Eventfulness for Interactive Video Alignment

JIATIAN SUN, Cornell University, USA
LONGXIULIN DENG, Cornell University, USA
TRIANTAFYLLOS AFOURAS, Meta AI, USA
ANDREW OWENS, University of Michigan, USA
ABE DAVIS, Cornell University, USA

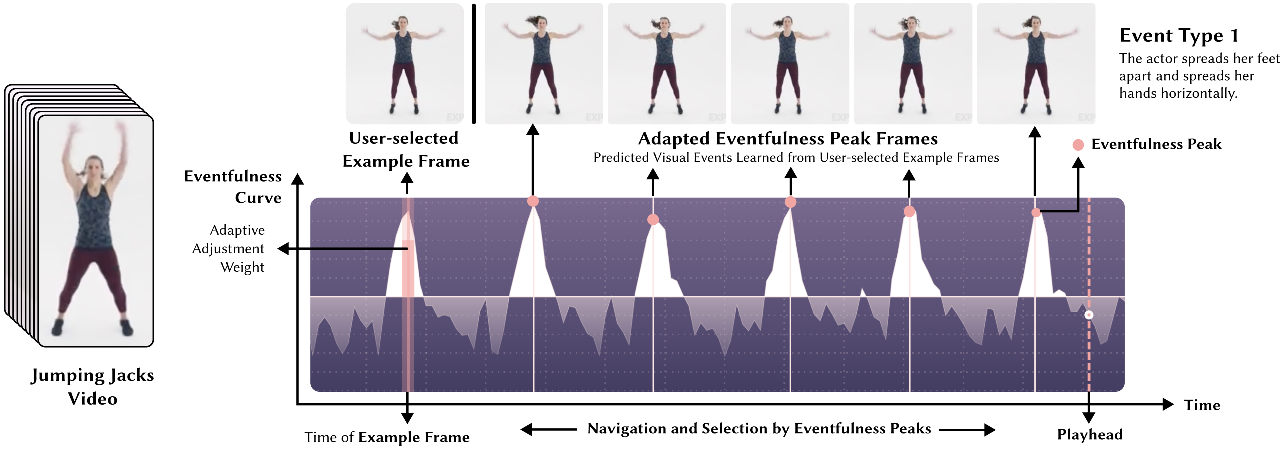
Event Type 1
The actor spreads her feet apart and spreads her hands horizontally.
User-selected Example Frame
Adapted Eventfulness Peak Frames
Predicted Visual Events Learned from User-selected Example Frames
Eventfulness Peak
Eventfulness Curve
Adaptive Adjustment Weight
Jumping Jacks Video
Time
Time of Example Frame
Navigation and Selection by Eventfulness Peaks
Playhead

Fig. 1. We learn a model of *eventfulness* in video, which represents the likelihood that different moments in video are the intended targets of synchronization tasks. Along with a general notion of eventfulness, we also learn descriptors for motion around each moment in video that can be used to adapt eventfulness based on a few representative, task-specific examples. Here we show that after providing example frames for the end poses of a single jumping jack, all of the subsequent corresponding poses are immediately identified as eventfulness peaks. These peaks can be navigated and edited (instead of video frames) making editing tasks much faster. We demonstrate our learned and adaptive eventfulness in novel tool for extracting and applying sound effects in video and for time-warping video based on a musical target.

Humans are remarkably sensitive to the alignment of visual events with other stimuli, which makes synchronization one of the hardest tasks in video editing. A key observation of our work is that most of the alignment we do involves salient localizable events that occur sparsely in time. By learning how to recognize these events, we can greatly reduce the space of possible synchronizations that an editor or algorithm has to consider. Furthermore, by learning descriptors of these events that capture additional properties of visible motion, we can build active tools that adapt their notion of eventfulness to a given task as they are being used. Rather than learning an automatic solution to one specific problem, our goal is to make a much broader class of interactive alignment tasks significantly easier and less time-consuming. We show that a suitable visual event descriptor can be learned entirely from stochastically-generated synthetic video. We then demonstrate the usefulness of learned and adaptive eventfulness by integrating it in novel interactive tools for applications including audio-driven time warping of video and the extraction and application of sound effects across different videos.

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: video, audio, alignment, editing, interaction, synth2real

ACM Reference Format:
Jiatian Sun, Longxiulin Deng, Triantafyllos Afouras, Andrew Owens, and Abe Davis. 2023. Eventfulness for Interactive Video Alignment. *ACM Trans. Graph.* 42, 4, Article 1 (August 2023), 10 pages. https://doi.org/10.1145/3592118

1 INTRODUCTION

One of the most frustratingly slow and tedious tasks in film making is synchronizing video and audio. Consider the act of manually synchronizing an audio clip with video through the lens of Fitts law [Fitts 1954], which states that the amount of time required

Authors' addresses: Jiatian Sun, js3623@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850; Longxiulin Deng, ld469@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850; Triantafyllos Afouras, afourast@robots.ox.ac.uk, Meta AI, New York City, New York, USA; Andrew Owens, ahowens@umich.edu, University of Michigan, Ann Arbor, Michigan, USA; Abe Davis, abedavis@cornell.edu, Cornell University, 107 Hoy Rd, Ithaca, NY 14853, Ithaca, New York, USA, 14850.

0730-0301/2023/8-ART1 $15.00
https://doi.org/10.1145/3592118

to move a cursor to a target is a function of the distance to the target divided by its width. Humans can detect misalignment of audio and video as small as one tenth of a second [Eg et al. 2015; Levitin et al. 2000], so if a 1-minute video clip spans 500 pixels on the timeline of an editing interface then the entire acceptable range of alignments with an audio clip (our target, in Fitts' terms) rests within the span of a single pixel. This makes the task of manually aligning audio with our video analogous to trying to click on a button that is less than one pixel wide. Plenty of alignment tasks in other applications call for this level of precision (e.g., centering 2D graphics on a presentation slide), but the tools we use to accomplish those tasks almost always employ some sort of snapping behavior to make the effective alignment target much larger. Professional audio editing tools similarly discretize possible points of synchronization based on musical metre or an onset envelope derived from audio (e.g., [Pro 2022]). But with no analogous mechanism to reduce the search space in video, many multimedia alignment tasks remain uniquely difficult for content creators. Our work addresses this challenge by learning a measure of *eventfulness*—the likelihood that each moment in a signal is the target of a synchronization task—for video. Building on this basic goal, we also learn descriptors of the motion around each moment in a video that let us adapt the notion of eventfulness to specific users and applications.

