Let the Beat Follow You - Creating Interactive Drum Sounds From Body Rhythm

Xiulong Liu
University of Washington
xl1995@uw.edu

Kun Su
University of Washington
suk4@uw.edu

Eli Shlizerman
University of Washington
shlizee@uw.edu

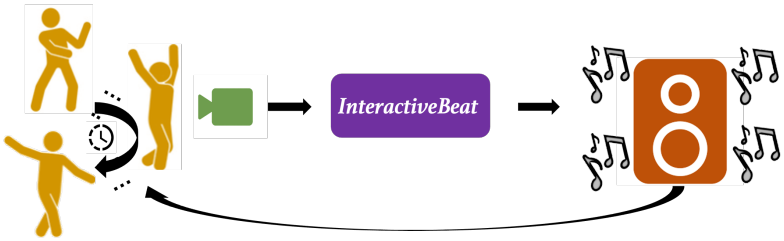

Figure 1. InteractiveBeat: A real-time system responding to human body movements captured by video camera by generating sounds that follow the rhythm of the movements. Please see supplementary videos and materials with sample results.

Abstract

It is often the case that human body movements include rhythmic patterns. A video camera system that captures these patterns and responds to them with rhythmic sounds or music, as these happen, could create a unique interactive experience. Creating such an experience requires a real-time translation of related visual cues into in-rhythm sounds and warrants novel real-time methods. In this work, we propose a novel learning-based system, called 'InteractiveBeat', which generates an evolving interactive soundtrack for a camera input that captures person's movements. InteractiveBeat infers body skeleton keypoints and translates them into drum rhythms using a series of sequence models. It then implements a conditional drum generation network for generating polyphonic drum sounds based on the rhythms. To guarantee real-time function, these models are integrated into a time-evolving pipeline with rules for updates. InteractiveBeat is trained and evaluated on a well-annotated large-scale dance database (AIST), and in addition, we collected a dataset of in-the-wild videos with people performing movements of various activities that correspond to background music. Furthermore, we develop a 'live' demo prototype of the system. Our evaluation results show that the system can generate interactive rhythmic drums more accurately than existing methods and achieves a non-cumulative latency of 34ms (approx. 30 fps). This allows InteractiveBeat to be synchronized with the video stream and react to real-time movements.

1. Introduction

We have an invisible metronome system in our brain and body [25, 31]. When we see a dance move, we often imagine a rhythmic sound that follows it. Also, when we hear rhythmic sounds, we sometimes instinctively start following them with synchronous movements. Such a natural experience involves synchronization of listening to the music, following the beat, and creating new movements. In recent years, the demand for real-time audio-visual creation tools has become more imminent as virtual and augmented reality platforms have gained traction. Instead of merely selecting from a predefined set of soundtracks, these tools have the potential to generate soundscapes dynamically, adapting in real-time to movements of the user. This fosters an evolving 'dialogue' between the user and the environment, bringing the immersive experience to a new higher level. Such real-time generative capabilities in audio-visual systems have broad applications, spanning from interactive dance workouts and immersive gaming, to therapeutic VR [54], where a tailored, responsive musical virtual environment can aid patients by adjusting to their movements and guiding them towards new movements.

Imagine a camera system where a person faces the camera and starts to make movements. The system reacts to movements with sounds in real-time corresponding to the rhythms of the movements. The auditory feedback from sounds helps the person to perceive the movements and their rhythm, and adapt to a particular desired rhythm or switch to another rhythm when proceeding with the movements.

This motivates the system that we propose in this work, *InteractiveBeat*. Beyond generating soundtracks for various movements, the system provides an immersive experience of real-time interaction with the soundtrack, where any person can create rhythmic sound effects with their bodies.

Generating rhythmic soundtracks from videos of human body movements is challenging because audio and visual modalities are not explicitly and uniquely related. Furthermore, implementing the generation in real-time adds to the challenges since the system must (i) be causal, with only past information used but not future, (ii) respond quickly to visual cues while dealing with excessive information which could be unrelated, and (iii) achieve plausible perceptual generation that aligns well with movements.

In this work, we address these challenges by developing a novel system, *InteractiveBeat*, which is a first-of-its-kind learning-based real-time vision-based system for rhythmic drum sound generation in response to human body movements being captured by a video camera. InteractiveBeat introduces (i) a learning-based approach that redesigns the traditional motion rhythm extraction algorithm (offline visual beat detection), enabling its seamless transition to a real-time operation, (ii) a style transfer module that maps motion rhythm to drums rhythm, (iii) a compact polyphonic drum generative model that translates rhythm to drum sounds. An overview of InteractiveBeat is shown in Fig. 2.

To complete the pipeline, we integrate real-time motion-estimation as the system's front-end, and design a producer-consumer workflow that includes updating rules to support real-time improvisation. Each component is implemented by compact networks and is able to run in real-time.

We train and evaluate our system on a well-annotated AIST dance database [53] and on a novel dataset of 'in-the-wild' clips from YouTube and TikTok with a total of 764 videos and 6+ hours of duration that we have collected. Furthermore, we implement a real-time pipeline to test the system. Objective metrics and human studies results show that InteractiveBeat reacts interactively to human motion with interesting sound, plausible synchrony, and minimal latency. In summary, in this work, our main contributions are:

- We propose a new application and task, learning-based real-time interactive rhythmic audio generation based on person's movements.
- The *InteractiveBeat* system that we introduce is a novel learning-based real-time drum sound generation system that is solely based on video camera input.
- InteractiveBeat implements a real-time pipeline that ensures synchronization with human motion while providing rich rhythmic drum sounds instead of plain beats.
- We collect a novel in-the-wild dataset of human movements with music in diverse visual scenes, and then separate its drum track from the original music to represent the rhythmic audio-visual scenes.

