7.

Our interpolation strategy in Section 5 does not make any assumption about the monotonicity of our warp curves, making interpolation backward in time just as easy as forward; we need only specify the appropriate control points and let our interpolation strategy do the rest. Based on this observation, we can generate outputs of arbitrary length by synthesizing random walks through our input.

Given a sequence $B = \{m_0, \dots, m_k\}$ of visual beats, we generate a new sequence B_u by taking a random walk through B according to an associated momentum parameter ϕ . Each iteration of the walk starts at a beat m_j , and takes either a forward step to m_{j+1} , or a backward step to m_{j-1} , adding its new location to B upon completion of the step. If the current location is m_0 , the next step will always be forward; and if it is m_k , the next step will always be backward. Otherwise, the probability of stepping in the same direction as the previous iteration is given by $0.5 + \phi$, and the probability of reversing direction is $0.5 - \phi$. In practice, we halt the random walk when the distance from its current location to m_k is equal to the number of remaining target beats, thereby filling in the rest of B with forward steps to ensure that the last target beat matches the last available source beat. Once the sequence B has been generated, synthesizing our output works the same as it did before.

When synthesizing unrolled output we always use the interpolation strategy from Equation 18 with $p < 1$ to ensure that the interpolated results are not symmetric around any of the visual beats. This makes the reversal of time less noticeable [Pickup et al. 2014].

7 APPLICATIONS AND RESULTS

We explore 4 applications of visual rhythm, each related to synchronizing motion in video with a musical target. Results, code, and additional material can be found on our project website.

7.1 Dance Retargeting

Our first application is designed to test our central hypothesis about visual beats independently of the vision-based algorithms and heuristics introduced in Section 4.

Our hypothesis states that aligning visual beats with musical beats creates the appearance of dance. If we assume the converse is also true—that dance indicates an alignment of musical beats with corresponding visual beats—we can test our hypothesis without any analysis of the visual signal we are manipulating.

Given a dance video, we first detect musical beats in the accompanying audio signal. We then treat those musical beats as visual

beats, and warp them into alignment with new music using the same approach outlined in Section 5. We used Equation 18 to produce our results, setting $\alpha = 0$ to maximize acceleration ahead of every visual beat.

Dance retargeting is easy, fast, and fairly robust. A simple script for retargeting YouTube videos, as well as several examples, can be found on our project website.

7.2 Dancification

We can use the algorithms from Sections 4 and 5 to warp video with irregularly-spaced (or sub-optimally-spaced) impacts into alignment with music. This can be done to make non-dance video appear to dance, or to “auto-tune” video that is already near-dance-like. For example, many videos can be found on the internet of animals moving in repetitive ways, put to music by human editors. The synchronization of these videos is usually a bit off, as any irregularity in the original motion will cause alignment to drift over time. Correcting for this drift manually is very tedious, and therefore rarely done; for example, we found more than ten different edits of the “turtle dance” video (Figure 2) on Youtube, but none differed by more than a constant speed factor from the others, leaving each of them out-of-synch with music by the end of the video. Our dancification code produces notably improved results, and can be used to tune the video to a variety of songs without manual effort (see supplemental material for results).

7.3 Accidental Dance

The objective function in Equation 16 gives us a way to score the set of available visual beats in a video, with higher scores suggesting better opportunity to create dance-like motion through warping. We use this to search through collections of video to find source material that can be turned into dance.

Using $V_{\mathcal{T}}$ from Equation 17, we write a slightly modified recurrence relation for visual beat selection:

$$C_v^*(m_i) = u_v(m_i) + \max_{j \in W(m_i)} \{C_v^*(m_j) + \gamma V_{\mathcal{T}}(m_i, m_j)\} \quad (25)$$

where for some fixed constant w , $W(m_i)$ contains all m_j with $(m_i - w) < m_j < m_i$. Separations between detected impacts greater than w then segment a video into disconnected components, each with their own optimal $C_v^*(m_i)$ and corresponding sequence of visual beats. Large window sizes result in longer source segments but allow for much higher rates of warping, which can look unnatural in some case. We typically use a small window of ~1-3 seconds, and sort the resulting segments according to their respective maximum scores. For each segment, we extract a separate video clip to use in retargeting, unfolding the result to the length of a target song as described in Section 6. To avoid creating large accelerations where the original motion in a video was more subtle, or where we are less certain about visual impact, we set the parameter p in Equation 18 proportional to u_v at each beat. This has the effect of only accentuating beats with high confidence.

