QUERY-BY-BEAT-BOXING: MUSIC RETRIEVAL FOR THE DJ

Ajay Kapur
Electrical and Computer Engineering
University of Victoria

Manj Benning
Electrical and Computer Engineering
University of Victoria

George Tzanetakis
gtzan@cs.uvic.ca
Computer Science
University of VIctoria

ABSTRACT

BeatBoxing is a type of vocal percussion, where musicians use their lips, cheeks, and throat to create different beats. It is commonly used by hiphop and rap artists. In this paper, we explore the use of *BeatBoxing* as a query mechanism for music information retrieval and more speci£cally the retrieval of drum loops. A classi£cation system that automatically identi£es the individual beat boxing sounds and can map them to corresponding drum sounds has been developed. In addition, the tempo of *BeatBoxing* is automatically detected and used to dynamically browse a database of music. We also describe some experiments in extracting structural representations of rhythm and their use for style classi£cation of drum loops.

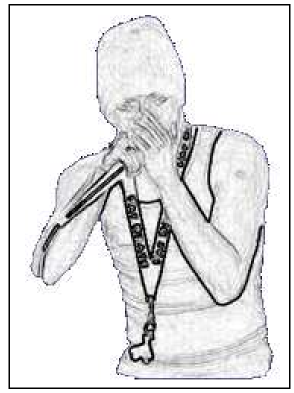

Figure 1. BeatBoxing

1. INTRODUCTION

Disc jockey (DJ) mixing, which £rst emerged in the early 1950's in Jamaica, is one of the earliest examples of music information retrieval (MIR), where a DJ retrieves prerecorded music from a set of records based on the mood and atmosphere of a night club and audience energy. Traditionally, a DJ uses a set of turntables in conjunction with a mixer to £lter appropriate music for the moment. In this paper, we present new tools for the modern DJ, enabling them to retrieve music with a microphone by *BeatBoxing*.

BeatBoxing is a type of vocal percussion, where musicians use their lips, cheeks, and throat to create different beats. It originated as an urban artform. The hip-hop culture of the early 1980's could seldom afford beat machines, samplers, or sound synthesizers. Without machine supplied beats to rap over, a new drum was created - the mouth. Generally, the musician is imitating the sound of a real drumset or other percussion instrument, but there are no limits to the actual sounds that can be produced with their mouth. As shown in Figure 1, the musician often covers his mouth with one hand to create louder, deeper sounds. A wide variety of sounds can be created with this technique enabling individual *BeatBoxers* to have different repertoires of sounds.

Most of existing work in MIR has either concentrated on melodic and pitch information in symbolic MIR or focused on timbral information in the case of audio MIR. Rhythmic information, although an important aspect of music, is frequently neglected. In this paper we focus on the retrieval and browsing of electronic dance music such as Drum & Bass, House, Rhythm & Blues etc. We believe these musical styles provide unique challenges and opportunities because their rhythmic characteristics are more important than their melodic and timbral aspects. In addition, the proposed techniques and applications can be used by experienced retrieval users that are eager to try new technologies, namely DJs. Furthermore, we want our developed MIR systems to be used in active live performance for mixing and browsing in addition to the traditional query/search model.

Based on these observations, the main goal of this work is to explore the use of *BeatBoxing* as a query mechanism for both retrieval and browsing. The paper is organized as follows: In section 2 we brie¤y describe existing digital tools for the DJ and related work in MIR. In section 3 we describe how the individual *BeatBoxing* sounds can be automatically identi£ed, our method of tempo extraction and explore the use of structured beat representations for style classi£cation. In section 4 data collection and various experiments in classi£cation and retrieval are described. In section 5 the implementation of the algorithms and two novel user interfaces for browsing music and processing *BeatBoxing* sounds are described. Finally, in section 6 we discuss conclusions, challenges and directions for future research.

