Realtime Classification of Hand-Drum Strokes

Michael Krzyzaniak
School of Arts, Media + Engineering
Arizona State University
mkrzyzan@asu.edu

Garth Paine
School of Arts, Media + Engineering
Arizona State University
garth.paine@asu.edu

ABSTRACT

Herein is presented a method of classifying hand-drum strokes in real-time by analyzing 50 milliseconds of audio signal as recorded by a contact-mic affixed to the body of the instrument. The classifier performs with an average accuracy of about 95% across several experiments on archetypical strokes, and 89% on uncontrived playing. A complete ANSI C implementation for OSX and Linux is available on the author's website [1] .

Author Keywords

NIME, Kiki, timbre, HRI, djembe, cajon, drum, darbuka, doumbek, bongos, percussion, frame drum, stroke, classification, signal processing, machine learning

ACM Classification

H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing—signal analysis, synthesis, and processing, I.5.2 [Pattern Recognition]: Design Methodology—classifier design and evaluation, feature evaluation and selection.

1. INTRODUCTION

Any percussion instrument is capable of producing an infinite variety of sounds depending upon where and how it is struck or actuated. In practice, however, players generally define a finite number of discrete methods of striking the drum (strokes), each with a characteristic timbre. This is especially true for drums that are played with bare hands. In general, rhythmic patterns on these instruments are largely defined by the sequence of strokes that comprise them. Therefore in order to characterize such a rhythm one must model the strokes in addition to the durations used to articulate it. The authors are currently developing a robot that plays djembe, which can analyze and respond appropriately to a rhythm that a human improvises on another hand drum. Such a robot may have uses in entertainment or therapy, and we are working under the hypothesis that its success in these domains will rely on its ability to respond to subtle nuances of the human's playing, such as stroke patterns. It is thus desirable for the robot to transcribe the human's strokes. Because we want the robot to 'learn' from the human, we want the human to play an instrument whose technique is similar to the robot's. For this reason we focus on instruments that are played with the hands (such as djembe or cajon) as opposed to sticks (such as snare drum), and whose strokes are not characterized by subtle manipulations of the fingers (such as tabla).

[1] `http://michaelkrzyzaniak.com/HandDrumClassifier/NIMEDemo.zip`

NIME'15, May 31-June 3, 2015, Louisiana State Univ., Baton Rouge, LA.

2. PREVIOUS WORK

Exhaustive studies have been successful at classifying percussive sounds [2, 5]. Such studies seek to grossly identify the instrument that produced the sound, but in the present case we seek to identify more subtle nuances within a single instrument. We also seek to use a smaller feature set that can be implemented to run efficiently in real-time while reducing the volume of the feature space, so that fewer training examples may be used. Hochenbaum and Kapur [3] detect nuances in a percussionist's playing via an accelerometer on their wrists. In their seminal robot Haile, Weinberg and Driscoll [7] use fundamental frequency estimation to estimate where (radially) a human strikes a large 'pow-wow' drum. Our work presented in section 3.2 below suggests that this one feature is not sufficient to categorize sounds in the current application. Sarkar [4] and Chordia and Rae [1] evaluate several methods for classifying tabla strokes, which are characterized by subtle digital manipulations. Tindale, Kapur and Fujinaga [6] perform a study similar to the present one focused on snare drum, which is played with drumsticks. Although both [4] and [6] are intended to be realtime, they both operate upon a few hundred milliseconds of audio, which limits the maximum speed of playing to a few Hz. The current study presents a full, working realtime implementation for hand drums that does not limit the player's repetition rate.