A key observation of our work is that, rather than offering an automatic solution to a specific alignment task, we can enable a broad range of more efficient interactive tools by learning a general and adaptive notion of eventfulness based heavily on low-level motion cues. We demonstrate this in the design of novel interactive interfaces for audio-driven time warping of video as well as the extraction and application of sound effects across different videos. Our contributions include:

- Introducing the idea of visual eventfulness for general interactive video alignment tasks.
- Showing that eventfulness can be learned entirely from stochastically generated synthetic video.
- A strategy for adapting eventfulness to specific tasks as part of interactive applications
- Demonstrating eventfulness in novel content creation tools for multimedia alignment tasks.

2 RELATED WORK

2.1 Synchronization, Alignment, & Snapping Behavior

Our work is perhaps closest in its goal to general alignment metrics used in other domains like 2D layout design and audio processing. More specifically, our goal is analogous to that of snapping behavior in 2D design tools (e.g., [Bier and Stone 1986; Ciolfi Felice et al. 2016]) and *novelty curves* (also sometimes called *onset envelopes*) from music information retrieval [Dixon 2006; Ellis 2007; Goto 2002; Grosche et al. 2010; Hu et al. 2017; Lerch 2012; McFee et al. 2015]. Also related are the concepts of *synchresis* from the arts [Chion et al. 1994], perceived synchrony from psychology [Dixon and Spitz 1980; Roseboom et al. 2009], and synchro-saliency from computer vision [Davis and Agrawala 2018], which all measure the strength of perceived synchronization between two signals. These can be modeled as conditional measures of eventfulness with respect to specific alignments of audio and video.

2.2 Learning to Synchronize Audio-visual Signals

The closest related work from vision and graphics is likely Davis and Agrawala [2018], which uses flow-based analysis to identify *visual beats* for synchronizing video with music. While their approach can work as a measure of eventfulness for videos with simple motion, it fails on complex real-world scenes including those with even minor camera motion. In contrast, we use a learned model of eventfulness trained on simulated motion that is more robust, and which can be easily adapt to specific tasks during use.

A variety of methods have evaluated the consistency of vision and sound. One line of work trains models to distinguish true audio-visual pairings from random pairings [Arandjelovic and Zisserman 2017] as a way of learning semantics. Other work learns to detect whether vision and audio are temporally aligned [Chen et al. 2021b; Chung and Zisserman 2016; Iashin et al. 2022; Korbar et al. 2018; Owens and Efros 2018] by training a model to distinguish between real and time-shifted examples. However, this work requires the audio and video come from the same scene. A similar line of work uses video textures to match an audio track [Narasimhan et al. 2022]. This however also requires that the audio in the input video already has co-occuring visual and audio events. Other work has synchronized music with body pose in dance video [Wang et al. 2020] or synchronizes speech with lips [Halperin et al. 2019]. While the goals of these methods are related, these approaches require detectors for specific body parts, while ours uses more generic motion cues and hence is not limited to a particular type of object. Other works use contrastive learning to retarget video at different speeds [Benaim et al. 2020] or learn to decompose video scenes into motion layers then and recompose them so that all motions appear temporally aligned [Lu et al. 2020].

2.3 Learning Motion from Synthetic Video

Previous work has learned optical flow from synthetically-generated video, where ground truth can be easily defined. Early work [Barron et al. 1994] proposed a simulation of a landscape sequence. Later work trained neural networks on simulations where 3D objects fly over images [Mayer et al. 2016], and 2D texture-mapped polygons [Dosovitskiy et al. 2015; Sun et al. 2021]. While we also learn about motion through 3D simulation, we use it to learn eventful motions, rather than optical flow.

2.4 Computational Video Editing & Dance

Our work is also related to other methods that incorporate automation into interactive tools for editing video, including [Berthouzoz et al. 2012; Davis and Agrawala 2018; Kopf et al. 2014; Leake et al. 2017]. Even closer to ours is work specific to aligning videos. Wang et al. [Wang et al. 2014] presented a method for interactively aligning multiple videos using a graph-based algorithm. Other work [Bazin and Sorkine-Hornung 2016] uses motion information to synchronize video of multiple people performing the same action using motion features. In contrast, our interactive tool is based on automatically-detected eventful keyframes that can easily be selected and aligned

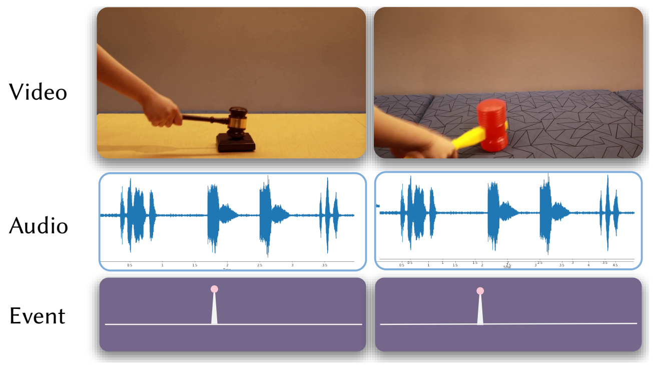

Fig. 2. **Eventfulness for audio-visual alignment**. Replacing the sound of the gavel with a squeak sound results in a video clip that can easily interpret, if the onsets are temporally aligned. This is in contrast with the more stringent criteria of synchronization [Chung and Zisserman 2016], which requires the signals come from the same underlying scene. Our representation allows users to interactively select these onsets and adapt them to a given task.

by a user. We also draw inspiration from related work dealing with visual rhythm and dance synthesis [Brick and Boker 2011; Chen et al. 2021a; chul Lee and kwon Lee 2005; Dyaberi et al. 2006; Kim et al. 2003; Liao et al. 2015; P. Chen et al. 2011].

3 EVENTFULNESS

Much of what sets our work apart from other work in audio-visual learning has to do with how we define *alignment*. Much of the related work in computer vision [Chung and Zisserman 2016] considers an audio signal to be synchronized with video if the corresponding samples from each occurred at the same time. This definition lends itself well to self-supervised learning, as most video is captured with a simultaneous audio stream [Korbar et al. 2018; Owens and Efros 2018]. However, this criteria can also be too stringent, for example in creative applications where otherwise unlikely pairings of audio and video are often the goal.