2. Related Work

Audio-visual learning, an emerging branch in multi-modal vision-based learning, that studies the relationship between audio and vision, has made significant progress in recent years. Numerous tasks including audio-visual correspondence [2,3,26,42], audio-visual event localization [52], audio-visual sound source separation [20,21,57,58], audio-visual navigation [7], audio-conditioned generation of human body movements [23,36,45], lips movements [51] and talking faces [33, 39, 59] are proposed. Learning-based vision to audio generation has also been explored. Image-to-audio generation leverages deep neural networks that take a single image as input and generate different types of audio (natural sounds, impact sounds, reverberation etc) in forms of spectrogram or audio waveform [8, 41, 46, 60].

Video-based audio generation explores the possibility of generating audio conditioned on different dynamic visual cues. When audio is set to music, prior works have shown that deep neural networks can predict the pressed keys of a top-down view of a piano performance [34] and then generate piano music correspondingly [48]. Later, Foley music [19] and Multi-instrumentalist Net [49] extend such generation to the music of different instruments conditioned on body movements. Rhythmic Net [50] and cmt [14] expand music generation from videos to a broader scope - videos that contain general human body movements or require background music. More recently, the emergence of novel deep generative models gave rise to approaches of music audio waveform generation from videos [61] [62] [47]. While these video-to-music generation systems can produce new music, their models are large by design, and the output at each step relies on global information (non-causal). These constraints preclude such systems from operating in real-time. Furthermore, the generated music is not suitable for interactive applications since the generative models strictly adhere to music rules, i.e., fixed tempo and bar-level structures, in contrast to human movements that are not necessarily restricted to these rules.

For real-time and interactive applications, it has been shown that a rule-based sensor system to correlate movement and sounds could potentially convert the sensed motion to MIDI [4, 5]. In particular, an artistic and pioneered vision-based real-time music system (VNS) [56] dating back to the late 90s, showed that lighting changes could be related to motion and these can be set through a set of rules to generate associated sounds. Similar works, such as [17], also proposed to relate visual changes to sound effects according to different rules. Following these works, later on, an interactive background music synthesis algorithm guided by visual content was introduced to synthe-

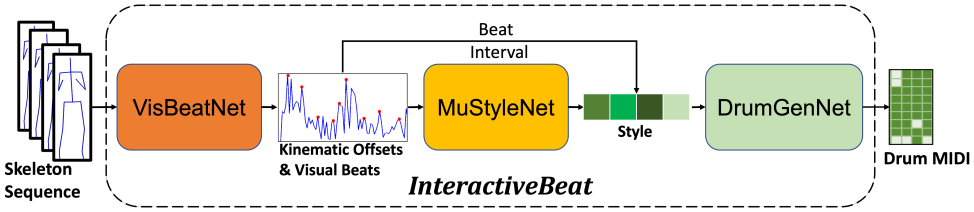

Figure 2. System Overview: Three stages of InteractiveBeat: (i) VisBeatNet for prediction of kinematic offsets and visual beats to estimate beat interval, (ii) MuStyleNet transfers kinematic offsets to 'style,' a vector representing drum rhythm, (iii) DrumGenNet translates 'style' to Drum MIDI in the next beat interval.

size dynamic background music for different scenarios [55]. These methods show the plausibility of the interactive correlation of movements and sound. Their limitations include reliance on delicate rules designed by music professionals or usage of music retrieval that cannot generate new music. In our work, we aim to generate new interactive soundtracks based on live camera input via a system that learns and generalizes such that it is applicable to body movements of various activities.

In the music generation domain, Musical Instrument Digital Interface (MIDI) has been used for efficient and high-quality music generation. Piano-roll [15] [22, 43] or event-based [27, 28, 30, 40] MIDI representations have been developed to generate music unconditionally in long sequences. In addition, when extra conditions are applied, deep generative network-based systems are capable of generating waveform from music attributes [16], and text descriptions [1, 9, 13, 29, 44]. For drum generation, GrooVAE [35] was developed to generate kick drums given conditional signals, including beat, downbeat, onset of snare, and Bass. RhythmicNet [50] improved the drum quality by designing a two-stage network system (a transformer and UNet) to model hits and 'style' separately. These methods cannot achieve real-time operation since they rely on long context length and incorporate large networks or multiple stages to generate music. We thereby implement a compact encoder-decoder network that infers the 2D drum roll matrix from a 1D rhythm sequence without long context.

3. Methods

We design our system, InteractiveBeat, to generate rhythmic drum sounds with core objectives of being real-time, aligned with human body movements, drum sounds with coherent rhythmic structures. To meet these objectives, InteractiveBeat consists of three neural network components:

- **VisBeatNet** predicts kinematic offsets, visual beats, and estimates tempo from a live stream of human motion.
- **MuStyleNet** transforms kinematic offsets into a drum 'style.'
- **DrumGenNet** synthesizes a polyphonic drum track based on the estimated tempo and inferred drum 'style.'

A real-time producer-consumer pipeline integrates the above network components with the motion estimation front-end.

This design is different than related recent work of RhythmicNet [50]. In particular,VisBeatNet and MuStyleNet components, as described below, implement a more effective way to bridge the motion rhythm and drum rhythm than 'Video2Rhythm' in RhythmicNet, and work in real-time. Furthermore, DrumGenNet adopts a similar network structure as the first stage of 'Rhythm2Drum' component of RhythmicNet, but with a more compact design to meet the real-time requirement.

