REALTIME SELECTION OF PERCUSSION SAMPLES THROUGH TIMBRAL SIMILARITY IN MAX/MSP

Arne Eigenfeldt
School for the Contemporary Arts
Simon Fraser University
Burnaby, Canada

Philippe Pasquier
School for Interactive Arts and Technology
Simon Fraser University,
Burnaby, Canada

ABSTRACT

A system for rating timbres for similarity from within a large database of percussion samples is described. Each sample in the database is pre-analysed using a Bark auditory modeler, and the three highest-rated bands are stored. In performance, live instrumental timbres are analysed, and comparisons are made within a heuristic search to create a list of similar and related samples.

1. INTRODUCTION

Despite significant research into timbral classification [8], most studies have been of limited use to composers, particularly those interested in realtime performance. The reasons for this vary, but include the inherent non-realtime nature, and its complexity; the former is being overcome, the latter less so. Despite this, composers of realtime music are beginning to use timbre as a control mechanism for interactive systems (see section 2.1).

One relevant question for live performance is how to handle large sample libraries in realtime; specifically, given a large database of commercial percussion samples, how can meaningful collections be derived, selected, and rearranged in performance?

Commercial sample libraries are usually grouped by geography (i.e. "Roots of South America", "Heart of Asia") and instrument (i.e. "tabla", "talking drum"). For commercial music purpose, such sample groups can often be freely substituted ("let's hear that percussion part with African samples…or South American"). Using samples only within such groupings eliminates the possibility of combining diverse samples from different groups; however, matching and labeling samples by hand is extremely time-consuming.

This is further complicated in performance, where laptoppers are often faced with selecting from menus of hundreds of sounds. Although a laptopper's rhythmic material is often in the form of pre-recorded loops, when playing back MIDI loops, the composer is faced with assigning specific samples to parts.

The described system is part of *Kinetic Engine* [1], a multi-agent performance system that generates complex and evolving ensemble rhythmic patterns, with minimal user interaction. In this system, all timbral choices are made within the software during performance; thus, an intelligent method of selection was required.

This research has two distinct goals:

- to treat samples individually, separate from instrument classification, and create new sample groupings based upon timbral similarity or dissimilarity;
- to interact with a live percussionist, in which the software would respond with similar or dissimilar timbres.

Section 2 will discuss related work, and its relationship to the described research; Section 3 will give a detailed description of the analysis; Section 4 will describe the realtime performance aspects; Section 5 will give the results of some testing, and Section 6 will offer some conclusions.

2. RELATED WORK

2.1. Realtime Timbre Analysis and Recognition

Timbre is becoming a potential control structure for realtime performance. Early work in this area was done by Lippe [2], who used Max and an IRCAM Signal Processing workstation (ISPW) to analyse timbre in performance.

More recently, Hsu [3] used realtime timbre recognition of saxophone to guide an interactive system. Ciufo [4] used Jehan's MSP external analyzer~ [5] to analyse incoming audio, which in turn influenced live audio processing. Jehan's MSP externals, used in this research, are allowing realtime composers to explore the potential of timbre as a control element.

Lastly, the recent appearance of MIR algorithms in ChucK [6] will, no doubt, precipitate many new works that involve timbral recognition.

2.2. Percussion Classification

Significant research [7] has been undertaken in transcribing audio, some of which involves the extraction of specific timbres. Gouyon et. al describe one such system that focuses upon percussion [9].

Early work in percussion transcription was done by Schloss [10], who classified several different conga strokes. Herrera et al. [11, 12] classified up to 33 different pitched and unpitched percussion instruments using 1976 different samples, and achieved an 85% recognition rate using a k-nearest neighbor algorithm. Tindale [13] classified different timbres produced by a snare drum, achieving a 95% recognition rate using a feed-forward neural network. Chordia [14] segmented and labeled tabla strokes, using methods that included neural networks, a multivariate Gaussian model, and tree-based classifiers.

2.3. Differences from Previous Work

The work described here focuses upon aspects of realtime performance, and is thus different from previous timbral recognition research. As the authors are composers, the intentions are also markedly different: rather than attempting to navigate a search space and return an exact match, the compositional interest is in *similar* timbres, rather than specific matches. In response to a performer's use of timbales, for example, the system response should not necessarily be limited to timbale samples, but timbres that have similar spectral content.

The research is loosely based upon perceptual models, but it does not make any claim that it is supported by subject testing. Instead, the authors wish to "play the composer card", and suggest that this work is based upon an auto-ethnographic analysis of our own listening, and is a codification of our compositional decision-making.

3. DESCRIPTION

In the present system, a majority of the computation is done prior to performance in an analysis of a sample database that consists of 118 diverse instruments and 1551 individual samples. These samples are exclusively non-pitched percussion, derived from a variety of commercial sample libraries. All sample durations are less than two seconds. During performance, portions of the analyses are organized into probability distributions for selection based upon timbral relationships.