To better understand this, imagine that we are given two videos, each with simultaneously-recorded audio (Figure 2). The first video depicts the strike of a wooden gavel, while the second depicts a similar action performed with a plastic squeaky toy. A content creator may wish to swap the sound of the hammer with the squeaky toy for comic effect—in this case, the result is comical precisely because the synchronization is salient despite contradicting our expectations of each object. That contradiction is illustrative of what sets our work apart from others on video synchronization: our goal is to predict what would be salient, not necessarily what is likely.

3.1 Using Eventfulness

Our high-level goal is to narrow the search space for video synchronization tasks. Even without directly solving such tasks, we can greatly increase a user's efficiency by reducing the number of likely solutions they need to consider. However, there is a trade-off between how much we narrow the search space and how well we generalize to different tasks—after all, different tasks may involve synchronizing with different events from the same video. To navigate this trade-off we need some sort of prior on the type of event that a user cares about. Our strategy here is to start with a very general and application-agnostic prior, then adapt this prior in real-time to the input of individual users.

3.2 Application-Agnostic Eventfulness

We can use the idea of salience to motivate our general notion of eventfulness. Consider the scenario where we are given an audio clip $\mathbf{a}$ and tasked with finding the temporal window $\mathbf{v}_a$ of a video that maximizes perceived synchrony with $\mathbf{a}$. We can define the event $\mathbf{e}_t$ that a human observer perceives synchronization between the two signals after pairing our audio with the window of video beginning at time t. Here it is best to imagine $\mathbf{e}_t$ as a continuous measure of the expected response over some sample population (e.g., a percentage of viewers in some perceptual study reporting that $\mathbf{a}$ appears synchronized with $\mathbf{v}_a$). We can maximize the perception of alignment by optimizing:

$$\underset{t}{\text{argmax}}\left(P(\mathbf{e}_t|\mathbf{v}_t, \mathbf{a}_t = \mathbf{a})\right) \tag{1}$$

Applying Bayes rule gives us:

$$\underset{t}{\text{argmax}}\left(\frac{P(\mathbf{a}_t = \mathbf{a}|\mathbf{e}_t, \mathbf{v}_t) \quad \overbrace{P(\mathbf{e}_t|\mathbf{v}_t)}^{\text{marginal eventfulness}}}{P(\mathbf{a}_t = \mathbf{a}|\mathbf{v}_t)}\right) \tag{2}$$

The terms $P(\mathbf{a}_t|\mathbf{e}_t, \mathbf{v}_t)$ and $P(\mathbf{a}_t|\mathbf{v}_t)$ encode our expectations of what the content of our video is supposed to sound like—for example, whether a wooden gavel should squeek. Determining if specific pairings of audio and video are plausible involves context-specific reasoning about a joint probability distribution involving multiple domains (audio and video). The other term, $P(\mathbf{e}_t|\mathbf{v}_t)$, represents the prior probability that a human would consider the video to be temporally aligned with random audio at time t. We can think of this as a marginal probability over all signals we might want to synchronize with. We consider this *marginal eventfulness* a default that can be further refined given more information about a target task.

Note that our definition here could be applied to any source and target stimuli that one can localize in a common domain. This highlights our connection to analogous alignment tasks in audio and 2D design applications. It also suggests that our metric for eventfulness could potentially be used for alignment with other signals such as haptic feedback, though we leave this to future work.

3.3 Adaptive Eventfulness

Part of what makes creative video alignment tasks so difficult is the lack of a unique or universal solution. Video is often full of distinct overlapping events that could each individually be the correct answer to a different alignment problem. As such, any single measure of eventfulness must balance generality against efficiency at any one task. To navigate this balance, we treat fine-tuning eventfullness to downstream tasks as a few-shot learning problem. Rather than learning a single measure of eventfulness, we predict it alongside several other properties of motion that form an *event descriptor* at each moment in a video. Given a downstream application, we can then fine-tune our definition of eventfulness within this feature space based on, e.g., descriptors for a small set of labeled representative events.

4 LEARNING VISUAL EVENTFULNESS

Our strategy involves learning a "default" notion of eventfulness along with additional motion features that can be used to adapt to different tasks. We learn these feature signals by predicting parameters used to animate a large number of stochastically-generated synthetic videos. For this work, we intentionally designed our training videos to be very abstract as a way to keep our learned features application-agnostic, but we note that one could easily fine-tune on representative videos from a particular application to improve results on specific tasks. More details and data about our training process can be found on our project website

We render our training videos in Unity [uni 2021]. Each video contains randomized 3D shapes textured with randomly selected images. Each shape moves according to keyframes generated by a stochastic process. Here we derive labels for the event descriptors we want to predict directly from the scene graph used during rendering. Our network is then trained to predict these descriptors directly from the frames of each video.

4.1 Event Descriptors

4.1.1 Marginal Eventfulness. Our estimate of marginal eventfulness is itself one dimension of the descriptor we predict at each moment in video. We derive this label from the timing of animation keyframes that coincide with discontinuous object motion. The logic here is that our most generalizable condition for perceived synchrony is that events should be localizable in time. We calculate the marginal eventfulness labels for a video as a time signal with impulses added at each keyframe. We then blur the resulting sparse time signal with a Gaussian kernel using a standard deviation equal to one video frame.

4.1.2 Additional Event Descriptor Labels. Our event descriptor needs to balance an ability to distinguish events with our desire to recognize when an unlabeled event is similar to a provided example. We also want the overall dimensionality of our feature space to be low enough for us to adapt the notion of eventfulness in real-time. Here we found some flexibility in precisely what features to use. Our main implementation uses 4 octaves of progressively more blurred versions of our marginal eventfulness curve, as well as aggregate measures of several other properties derived from our generated motion. This includes sums over the velocity and acceleration of objects in camera space, with positive and negative motion in x and y each measured as separate dimensions. This adds 12 dimensions to our marginal eventfulness for a full 13-dimensional descriptor.