3. IMPLEMENTATION

3.1 Onset Detection

In order to classify strokes in real time, it is first necessary to detect where in the signal the strokes occur. In the current context (a contact-mic affixed to a single percussion instrument), even the most naive onset-detection algorithm, such as amplitude thresholding, would likely suffice for a laboratory experiment. However, we implemented a more robust algorithm – a variant of the one described in [7]. Specifically, we perform an STFT on the incoming audio stream. At each analysis frame, we identify the bins whose magnitudes have increased since the previous frame, and accumulate the amount of positive change over all of the bins. The resulting values, computed at each successive analysis frame, serve as an onset-strength signal (OSS). Noise is removed from the OSS by low-pass filtering. Percival uses a 14th order filter which will introduce a delay of at 40 milliseconds at a sample rate of 44.1 kHz. In order to minimize

this delay, in our implementation the filter-order was hand-tuned to the lowest acceptable value, 4, resulting in 12 ms delay. Peaks are picked from the OSS as in [1]. Namely, onsets are identified at any local maximum in the OSS that is above a user-defined threshold. Our classifier uses the first 50 milliseconds of audio after the peak is identified. This value is short enough to exceed the maximum repetition rate that a human percussionist can sustain (roughly 16 Hz), but long enough to capture most flams as single events, which is how they are typically used. Although a 50 millisecond analysis latency is noticeably late in musical contexts, the authors are developing a predictive music generation algorithm which will correct for this and other delays introduced by the robotic system.

3.2 Feature Selection

Preliminary analysis of the three fundamental djembe strokes, bass, tone, and slap [5], indicated that the frequency distribution is different for each stroke; Namely, more energy is in the higher part of the spectrum for tone as opposed to bass, and again for slap as opposed to tone, as can be seen in Figure 1.

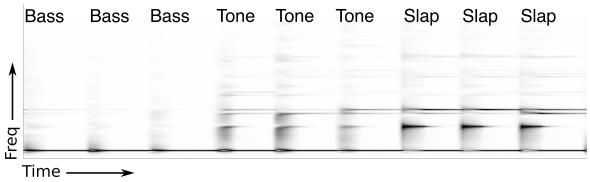

Figure 1: Spectrogram of djembe strokes showing different energy distributions in different strokes.

For this reason, spectral centroid was chosen as a classifier feature (see [6] for definitions of the features used henceforth). It was furthermore hypothesized that other spectral features such as spread, skewness, and kurtosis might be distinct for each stroke, and further analysis of several hand drums revealed this to be broadly correct, as can be seen in Figure 2.

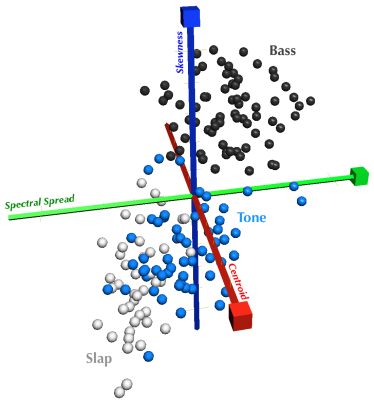

Figure 2: Cajon Stokes – Spectral centroid, spread, and skewness on 50 millisecond samples of untrived playing.

However, it was also found that, in some cases, these features were linearly dependent upon one another (an increase in one is always accompanied by an increase in another, obviating the need for both). Nonetheless, in most cases, the performance of the classifier was marginally degraded by the systematic exclusion of any one of these items, so they were all included in the model. Additionally, spread, skewness and kurtosis can be computed in a single pass through the spectrum, so if any one is included, the computational overhead of including all is small. These spectral features are all computed using an STFT (N=1024, Hann, hop=256), and averaged over the duration of the 50 ms sample. The frequency bins of all four spectral features were logarithmically weighted to model their perceptual interpretations. These features alone still leave some overlap in the stroke categories. It was hypothesized that amplitude and noisiness might account for some of this overlap. This was also found to be true, as can be seen in Figure 3, so RMS amplitude and zero-crossing rate were included as features.

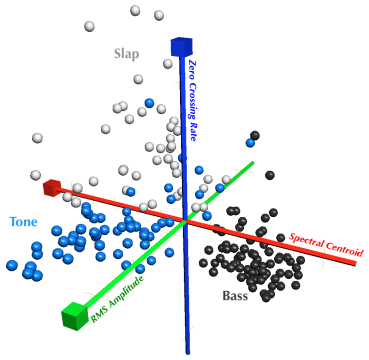

Figure 3: Cajon Stokes – Spectral centroid, ZCR and Amplitude on 50 millisecond samples of untrived playing.