3.1. VisBeatNet

To predict motion rhythm and to estimate the tempo, we develop a novel approach, *VisBeatNet*. This is inspired by an optical flow-based visual beats prediction method [11]. While the original method relies on offline operations such as windowing, filtering, and dynamic programming optimization, we adapt it for real-time applications. Specifically, *VisBeatNet* employs a compact neural network that is trained to predict visual beats in real-time. The training uses ground truth derived from a robust pre-computation of visual beats. In the following, we review the background of **visual beats pre-computation** and then describe **VisBeatNet**, our solution for real-time application.

3.1.1 Visual Beats Pre-computation

We use a real-time human pose estimator (OpenPose) [6] to extract the 2D skeleton key-points from a real-time video stream, and pre-compute the visual beats ground truth via: computing the Directogram, converting the Directogram to Kinematic Offsets, and performing dynamic programming to obtain the Visual Beats. Each stage is detailed below.

- **Computing the Directogram from Skeleton Sequence:** Skeleton sequence is considered as a three-dimensional tensor $S \in \mathbb{R}^{T \times J \times 2}$ where T is the number of frames,

J is the number of keypoints, and the last dimension indicates x and y coordinates. By computing the first order difference of this 3D skeleton tensor, $\Delta S_t = S_t - S_{t-1}$, we capture motion at each frame. Using polar coordinates of the last dimension and splitting the full circle $(0, 2\pi)$ into N equal bins, we assign the motion magnitude of every key point into one of the bins according to its motion angle. The motion magnitudes of each bin are summed to obtain the Directogram $D_G(t, \theta)$

$$D_G(t,\theta) = \sum_j \Delta S_t(j) \mathbb{1}_\theta(\angle S_t(j)), \text{ where}$$

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

- **Converting the Directogram to Kinematic Offsets:** Kinematic Offsets represent motion changes according to deceleration. The deceleration is computed through the negative first order difference of the Directogram ΔD_G to obtain Motion Flux M, which represents the deceleration in various directions. Low-pass filters are applied to M to filter noise. To find the Kinematic Offsets, negative values in M are removed and the mean over each frame is computed which constitutes K. Top 1% peaks in K are then used and normalized to $[0, 1]$ range to obtain the smoothened Kinematic Offsets.
- **Obtaining the Visual Beats from Kinematic Offsets:** Kinematic Offsets, a continuous signal, is converted to a *binary sequence* that indicates whether there is a significant change in the human motion. We call the sequence *Visual Beats*. Using dynamical programming with the objective 'beat score function' the Visual Beats are computed by finding a set of local peaks of Kinematic Offsets having close or equal interval

$$V(\mathbf{m}) = \sum_{j=1}^{n} u(m_j) + \alpha \sum_{j=1}^{n-1} V_T(m_j, m_{j+1}), \tag{1}$$

$$V_T(m_j, m_{j+1}) = \frac{T[\text{bin}(m_{j+1} - m_j)]}{T_{max}} - 1.0, \tag{2}$$

where u is the Kinematic Offsets value of the candidate beat to encourage strong visual impacts. $\{m_j\}_{j=1}^{n} \in \mathbf{m}$ is a subset of candidate beats. $V_T(m_j, m_{j+1})$ penalizes the deviation from optimal tempos within a local window to encourage equal-spacing beats and it computes time-dependent autocorrelation function T on Kinematic Offsets to measure the deviation. α balances the weight between two terms and T is the autocorrelation average within local time window.

3.1.2 VisBeatNet: A real-time network for visual beats

We introduce VisBeatNet, a light-weight neural network designed to predict Kinematic Offsets and Visual Beats from a motion sequence, as visualized in Fig. 3. This architecture is built upon two uni-directional Gated Recurrent Units (GRUs), RN1 and RN2.

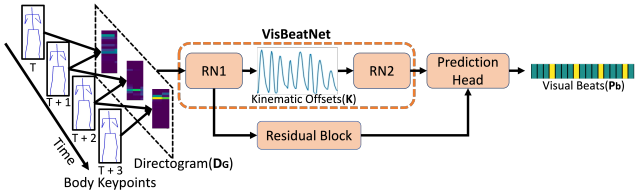

Figure 3. Detailed Schematics of VisBeatNet.

- **RN1 (Kinematic Offsets Prediction)**: RN1 predicts the Kinematic Offsets K auto-regressively from Directogram D_G as the given context.
- **RN2 (Visual Beats Prediction)**: RN2 takes both Kinematic Offsets K and residual connection from hidden states of prediction window in RN1 as inputs and outputs the Visual Beats P_b distribution via a linear prediction head.

Training. We train VisBeatNet using the pre-computed Kinematic Offsets and Visual Beats as theground truth. The training applies teacher-forcing to RN1, and employs two loss terms: (i) Mean Square Error (MSE) between the predicted Kinematic Offsets and the ground truth, and (ii) Weighted binary cross-entropy between the predicted beat distribution and the actual Visual Beats.

Post-processing using B-HMM. After training, the Visual Beats distribution P_b is processed by a pre-built beat Hidden Markov Model (B-HMM). Utilizing the Viterbi algorithm, the B-HMM yields the final output: a list of beat times T_b, specifying when each beat occurs.