2. RELATED WORK

Research in building novel digital systems for DJ's is a growing area. There are a number of commercial products such as *Final Scratch* [1] by Stanton, which is a turntable controller that uses special records to send position sensor data to the computer. *Tracktor* [2] by Native Instruments is a powerful software that includes graphical waveform displays, tempo recognition, automatic synronization, real-time time stretching, and ten cue points for live mixing of MP3, WAV, AIFF, and audio CD formats.

Academic research on building tools for the DJ is also becoming more commonplace. *AudioPad* [1] and *Block Jam* [2] are both performance tools for controlling playback of music on sample based sequencers. *Mixxx* [3] is software used both in realistic performance setting and as a means to study DJ interface interaction.

Another important area of in¤uence is automatic rhythm analysis. Initial work in this area such as [4, 5] concentrated on the extraction of tempo but more recent work has looked into extracting more detailed information. The classi£cation of ballroom dance music based on rhythmic features is explored in [6]. The extraction and similarity of rhythmic patterns independently of the actual sounds used to produce them is explored in [7] using a Dynamic Programming approach. The classi£cation of different percussive sounds using the ZeroCrossing Rate is described in [8]. The idea of using the voice as a query mechanism is explored in the different context of Indian tabla music in [9]. Finally, our approach to Query-by-Beat-Boxing although based on rhythm rather than melodic information shares some similarities with query-by-humming systems such as [10, 11].

On the application side, an important in¤uence has been the idea of a music browsing space where the visual information is correlated with music similarity and relations. Examples include the exploration of music collections by using visual representations of Self-Organizing Maps [12], using a fast version of multidimensional scaling (MDS) called Fast Map in [13] and the use of direct soni£cation in the Sonic Browser [14].

3. AUDIO ANALYSIS AND CLASSIFICATION

The £rst step in Query-by-BeatBoxing is to identify the individual vocal percussion sounds. This stage roughly corresponds to the pitch detection-segmentation stage in Query-by-Humming. Audio drum loops are signi£cantly different for vocal *BeatBoxing* loops and therefore require different analysis methods. Because our goal is to be able to retrieve from databases of drum loops, we need to be able to convert audio drum loops into some representation that can be used for similarity matching between those different types of signals.

[1] `http://www.finalscratch.com` (April 2004)
[2] `http://www.native-instruments.com` (April 2004)

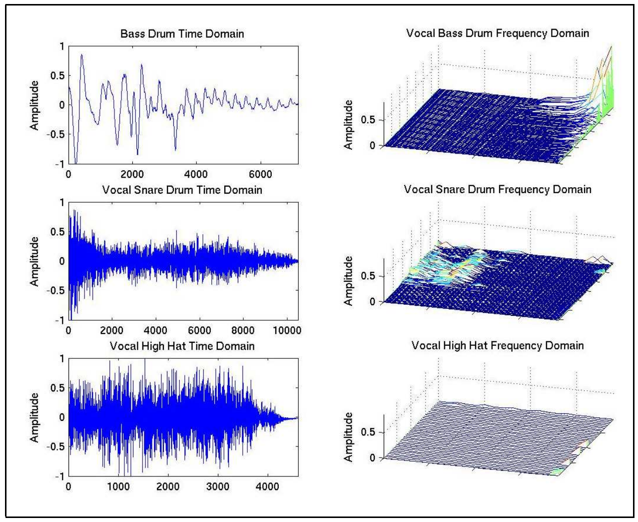

Figure 2. Graphs showing time and frequency domain of vocal bass drum, snare drum and high hat

3.1. BeatBoxing sound identi£cation

Most commonly *BeatBoxing* techniques include sounds which imitate a real drumset such as bass drum, snare drum, and high-hat. However, advanced vocal percussion has no limits to the sounds that can be produced, including noises such as simulated turntable scratches and humming-along the beat. In our experiments, three general types of of beat boxing sounds were analyzed and classi£ed. The £rst is a bass drum vocal hit that is characterized by lower frequency coming from the chest of the performer. The second is a snare drum vocal hit that is created by the quick pass of air through the teeth. The third is a high-hat vocal hit, which is characterized by a 'S' sibilance sound, created by the tongue arching upward to the roof of the mouth. Figure 2 shows graphs of the time and frequency domain plots for these three types of vocal hits.

