Triggering Sounds From Discrete Air Gestures: What Movement Feature Has the Best Timing?

Luke Dahl
Center for Computer Research in Music and Acoustics
Department of Music, Stanford University
Stanford, CA
lukedahl@ccrma.stanford.edu

ABSTRACT

Motion sensing technologies enable musical interfaces where a performer moves their body "in the air" without manipulating or contacting a physical object. These interfaces work well when the movement and sound are smooth and continuous, but it has proven difficult to design a system which triggers discrete sounds with precision that allows for complex rhythmic performance.

We conducted a study where participants perform "air-drumming" gestures in time to rhythmic sounds. These movements are recorded, and the timing of various movement features with respect to the onset of audio events is analyzed. A novel algorithm for detecting sudden changes in direction is used to find the end of the strike gesture. We find that these occur on average after the audio onset and that this timing varies with the tempo of the movement. Sharp peaks in magnitude acceleration occur before the audio onset and do not vary with tempo. These results suggest that detecting peaks in acceleration will lead to more naturally responsive air gesture instruments.

Keywords

musical gesture, air-gestures, air-drumming, virtual drums, motion capture

1. INTRODUCTION

1.1 Air-controlled Instruments

We typically think of a musical instrument as an artifact under manipulation by a human for the purposes of making musical sound. In most acoustic instruments the energy for producing sound is provided by human movement in direct contact with the instrument: striking a drum, bowing or plucking a string, blowing air through a flute. In instruments where the acoustic energy is not provided by the player, such as a pipe organ or the majority of electronic and digital instruments, control of the instrument relies on manipulation of a key, slider, rotary knob, etc.

With the advent of electronic sensing it became possible to control an instrument with gestures "in the air." Early examples include the Theremin, which is controlled by empty-hand movements in space, and the Radio Baton [8] and Buchla Lightening, which sense the movement of hand-held batons.

NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.

The recent proliferation of affordable motion sensing technologies (e.g. the Microsoft Kinect) has led to a surge in "air-controlled" new musical interfaces where a performer moves their body without manipulating or contacting a physical object. These interfaces seem to work well when the movement and control of sound are smooth and continuous. However, in our experience and observations it has proven difficult to heuristically design a system which will trigger discrete sounds with a precision that would allow for a complex rhythmic performance. In such systems the relationship between a gesture and the timing of the resulting sound often feels wrong to the performer.

1.2 Air-Gestures

We define *air-gestures* as purposeful movements a performer makes with their body in free space in order to control a sound-generating instrument that is designed to have an immediate response. *Discrete air-gestures* are meant to trigger a sonic event at a precise time, and are contrasted with *continuous air gestures* in which some movement quality (e.g. the height of the hand) is continuously mapped to some sonic variable (e.g. a filter cutoff frequency.)

In popular usage *air-drumming* refers to miming the gestures of a percussionist in time to another musician's (usually prerecorded) performance. For the sake of this research we expand this to include gestures in which a performer mimics the striking of an imaginary surface in order to trigger a sound with a sudden attack. Air-drumming is not the only type of discrete air-gesture. For example, jerky movements, such as in the dance style known as "popping and locking" might also be used to trigger sounds.

1.3 Motivation and overview

The goal of this research is to improve the design of air-controlled instruments so that discrete air-gestures can be reliably triggered with timing that feels natural to the performer. To this end, we conducted a study of air-drumming, in which participants gesture in time to simple rhythmic sounds.

We record participants' movements in a motion capture system, and analyze this data to address the following question: What aspect of the air-drummer's movement corresponds to the sound? In other words, we assume that when a person makes a discrete air-gesture, they do something with their body to create an internal sense of a discrete event, and that they intend this event to correspond to the sonic event of a drum sound. We want to know what this "something" is, and to characterize its timing with respect to the sonic event.

We examine two candidate movement events: a *hit* is the moment where the hand suddenly changes direction at the end of a strike gesture, and *acceleration peaks*, which occur before the hit as the hand decelerates. We analyze

the timing of these events with respect to the onset time of their corresponding audio events.

Our air-drummers mime in time to a musical performance, but we assume that the correspondence between gesture and sound would be the same if they were triggering the sounds themselves. Thus we expect the results of our analysis to reliably describe sound-generating discrete air-gestures and be useful in improving the timing of air-instruments. In order to test this assumption we compared movements in time to prerecorded sounds with movements in time to percussive sounds made vocally by the participants themselves.