While Visual Beats capture moments of strong visual impact in human motion, they inherently lack the drum rhythm, resulting in unnatural sounds when they are directly translated to audio, i.e the direct approach, termed **Mono**, directly overlays a monophonic drum sound atop of the visual beats, utilizing an auto-regressive drum notes generator. To address this limitation, we propose a refined approach which achieves more natural drum sounds, termed **Poly**, which periodically updates the tempo with VisBeatNet. Subsequently, a polyphonic drum language model (Section 3.3) conditioned on 'style,' a learning-based drum rhythm (Section 3.2), are applied.

3.2. MuStyleNet: Style Transfer for Drum Rhythm

The essence of drum audio lies in its rhythm - the pattern of drum onsets within each frame. In a realistic drum track every frame contains multiple onsets of different drum kits. Extracting the strongest onset in each frame gives rise to what we define as the drum 'style', a vector representing the rhythm of the drum audio.

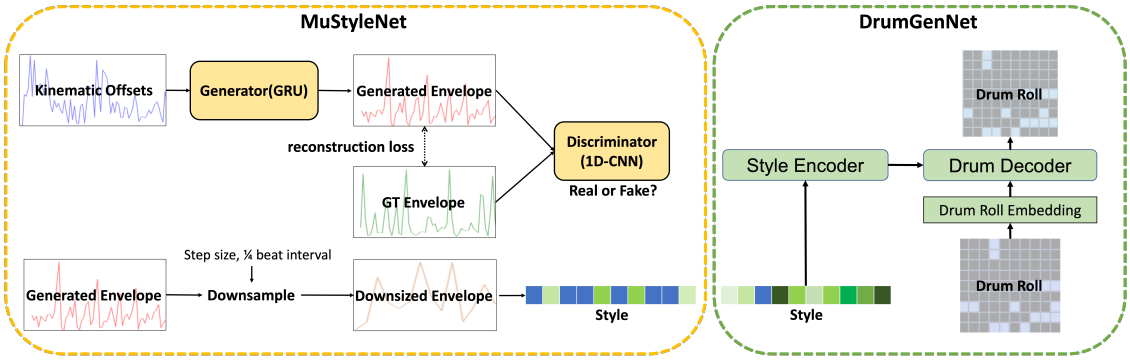

Figure 4. Detailed schematics of MuStyleNet (left) and DrumGenNet (right)

To obtain the'style', it is crucial to establish a relationship between 'style' and kinematic offsets, which represent the rhythm of body movements. There are multiple plausible ways to transform these offsets into the drum 'style'. A straightforward method was presented in 'RhythmicNet'. This method combined predicted music beats with motion peaks based on spectral analysis. Since drum rhythms contain regular patterns, while motion rhythms do not, it is unclear how to obtain associated 'styles'. Indeed, the crude approximation for the 'styles' can result in unnatural drum rhythms when used as the conditional input to the drum generation network.

Towards this end, we propose an adversarial style transfer module, *MuStyleNet*, which learns to translate the Kinematic Offsets into drum 'style' using a Generative Adversarial Network [24]. As illustrated in Fig. 4, the network takes kinematic offsets $K = \{K_1, K_2, ..., K_t\}$ as input and outputs the corresponding onset envelope $O = \{O_1, O_2, ..., O_t\}$. The generated envelope is given to the discriminator component to decide whether the envelope is real or fake. The GAN objective is defined by:

$$\min_G \max_D \mathbb{E}_{O\sim\mathcal{O}}[\log D(O)] + \mathbb{E}_{\hat{O}\sim\hat{\mathcal{P}}}[log(1 - D(G(K)))]. \quad (3)$$

Once the generated onset envelope is obtained, O is accumulated by step size st to obtain $O_{acc}(k) = \sum_{t=T_{s_k}}^{T_{s_{k+1}}} O_t$, where st is determined by the beat interval estimated in VisBeatNet and $T_{s_{k+1}} = T_{s_k} + st$. For consistency with the drum MIDI dataset that is used in the next stage, we fix st to be an interval of a quarter beat, and normalize the compressed onset envelope to obtain the drum 'style.'

3.3. Drum Generation

DrumGenNet This stage translates the 1D drums matrix, obtained in the previous stage, into polyphonic drum sounds. In contrast to the 2-stage Rhythm2Drum network in RhythmicNet [50], we keep the first stage network (with fewer number of layers and hidden size) that translates 1D drum rhythm into 2D drum hits matrix and due to real-time constraints discard the UNet that generates velocity and offset matrices. Furthermore, the 1D rhythm inout is continuous rather than binary, and thereby we use the continuous values as velocity for all drum hits at each time step. These changes make the drum generation network compact with the inference overhead compatible for real-time. We train DrumGenNet using a cross-entropy loss.

3.4. Real-Time Pipeline

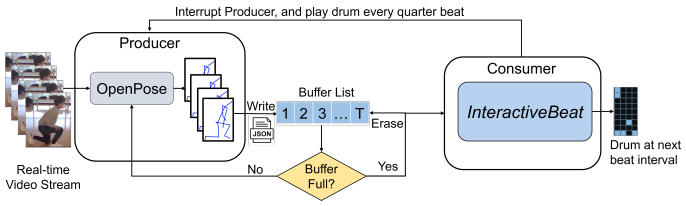

Figure 5. Real-Time Pipeline: A two-threaded pipeline with the Producer' and Consumer'. The Producer uses Openpose [6] to extract the skeleton sequence from live video and buffers it. When full, the sequence moves to the Consumer thread. The Consumer runs InteractiveBeat to produce drum MIDI for the next beat interval, dynamically updating. The Producer continues to add sequence to the buffer, pausing only for drum sound playback every quarter beat.

