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

1. Additional Implementation Details

1.1. Visual Beats Rules

To *train* VisBeatNet and determine the tempo of the drums, we obtain the visual beats with an offline method. Candidate visual beats are inferred by the following rule: 1) The kinematic offset of a visual beat must be the local peak within a window of 0.25s. 2) The minimum time wait after previous peak 0.25s. 3) The visual beat must have a value of kinematic offset above the local mean of $0.25s$ window by a threshold value 0.015. To perform optimization using dynamic programming on visual beats, the time-dependent auto-correlation is calculated with a window size of 4 seconds, hop size of 1 video frame, and normalized by its column maximum. The parameter α which balances the terms in the objective function is set to 0.02.

1.2. Full Pipeline

For real-time operation of the system, we integrate all components such as motion estimation, visual beat prediction, style transfer, and drum Midi generation into a single pipeline. We design the pipeline to work in a producer-consumer mode, where the producer thread reads the extracted body skeleton key points from motion estimation module and sends the skeleton key points to a buffer list that is shared with the consumer thread.

The flow of the pipeline is as follows: Motion Estimation Module infers body key-points from the camera stream, and saves results of each frame into a json file in a folder shared with the producer thread. When a new file is saved to the folder, the Producer Thread reads the file and puts the skeleton key points data into a buffer list that is shared with the Consumer. The Producer Thread and the Consumer Thread are sharing a condition variable to coordinate with each other. When the Consumer Thread obtains enough skeleton key points frames determined by the inference window T_i, it starts the drum generation by:

- Computing the directogram D_G and feeding it into the VisBeatNet to predict the kinematic offsets K and visual beats distribution P_b for the next beat interval, and update the tempo.
- Translating the kinematic offsets into a 'style': 1D drum matrix S_f aggregating the drum onset envelope by a step size of $ssize$.
- Generating the drum Midi Midi_next based on style S_f. The full pipeline is formulated in Algorithm [1].

2. Human Evaluation Details

For all human evaluations, we conducted surveys using Amazon Mechanical Turk (AMT), which ensures the efficient management of tasks and timely responses from a large pool of participants. On AMT, each individual task is referred to as a Human Intelligence Task (HIT). A HIT represents a single unit of work that a participant, or 'worker', can accept, complete, and submit for compensation.

2.1. Survey Construction

We began by constructing paired videos samples using the methods described in Tables 1, 2, 4, and 5 from main text, for both the AIST and 'in-the-wild' datasets. For the experiments detailed in Sections 4.3 and 4.4 (corresponding to Tables 1 and 2), we selected 30 videos from each dataset (samples are shared for these 2 experiments). Similarly, for the drum generation experiment (Section 4.5 with Table 4 in main text), 30 videos were chosen from each dataset. For the real-time evaluation (Section 4.6 in main text), a total of 30 videos are selected for AIST and 'in-the-wild' set. To mitigate potential biases, we randomized the order of videos generated by different methods across all experiments.

2.2. Survey Deployment

We created 4 distinct surveys on AMT, corresponding to the 4 experiments. For all surveys, participants were required to be Masters on the platform, have an approval rate $> 95\%$, and possess a background in music or dance with a minimum of 1 year's experience. Additionally, we restricted participation to those based in the United States to ensure language proficiency in English, without collecting further personal details. Responses outside the 50-second to 10-minute window were excluded.

To ensure that the participants are clear about the goal of the tasks, the following instructions were provided:

Algorithm 1: InteractiveBeat Pipeline

Require: Skeleton Save Path $path$, Inference Window T_i, Limit Time T_l, Maximum Wait Time T_w;
Initialize empty lists and a timer and a thread Condition Variable: skeleton buffer l_b, skeleton array l_a, kinematic list l_k, $curTime$ = getTime(), $cond$ = Threading.Condition(), end_signal = EmptyObject() ;

OpenPose Subprocess:
while *webcam.isOpen()* **do**
  $skeleton$ = OpenPose($webcam$);
  save(skeleton, $path$);
end

Producer Thread:
while *True* **do**
  if *checkNewFile($path$)* **then**
    if *isFirstFrame* **then**
      $global\_timer$ = getTime();
    end
    file = get_newest_file($path$);
    new_skeleton=read_json(file); append(l_b, new_skeleton);
    start_wait_timer();
    cond.Notify();
  end
  if $wait\_time() > T_w$ ***or*** $curTime > T_l$ **then**
    append(skeleton_buffer, end_signal);
    break;
  end
end

Consumer Thread:
while *True* **do**
  while *($len(l_b) < T_i$)* **do**
    cond.Wait();
  end
  if $l_b[-1] == end\_signal$ **then** break;
  $l_a, l_b = l_b, []$;
  D_G = Compute Directogram(l_a);
  K, P_b = VisBeatNet(D);
  b_l = B-HMM(P_b); b_{int} = beat_interval(b_l);
  $ssize = b_{int}/4$; S_f = MuStyleNet(K, ssize);
  append(l_k, K);
  Midi_next = DrumGenNet(S_f);
  playMidi(Midi_next, T_o);
end

- For each pair of comparisons, please watch and/or listen to the samples from beginning to end.
- For videos, please pay attention to whether the sound reacts to body moves adequately or with delay, whether there are strong impact sounds when there is no motion. If such scenarios frequently appear, then the alignment between motion and sound is not adequate.
- For generated drum tracks, please focus on whether the drum sounds are coherent, natural and show rhythmic structure rather than plain beats.
- Once a video sample has been evaluated, it should not be reassessed again from the same "HIT group". This ensures that each vote for the same sample comes from unique participants.