All features are normalized across the training set by subtracting the mean and dividing by the standard deviation. This prevents the feature with the largest scale from dominating the classifier.

3.3 Realtime Classification

Following [8], we implement a kNN classifier. Although the time-complexity of this algorithm is high, it by no means precludes realtime operation in this case. A full analysis is beyond the scope of this paper. Nonetheless, kNN, not including distance calculations, can be implemented to run in $\mathcal{O}(N) + \mathcal{O}(k)$ time on average (Quicksort by distance, then Quicksort lowest k by category), where N is the number of training examples, k is the number of neighbors, and the symbol $\mathcal{O}()$ is shorthand for the word 'operations'. However, the implementation provided here has a higher complexity in k, and runs in about $\mathcal{O}(Nk) + \mathcal{O}(k^2/2)$. This was tested to run about 10 times faster than the former for values of $k < 10$, and twice as fast for very high values of k up to 350. This is owing to much simpler 'operations' despite a higher time-complexity. In any event, the classifier's accuracy was generally not found to improve for values of k greater than about 5, so the added complexity is small in practice. In both cases, the actual computation time of kNN is dominated by calculating the distances, which has a complexity of $\mathcal{O}(Nf)$, where f is the number of features in the model. Because f is fixed by the model, the goal would be to reduce N, which could be done through a variety of techniques, such as clustering. However, even this was not necessary; In practice, the classifier was found to require between about 14 and 32 microseconds to run on a 2.6 GHz Intel i7, for $k = 5$ and $N = 100$. On the other hand, feature calculation, including STFT, required between about 1405 and 2220 microseconds. These calculations can run,

at most, every 50 milliseconds (the amount of audio used by this algorithm), which would consume at most about 4% of CPU time. Indeed, the operating system reported the CPU usage of the classifier to be under 7% during a barrage of strokes, and the excess is consistent with the onset-detector's computation time.

4. EVALUATION

Several experiments were designed to test the classifier's efficacy under a variety of conditions. The first few experiments will go deep and analyze a single instrument, the djembe, in a variety of contexts. Subsequently, a broad assessment of the generalizability of the model will be made by testing it on several instruments. Because in practice we desire to capture only the sound of the instrument played by the human, while eliminating the sound of the robot and other extraneous sounds, all instruments in this study are recorded using a piezo-disc contact-mic coupled with a high-impedance amplifier.

4.1 The Ecological Case

The first experiment was designed to be simple but ecologically valid, representing how the classifier is intended to be used. A contact-mic was affixed to a djembe. The frequency sensitivity of the mic is highly dependent upon its placement, and a location near the bottom of the drum seemed to capture a good mix of low and high frequencies. Because the curvature of the djembe is not amenable to a flat piezo disc, the disc was coupled to the drum via a small amount of putty. The classifier was trained by playing 20 archetypical examples of each stroke – bass, tone, and slap – in succession. A rhythmic sequence of 125 strokes (49 bass, 47 tone and 30 slap) was then played, and the onset detector and classifier's performance were evaluated. The onset detector correctly identified all onsets and gave no false positives. The classifier was 95% accurate for $k = 2$. The signal data was also recorded and fed back into the classifier for each k from 1 to 5. The worst case was $k = 5$ with accuracy of 86%, as can be seen in Table 1 .

Table 1: Classifier confusion matrix for uncontrived djembe strokes (worst-case scenario where $k = 1$; Columns labelled by software; Rows by performer).

	Bass	Tone	Slap
Bass	**41**	8	0
Tone	0	**44**	2
Slap	0	4	**26**

The confusion between tone and slap were attributable to the same strokes for each value of k. These strokes were aurally ambiguous to the authors as well. The variation in accuracy as a function of k was attributable to variation in confusion between bass and tone. This is likely due to tie-resolution for ambiguous strokes, which could be improved with a larger training set. It should be noted that leave-one-out cross-validation on the training set indicated very high accuracy: 100% for $1 \leq k \leq 3$. The lower accuracy on the independent set of observations is probably because strokes used in actual rhythms are somewhat less consistent than their archetypical counterparts, owing to timing constraints, expressive variability, noise in the human motor control system, and the physics of the vibrating drum head.