Our main supplemental video contains a montage of accidental dancing results generated from 2012 and 2016 presidential and primary debate footage, with individual results linked from our project website.

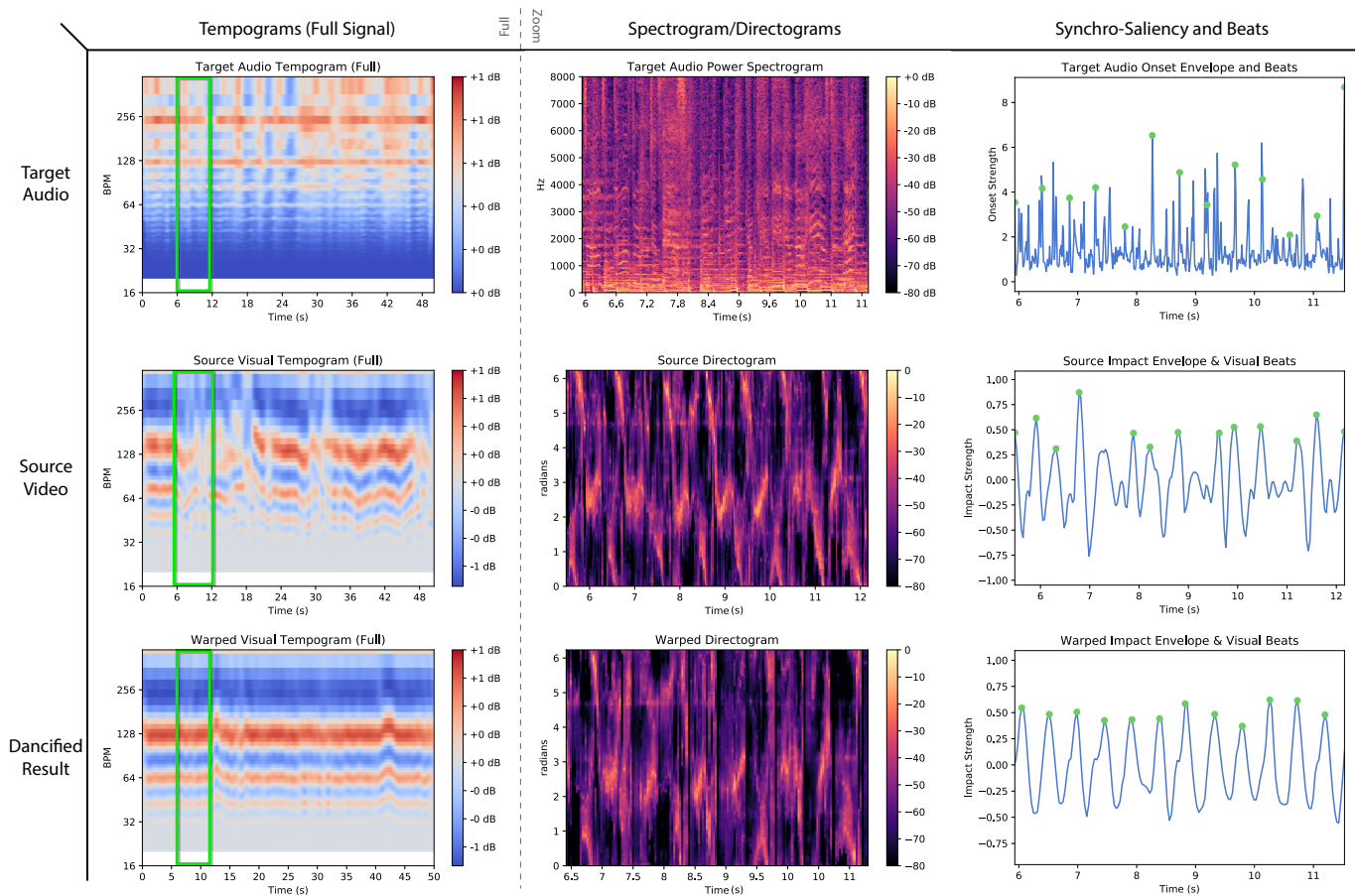


Fig. 7. **Dancification as a transformation of visual metric structure** — The top row visualizes a tempogram (left), power spectrogram (middle), and an onset envelope (right) for the song *Canned Heat* by Jamiroquai. The spectrogram and onset envelope correspond to the boxed region of the tempogram, and green dots on the onset envelope show the locations of musical beats. The middle row visualizes the corresponding visual complements for the turtle video [YouTube:shubhgupta91 2015]. The bottom row shows what those same complements look like after retargeting the video to *Canned Heat*. From the left column we can see that the music has a mostly constant tempo around 128bpm, while the visual tempo of our source video varies quite a bit over time. After dancification, visual tempo is more constant and has been shifted to match that of the target audio. In the right column we can see how dancification shifts visual beats that are irregularly distributed in the source video into alignment with the tempo of the target audio.

7.4 Visual Instrument

In one more experiment we explored the use of visual beats as temporal control points for user-driven manipulation of video. For this we selected three short videos: one of a cat meowing, a second featuring a cat quickly wiggling, and a third featuring a cat performing a begging trick. For each of these videos we detected visual beats, from which we designated a subset of alternating ‘on’ and ‘off’ beats. We then assigned each video to a MIDI instrument, corresponding its respective ‘on’ beats with the onset of MIDI notes, and ‘off’ beats with the release of those notes, using unfolding to allow for arbitrarily long input. The result is a set of virtual ‘puppets’ that can be controlled by the user.

To demonstrate this, we played each of the three MIDI instruments along with a different track of the song *Eye of the Tiger* by the band Survivor. The resulting video, entitled ‘Eye of the Housecat’, can be found on our project website.

8 DISCUSSION

This paper demonstrates how to create and manipulate the appearance of dance in video based on an analysis of visible motion that mirrors that of musical rhythm in audio.