4.2 Synthesizing Training Data for Eventfulness

4.2.1 Synthesizing Motions. The synthetic video generation process is summarized in Figure 3. In order to create a video, we first randomly sample a static background image from ImageNet [Deng et al. 2009]. We then render a number of moving objects with randomized geometry and textures in front of this background. The resulting videos are somewhat similar to those used in synthetic optical flow datasets [Dosovitskiy et al. 2015], but designed for learning eventfulness over multi-frame time horizons rather than the two-frame setting of flow fields.

The geometry of each object is created by applying noise to scaled versions of various basic primitive shapes (e.g., ball, cube, rod, cylinder, and humanoid) using a displacement map. The texture of each object is selected randomly from ImageNet. We sample $n \in \{1, 2, 3, 4, 5\}$ objects in every scene.

To generate motion trajectories for each object, we first sample a finite set of event times $E = \{e_0, e_1, e_2, \ldots, e_n\} \in \mathbb{R}$. These event times will serve as the labels for our marginal eventfulness. To create motion with discontinuities at each event, we assign positions to each event in sequence, ensuring at each event that the direction to the subsequent event differs by some threshold from the direction to the previous event. Once the position of each event keyframe is chosen, we interpolate between event positions using a multi-segment cubic Bezier spline. To avoid large motion discontinuities far away from keyframes, we limit the control points for each Bezier segment to the axis-aligned bounding box defined by the endpoints of that segment. Moving objects are allowed to intersect and pass through each other.

4.2.2 Randomization. We randomize many aspects of our simulated videos to generate diverse training data (Figure 3). This includes the 3D geometry, pose, and texture of each object. We also randomize lighting with different combinations of spot lights, directional lights, point lights, rectangle lights, and disc lights of random colors. Camera motion is also randomized and synthetic camera shake is added to model video captured with hand shake. We model this camera shake by applying a randomized shake force to the camera counterbalanced by a restoration force. Each video is rendered with motion blur and anti-aliasing enabled. For the results in this paper, we trained on 2000 distinct 30-second clips totaling over 16 hours of video. Then, during training, we further augmented these clips with random spatial cropping and color jittering over brightness, contrast, hue, and saturation, as well as additional simulated camera shake applied to the field of view of each video.

4.3 Eventfulness Prediction Model

Let the input to the model be a video clip of T frames, $v \in \mathbb{R}^{T \times H \times W \times 3}$. To extract spatiotemporal visual features and per-frame event predictions, we use a variation of the popular (2+1)D ResNet-18 architecture [Tran et al. 2018], where we have removed several of the temporal strides in order to maintain dense outputs on the temporal dimension. The network consists of a deep stack of alternating spatial and temporal convolutions with residual connections [He et al. 2016]. The output of the network is a matrix $\mathbf{y}(\mathbf{v}) \in \mathbb{R}^{T \times N}$ of our N features estimated at each frame $t \in \{1 ... T\}$, which can equivalently be viewed as a time signal of per-frame descriptors $\mathbf{y}(\mathbf{v}) = [y_0, y_1, ..., y_T]$.

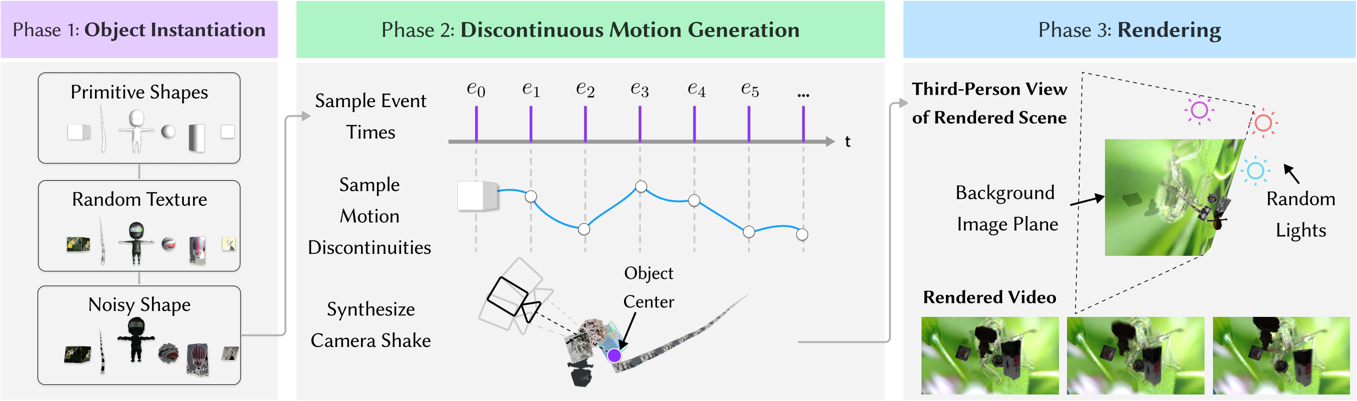
Phase 1: Object Instantiation
Primitive Shapes
Random Texture
Noisy Shape
Phase 2: Discontinuous Motion Generation
Sample Event Times
e_0 e_1 e_2 e_3 e_4 e_5 ...
t
Sample Motion Discontinuities
Synthesize Camera Shake
Object Center
Phase 3: Rendering
Third-Person View of Rendered Scene
Background Image Plane
Random Lights
Rendered Video

Fig. 3. **Synthesizing Training Data**. We learn a model of visual eventfulness from stochastically generated synthetic video. Our process for generating videos has three phases. First we generate geometry by selecting from a set of primitives, then adding random geometric noise through displacement maps and texturing geometry with random images from ImageNet [Deng et al. 2009]. In the second phase, we sample motion trajectories for scene objects and the camera. Each scene object follows a path determined by randomly generated keyframes, while camera motion follows a simple randomized model of camera shake. In the third phase we render the generated objects according to the generated motion curves under randomized lighting and over a randomly selected background image.