4.2 loudness

In the previous experiment, the drum was played at a moderate loudness with natural metric accents. Another experiment was conducted to test the accuracy of the classifier when extreme variations in loudness were present. In this experiment, 30 strokes of each category (bass, tone, slap) were recorded on djembe. Of these, 10 were played very softly, 10 intermediate, and 10 very loud. In this case, even after hand-tuning the threshold, the onset detector failed to detect three strokes in the quietest category and spuriously detected two false positives (immediately following a true positive) in the loudest category. The spurious false positives were removed from the data, and no attempt was made to recover the missed strokes. Leave-one-out cross-validation was performed on the data for all values of k, treating them as three stroke categories. The accuracy is slightly improved by choosing k a few greater than 1, and then gradually decreases with increasing k, as can be seen in Figure 4. The classifier was, on average, 93% accurate for $1 \leq k \leq 10$.

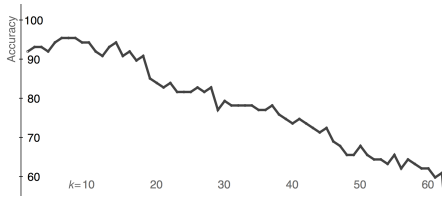

Figure 4: Classifier accuracy as a function of k.

The shape of this curve is representative of all the data sets analyzed.

4.3 Extended Techniques

Although bass, tone, and slap are the core strokes of djembe technique, skilled players of this and other hand drums define and use many more strokes, which are typically subtler variations on the core three. This experiment tested the classifier's accuracy on a set of 7 strokes: bass, tone, slap, muted slap (dampened with free hand during stroke), closed slap (the striking hand remains on drumhead after stroke), closed bass (ditto) and flam (two quick slaps in rapid succession, taken as a single gestalt). Not all of these are proper to djembe technique, but are characteristic of several Latin American and African instruments. 50 examples of each were played, and cross-validated. The classifier was 90.6% accurate for $k = 5$, and on average 88.2% accurate for for $1 \leq k \leq 10$. The confusion was as Table 2 for $k = 4$.

Table 2: Classifier confusion matrix for archetypical djembe strokes ($k = 4$; Columns as labelled by software; Rows as labelled by performer).

	Slap	Tone	Bass	Closed Slap	Muted Slap	Closed Bass	Flam
Slap	**44**	3	0	0	0	0	3
Tone	5	**45**	0	0	0	0	0
Bass	0	0	**50**	0	0	0	0
Closed Slap	1	0	0	**47**	0	0	2
Muted Slap	0	0	1	0	**46**	3	0
Closed Bass	0	0	1	2	5	**42**	0
Flam	2	0	0	11	0	0	**37**

It is interesting to note that the plurality of confusion, 46% of it, involved flam. This was unexpected because flam was hypothesized to contain much more energy over the sample than other strokes, owing to the second attack,

which should make it easily identifiable.

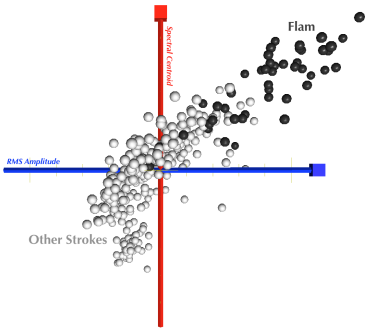

Figure 5: Djembe Stokes – Flams in Spectral Centroid - RMS Amplitude space

Although this was true, as is seen in Figure 5, the effect had great variance, which, on the low end, caused many flams to intermingle with the other stroke categories. This is suspected to be a side-effect of the temporal granularity of the onset-detector rather than an acoustic property of the strokes, although further analysis is needed.

4.4 Different Instruments

The previous experiments focused on djembe in order to give a complete picture of the classifier's performance on a single instrument. However, the classifier was designed to be more general, so another set of experiments tested several instruments.