3.1. Database Organisation

An automated patch was written in MaxMSP to automatically add samples to the database. It mixes any stereo files to mono, normalizes the samples, performs the Bark analysis (see Section 3.1.1), saves the samples in the correct format (AIFF), and adds the sample to the three sample arrays files (see Section 3.1.2).

3.1.1. Bark analysis

A Bark analysis, using Jehan's auditory model spectrum analyser[1], is performed on each sample. The Bark analysis [15] provides perceptually meaningful data, corresponding to the first 25 critical bands of hearing. Furthermore, when compared to standard FFTs, the analysis itself already provides useful data reduction.

Peak levels for each of the 25 bands are held, thereby creating a static line spectrum (see Fig.1). The decision to use a non-dynamic vector was initially due to the nature of the transient percussion timbres. Admittedly, even with timbres of short duration, a great deal of information is still lost; however, the non-dynamic vector provided the required differentiation between samples.

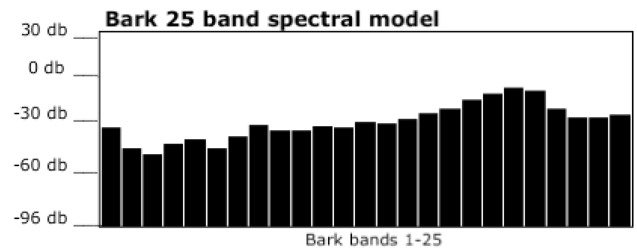

Figure 1. Bark analysis of Khanjari.

The three highest Bark bands are normalized and stored (see Table 1).

Band	Amplitude
19	1.0
18	0.96
20	0.91

Table 1. Normalized Bark amplitudes for the three highest bands of Khanjari of Fig. 1.

3.1.2. Creation of sample arrays

Once the entire sample database has been analysed, several arrays are created. These include:

-sample_DB, which contains the filepaths to individual samples;

-spectrum_DB, which contains each sample's Bark analysis;

-bands_by_sample, which contains pointers to the sample_DB, sorted by Bark energy bands. This allows access to all samples that have peak energy in a specific band.

For example, the first Khanjari sample has an index of 760 in sample_DB, and, as shown in Table 1, contains peak energy in bands 19, 18, and 20. Therefore, in bands_by_sample, the index 760 is entered in indices 19, 18, and 20 (along with all other samples that have peak energy in those bands).

4. REALTIME SEARCH USING HEURISTICS

It is possible to select samples for performance using approximate spectral bandwidths, from which the software

[1] bark~, available at http://web.media.mit.edu/~tristan/

can use Gaussian probabilities to generate a random input vector as its three input bands (see Fig. 2).

Figure 2. Defining a spectral band from which to choose a sample. Moving the sliders alters the displayed bandwidth to the right. Clicking on the "new" button will generate a random input vector from within the chosen bandwidth.

Another approach is to select samples based upon realtime analysis of incoming audio timbre, using Jehan's analyser~ to derive the 3 highest Bark bands. All samples in the database that contain energy in the same bands as the incoming timbre, as well as in adjacent bands, are selected from bands_by_sample. This sample list, which can reach several hundred items, is then incremented randomly (in order to extract different results given the same input values), and bands are compared.

4.1.1. Similarity comparisons

A "fuzzy list" is created around the incoming bands, using adjacent bands: for example, given incoming bands of (5 9 14), a fuzzy list is generated of (4 6 8 10 13 15). Direct matches (i.e. 5, 9, or 14) are summed; matches to the fuzzy list are summed and scaled by 0.66 - a hand-tuned value that created the ordering in Table 2. The scores are summed and divided by three to create a closeness rating (see Table 2).

Direct matches (n x 1.0)	Fuzzy matches (n x 0.66)	Rating ($\sum$ / 3)
3	0	1.0
2	1	0.89
1	2	0.77
2	0	0.67
0	3	0.66
1	1	0.55
1	0	0.33
0	2	0.44
0	1	0.22

Table 2. Closeness ratings based upon matching bands

4.2. Search Heuristics

Since the search space can contain several hundred items, it was found that evaluating it in its entirety to find the best matches (using a 16-nearest neighbour implementation) took too long: often taking several seconds (see section 5). Therefore, a heuristic algorithm was created to find enough (16) acceptable solutions.

When the search first begins, only those ratings (see Table 2) of 0.67 and above are acceptable; after 250 ms of searching, this is enlarged to include samples of 0.66 and above, and after 1000 ms of searching, it is enlarged to include 0.55 and above. The database is large enough that this final criteria of at least one direct and one "close" match (creating the rating of 0.55) provided the required number of samples.

4.3. Dissimilarity

Choosing dissimilar timbres to a given Bark set is simply a matter of creating an inverse probability vector around the incoming three bands, then choosing three new bands using quantile probabilities from this vector, and finally searching for timbres with the new bands.