We motivated much of our work with a speculative model of saliency for music and dance in Section 1, introducing heuristics to approximate components of that model throughout the paper. Here we discuss limitations of our model and the heuristics we use to approximate it, as well as opportunities for future work.

8.1 Limitations

Visible impact is a very simple heuristic for local saliency in video, but it has many problems. For example, as we calculate it, visible impact makes no distinction between the motion of a central subject and background or camera motion. As a result, even minor camera motion—which is easily ignored by human viewers—can create large

visual impacts and lead to the selection of incorrect visual beats. This is easy to see in our supplemental results on unstabilized footage captured with hand-held cameras. Stabilization and a data-driven prior on image saliency would likely help with this problem. Other motion estimation approaches (e.g., point-based motion tracking) could also make addressing these problems easier by allowing more direct measurement of visible acceleration. A completely data-driven estimate of local saliency could also work, given the right training strategy and data.

Because we use musical beats as targets for warping, our results are susceptible to errors in musical beat tracking. As an aggregated attribute, tempo is generally easier to estimate than individual beats. For this reason, our results occasionally follow the correct tempo, but appear phase-shifted from the beat. This could be fixed with better musical beat tracking, or a manually-specified target (e.g., MIDI or other annotated music).

The dynamic programming formulation of Equation 16 is designed to find the optimal subset of visible impacts within a segment of video to use as visual beats. This is done by maximizing an objective over all subsets using dynamic programming. While all visible impacts are considered during this optimization, only those selected as part of the optimal set directly influence the resulting score. Optimality of the chosen subset imposes some limit on the strength of spurious impacts between visual beats, but our objective does not explicitly penalize such impacts. This can result in some distracting artifacts where locally salient motion is warped to seemingly random times. It should be possible to add consideration of such spurious impacts to our objective at the cost of additional computation.

8.2 Future Work

Our model of saliency in Section 1 is largely speculative. Perceptual studies could help refine and validate that model.