4.4.1 Cajon

In one experiment, 20 archetypical strokes from each of 4 categories – bass, tone, slap, and muted slap – were played on a Peruvian cajon (wooden box) without snares. Cross-validation revealed an average accuracy of 93% for $1 \leq k \leq 10$. In another experiment, an uncontrived rhythmic sequence of 168 strokes from three categories was played on cajon. Each stroke was manually labelled (bass, tone, slap) and given to the classifier for analysis. Cross validation on this set yielded an average accuracy of 87% for $1 \leq k \leq 10$. As with the djembe, the lower accuracy on performed strokes as apposed to contrived ones is likely attributable to greater variability in the acoustic content of the strokes. Generally, archetypical strokes should probably be used as training examples. Figures 2 and 3 depict this dataset.

4.4.2 Darbuka

In this experiment, 30 archetypical strokes from each of 3 categories – doum, tek, and pa – were played on darbuka (ceramic goblet drum). The classifier was on average 96% accurate for $1 \leq k \leq 10$.

4.4.3 Frame Drum

Furthermore, 30 archetypical strokes from each of 3 categories – doum, tek, and pa – were played on a small frame drum (hoop and animal hide) with unusually thick skin. Cross-validation indicated an average of 95% accuracy for $1 \leq k \leq 10$.

4.4.4 Bongos

Several percussion instruments, such as bongos, are actually two separate drums of different pitch, optionally joined together by a center block and bolts. While it would in principle be possible to treat each drum separately, with a separate contact-mic and separate set of training examples for each, we wanted to know to what extent such instruments could be analyzed as a single unit. Therefore, a contact-mic was placed on the center block joining a pair of bongos. 30 exemplary strokes from each of 5 categories – open and closed strokes on the larger drum, and open, closed, and muted strokes on the smaller drum – were played. Cross validation yielded an average accuracy of 94% for $1 \leq k \leq 10$. The majority of the confusion was between the open and closed strokes on the larger drum. This is suspected to be due in part to the placement of the contact-mic which was not acoustically ideal but provided a strong signal for both drums. It is hypothesized that the accuracy could be increased by using two separate microphones, one placed directly on the body of each drum. Such microphones could be soldered together in parallel and serviced by a single amplifier and set of classifier training examples.

5. CONCLUSION AND FUTURE WORK

In conclusion, we have found the provided classifier to work with relatively high accuracy on a variety of instruments. In practice, its correspondence to human perception is acceptably high for the intended application, i.e. collaborative music making with a musical robot. Future work will use a variant of this algorithm to allow percussion robots to perform auto-calibration. If a human played several archetypical strokes on the instrument as training examples, then the robot could search its control-parameter space (impact angle, velocity, hand tension, etc...) for a point that yielded the lowest self-classification error.

6. ACKNOWLEDGEMENTS

Thanks to the School of School of Arts, Media + Engineering at Arizona State University for ongoing support. Thanks to Simone Mancuso for generously allowing us to use his instruments. We give reverence to the animals whose skin was used in the manufacture of some of the instruments in this study.

7. REFERENCES

[1] P. Chordia and A. Rae. Tabla gyan: A system for realtime tabla recognition and resynthesis. In *Proc. Int. Comput. Music Conf.(ICMC)*, 2008.

[2] P. Herrera, A. Yeterian, and F. Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *Music and Artificial Intelligence*, pages 69–80. Springer, 2002.

[3] J. Hochenbaum and A. Kapur. Drum stroke computing: Multimodal signal processing for drum stroke identification and performance metrics. In *International Conference on New Interfaces for Musical Expression*, 2012.

[4] M. Sarkar. *Tablanet: a real-time online musical collaboration system for Indian percussion*. PhD thesis, Massachusetts Institute of Technology, 2007.

[5] J. Silpanpää. Drum stroke recognition. Technical report, Tampere University of Technology, 2000.

[6] A. Tindale, A. Kapur, and I. Fujinaga. Towards timbre recognition of percussive sounds. *Submitted to ICMC 2004*, 2004.

[7] G. Weinberg, S. Driscoll, and M. Parry. Haile–an interactive robotic percussionist. *Ann Arbor: Scholarly Publishing Office, University of Michigan Library*, 2005.