One important observation is that the spectral and dynamic characteristics of the vocal drum sounds are not directly similar to the corresponding real drum sounds so an audio feature extraction and classi£cation stage is required to identify the sounds. The produced vocal percussive sounds have short duration (average 0.25 seconds) and therefore a single feature vector is computed for the duration of the sound.

For the feature extraction we experimented with a variety of feature sets proposed in the literature. The following features were considered:

- Time Domain features: ZeroCrossings, Root-Mean-Squarred Energy (RMS) and Ramp Time
- Spectral Domain features: Centroid, Rolloff, and Flux
- Mel-Frequency Cepstral Coef£cients (MFCC) [15]
- Linear Predictive Coef£cients (LPC) [16]

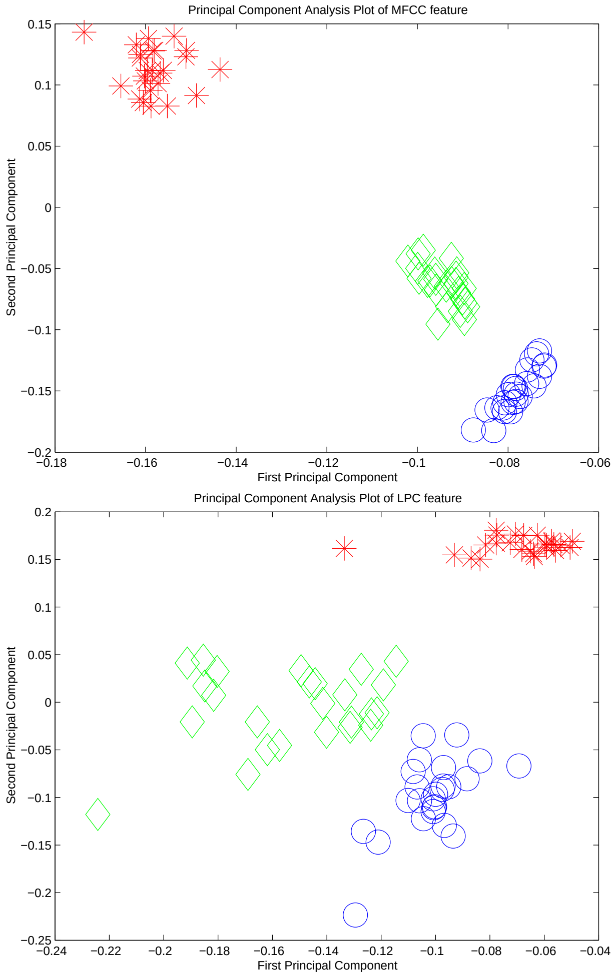

Figure 3. Scatter plots for feature analysis showing three clusters (bass (star), snare (circle), high hat (rectangle)

- Wavelet-based features: Means and standard deviations of wavelet coef£cients in each subband [17]

An analysis of the classi£cation ability of each feature set was performed by training machine learning classi£ers as well as examining scatter plots of the corresponding data. The best single dimensional features were Zero-Crossing, Spectral Centroid and Rolloff. LPC and MFCC coef£cients performed better than the wavelet-based features. Figure 3 shows two-dimensional scatter plots of the two highest principal components of the LPC and MFCC multi-dimensional features. The three classes of interest are clearly separated visually. Classi£cation results are provided in section 4.

3.2. Rhythm Analysis

Audio drum loops are signi£cantly different from vocal *BeatBoxing* sounds. Although a method based on individual percussion sound identi£cation such as the one described in subsection 3.1 could also be utilized for audio drum loop analysis; our initial experiments in that direction showed that this is not the case.