4.4 Training

We train our model to regress the feature signals for each training video. Given training data $\mathcal{D}$ consisting of pairs $(\mathbf{v}, \mathbf{y}^*)$ of video clips $\mathbf{v}$ and ground truth event feature targets $\mathbf{y}^* \in \mathbb{R}^{T\times N}$, the model is trained to minimise the Mean Squared Error (MSE):

$$\mathcal{L} = \mathbb{E}_{(\mathbf{v},\mathbf{y}^*)\in\mathcal{D}} \sum_{t=1}^{T} \left\| y_t - y_t^* \right\|^2. \tag{3}$$

5 INTERACTIVE TOOLS

We observe that, while different applications may care about different events in video, the distribution of such events tends to be sparse. Our approach to incorporating eventfulness in interactive tools reflects this observation. We start with marginal eventfulness as a default prior on the distribution of events a user wants to select, then we refine this prior in real-time as the user interacts with the application. The refining process can be seen as a kind of few-shot learning. For this, we view the predicted values $\mathbf{y}(\mathbf{v})$ as feature descriptors for each moment in video.

We demonstrate eventfulness in three novel interactive tools: the first two are designed to help with the extraction and application of sound effects to and from video. The third is designed to help time-warp one video based on events in another video or audio signal, which we use to perform dancification [Davis and Agrawala 2018]. The role of eventfulness in all of these applications is to reduce the space of synchronization events that a user must search through for each associated task.

5.1 Adaptive Event Navigation and Selection

Each of our three tools involves selecting events in a video timeline, and each of our interfaces shares some common layout and functionality for this part of the task (see Figure 4). We use eventfulness to reduce the search space of events, which we do first by visualizing eventfulness on the video timeline and second by allowing direct navigation to and between peaks of the eventfulness signal using the keyboard. Users can also adjust the eventfulness signal by selecting an example event and either increasing or decreasing its weight, which will adjust the curve response at similar events accordingly.

5.1.1 Timeline GUI. Users see a two-row timeline panel displayed below their video. The top row shows a zoomed-out view of the current adaptive eventfulness curve for the entire video. This first row acts as a mini-map, with an adjustable highlighted region that is scaled up and displayed at higher resolution in the second row. Users can navigate the timeline with their keyboard, mouse, or some combination of the two. Keyboard navigation provides the option of navigating by consecutive frames or by peaks of the current eventfulness signal. The eventfulness signal is normalized to a fixed range with negative values displayed in a slightly darker color. By default, we initialize the eventfulness signal to our estimate of marginal eventfulness. Users can perform three different selection actions at the current event: one performs a task associated with the current application (e.g., adding a sound to the current event), a second marks the current event as a positive example, and a third marks the current event as a negative example. Users can also increase or decrease the extent to which the eventfulness curve adapts according to these positive examples.

5.1.2 Adapting Eventfulness. Given a set of positive and negative example events, the goal of adapting our eventfulness signal should be to increase the response of events that are similar to our positive examples and decrease the response of events that are similar to our negative examples. For interactive use, we would also like for adaptation to happen smoothly and at interactive rates. Here, there are many approaches one could take, but we opt for a very simple strategy where the current eventfulness signal is calculated by first

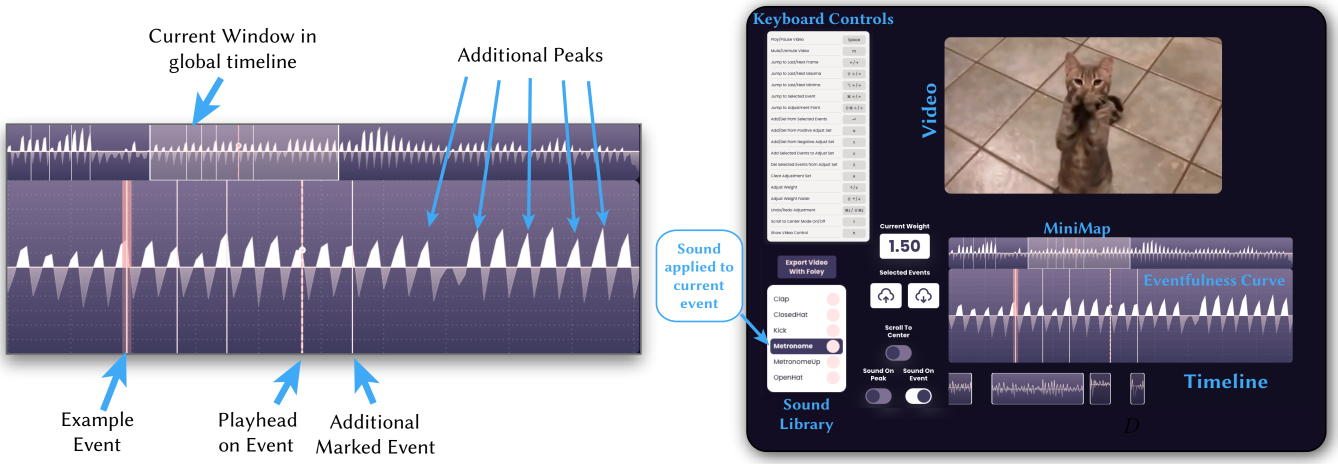

Fig. 4. **Adaptive Event Navigation** (left) & **Foley Sound Effect Tool** (right). Our interface for navigating and selecting events (left) shows the current eventfulness curve displayed over a video timeline. Users can navigate using standard timeline controls, or by jumping between consecutive eventfulness peaks. They can also indicate example events and adjust the weight of those events. Our Foley application (right) further lets users extract sound effects from selected events or add selected sound effects to events.