1.4 Related Work

1.4.1 Discrete air-gesture performance systems

The Radio Baton [8] senses the spatial location of two handheld wands. To trigger discrete sound onsets it uses a spatial height threshold which corresponds to the height at which the baton contacts the surface of the sensor, giving the user tactile feedback. So we say that while the Radio Baton senses continuous air-gestures, the discrete events which it enables are not air-gestures.

A number of systems which trigger percussion sounds from air-gestures have been implemented as part of real-time performance systems. Havel and Cathrine [4] track sensors on the ends of drum sticks and use a velocity threshold to trigger sounds. They also note that peak velocity amplitude is correlated to the time between strikes.

Kanke et al. [6] use data from acceleration and gyro sensors in drum sticks to differentiate between striking a real percussion instrument and an air-instrument. Strikes are registered when the acceleration exceeds a threshold.

1.4.2 Studies of discrete air-gestures

A few studies of discrete air-gestures have been conducted. Patola et al. [7] studied participants striking a virtual drum surface in time to a metronome click, and compared the use of a physical stick held in the hand with a virtual stick. Drum sounds were triggered when the tip of the stick first intersected a virtual horizontal drum surface at a specific location. Amongst their findings was that drum hits lagged behind metronome clicks by 20 ms, which they attribute to the "perceptual attack time" of the clap sound they were using.

Collicutt et al. [1] compared drumming on a real drum, on an electronic drum pad, with the Radio Baton, and with the Buchla Lightening II. In all cases they track the height of the hand (even though their participants held sticks), and use vertical minima to determine when strikes occurred. However, they note that this did not work for a subject whose strikes corresponded to smaller minima before the actual minimum. (We also find in our study that strikes do not always correspond to minima.) They found that using the Lightening had the second best timing variability, and attribute this to the different way in which users control their movements when there is no tactile event.

1.4.3 Studies of real drum gestures

S. Dahl [3] made motion capture recordings of drummers playing a simple rhythm on a real snare drum, and found that subjects raised the stick higher in preparation for accented strikes and that preparatory height correlated with higher peak velocity. For their analysis they detect hits as points which satisfy two criteria: the local minima of stick tip height must pass below a threshold, and the difference between two subsequent changes in vertical velocity (roughly equivalent to the 3rd derivative of position, also known as "jerk") must surpass a threshold.

Figure 1: The stimulus rhythm. Slow notes are labeled 'S' and fast notes 'F'

1.4.4 Sensorimotor synchronization

Air-drumming in time to music is a form of synchronizing movements to sound. Research into sensorimotor synchronization goes back decades [9], and one of the primary findings is that when tapping in time to an audible beat (usually a metronome click), most people tap before the beat. This "negative mean asynchrony" is often a few tens of milliseconds, but may be as great as 100 ms. This is relevant to our work because we assume that, much like the subjects in tapping experiments, our air-drummers are synchronizing some physically embodied sensation to the beat.

As far as we know, the research we described in this paper is the first detailed empirical analysis of drumming gestures in time to percussive sounds.

2. STUDYING AIR-DRUMMING GESTURES

2.1 Experiment

The goal of our study is to understand what people do when they perform air-drumming gestures in time to rhythmic sounds, and how their movements correspond to the sounds. The ultimate aim is to use these results to design better discrete air-gesture-controlled instruments.

2.1.1 The tasks

We recorded the movements of people making air-drumming gestures in time with a simple rhythm described below. Participants performed two tasks. Task 1 is to gesture in time with a recording of the rhythm. They were asked to gesture *as if striking a drum* somewhere in front of them with a closed empty right hand, and to act *as if they are performing* the sounds they hear. Since we are interested in gestures someone might make while performing an air-instrument in free space, we did not provide further specification as to the location or style of the strike.

Task 2 is to vocalize the rhythm while gesturing as if they are performing on a drum somewhere in front of them. They create the rhythm themselves by saying the syllable "da" or "ta" for each drum hit. These tasks are very different: the first is to synchronize one's movement to an external sound, and the second is to simultaneously make sounds and gestures which coincide. Neither of these are the task we are interested in, i.e. playing sounds with discrete air-gestures. By comparing the performance of these two tasks we hope to understand whether one is a better proxy (section 2.3.1.)