The main reasons are: 1) audio drum loops, unlike vocal percussion, contain a large variety of different sound samples, and 2) there is signi£cant overlap in time between the individual drum sounds. Therefore, a different approach was followed in the analysis of drum loop sounds.

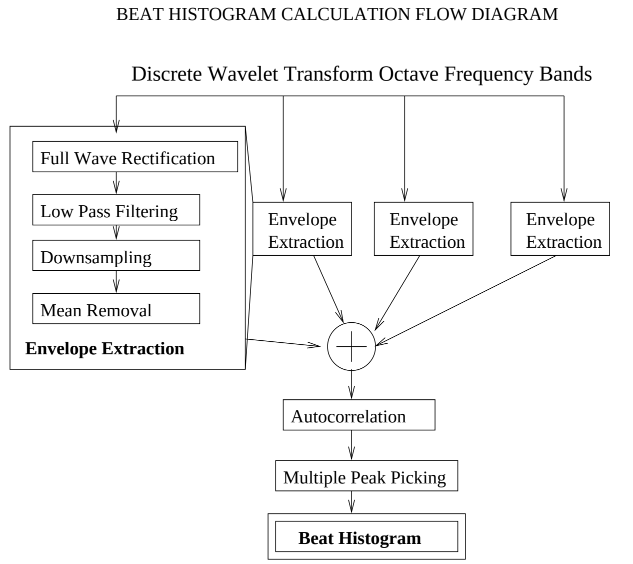

Figure 4. Beat Histogram Calculation Diagram

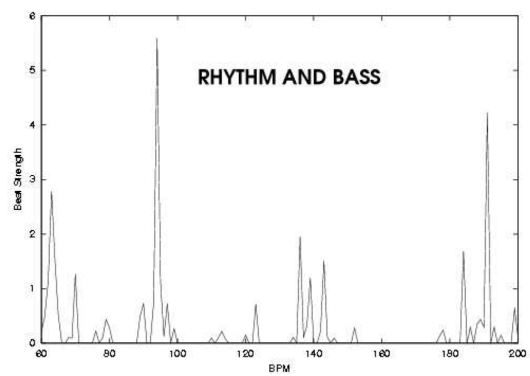

Figure 5. Beat Histogram

In order to analyze the drum loops, the signal is separated into different frequency bands using a Discrete Wavelet Transform (DWT). The envelope of each band is calculated using Full Wave Recti£cation, Low Pass Filtering and Normalization. This front-end is based on the method for the calculation of Beat Histograms described in [18]. The Beat Histogram (BH) shows the distribution of various beat periodicities of the signal. For example a piece with tempo 60 Beats-per-Minute (BPM) would exhibit BH peaks at 60 and 120 BPM (quarter and eight notes respectively). Figure 4 shows a schematic diagram of the this calculation. Figure 5 shows a BH for a piece of Rhythm and Blues music (notice the peaks at 96 BPM (main tempo) and 192 BPM).

Figure 6. Original signal followed by low and high wavelet bands showing the separation of bass drum sounds from high-hat sounds

The main peak of the BH (subject to some heuristics) is selected as the tempo of the beat boxing signal or the drum loop that is processed. This automatically detected tempo information is used in *Musescape* for live browsing of drum loops and dance music as described in section 5. In addition, several features characterizing the BH can be computed and used for subsequent analysis such as similarity retrieval and classi£cation. The BH features described in [18] are utilized in this paper.

In addition to calculating the BH, each subband of the DWT can be processed separately to identify the tracks of individual drum sounds. Figure 6 shows three waveforms displays of a audio drum loop. The top waveform is the original signal. The second waveform contains a low frequency subband of the wavelet decomposition and the third waveform contains a high frequency subband. It easy to see (and hear) that the low frequency band contains mostly the bass drum sounds and the high frequency band contains mostly the high-hat sounds. The advantages of using the subband approach for detecting the drum tracks include: handling of sound overlap (a high-hat sound that is played at the same time as the bass drum sound is still identi£ed) and that no classi£cation model based on speci£c sounds is utilized.