Several of the heuristics in our work sample a much larger design space of possible analyses for music and dance. For example musical onsets and visible impacts are one pair of synchro-salient complements for approximating local saliency in music and dance, but others may lead to different interpolation strategies, or be suitable for different styles of dance.

We are excited to explore new applications of visual rhythm and beat. For example, we would like to automate synchronization of several videos with different sources or instruments as an extension of what we do with MIDI in Section 7.4, and are also interested in using our work to analyze video content—for example, to evaluate dancing quality, or characterize different dance styles in video.

Much of our work could also be adapted to synchronize video with non-musical targets. Our visual instrument application takes one step in this direction by letting users control the target of warping.

8.3 Conclusion

By showing that we can adapt rhythmic analysis developed for audio to the tasks of analyzing and synthesizing dance, our work provides exciting new opportunities to build creative tools for manipulating audio and video.

ACKNOWLEDGMENTS

We thank all of the dancers — be they people, cats, dogs, turtles, or anything else. Our research is supported by The Brown Institute for Media Innovation.

REFERENCES

- Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. 2012. Selectively De-Animating Video. *ACM Transactions on Graphics* (2012). <http://graphics.berkeley.edu/papers/Bai-SDV-2012-08/>
- Jean Charles Bazin and Alexander Sorokin-Hornung. 2016. ActionSnapping: Motion-Based Video Synchronization. In *ECCV*.
- Floraine Berthouzou, Wilmot Li, and Maneesh Agrawala. 2012. Tools for Placing Cuts and Transitions in Interview Video. *ACM Trans. Graph.* 31, 4, Article 67 (July 2012), 8 pages. <https://doi.org/10.1145/2185520.2185563>
- Sebastian Böck and Gerhard Widmer. 2013. Maximum Filter Vibrato Suppression for Onset Detection.
- Thaddeus L. Bolton. 1894. Rhythm. *The American Journal of Psychology* 6, 2 (1894), 145–238. <http://www.jstor.org/stable/1410948>
- Timothy R. Brick and Steven M. Boker. 2011. Correlational Methods for Analysis of Dance Movements. *Dance Research* 29, supplement (2011), 283–304. <https://doi.org/10.3366/drs.2011.0021> arXiv:<https://doi.org/10.3366/drs.2011.0021>
- Kevin Burg and Jamie Beck. 2012. Cinemagraphs. (2012). <http://cinemagraphs.com/>
- M. Chion, C. Gorbman, and W. Murch. 1994. *Audio-vision: Sound on Screen*. Columbia University Press. <https://books.google.com/books?id=BBs4Arfm98oC>
- Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H. Salesin, and Richard Szeliski. 2005. Animating Pictures with Stochastic Motion Textures. *ACM Trans. Graph.* 24, 3 (July 2005), 853–860. <https://doi.org/10.1145/1073204.1073273>
- Hyun chul Lee and In kwon Lee. 2005. Automatic Synchronization of Background Music and Motion. In *in Computer Animation, in Computer Graphics Forum, Volume 24, Issue 3* (2005. 353–362).