The stimulus rhythm is shown in figure 1. We are interested in whether people's gestures are different when performed at different speeds, and so the rhythm is designed to have an equal number of 'slow' notes (quarter notes with rests in between), and 'fast' notes (quarter notes with no rests.) We compare these two cases in section 2.3.2.

For task 1 the rhythm was played by the sound of a synthesized tom drum at a tempo of 100 beats per minute (where a beat is one quarter note.) For task 2 participants heard a 4-beat metronome count at 100 bpm, after which they performed and vocalized the rhythm without audio accompaniment.

For each trial participants perform the rhythm four times

Figure 2: A participant with markers and motion capture cameras

in succession without stopping. Two trials are recorded for each task, resulting in a total of 8 repetitions of the rhythm for each task.

A third task was recorded in which participants performed a similar rhythm which has notes of two dynamic levels (accented and unaccented.) The analysis and comparison of these cases will be described in a future publication.

2.1.2 Equipment

Participants were outfitted with fourteen reflective markers on their right arm and upper torso (figure 2), and their movements were recorded at 200 frames per second by a Motion Analysis motion capture system with twelve cameras mounted around the participant.

Participants could read the rhythm on a music stand 1 meter to their front right. For task 1 the rhythm was played over a Behringer MS40 studio monitor placed 1 meter to the front left. For task 2, a metronome count-off was played over the studio monitor, and participants' vocalizations were recorded by an AKG C414 microphone placed 1 meter in front of them. All sounds were recorded into the motion capture system at 20kHz via an analog input. Stimulus and metronome sounds were played from Ableton Live and initiated at the beginning of each trial by the experimenter.

2.1.3 Participants

We recruited ten participants with the requirement that they have some experience playing a musical instrument and that they be able to read music. The participants were five females and five males, ranging in age from 22 to 57 years, with a median age of 23.5 years. All were right-handed. They reported between 13 and 48 years of musical experience, with a median of 16 years. Four participants had formal dance training, and these reported receiving 3 to 7 years of training. Before recording data it was verified that each participant could read the simple rhythms and perform the desired tasks.

The procedure was approved by the internal review board for human subjects research at Stanford University.

2.2 Analysis

2.2.1 Detecting audio onsets

The first stage of analysis is to determine the onset times of the audio events (the drum or vocal sounds) for each trial. These onset times will act as a reference against which we compare the timing of the movement features.

To detect audio onsets we pass the squared audio signal in parallel through two DC-normalized one-pole low-pass filters. These are used to estimate two energy envelopes of the audio where one is "fast", with a time constant of 0.5 ms, and the other is "slow", with a time constant of 10 ms. When the ratio of the fast estimate over the slow estimate exceeds a threshold, we register a potential onset. Similar techniques have been used to detect the first arrival time of echoes in geophysical prospecting [2], and have been adapted for detecting audio onsets [5]. We remove potential onsets for which the slow estimate is very low (these are false events in the background noise), and those which occur within 200 ms of an earlier onset (in order to keep only the first moment of attack.)

2.2.2 Detecting hits

The first movement feature we examine is the end of the striking gesture, which we refer to as a *hit*. In a real drum strike the hit would correspond to the moment when the drum stick hits the head of the instrument, imparts energy into the instrument thus initiating the sound, and rebounds in the opposite direction from which it came.

For a striking gesture in free space, where no physical contact is made, where is the end of the strike? As Collicutt et al. discovered [1], and as we found when inspecting our own data, the hit does not necessarily correspond to the moment when the minimum height is reached. Furthermore, we do not restrict our participants' movements to any particular plane or direction (they are instructed to act as if they are striking an invisible drum "somewhere in front of them"). Thus we define a hit as *the moment at the end of a striking gesture where the hand suddenly changes direction.*

To that end we designed a sudden-direction-change detector. The design takes inspiration from the onset detector described in section 2.2.1, which compares slow and fast estimates of audio energy. Our direction-change detector uses a slow and fast estimate of the hand's 3D velocity vector. The intuition is that during a sudden change of direction, the slow estimate will lag behind the quickly reacting fast estimate, and the angle between these two estimate vectors will be large. Upon inspecting our data we found that the moment we believed was the hit most reliably corresponded to a positive peak in the rate of change of this angle.