If the inverse selected bands are probabilistically drawn only once, the selected timbres will all be similar; however, the same inverse probability vector can generate several different sets, each of which would be dissimilar to the original, yet with a likelihood of dissimilarity to one another as well.

5. TESTING

The following tests were done, comparing the heuristic search described, with a 16-nn implementation that returned the 16 top results for each query, examining the entire database, and comparing all three Bark bands and energies.

The first test used the same 10 random samples from the database as queries (see Table 3).

	Min	Max	Mean
Heuristic	163	981	376
16-nn	2827	8650	6293

Table 3. Time in ms to generate 16 results, using the same 10 random queries from the database.

While this demonstrates that the proposed heuristic algorithm is faster than the 16-nn, of greater significance is the number of high ratings produced by the former; this is shown in Table 4.

Min	Max	Mean
.64	.89	.74

Table 4. Rating the scores of the 16 selected timbres, using Table 2 rating scheme.

The second test generated three random Bark bands and amplitudes (see Table 5).

	Min	Max	Mean
Heuristic	174	1317	867
16-nn	881	8612	3656

Table 5. Time in ms to generate 16 results, using random bands and amplitudes.

In comparison to the first test, the heuristic method took longer to find acceptable solutions; however, even though the queried timbres are potentially outside the database, the algorithm still produced acceptable results (see Table 6).

Min	Max	Mean
.42	.81	.6

Table 6. Rating the scores of the 16 selected timbres from the second test, using table 3 rating scheme.

6. CONCLUSIONS

A system for rating timbres for similarity from within a large database of percussion samples was described, together with a heuristic search algorithm that is useable in (realtime) performance. Major differences from existing MIR-type timbre recognition software include realtime capabilities, and a heuristic search that returns different results given the same query, as well as one being based upon time, rather than the size of the search space. Furthermore, the ability to select dissimilar timbres from incoming timbral analysis is also possible.

Ongoing and future work includes the ability to analyse samples longer than 2 seconds, which would entail a dynamic timbral representation, rather than the current static spectrum. Visualisation of sound similarities is already being undertaken through the application of a self-organising map for timbral similarity and selection from a database of melodic loops and soundscape recordings.

This software, along with *Kinetic Engine*, was created in Max/MSP, and is available at the first author's website: www.sfu.ca/~eigenfel/research.html

7. REFERENCES

[1] Eigenfeldt, A. "Drum Circle: Intelligent Agents in Max/MSP", in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, 2007.

[2] Lippe, C. "A Composition for Clarinet and Real-Time Signal Processing: Using Max on the IRCAM Signal Processing Workstation", in *Proceedings of the 10th Italian Colloquium on Computer Music,* Milan, Italy, 1993.

[3] Hsu, W. "Managing Gesture and Timbre for Analysis and Instrument Control in an Interactive Environment", in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Paris, 2006.

[4] Ciufo, T. "Beginner's mind: an environment for sonic improvisation", in *ICMC,* Barcelona, 2005.

[5] Jehan, T., Schoner, B. "An Audio-Driven Perceptually Meaningful Timbre Synthesizer", in *ICMC,* Berlin, 2001.

[6] Fiebrink, R., Wang, G., Cook, P. "Support for MIR Prototyping and Real-time applications in the ChucK Programming Language", in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, 2008.

[7] Tzanetakis, G., Cook, P. "MARSYAS: A Framework for Audio Analysis" in *Organized Sound, Cambridge University Press* 4(3), 2000.

[8] Herrera, P., Amatriain, X., Batlle, E., Serra, X. "Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques", in *ISMIR*, Plymouth, MA, 2000.

[9] Gouyen, F., Pachet, F., Delerue, O. "On the Use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds", in *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00)*, Verona, Italy, 2000.

[10] Schloss, W.A. *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High-Level Analysis*, Ph.D. Thesis, Stanford Univ., 1985.

[11] Herrera, P., Yeterian, A., Yeterian, R., Gouyon, F. "Automatic classification of drum sounds: a comparison of feature selection and classification techniques", in *Proceedings of the 2nd International Conference on Music and Artificial Intelligence*, 2002.

[12] Herrera, P., Dehamel, A. Gouyon, F. "Automatic labeling of unpitched percussion sounds", in *Proceedings of the Audio Engineering Society*, 2003.

[13] Tindale, A., Kapur, A., Tzanetakis, G., Fujinaga, I. "Retrieval of percussion gestures using timbre classification techniques", in *ISMIR*, Barcelona, 2004.

[14] Chordia, P. "Segmentation and Recognition of Tabla Strokes", in *ISMIR*, London 2005.

[15] Jehan, T. *Creating Music by Listening*. PhD thesis, MIT Media Lab, Cambridge, MA, 2005.