- Laura K. Cirelli, Christina Spinelli, Sylvie Nozaradan, and Laurel J. Trainor. 2016. Measuring Neural Entrainment to Beat and Meter in Infants: Effects of Music Background. *Frontiers in Neuroscience* 10 (2016), 229. <https://doi.org/10.3389/fnins.2016.00229>
- H. Cowell and D. Nicholls. 1996. *New Musical Resources*. Cambridge University Press. <https://books.google.com/books?id=BeLDXA-7TtAC>
- Abe Davis, Katherine L. Bouman, Justin G. Chen, Michael Rubinstein, Oral Buyukozturk, Fredo Durand, and William T. Freeman. 2017. Visual Vibrometry: Estimating Material Properties from Small Motions in Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (April 2017), 732–745. <https://doi.org/10.1109/TPAMI.2016.2622271>
- Abe Davis, Katherine L. Bouman, Justin G. Chen, Michael Rubinstein, Fredo Durand, and William T. Freeman. 2015a. Visual Vibrometry: Estimating Material Properties from Small Motion in Video. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Abe Davis, Justin G. Chen, and Frédo Durand. 2015b. Image-space Modal Bases for Plausible Manipulation of Objects in Video. *ACM Trans. Graph.* 34, 6, Article 239 (Oct. 2015), 7 pages. <https://doi.org/10.1145/2816795.2818095>
- Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J. Mysore, Frédo Durand, and William T. Freeman. 2014. The Visual Microphone: Passive Recovery of Sound from Video. *ACM Trans. Graph.* 33, 4, Article 79 (July 2014), 10 pages. <https://doi.org/10.1145/2601097.2601119>
- Simon Dixon. 2006. Onset detection revisited. In *In Proceedings of the 9th international conference on digital audio effects*. 133–137.
- V. Dyaberi, H. Sundaram, T. Rikakis, and J. James. 2006. The Computational Extraction of Spatio-Temporal Formal Structures in the Interactive Dance Work '22'. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*. 59–63. <https://doi.org/10.1109/ACSSC.2006.356583>
- Daniel P. W. Ellis. 2007. Beat Tracking by Dynamic Programming. *Journal of New Music Research* 36, 1 (2007), 51–60. <https://doi.org/10.1080/09298210701653344> arXiv:<https://doi.org/10.1080/09298210701653344>
- Masataka Goto. 2002. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. 30 (09 2002).
- P. Grosche, M. Muller, and F. Kurth. 2010. Cyclic tempogram – A mid-level tempo representation for musicsignals. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 5522–5525. <https://doi.org/10.1109/ICASSP.2010.5495219>
- Xiao Hu, Jin Ha Lee, David Bainbridge, Kahyun Choi, Peter Organisciak, and J. Stephen Downie. 2017. The MIREX Grand Challenge: A Framework of Holistic User-experience Evaluation in Music Information Retrieval. *J. Assoc. Inf. Sci. Technol.* 68, 1 (Jan. 2017), 97–112. <https://doi.org/10.1002/asi.23618>
- Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. 2009. Learning to Predict Where Humans Look. In *IEEE International Conference on Computer Vision (ICCV)*.
- Tae-hoon Kim, Sang Il Park, and Sung Yong Shin. 2003. Rhythmic-motion Synthesis Based on Motion-beat Analysis. *ACM Trans. Graph.* 22, 3 (July 2003), 392–401.