For real-time operation of the system, we integrate all components, i.e., motion estimation, beat inference, style transfer and drum generation, into a single and efficient pipeline. We design the pipeline to work in a producer-consumer mode, where the thread of the producer reads the body keypoints from motion estimation module, and sends them to a buffer list shared with the consumer thread. When the thread of the consumer obtains enough frames with keypoints, InteractiveBeat starts the inference by computing the Directogram and feeding it into the neural network components to generate the 'style'. The 'style' is provided to DrumGenNet to obtain associated drum MIDI. Meanwhile, InteractiveBeat dynamically updates the tempo every beat interval to keep up with the human motion. The Interac-

tiveBeat real-time pipeline is illustrated in Fig. 5. Further implementation details of the pipeline are illustrated and explained in Supplementary Materials.

4. Experiments

4.1. Datasets

We use multiple datasets to train and evaluate InteractiveBeat. We use the AIST Dance Video Database [53], as laboratory dataset. This dataset is a large-scale collection of dance videos recorded in a studio. The database includes 10 dance genres, with each genre 1080 videos. We split samples into train/validate/test sets by 80/10/10.

In addition, we collected a novel in-the-wild dataset, from YouTube and TikTok, which includes 764 videos of individuals performing diverse movements such as dance, sports, and aerobics. It includes soundtracks with different music genres of pop, hip-hop, jazz, EDM, and classical and in total spans 6 hours. Each frame of the video maintains a solo subject. The distance and orientation of the performer to the camera may vary. Performers from multiple ethnicities, cultures and skills are included. We apply Demucs [12], a source separation model to extract rhythmic sections of raw audio, and then remove silent sections. The dataset provides links to original video, percussion tracks, rhythmic video segments, and body keypoints computed with OpenPose [6]. The split of the dataset is 80/10/10 for training/validation/testing.

VisBeatNet is trained on pre-computed visual beats from AIST videos and 'in-the-wild' datasets. To train MuStyleNet, we manually sample 100k kinematic offset curves from AIST and from 'in-the-wild' videos, and then pair them with drum samples with onset envelope extracted in Groove MIDI Dataset [22] (a large-scale drum dataset with 9 canonical drum categories) by using a dynamic time warping method proposed by [11]. DrumGenNet is trained on the Groove MIDI Dataset.

4.2. Implementation Details

All models are implemented in PyTorch. In particular, skeleton key points are obtained using OpenPose [6] to extract 17 2D-keypoints of body joints. We set the number of bins to be 18 for the Directogram computation, such that each direction aggregates 20 degrees range of motion magnitude.

For VisBeatNet, RN1 and RN2 are GRUs of hidden sizes of 64 and 32, respectively. During training, a 3-second Directogram is fed as context, and the Kinematic Offsets and Visual Beats are predicted for the next second.

For MuStyleNet, we implement a GAN consisting of a 2-layer unidirectional GRU with the hidden size of 64 as generator, and a 3-layer 1D-CNN as the discriminator. The network is trained with MSE loss for reconstruction and Wasserstein loss [18] as an adversarial loss.

DrumGenNet implements a 2-layer transformer encoder-decoder with hidden size 64 and 1 attention head. The context length is 3 beat intervals and the prediction length is a single beat interval.

	AIST dataset		'in-the-wild' dataset	
	Visual Beats	Music Beats	Visual Beats	Music Beats
Votes	46.7%	53.3%	63.7%	36.3%

Table 1. Preference of Visual Beats v.s Music Beats.

4.3. Validation of Visual Beats as Ground Truth

To validate the effectiveness of using visual beats as ground truth, we conducted a user study comparing pre-computed visual beats against music beats obtained from the audio of the videos. We collected videos from AIST and 'in-the-wild' for evaluation (30 from each). The question, "Which beats do you think are better in sync with body moves?" is asked. As shown in Table 1, Visual Beats are preferred over Music beats. For 'in-the-wild', the preference is of large margin of **+27.4%** and for AIST the preference is of smaller margin of 6.6%. We suspect that the difference in the margin is due to data processing and annotation. AIST includes precise annotations and alignment of well-annotated music beats with dance movements while 'in-the-wild' dataset relies on music beats extracted with libraries, such as Demucs + Librosa, since there are no annotations. This process can introduce errors.

4.4. Visual Beats Prediction Comparison

Learning-based Visual Beats. To compare visual beats, we train stage 1 of the Video2Rhythm component in RhythmicNet [50] and compare it against VisBeatNet. We use pre-computed visual beats as ground truth,. The original implementation of Video2Rhythm uses a bi-directional graph-transformer with SSM modules, which is non-causal and cannot be applied in real-time. Direct adaption would be adding a causal transformer decoder to it that decodes the Kinematic Offsets with auto-regression and a linear layer for Visual Beats estimation. We use a 1-layer decoder with same hidden size and number of attention heads as RhythmicNet. Given a 3-sec input, we predict the next 1 second, and repeat the process for 3 times on a rolling basis to collect 3-sec Visual Beats predictions for evaluation 2.

Rule-based Visual Beats. We also implement a rule-based Visual Beats prediction baseline that satisfies the constraints of maximum within a pre-defined window size of 0.25s, minimum time wait after previous peak 0.25s, and above the average within the window than a threshold, which is set to 0.015. These parameters are the same as the ones used to extract candidate Visual Beats during Visual Beats pre-computation.