Here is a detailed description of our sudden-direction-change detector:

1. From the motion capture data extract the position data for the hand (using the marker on the back of the hand at the base of the third finger.) This is represented as three coordinates (x, y, and z) over time, where x is the direction the participant is facing, and z is upward.

2. Smooth the position data in each dimension by approximating each point as a weighted least-squares quadratic fit of the point and its seven neighbors on either side.[1]

3. Calculate the 3D velocity vector of the hand, $\mathbf{v}_{hand}$, as the first difference of the smoothed hand position.

[1]Thanks to Jonathan Abel for suggesting this technique.

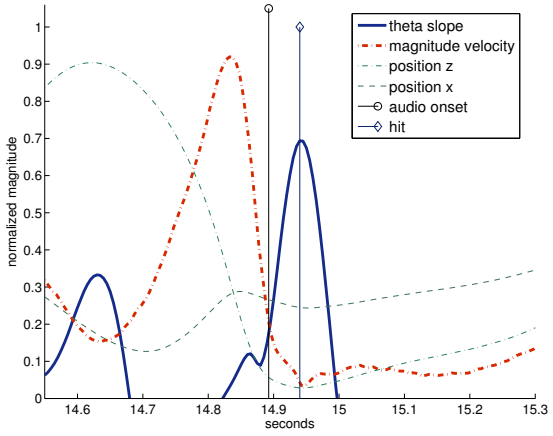

Figure 3: Detecting the hit for a strike gesture

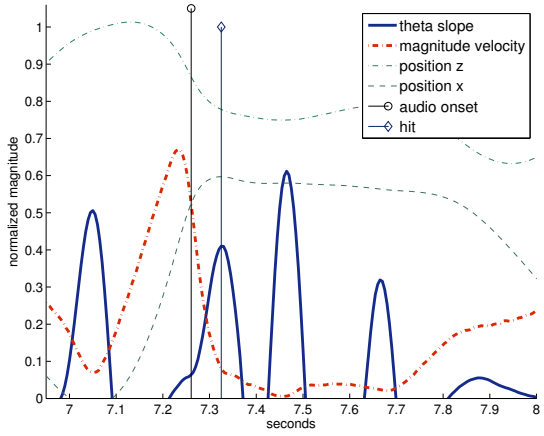

Figure 4: Detecting the hit for a more complex strike gesture.

4. Create two smoothed versions of the velocity vector by passing it through two "leaky integrators" (i.e. DC-normalized one-pole lowpass filters.) One, $\mathbf{v}_{slow}$, has a time constant of 100 ms, and the other, $\mathbf{v}_{fast}$ has a time constant of 5 ms. These are implemented as recursive filters on the 3D velocity vector according to the following difference equations:

$$\mathbf{v}_{fast}[n] = (1 - a_{fast})\mathbf{v}_{hand}[n] + a_{fast}\mathbf{v}_{fast}[n-1]$$
$$\mathbf{v}_{slow}[n] = (1 - a_{slow})\mathbf{v}_{hand}[n] + a_{slow}\mathbf{v}_{slow}[n-1]$$

where a_{slow} and a_{fast}, are the pole locations corresponding to the slow and fast time constants.

5. At each time point n calculate the angle θ between $\mathbf{v}_{slow}$ and $\mathbf{v}_{fast}$:

$$\theta[n] = \cos^{-1}\left(\frac{\langle \mathbf{v}_{slow}[n], \mathbf{v}_{fast}[n]\rangle}{\|\mathbf{v}_{slow}[n]\| \cdot \|\mathbf{v}_{fast}[n]\|}\right)$$

6. Calculate θ_{slope} as the first difference of θ.

7. Find all peaks of θ_{slope} which exceed a threshold. We consider the times of these peaks as the moment when a movement changed direction and we store them as candidate hit times.

We then want to find the change of direction associated with each strike gesture. That is we want those direction changes which occur after a fast movement and near to an audio onset. To find the hit for each audio onset we apply the following algorithm:

1. Since a hit occurs after a fast movement of the hand, we find all peaks of the magnitude hand velocity, $\|\mathbf{v}_{hand}\|$, which exceed a certain threshold.