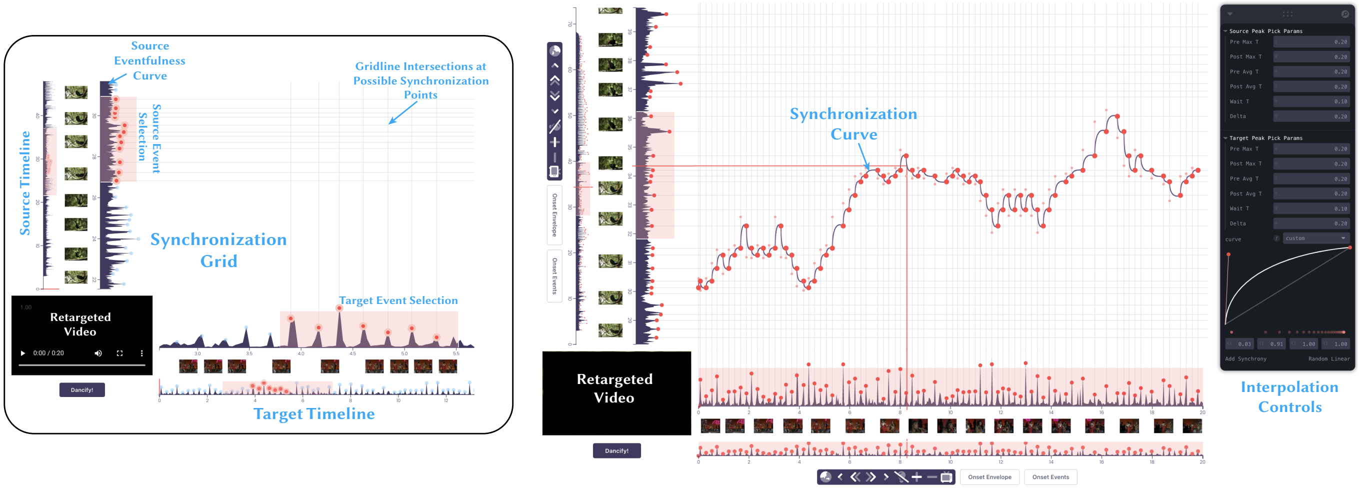

Fig. 5. **Time Warping Tool** Our tool lets users time warp a selected source video based on a mapping of events to those of a target signal. In the example shown here, the target signal is a piece of music, and target events are the beats of that music. The synchronization grid visualizes all possible synchronizations of a source video event with a target beat. The synchronization curve shown in the right screenshot is defined by a sequence of these synchronization points, which are then interpreted as keyframes. Our interface further lets users define the interpolation between keyframes using standard animation tweening curves.

projecting the features $\mathbf{y}(\mathbf{v})$ onto some unit weight vector $\vec{\mathbf{w}}$, and then re-scaling the resulting time signal to a fixed value range. We can think of this as representing the eventfulness of each frame by the projection of its features onto the descriptor of some representative event. Our default starting weights are given by the unit vector with a 1 in the feature dimension corresponding to marginal eventfulness and 0's in all other dimensions. As a user adjusts eventfulness, we interpolate this weight vector toward the normalizes sum of positive example descriptors minus negative example descriptors. Figure 6 shows an example video featuring a swinging mouse. Here, three different users could be interested in three different events: for example, one may be interested in events where the mouse reaches the left peak of its swing, another in events where it reaches the bottom of the swing, and a third may be interested in events at the right peak of its swing. In each case, with just a few

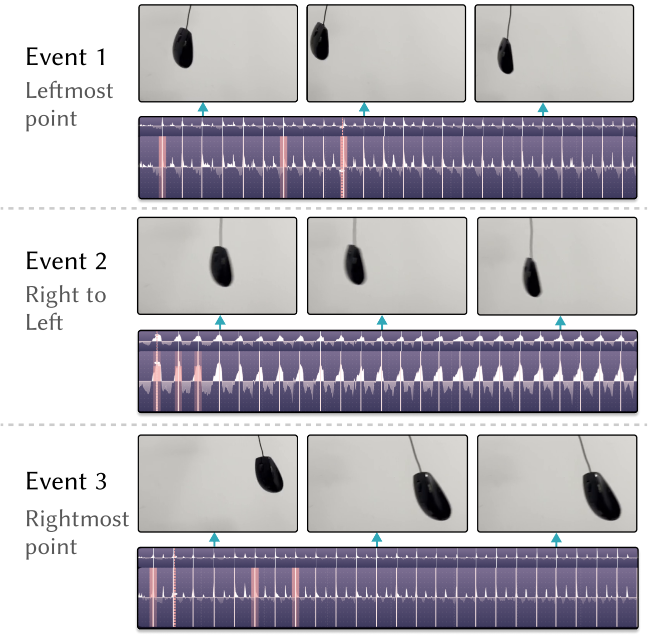

Fig. 6. **Adaptive Eventfulness**. Here we show three different eventfulness curves adapted to three different types of events within a scene. The example scene here is quite simple: a computer mouse swings in a repetitive pendulum motion. Each curve is adapted to three provided examples events. The first and third curves select for the leftmost and rightmost parts of the swing, respectively, while the second curve selects for events that are mid-swing from right to left. In each case, the provided examples are enough to highlight corresponding events throughout the rest of the video. We also see that the eventfulness peaks are sharper in the first and last curves, which reflects the fact that the extrema of the swing are more localizable.

(2-3) example events you can adjust the eventfulness to isolate the desired events in the video.

In addition to visual eventfulness, our interface provides users the option of selecting events based on what we call *estimated synchrony*, which is the element-wise product of the visual eventfulness curve with the audio onset envelope. Estimated synchrony is a good way to search for events that are simultaneously salient in both audio and video.

5.2 Foley Extraction

Our first application is designed to help with the extraction of sound effects from video for later use. While some recent commercial tools offer features that use detected audio onsets to navigate and align audio, our tool has the added benefit of analyzing visual eventfulness. We can use visual eventfulness directly to find certain events, or we can use estimated synchrony as a way to filter for moments that are salient in both audio and video. For example, one of the results in our supplementary material comes from a video interview with a Foley artist [Ament 2014]. In the video, the artist discusses his work while demonstrating how various sounds are made, resulting in a mix of audio onsets caused by speech and by the sound effects he is demonstrating. Here, we can use estimated synchrony to filter for events that are both audio onsets and visual eventfulness peaks, which helps to filter out dialogue and isolate events like swords clashing.

5.3 Foley Application

Our second tool is for inserting sound effects at selected events in videos. Here, users can make direct use of output from our first tool, or select from a bank of existing sound effects. Users navigate the timeline and select events where sound effects should be added. For each selected event, they can also change what sound should be added. For example, one example in our supplemental material shows a user adding metal sword clashes to a video of a child practicing kendo. In general, we found that when we are replacing an existing sound, estimated synchrony (the product of visual eventfulness and an audio onset curve) is often best for event selection. When we are adding sound to silent parts of a video, we found that visual eventfulness obtains better performance, since the existing audio may not always be reliable.