Beat Objective Evaluation. We follow the rubrics proposed for musical beat tracking [10] to evaluate Visual Beats prediction. We compute the performance in terms of Precision, Recall, F-score measure, and Cemgil's score (Cem). We also compute the Beat Alignment Score [37] which measures the correlation between motion and music. As shown in Table 2, VisBeatNet achieves on-par performance with RhythmicNet on visual beats prediction, while using only 2 GRU layers (*30k parameters*) with *5ms* inference time compared to the RhythmicNet (*800k parameters*) with *90ms* inference time. In Section 4.6, we show that such inference time introduces large system latency which hinders real-time application. Further, it is noteworthy that the rule-based approach achieves reasonable performance in recall score. However, its precision is low, which generates excessive beats even when no movements appear.

Beat Subjective Evaluation. We compare the predicted visual beats by VisBeatNet against the ground truth visual beats to evaluate the perceptual gap between the prediction and the ground truth. As shown in Table 3, the gap is of 11.4% or 6% for AIST and 'in-the-wild' datasets respectively, which demonstrates the relative effectiveness of VisBeatNet. The gap for 'in-the-wild' is smaller than the gap on AIST. The reason is that the pre-computed visual beats for 'in-the-wild' videos are more noisy than AIST videos due to the quality of the body keypoints inputs, but in terms of alignment, visual beats align better with movements than accompanying music beats, as we show in Table 1.

4.5. Drum Generation Comparison

As we discuss in Section 3.1, the drums can be generated by whether directly adding monophonic drum notes to the visual beats, or applying a polyphonic drum generative model conditioned on drum 'styles'. For *monophonic* drum, we train a 1-layer GRU model, with hidden size 64 on Groove MIDI dataset to auto-regressively generate monophonic drum notes. *Polyphonic* drums are generated using 'DrumGenNet.' We generate the drum sounds using 'styles' extracted from the videos rather than real drum rhythm extracted from drum tracks. This is important because the drum generator input comes from the rhythm in motion modality rather than drums, while the evaluation of the original 'Rhythm2Drum' stage of 'RhythmicNet' [50] ignores this crucial point. To be clear, we name the baseline methods as 'Mono' or 'Poly' method as follows.

Rule-based-Mono uses the rule-based visual beats described in Section 4.4 to generate 'style' patterns, and use 'Mono' to generate drum sounds.

RhythmicNet-Mono use the 'Video2Rhythm' stage of 'RhythmicNet' to generate 'style' patterns, and use 'Mono' to generate drum sounds.

RhythmicNet-Poly uses 'Video2Rhythm' for 'style', and 'DrumGenNet' to generate drums. The tempo is estimated by 'Video2Rhythm'.

InteractiveBeat uses 'MuStyleNet' to generate 'style' from Kinematic Offsets, and 'DrumGenNet' to generate drum. The tempo is estimated by VisBeatNet.

We use two *audio* objective metrics to compare the drum sound quality.

1) FID was introduced to evaluate image quality in GANs. It is adapted in audio-visual domain for audio spectrograms of generated soundtracks. It measures the distance between InceptionV3 pre-classification feature distributions for real and generated samples. By adapting InceptionV3 input for a 2D magnitude spectrogram and training it on GrooveMIDI for classifying 12 drum genres, we extract 2048-sized vectors from the last layer for both sets of samples. FID is then computed from these vectors.

2) NDB metric is used to evaluate the diversity of generated samples; the lower the NDB score, the better the diversity. Following RhythmicNet [50], We select k = 50 for k-means algorithm to cluster Voronoi cells in log-spectrogram space.

For each baseline, we generate 5000 samples separately from the test set of AIST and 'in-the-wild' to perform the objective evaluations. As shown in Table 4, InteractiveBeat achieves better drum quality than 'RhythmicNet'. A polyphonic drum generative model with a clear bar-level structure is necessary to produce quality drum sounds.

We also perfrom a perceptual experiment where we select 30 videos from each of the datasets AIST and 'in-the-wild' and ask the raters to compare the drum sounds for different methods. In particular, we ask: "Which drum track sounds most natural, coherent, and rhythmic?" As shown in Table 4, InteractiveBeat consistently outperforms other baselines by a large margin, **+18.6%** and **+13.7%**, for AIST and 'in-the-wild' respectively.

4.6. Real-Time System Evaluation

To evaluate the real-time system, we use latency as the main metric and evaluate the baselines (Rule-based-Mono, RhytmicNet-Mono,RhytmicNet-Poly) along with InteractiveBeat. The system latency can be analyzed based on the following aspects:

Camera Frame Rate (CFR): Our off-the-shelf web camera for experiments operates at $CFR = 30fps$.

OpenPose Inference Speed (OIS): On a TitanX GPU, OpenPose achieves $OIS = 70fps$ on AIST and 'in-the-wild' videos.

Network Inference Speed (NIS): This is the inference speed of all networks combined.

Causal v.s Non-causal (C/NC): generate the drum for the present with delay (NC) v.s generate the drum for the next interval, compensating for delay (C).