2. For each of these peaks we find the next candidate hit time (i.e. a large peak in θ_{slope} as described above.)

3. To prevent choosing changes of directions that occur after a preparatory upwards movement, we remove hits for which the distance between the hand and the shoulder is less than a threshold.

4. For each audio onset we find the hit candidate which is closest in time, and define this as the hit time.

Does this method find the correct moment where a hit occurs? There is no way to know for sure because the "hit" does not exist in any objective sense, i.e. we have no ground truth. Figure 3 shows the detected hit time for a slow note by one participant. Since the striking gesture happens primarily in the x and z directions we plot those position components. We see that the hit happens at extrema in both these dimensions, and that the hit coincides with a distinct minimum in the magnitude velocity.

However the striking gesture of another participant, shown in figure 4, is more complex. This participant tended to add a sharp hook to the end of their strike. This can be seen by examining the position data, and is detected as multiple direction changes (large peaks in θ_{slope}). Our algorithm chooses the first such peak which corresponds to an extrema in the x direction and a sudden change of slope in magnitude velocity.

2.2.3 *Acceleration Peaks*

While examining the data we noticed that large peaks in the magnitude acceleration often occur close to the audio onset. For an unimpeded movement, acceleration of the hand is the result of a muscular force, and so we hypothesize that an acceleration peak may correspond to the internal movement event that air-drummers create to correspond with the sound. (In fact these peaks are decelerations as the participant sharply brakes their strike.) In order to pick the highest peak corresponding to each strike, we employ the following algorithm:

1. Calculate the acceleration vector, $\mathbf{a}$, as the first difference of the velocity vector calculated in step 3 above.

2. Calculate magnitude of the acceleration vector, $\|\mathbf{a}\|$.

3. Look for times where $\|\mathbf{a}\|$ first exceeds threshold $AccThr_{high}$, and call these T_{up}.

4. For each T_{up} find the next point where $\|\mathbf{a}\|$ passes below threshold $AccThr_{low}$, and call these T_{down}.

5. For each interval $[T_{up}, T_{down}]$ find the time of the highest peak in $\|\mathbf{a}\|$, and save this as a prospective acceleration peak.

6. For each audio onset find the prospective acceleration peak that is nearest in time, and define this as the acceleration peak time for that onset.

Figure 5 shows the acceleration peak for the strike gesture for a slow note. We can see that it occurs much closer to the audio onset than the corresponding hit.

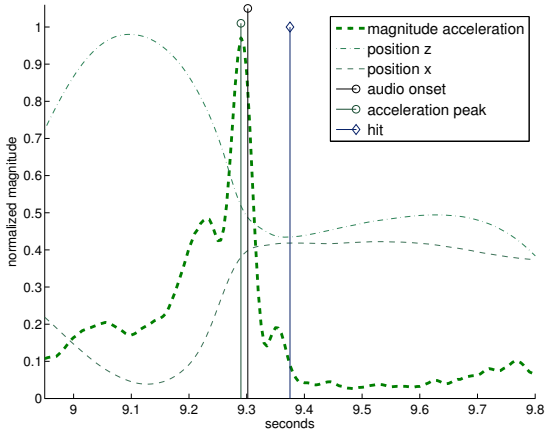

Figure 5: Detecting the acceleration peak for a strike gesture.

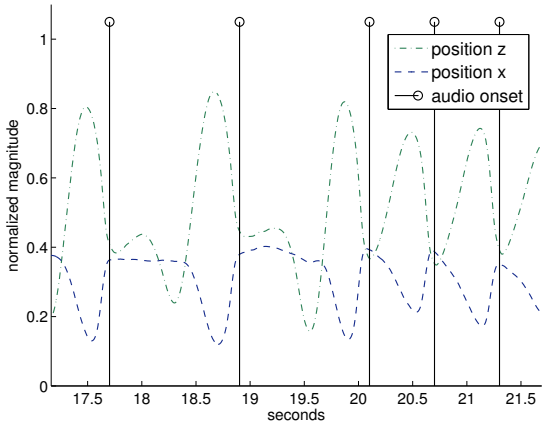

Figure 6: Position data for 2 slow notes and 3 fast notes.