5.4 Dynamic Video Synchronization

Our third application deals with the challenging task of dynamically warping a video into alignment with some target signal. In prior work, this has been accomplished through optimization of a continuous objective [Wang et al. 2014]. Here we address scenarios closer to the dancification explored in Davis and Agrawala [2018], where discrete visual events are to be synchronized with discrete events in a target signal (e.g., aligning visual events with beats in a piece of music). Our interface plots the events of a source video against those of a target signal (Figure 5). The target signal can be another video or a piece of audio—for dancification we use musical beats for target events. The source video's timeline is mapped to the y axis of the interface, and the target signal is mapped to the x axis. We extend horizontal lines from each source event and vertical lines from each target event to create a grid. Each intersection of lines in this grid represents a possible synchronization between a source event and a target event. Users can double-click a grid intersection to add the corresponding synchronization constraint. Each target event can have at most one synchronization point, but source events can be synchronized with multiple target events. We connect the selected synchronization points to form an animation curve, with each point acting as a keyframe. Users can also control the interpolation between keyframes by adjusting standard Bezier curve controls just as they would in traditional animation software. We also give users the option of automatically generating a random walk over a selected range of source and target events. See our supplemental material for a demonstration.

6 EVALUATION

We found adaptive eventfulness to be useful in each of our three applications. The best way to experience these results is by exploring the supplemental demos and applications on our project website. We also show examples of visual eventfulness being used to synchronize different videos of dancers performing similar choreography. In addition to those qualitative results, we use three labeled datasets

Table 1. **Performance of our proposed method on event detection.** We use average precision to compare different methods in detecting labeled events. We use two variations of our outputs: marginal eventfulness only (ME) and adaptive eventfulness (AE), which uses 3 ground truth events to reweigh the eventfulness curve. VisBeat is from Davis and Agrawala [2018] while the SparseSync is an audio-video alignment network from Iashin et al. [2022]. SparseSync-O is generated by evaluating the audio-video alignment score of the original audio clip and video clip contained in a sliding window over the timeline. Similarly, SparseSync-S is generated by evaluating the audio-video alignment score of a synthetic impulse sound and the original video clip contained in a sliding window over the timeline.

Method	Greatest Hits	Bouncing Ball	Videos in the Wild
VisBeat	0.10	**0.57**	0.22
SparseSync-O	0.10	0.02	0.10
SparseSync-S	0.11	0.11	0.09
Ours (ME)	0.14	0.27	0.10
Ours (AE)	**0.14**	0.56	**0.31**

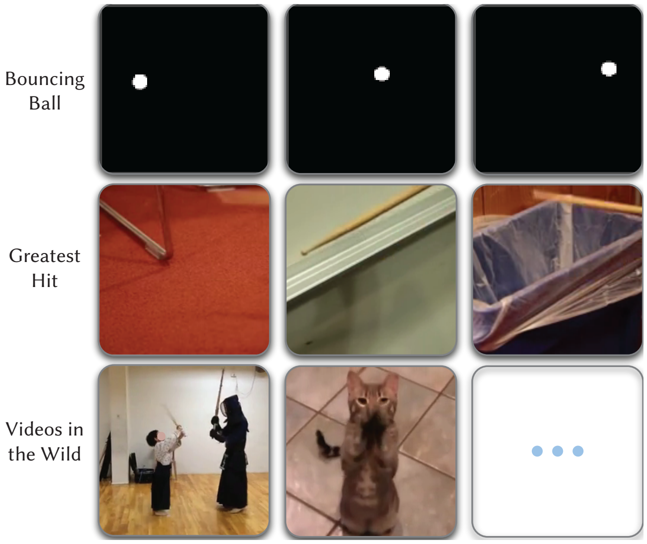

Fig. 7. **Example frames from the three datasets used for evaluation**.

to examine our learned eventfulness metric and compare it with metrics from previous work here.

6.1 Quantitative Evaluation

The application-dependent nature of eventfulness makes quantitative evaluation difficult, but we can examine how correlated marginal eventfulness is with some known events in labeled video, and we can examine how much this correlation increases when we adapt to example events. We can also compare these with alternative metrics from previous work, including visible impact [Davis and Agrawala 2018] and SparseSync Iashin et al. [2022]. We do this on three datasets. Example frames from each dataset are shown on Figure 7.

Bouncing Ball. This is an extremely simple toy dataset created by simulating a 2D ball as it bounces around a rectangular window. The ball moves at a constant speed and changes direction at each bounce. Ground truth events are marked whenever the direction of the ball changes. We synthesize 200 videos, each 30 seconds long with variations in ball speed and size.

The Greatest Hits Dataset [Owens et al. 2016]. contains recordings of a person interacting with various objects in indoor and outdoor scenes by hitting or scratching them with a drumstick. The dataset comprises 977 videos which contain 46,577 actions in total. Each hit or scratch generates a clear sound and can be identified by an audio onset. We obtain pseudo-ground truth event times by detecting these onsets [McFee et al. 2015].

Videos in the Wild. We also collected several "in the wild" videos from YouTube, focusing on those containing complex, semi-repetitive motions. These videos are particularly challenging since salient visual events are often uncorrelated with audio onsets, and different applications could potentially call for synchronizing with different events from the same video. We invited 10 users to label 3 videos in this dataset, instructing each user to select events that were similar to two provided example events from each video.

6.2 Quantitative Results

In Table 1, we show the results of using marginal and adaptive eventfulness to predict labeled events. Adaptive eventfulness is evaluated by randomly selecting 3 ground truth events from each video and using them as positive examples to reweigh eventfulness. Additionally, we compared with [Davis and Agrawala 2018], which uses hand-crafted flow-based heuristics to detect motion discontinuities, and with SparseSync Iashin et al. [2022], a state-of-the-art synchronization network. We evaluate each metric using average precision, using an approximately 150ms threshold, similar to Owens et al. [2016]. We found that adapting to positive examples significantly improves results. Our adapted model outperforms all other baselines on both real video datasets. Davis and Agrawala [2018] performs best on the *Bouncing Ball* videos, as they are synthetic videos that are designed to meet the assumptions made by the visible impact heuristic. The performance of SparseSync reflects the difference between the synchronization task and detecting eventfulness.

7 DISCUSSION

7.1 Limitations & Future Work

Our learned eventfulness metric is intentionally permissive to accommodate a broad range of potential use cases. Training on abstract synthetic videos was one way to favor this kind of generality. However, given the data and resources, one could train on a rich and varied set of real videos to learn a descriptor space capable of more precise fine-tuning. Along similar lines, our current eventfulness metric is designed to be largely invariant to specific image content, instead relying on the characteristics of motion around each moment in video. In many real applications, it may be useful to isolate events based on the appearance of their accompanying frames.