For *CFR* and *OIS*, *CFR*= 30fps is the minimum possible latency (L_{cam} = 33ms) of the system , while OpenPose

	Num Params	Inference time	AIST Dataset					'in-the-wild' Dataset				
			Pr ↑	Rec ↑	Cem ↑	F↑	B-Aln ↑	Pr ↑	Rec ↑	Cem ↑	F↑	B-Aln ↑
Rule-based	-	**1ms**	38.62%	57.31%	33.53%	45.10%	0.286	39.25%	56.96%	33.42%	45.7%	0.2813
RhythmicNet [50]	800k	90ms	**69.78%**	**49.60%**	45.93%	**59.45%**	**0.4147**	**67.25%**	51.29%	44.52%	**55.70%**	0.4098
VisBeatNet	**30k**	5ms	69.01%	**50.71%**	46.02%	58.19%	0.4032	66.17%	**51.91%**	44.31%	55.43%	**0.4175**

Table 2. Visual Beat prediction evaluation on AIST dance dataset(lab environments) and 'in-the-wild' dataset. The abbreviation of each component stands for: Pr(Precision), Rec(Recall), F (F-score measure), Cem (Cemgil's score), B-Alg(Beat Alignment Score).

	AIST dataset		'In-the-wild' dataset	
	VisBeatNet	GT Visual beats	VisBeatNet	GT Visual beats
Votes	44.3%	**55.7%**	47.0%	**53.0%**

Table 3. Visual beats prediction v.s GT perceptual preference.

	AIST dataset			'In-the-wild' dataset		
	FID↓	NDB↓	Votes↑	FID↓	NDB↓	Votes↑
Rule-based-Mono	68	49	3.7%	72	49	3.0%
RhythmicNet-Mono [50]	66	49	10.3%	70	49	12.7%
RhythmicNet-Poly [50]	47	46	33.7%	49	47	35.3%
InteractiveBeat	**37**	**41**	**52.3%**	**43**	**41**	**49.0%**

Table 4. Audio quality metrics(NDB, FID) and soundtrack preference between InteractiveBeat and other baselines.

Methods	Causal or Non-Causal(C/NC)	Total Latency(ms)↓	Votes↑
Rule-based-Mono	NC	158	3.7%
RhythmicNet-Mono [50]	NC	133	11.3%
RhythmicNet-Poly [50]	C	103	35.3%
InteractiveBeat	C	**34**	**49.7%**

Table 5. Real-Time evaluation on InteractiveBeat v.s other baseline methods.

achieves faster speed (70fps). For *NIS* and *C/NC*, we describe the integration of different network choices into the real-time pipeline, and analyze the total latency. A summary of total latency is shown in Table 5 and we summarize the latency metrics below.

- **Rule-based-Mono**: A non-causal method that directly adds monophonic drums to the 'style'.

  Inference time: $L_{ni} = 125ms$ (visual beats are determined after 125ms, half of the window size (0.25s).
  Total latency: $L_{total} = L_{cam} + L_{ni} = 158ms$.
- **RhythmicNet-Mono**: A non-causal method with the Video2Rhythm stage of 'RhythmicNet' [50]. It uses a 3-sec input window for real-time adaptation and checks for visual beats in the latest 60ms.

  Inference time: $L_{ni} = 70ms$.
  'Style' check delay: $T_r = 30ms$.
  Total latency: $L_{total} = L_{cam} + L_{ni} + T_r = 133ms$.
- **RhythmicNet-Poly**: A causal method which compensates for camera frame rate delay by forecasting the style' for the next interval. Apply a 3-sec Video2Rhythm inference window followed by a 1-layer GRU forecaster.

  Inference time: Video2Rhythm & forecaster: $81ms$, 'DrumGenNet': $22ms$.
  Total latency: $L_{total} = L_{ni} = 103ms$.
- **InteractiveBeat**: Our method for causal forecasting visual beats in the next interval.

  Inference time: Directogram calculation ($1ms$), VisBeatNet ($5ms$), MuStyleNet ($6ms$) and DrumGenNet ($22ms$).
  Total latency: $L_{total} = L_{ni} = 34ms$.

As shown in Table 5, InteractiveBeat achieves significantly lower latency than other baselines due to compact networks design that reduces inference speed and a causal scheme which compensates for the delay. Notably, 'RhythmicNet-Poly' is an adaption from 'RhythmicNet' with addition of real-time constraint. Our results show that merely adding such constraint without change of network design would still correspond to larger latency.

To further evaluate the operation in real-time in terms of its perceptual experience, we conducted a human study where we generated drum sounds for a total of 30 videos from AIST and 'in-the-wild' set in real-time for each method. A question "Which drum soundtrack do you prefer, considering the alignment with body movements and latency?" was presented to raters. As shown in Table 5, InteractiveBeat receives most votes, higher by **+14.4%** than the second top pipeline of 'RhythmicNet-Poly'.

5. Discussion and Conclusion

In this work, we propose a real-time system, InteractiveBeat, for interactive generation of sounds that accompany person's body movements being captured by a camera. InteractiveBeat introduces a series of compact models to generate drum sounds in real-time with low latency. Quantitative experiments and human evaluation studies show that the system can achieve a correspondence between movements and the emitted sounds. The current system focuses exclusively on generating drum sounds to achieve optimal synchrony between body movements and sounds. A plausible future extension of the system could be extension to generate 'organic' music that will result with melodic soundtracks and further promote the perecpetual experience of interaction between music and movement. Additionally, system's current latency of a 34ms could be further optimized. For advanced VR applications, 10-20ms latency is desired [38], and for musical performances, latency of 10ms is required for natural perceptual experience [32].