2.2.4 *Timing Statistics*

The audio onset times, hit times, and acceleration peak times were calculated for each note, as described above. For each hit time and acceleration peak time we subtract the associated audio onset time to get the time offset (or asynchrony) between the audio event (the onset of the sound) and the detected movement event (a hit or an acceleration peak). A negative offset means the movement event preceded the audio event, and a positive offset means it came after. All subsequent analysis is performed on these offsets.

Since there were two trials for each task, we aggregate the data from each trial for each participant, and then split the data into the slow note and fast note conditions. For each participant this leads to a total of 40 events for each condition (5 events per 4-bar rhythm for each condition $\times$ 4 repetitions of the rhythm $\times$ 2 trials per task.)

In order to reject bad data due to detector errors or participant mistakes, we remove events whose offset is greater than half the time between notes (600 ms for slow notes, 300 ms for fast notes.) We then reject as outliers events which lie more than two standard deviations from the mean for each condition for each participant. This led to the rejection of 21 slow hits, 21 fast hits, 18 slow acceleration peaks, and 23 acceleration peaks (out of 400 total for each case.)

For the following results we want to know whether various conditions differ in the greater population. We compute the mean (or standard deviation) of each participant's offset times for the conditions we wish to compare. To infer whether the two conditions differ in the population, we conduct a two-sided paired-sample T-test of the ten participants' means (or standard deviations) for the two conditions.

For example, to compare whether for task 1 the standard deviation of hit times is different between slow notes and fast notes, we first find the standard deviation of each participant's slow hits for task 1. Then we find the standard deviations of each participant's fast hits for task 1. We now have 10 sample standard deviations for each condition, and on these we conduct a T-test with 9 degrees of freedom.

2.3 Results

2.3.1 *Which task is better?*

The first question we want to answer is whether there is any important difference between the two tasks the participants performed. Task 1 was to gesture in time to drum sounds, and task 2 was to vocalize drum sounds while gesturing along with them.

We compared both the means and standard deviations of task 1 and task 2 across all conditions (the four combinations of slow notes, fast notes, hits and acceleration peaks.) We found two significant differences: for fast notes both hits and acceleration peaks came slightly earlier in task 1, and for slow notes both hits and acceleration peaks had slightly higher standard deviations in task 1.

The differences were small, and the findings do not make a compelling case for either task being better. However task 1 is a simpler activity, so for the remainder of this paper we use only the data from task 1. As a sanity check the analyses described below were also performed on task 2, and none of the results contradict those reported here.

2.3.2 *Are slow and fast notes different?*

For most of our participants the gestures for slow notes had pauses or bounces between them, while the gestures for fast notes were simpler and more sinusoidal (see figure 6.)

For the sake of reliably triggering discrete air-gestures, we care whether the offset times for hits and acceleration peaks are different for slow and fast notes. If they are, then an instrument which triggers sounds from discrete air-gestures needs to somehow take into account the tempo and rhythmic level of the intended notes.

We break this question into four separate tests:

(a) For hits, do slow and fast notes have different mean offsets? Yes, slow hits are much later than fast hits ($T(9) = 4.5366, p = 0.0014$.)

(b) For hits, do slow and fast notes have different standard deviations? No.

(c) Do slow and fast notes have different mean offsets for acceleration peaks? No, the difference between fast and slow hits does not exist for acceleration peaks.

(d) Do slow and fast notes have different standard deviations of the offset for acceleration peaks? Yes, but only slightly ($T(9) = 2.5592, p = 0.0307$.)

Table 1 shows the mean offsets across all participants for all four cases. In summary, slow hits will on average occur after the audio onset, while hits detected for fast notes will fall much closer to the audio onset. For acceleration peaks, even though no significant difference was detected, the means for fast notes do precede the means for slow notes.

Table 1: Mean offsets across all participants

	Fast Notes	Slow Notes
Hits	-3 ms	44 ms
Acceleration Peaks	-32.9 ms	-13.9 ms

2.3.3 How are hits and acceleration peaks different?

Next we want to know how the offsets for hits and acceleration peaks differ:

(a) Do hits and acceleration peaks have different mean offsets for slow notes? Yes ($T(9) = 4.8440, p = 0.0009$.)

(b) Do hits and acceleration peaks have different mean offsets for fast notes? Yes ($T(9) = 4.5294, p = 0.0014$.)