- <https://doi.org/10.1145/882262.882283>
- Timothy R. Langlois and Doug L. James. 2014. Inverse-Foley Animation: Synchronizing rigid-body motions to sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)* 33, 4 (Aug. 2014). <https://doi.org/10.1145/2601097.2601178>
- Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational Video Editing for Dialogue-driven Scenes. *ACM Trans. Graph.* 36, 4, Article 130 (July 2017), 14 pages. <https://doi.org/10.1145/3072959.3073653>
- Alexander Lerch. 2012. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics* (1st ed.). Wiley-IEEE Press.
- Zicheng Liao, Yizhou Yu, Bingchen Gong, and Lechao Cheng. 2015. audeosynth: Music-Driven Video Montage. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015).
- Feng Liu, Yuzhen Niu, and Michael Gleicher. 2009. Using Web Photos for Measuring Video Frame Interestingness. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2058–2063. <http://dl.acm.org/citation.cfm?id=1661445.1661774>
- LumBeat. 2013. 60 BPM Metronome. (Feb 2013). https://www.youtube.com/watch?v=gsJEMH_emBM
- Brian McFee, Matt McVicar, Oriol Nieto, Stefan Balke, Carl Thome, Dawen Liang, Eric Battenberg, Josh Moore, Rachel Bittner, Ryuichi Yamamoto, Dan Ellis, Fabian-Robert Stoter, Douglas Repetto, Simon Waloschek, CJ Carr, Seth Kranzler, Keunwoo Choi, Petr Viktorin, Joao Felipe Santos, Adrian Holovaty, Waldir Pimenta, and Hojin Lee. 2017. librosa 0.5.0. (Feb. 2017). <https://doi.org/10.5281/zenodo.293021>
- Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python.
- K. McPherson. 2006. *Making Video Dance: A Step-by-step Guide to Creating Dance for the Screen*. Routledge. <https://books.google.com/books?id=b3hVewAACAAJ>
- Trista P. Chen, Ching-Wei Chen, Phillip Popp, and Bob Coover. 2011. Visual Rhythm Detection and Its Applications in Interactive Multimedia. 18 (01 2011), 88–95.
- Aniruddh D. Patel and Steven M. Demorest. 2013. 16 - Comparative Music Cognition: Cross-Species and Cross-Cultural Studies. In *The Psychology of Music (Third Edition)* (third edition ed.), Diana Deutsch (Ed.). Academic Press, 647–681. <https://doi.org/10.1016/B978-0-12-381460-9.00016-X>
- Aniruddh D. Patel, John R. Iversen, Micah R. Bregman, and Irena Schulz. [n. d.]. Experimental Evidence for Synchronization to a Musical Beat in a Nonhuman Animal. *Current Biology* 19, 10 (2017/11/14 [n. d.]), 827–830. <https://doi.org/10.1016/j.cub.2009.03.038>
- L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman. 2014. Seeing the Arrow of Time. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Y. Pritch, A. Rav-Acha, and S. Peleg. 2008. Nonchronological Video Synopsis and Indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 11 (Nov 2008), 1971–1984. <https://doi.org/10.1109/TPAMI.2008.29>
- Bruno H. Repp and Yi-Huang Su. 2013. Sensorimotor synchronization: A review of recent research (2006–2012). *Psychonomic Bulletin & Review* 20, 3 (01 Jun 2013), 403–452. <https://doi.org/10.3758/s13423-012-0371-2>
- Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. 2000. Video Textures. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 489–498. <https://doi.org/10.1145/344779.345012>
- C. Turk. 2002. *Effective Speaking: Communicating in Speech*. Taylor & Francis. <https://books.google.com/books?id=afITagAAQBAJ>
- Ubisoft. 2013. Just Dance Kids 2 I Am A Gummy Bear. (May 2013). <https://www.youtube.com/watch?v=HbZosS4dX3g>
- C. Vernallis. 2004. *Experiencing Music Video: Aesthetics and Cultural Context*. Columbia University Press. <https://books.google.com/books?id=DjDIw2pxjiMC>
- Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. 2006. The Cartoon Animation Filter. *ACM Trans. Graph.* 25, 3 (July 2006), 1169–1173. <https://doi.org/10.1145/1141911.1142010>
- Oliver Wang, Christopher Schroers, Henning Zimmer, Markus Gross, and Alexander Sorkine-Hornung. 2014. VideoSnapping: Interactive Synchronization of Multiple Videos. *ACM Trans. Graph.* 33, 4, Article 77 (July 2014), 10 pages. <https://doi.org/10.1145/2601097.2601208>
- Shen-Zheng Wang, Yung-Sheng Chen, Shih-Hung Lee, and C.-C. Jay Kuo. 2008. Visual Tempo Analysis for MTV-Style Home Video Authoring. In *Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 2 - Volume 02 (CISP '08)*. IEEE Computer Society, Washington, DC, USA, 192–196. <https://doi.org/10.1109/CISP.2008.721>
- David White, Kevin Loken, and Michiel van de Panne. 2006. Slow in and Slow out Cartoon Animation Filter. In *ACM SIGGRAPH 2006 Research Posters (SIGGRAPH '06)*. ACM, New York, NY, USA, Article 3. <https://doi.org/10.1145/1179622.1179626>
- Andrew Witkin and Zoran Popovic. 1995. Motion Warping. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. ACM, New York, NY, USA, 105–108. <https://doi.org/10.1145/218380.218422>
- WSJDigitalNetwork. 2012. Best Moments of First Obama/Romney Debate. (Oct 2012). <https://www.youtube.com/watch?v=QQC0nz0t9F4>
- YouTube:shubhgupta91. 2015. Turtle dancing at Satisfaction HD. (May 2015). https://www.youtube.com/watch?v=YE6_WbI0YLk
- Jean yves Bouguet. 2000. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs* (2000).
- Zumba with Layryn. 2014. "Danza Kuduro" Zumba Routine. (Jun 2014). <https://www.youtube.com/watch?v=gH20VFWEMdM>