References

[1] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. Musiclm: Generating music from text, 2023. 3

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017. 2

[3] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, pages 892–900, 2016. 2

[4] Tamara Berg, Debaleena Chattopadhyay, Margaret Schedel, and Timothy Vallier. Interactive music: Human motion initiated music generation using skeletal tracking by kinect. In *Proc. Conf. Soc. Electro-Acoustic Music United States*, 2012. 2

[5] Frédéric Bevilacqua, Lisa Naugle, and Isabel Valverde. Virtual dance and music environment using motion capture. In *Proc. of the IEEE-Multimedia Technology And Applications Conference, Irvine CA*, 2001. 2

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 3, 5, 6

[7] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European Conference on Computer Vision*, pages 17–36. Springer, 2020. 2

[8] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 29:8292–8302, 2020. 2

[9] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, 2023. 3

[10] Matthew EP Davies, Norberto Degara, and Mark D Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009. 7

[11] Abe Davis and Maneesh Agrawala. Visual rhythm and beat. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2532–2535, 2018. 3, 6

[12] Alexandre Défossez. Hybrid spectrogram and waveform source separation. In *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021. 6

[13] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. 3

[14] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2037–2045, 2021. 2

[15] Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3

[16] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*, 2017. 3

[17] Sidney Fels and Kenji Mase. Iamascope: A graphical musical instrument. *Computers & Graphics*, 23(2):277–286, 1999. 2

[18] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a wasserstein loss. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2053–2061, Cambridge, MA, USA, 2015. MIT Press. 6

[19] Chuang Gan, Deng Huang, Peihao Chen, Joshua B. Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *ECCV*, 2020. 2

[20] Chuang Gan, Deng Huang, Hang Zhao, Joshua B Tenenbaum, and Antonio Torralba. Music gesture for visual sound separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10478–10487, 2020. 2

[21] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–53, 2018. 2

[22] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning (ICML)*, 2019. 3, 6

[23] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2019. 2

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5

[25] Jessica A. Grahn and Matthew Brett. Rhythm and Beat Perception in Motor Areas of the Brain. *Journal of Cognitive Neuroscience*, 19(5):893–906, 05 2007. 1

[26] David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2016. 2

[27] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized pi-

ano music modeling and generation with the maestro dataset. *arXiv preprint arXiv:1810.12247*, 2018. 3

[28] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2018. 3

[29] Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, Jesse Engel, Quoc V. Le, William Chan, Zhifeng Chen, and Wei Han. Noise2music: Text-conditioned music generation with diffusion models, 2023. 3

[30] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020. 3

[31] Richard B Ivry and Rebecca MC Spencer. The neural representation of time. *Current Opinion in Neurobiology*, 14(2):225–232, 2004. 1

[32] Robert H. Jack, Adib Mehrabi, Tony Stockman, and Andrew Mcpherson. Action-sound latency and the perceived quality of digital musical instruments. *Music Perception*, 2018. 8

[33] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. You said that?: Synthesising talking faces from audio. *International Journal of Computer Vision*, 127(11-12):1767–1779, 2019. 2

[34] A Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1838–1842. IEEE, 2020. 2

[35] Stefan Lattner and Maarten Grachten. High-level control of drum track generation using learned patterns of rhythmic interaction. In *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 35–39. IEEE, 2019. 3

[36] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems*, pages 3586–3596, 2019. 2

[37] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++, 2021. 7

[38] Simone Mangiante, Guenter Klas, Amit Navon, Zhuang GuanHua, Ju Ran, and Marco Dias Silva. Vr is on the edge: How to deliver 360° videos in mobile networks. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, VR/AR Network '17, page 30–35, New York, NY, USA, 2017. Association for Computing Machinery. 8

[39] Tae-Hyun Oh, Tali Dekel, Changil Kim, Inbar Mosseri, William T Freeman, Michael Rubinstein, and Wojciech Matusik. Speech2face: Learning the face behind a voice. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7539–7548, 2019. 2

[40] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, 2020. 3

[41] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016. 2

[42] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *European conference on computer vision*, pages 801–816. Springer, 2016. 2

[43] Adam Roberts, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. A hierarchical latent vector model for learning long-term structure in music. In *International Conference on Machine Learning*, pages 4364–4373. PMLR, 2018. 3

[44] Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion, 2023. 3

[45] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7574–7583, 2018. 2

[46] Nikhil Singh, Jeff Mentch, Jerry Ng, Matthew Beveridge, and Iddo Drori. Image2reverb: Cross-modal reverb impulse response synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 286–295, 2021. 2

[47] Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, and Timo I. Denk. V2meow: Meowing to the visual beat via music generation, 2023. 2

[48] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio generation for a silent performance video. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[49] Kun Su, Xiulong Liu, and Eli Shlizerman. Multi-instrumentalist net: Unsupervised generation of music from body movements. *arXiv preprint arXiv:2012.03478*, 2020. 2

[50] Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? *Advances in Neural Information Processing Systems*, 34, 2021. 2, 3, 5, 6, 7, 8

[51] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2

[52] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018. 2

[53] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *ISMIR*, pages 501–510, 2019. 2, 6

[54] Lucia Valmaggia. The use of virtual reality in psychosis research and treatment. *World Psychiatry*, 16(3):246, 2017. 1

[55] Yujia Wang, Wei Liang, Wanwan Li, Dingzeyu Li, and Lap-Fai Yu. Scene-aware background music synthesis. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1162–1170, 2020. 3

[56] Todd Winkler. Creating interactive dance with the very nervous system. In *Proceedings of Connecticut College Symposium on Arts and Technology*, 1997. 2

[57] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1735–1744, 2019. 2

[58] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 570–586, 2018. 2

[59] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 2

[60] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to sound: Generating natural sound for videos in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3558, 2018. 2

[61] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized gan for complex music generation from dance videos. In *The European Conference on Computer Vision (ECCV)*, 2022. 2

[62] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. In *International Conference on Learning Representations (ICLR)*, 2023. 2