(c) Do hits and acceleration peaks have different offset standard deviations for slow notes? Yes, but only slightly ($T(9) = 3.2287, p = 0.0103$.)

(d) Do hits and acceleration peaks have different offset standard deviations for fast notes? Yes, but only slightly ($T(9) = 2.4022, p = 0.0398$.)

It's no surprise that hits and acceleration peaks have significantly different mean offsets. Acceleration peaks, as we've defined them, should always occur before their associated hit.

A better question is, by how much? For slow notes we find that acceleration peaks precede hits by between 31 and 85 ms (this is the 95% confidence interval for the paired-sample T-test.) For fast notes acceleration peaks precede hits by between 15 and 45 ms. That is, the difference between hits and acceleration peaks is smaller for fast notes.

3. DISCUSSION

3.1 Hits vs. acceleration peaks

If you wanted to design a system to trigger sounds with air-drumming gestures that has a timing that feels natural to the user, which movement feature would you use?

It is interesting that when comparing standard deviations, either between hits and acceleration peaks (section 2.3.3, tests b and c), or between slow and fast notes (section 2.3.2, tests b and d), the few significant differences found were small. This suggests that either feature would have similar noise or jitter.

For a real-time system, acceleration peaks are better because they occur on average before the time of the audio event (see table 1), and don't vary as much with note speed (section 2.3.2, tests a and c.)

The hit and acceleration peak detection algorithms (sections 2.2.3 and 2.2.2) are not designed for real-time use. Both use thresholds which are calibrated to the range of the related variable over the length of a recorded trial. And the algorithm for choosing peaks relies on future knowledge. Thus for real-time applications these algorithms would need to be revised to work using only previous information.

3.2 Other applications of these results

The research described here, and future similar research into coordination between music and movement features, may have application to other musical interactions. For example hyper-instruments (traditional instruments that have been augmented with various sensors whose data is used to control computer based sound-processing) may be designed to more precisely trigger discrete audio events from gestures made with the instrument.

Similarly, systems which control musical processes from the movements of dancers may also be made to have better timing with respect to the dancer's internally perceived sense of discrete movement events.

3.3 Future Work

There are a number of ways this work can be developed and extended. We have not yet analyzed the individual differences between participants, and we would like to understand how dynamic level affects air-drumming gestures.

We currently compare notes at two rhythmic levels. To better understand how tempo or rhythmic level affects the timing of movement features with respect to the desired sound, we would need to run further studies with multiple tempos and more complex rhythms.

We expect that further analysis of our movement data may reveal other movement features which more reliably indicate the correct time of the player's intended sounds.

Lastly, it may be useful to study other non-striking discrete air-gestures, such as triggering a sound by bringing some part of one's body to a sudden halt, which is different than the drumming gestures studied here which usually have a rebound.

4. ACKNOWLEDGMENTS

Thanks to Professor Takako Fujioka for use of the Stanford NeuroMusic Lab and for invaluable advice.

5. REFERENCES

[1] M. Collicutt, C. Casciato, and M. M. Wanderley. From real to virtual: A comparison of input devices for percussion tasks. In *Proceedings of NIME*, pages 4–6, 2009.

[2] F. Coppens. First arrival picking on common-offset trace collections for automatic estimation of static corrections. *Geophysical Prospecting*, 33(8):1212–1231, 1985.

[3] S. Dahl. Playing the accent-comparing striking velocity and timing in an ostinato rhythm performed by four drummers. *Acta Acustica united with Acustica*, 90(4):762–776, 2004.

[4] C. Havel and M. Desainte-Catherine. Modeling an air percussion for composition and performance. In *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 31–34. National University of Singapore, 2004.

[5] J. Herrera and H. S. Kim. Ping-pong: Using smartphones to measure distances and relative positions. *Proceedings of Meetings on Acoustics*, 20(1):–, 2014.

[6] H. Kanke, Y. Takegawa, T. Terada, and M. Tsukamoto. Airstic drum: a drumstick for integration of real and virtual drums. In *Advances in Computer Entertainment*, pages 57–69. Springer, 2012.

[7] T. Mäki-Patola. User interface comparison for virtual drums. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 144–147. National University of Singapore, 2005.

[8] M. V. Mathews. Three dimensional baton and gesture sensor, Dec. 25 1990. US Patent 4,980,519.

[9] B. H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992, 2005.