Our goal was to obtain 10 unique votes for each video sample. To achieve this, we bundled 10 HIT tasks for each video sample into a HIT group. After completing a HIT from a group, participants were assigned a qualification, preventing them from accessing HITs from the same group again. This setup ensured unique votes for each sample without explicitly controlling the number of distinct participants.

2.3. More Survey Results

To measure intra-sample agreement across participants, we use the "percentage agreement" metric, defined as the ratio of votes for the most popular choice to the total votes for that sample.

The detailed results for each survey are as follows:

- **Preference of Visual Beats vs. GT Music Beats Experiment:** For AIST dataset, we received 300 valid votes (1.25 minutes rating duration on average) coming from 42 distinct participants, with 7.14 ± 2.7 ratings per participant, and a percentage agreement of $78.2\% \pm 5.6\%$ per video sample. For 'in-the-wild' dataset, we received 300 valid votes (1.67 minutes rating duration on average) from 45 distinct participants, with 6.67 ± 2.2 ratings per participant, and a percentage agreement of $72.6\% \pm 7.8\%$ per video sample.
- **Visual Beat Comparison Experiment:** For AIST dataset, we received 300 valid votes (1.4 minutes rating duration on average) coming from 38 distinct participants, with 7.89 ± 2.9 ratings per participant, and a percentage agreement of $74.0\% \pm 4.1\%$ per video sample. For 'in-the-wild' dataset, we received 300 valid votes (1.7 minutes rating duration on average) from 36 distinct participants, with 8.33 ± 3.1 ratings per participant, and a percentage agreement of $71.2\% \pm 8.6\%$ per video sample.
- **Drum Generation Experiment:** For AIST dataset, we received 300 valid votes (2.2 minutes rating duration on average) coming from 40 distinct participants, with 7.5 ± 1.7 ratings per participant, and a percentage agreement of $81.3\% \pm 5.1\%$ per video sample. For

'in-the-wild' dataset, we received 300 valid votes (2.4 minutes rating duration on average) from 43 distinct participants, with 6.98 ± 1.5 ratings per participant, and a percentage agreement of $77.6\% \pm 6.9\%$ per video sample.

- **Real-time Experiment:** We received 300 valid votes (2.7 minutes rating duration on average) coming from 44 distinct participants, with 6.81 ± 1.6 ratings per participant, and a percentage agreement of $73.8\% \pm 5.5\%$ per video sample.

3. Utility of InteractiveBeat System

To test the usefulness and entertainment aspect of InteractiveBeat, we invited 5 guests to try our system with live demo, and rate their response with respect to 'utility level' as well as ask them to justify their ratings with comments. The utility level ranges from 1 to 5 (1 - 'not fun and bad experience', 2 - 'indifferent', 3 - 'a bit fun but not good enough', 4 -'fun with reasonable quality', 5 - 'super fun with amazing quality'. In the experiment, we asked the guests to think of their favorite moves ahead of time and let them move freely in front of the camera. As they started moving, the drum sounds began playing in response to their movements interactively. Each guest moved according to their rhythm and could change their motion speed, stop abruptly and then start to move again. Each guest performed movements for 1 minute to experience the system. We asked them to comment on how their experience was with the system and listed the ratings along with the comments below.

Guest 1: Rating: 5. Comments: It's been fascinating how the system converted my body movements into drum sounds. I find it a very unique way to explore rhythm and express myself through movement, and this could be a fun and creative exercise for people who love dancing.

Guest 2: Rating: 4. Comments: Grooving along and experimenting with different moves to see how they affected the drum sounds is a cool experience. Anyone can stand in front of the camera and explore different ways to interact with the computer algorithm and get interesting sound feedback from it, which definitely adds an extra layer of enjoyment to the moves.

Guest 3: Rating: 4. Comments: I am inspired by this novel concept of "making sounds as you move". I tried various styles of movement to interact and produce different drum sounds. Such an installation is a really creative tool where I can express myself through my moves.

Guest 4: Rating: 4. Comments: The live demo is a fresh experience for me and it's the first time that I hear the rhythm of my motion. While it may not be perfect in its responsiveness, it still generates an enjoyable rhythm that complements my moves. I believe in its potential application in gyms where you can listen to the funny sounds generated by the system while doing workouts.

Guest 5: Rating: 4. Comments: The drum sound system is an interesting way to engage with sound through moving my body. Overall, it succeeded in capturing the important parts of my movements and most of the sounds agree with my expectation.

4. Limitations

As discussed in Discussion and Conclusion section, InteractiveBeat is uncapable of generating soundtrack that consists of additional instruments than drum, in particular those with melodies, like guitar or piano. A key challenge is that a typical musical note lasts for a long duration than a short impulse like drum sound. The forecasting scheme defined in InteractiveBeat only predicts a short window of drum hits to reflect the instantaneous motion rhythm rather than notes with smooth transition. While it seems that the prediction window could be made longer to predict melodies for the next bar, we found empirically that such generated melodies regardless of guitar or piano did not show plausible alignment with the movements. This suggests that another approach could be required for interactive generation of melodies that align with the rhythm of movements.

Moreover, soundtracks of instruments, such as piano or guitar, involve long-term dependencies and more delicate music rules like chord progression, music scales, the theme of music etc. Such complexities must be traded off when real-time constraints are considered since the priority of the system is responding to the motion rhythm rather than comply with such music rules. To avoid possible conflicts between the two, the motion rhythm of the performer needs to be constrained such that the style of their moves would show a regular pattern and is conformed with the target music style or genre. A plausible approach could be to ask users to select the styles of the movements they intend to perform, and the system will be prepared to fit users preference and generate the corresponding music soundtracks that are both align with the users intended movements and adhere to the music rules of selected styles or genres. Incorporation of additional constraints and extension of InteractiveBeat to support them could be a viable future direction toward enriching and further adapting the generated output.