In this work we focused mostly on applications involving alignment of audio and video. However, the notion of adaptive visual eventfulness could be extended to the exploration and analysis of more arbitrary events in video, which could have applications in interactive search and labeling of events, for example, in scientific applications.

7.2 Conclusions

We have proposed adaptive visual eventfulness as a useful tool for interactive multimedia alignment. We showed that a useful adaptive eventfulness metric can be learned from synthetic data and applied to real videos. We also presented novel interactive tools that use adaptive eventfulness to align media in a variety of ways.

ACKNOWLEDGEMENT

This work was partially funded by a generous gift from Meta.

REFERENCES

2021. Unity Game Engine. https://unity.com.

Vanessa Theme Ament. 2014. *The Foley grail: The art of performing sound for film, games, and animation.* Routledge.

Relja Arandjelovic and Andrew Zisserman. 2017. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision.* 609–617.

John L Barron, David J Fleet, and Steven S Beauchemin. 1994. Performance of optical flow techniques. *International journal of computer vision* 12, 1 (1994), 43–77.

Jean-Charles Bazin and Alexander Sorkine-Hornung. 2016. Actionsnapping: Motion-based video synchronization. In *European Conference on Computer Vision.* Springer, 155–169.

Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. 2020. SpeedNet: Learning the Speediness in Videos. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), 9919–9928.

Floraine Berthouzoz, 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

Eric A. Bier and Maureen C. Stone. 1986. Snap-Dragging. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '86).* Association for Computing Machinery, New York, NY, USA, 233–240. https://doi.org/10.1145/15922.15912

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

Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. 2021b. Audio-visual synchronisation in the wild.

Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. 2021a. ChoreoMaster: Choreography-Oriented Music-Driven Dance Synthesis. *ACM Trans. Graph.* 40, 4, Article 145 (jul 2021), 13 pages. https://doi.org/10.1145/3450626.3459932

M. Chion, C. Gorbman, and W. Murch. 1994. *Audio-vision: Sound on Screen.* Columbia University Press. https://books.google.com/books?id=BBs4Arfm98oC

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.

Joon Son Chung and Andrew Zisserman. 2016. Out of time: automated lip sync in the wild. In *Asian conference on computer vision.* Springer, 251–263.

Marianela Ciolfi Felice, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2016. Beyond Snapping: Persistent, Tweakable Alignment and Distribution with StickyLines. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16).* Association for Computing Machinery, New York, NY, USA, 133–144. https://doi.org/10.1145/2984511.2984577

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

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 248–255.

Norman F Dixon and Lydia Spitz. 1980. The Detection of Auditory Visual Desynchrony. *Perception* 9, 6 (1980), 719–721. https://doi.org/10.1068/p090719 arXiv:https://doi.org/10.1068/p090719 PMID: 7220244.

Simon Dixon. 2006. Onset detection revisited. In *In Proceedings of the 9th international conference on digital audio effects.* 133–137.

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision.* 2758–2766.

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

Ragnhild Eg, Carsten Griwodz, Pål Halvorsen, and Dawn Behne. 2015. Audiovisual Robustness: Exploring Perceptual Tolerance to Asynchrony and Quality Distortion. *Multimedia Tools Appl.* 74, 2 (jan 2015), 345–365. https://doi.org/10.1007/s11042-014-2136-6

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

P. M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental PSychology* 74 (1954), 381–391.

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

Tavi Halperin, Ariel Ephrat, and Shmuel Peleg. 2019. Dynamic Temporal Alignment of Speech to Lips. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 3980–3984. https://doi.org/10.1109/ICASSP.2019.8682863

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 770–778.

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

Vladimir Iashin, Weidi Xie, Esa Rahtu, and Andrew Zisserman. 2022. Sparse in Space and Time: Audio-visual Synchronisation with Trainable Selectors. *arXiv preprint arXiv:2210.07055* (2022).

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

Johannes Kopf, Michael F. Cohen, and Richard Szeliski. 2014. First-Person Hyper-Lapse Videos. *ACM Trans. Graph.* 33, 4, Article 78 (jul 2014), 10 pages. https://doi.org/10.1145/2601097.2601195

Bruno Korbar, Du Tran, and Lorenzo Torresani. 2018. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230* (2018).

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.

Daniel J. Levitin, Karon MacLean, Max Mathews, Lonny Chu, and Eric Jensen. 2000. The perception of cross-modal simultaneity (or "the Greenwich Observatory Problem" revisited). *AIP Conference Proceedings* 517, 1 (2000), 323–329. https://doi.org/10.1063/1.1291270 arXiv:https://aip.scitation.org/doi/pdf/10.1063/1.1291270

Zicheng Liao, Yizhou Yu, Bingchen Gong, and Lechao Cheng. 2015. audeosynth: Music-Driven Video Montage. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015).

Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T Freeman, and Michael Rubinstein. 2020. Layered neural rendering for retiming people in video. In *SIGGRAPH Asia.*

Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. 2016. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 4040–4048.

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.

Medhini Narasimhan, Shiry Ginosar, Andrew Owens, Alexei A Efros, and Trevor Darrell. 2022. Strumming to the Beat: Audio-Conditioned Contrastive Video Textures. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 3761–3770.

Andrew Owens and Alexei A. Efros. 2018. Audio-Visual Scene Analysis with Self-Supervised Multisensory Features.

Andrew Owens, Phillip Isola, Josh H. McDermott, Antonio Torralba, Edward H. Adelson, and William T. Freeman. 2016. Visually Indicated Sounds. 2405–2413.

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.

Logic Pro. 2022. Logic Pro X.

Warrick Roseboom, Shin'ya Nishida, and Derek H Arnold. 2009. The sliding window of audio–visual simultaneity. *Journal of vision* 9, 12 (2009), 4–4.

Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. 2021. AutoFlow: Learning a Better Training Set for Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 10093–10102.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. 6450–6459. https://doi.org/10.1109/CVPR.2018.00675

Jianren Wang, Zhaoyuan Fang, and Hang Zhao. 2020. Alignnet: A unifying approach to audio-visual alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.* 3309–3317.

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 (jul 2014), 10 pages. https://doi.org/10.1145/2601097.2601208