One obvious question is whether a similar wavelet analysis could be applied to the *BeatBoxing* signals. Indeed it can be used but the main reason we choose not to do so is that the wavelet analysis approach is more computationally intensive and doesn't give any better results than the individual sound identi£cation method. In a similar fashion to the query-by-humming approach, the processing of the query has to be fast but the targets (in this case the drum loops) can be preprocessed beforehand. Therefore query processing time is an important concern but target processing time is not as important.

3.3. Structural Representations - Matching

Once the vocal percussion signals and drum loops have been analyzed then we would like to develop methods for content-based similarity retrieval and classi£cation. In order to experiment with various algorithms the following three tasks were chosen: 1) retrieval using as query a drum loop at a different tempo from the one contained in the target database, 2) retrieval using as query a vocal rendition of a particular drum loop from a target database of audio drum loops, and 3) classi£cation of drum loops into 4 styles (described in section 4).

Our £rst attempt in that direction was using features computed using the BH representation proposed in [18]. Although, this approach works for music retrieval and musical genre classi£cation, the results were not particularly good for our task. We believe this is due to the fact that drum loop classi£cation requires more detailed information than the BHs provide. BHs are good at telling apart HipHop from Rock music but don't contain the detailed information required to identify or classify a particular drum pattern. Some results of style classi£cation of drum loops using features based on the BH are presented in section 4. The results are signi£cantly better than random but there is room for improvement.

Another approach that has been proposed in the literature [7] is the use of dynamic programming to time-align trajectories of feature vectors to detect similar drum patterns. Our initial experiments with this approach were not encouraging. We believe the main reason is that the spectral characteristics of vocal percussion sounds are very different from the characteristics of actual drum loop sounds. In addition, this approach suffers from the drawback of not directly handling the overlap of percussive sounds.

We are currently exploring the separate extraction of features on each band for classi£cation and similarity retrieval. Preliminary results are encouraging but a full scale evaluation hasn't yet been conducted.

4. EXPERIMENTS

4.1. Data collection

For *BeatBoxing* vocal hit identi£cation, a total of 75 sound£les were recorded by two different "beatboxers": 25 vocal bass drums, 25 vocal snare drums and 25 vocal high hats. For retrieval experiments, we created a database of 200 sound£les of four genres of dance music, typically played by DJ's: Drum & Bass (DnB), House, Rhythm & Blues (RnB), and Reggae (Dub). These sound£les were obtained using pre-recorded loops from *Dr. Rex Drum Sequencer* in *Reason* [3]. Each of the 100 loops chosen for the experiments has a default tempo at which it is normally played. For each of the four genre's, 25 samples of loops at the default tempo were recorded, as well as 25 samples of a time stretched or shrunk version at 120 BPM to use for tempo-invariant recognition analysis. These £les were also recorded at 44100 Hz.

[3] `http://www.propellerheads.se` (March 2004)

	zcr	spc	spr	lpc	mfcc
Vocal Bass Drum	100	100	92	100	88
Vocal Snare Drum	100	96	92	100	88
Vocal High Hat	92	88	96	88	92
Overall	97.3	94.7	93.3	96	89.3

Table 1. Percentages of classi£cation accuracy for *BeatBoxing* sounds (zcr = ZeroCrossing, spc,r = Spectral Centroid,Rolloff)

Furthermore, two professional *BeatBoxers* performed 12 selected beats (3 for each genre). Two versions for each beat were recorded: (1) listening with headphones to the corresponding *Dr. Rex* loop at default tempo and recording the performance, and (2) performing a memorized beat without any metronome.

All the voice recordings were recorded using an AKG C1000 microphone into a *Protools DIGI 002* sequencer at a sampling rate of 44100 Hz. All £les were normalized before analysis and experimentation.

4.2. Classi£cation

Table 1 shows some representative classi£cation percentage accuracy results for the identi£cation of individual vocal *BeatBoxing* sounds. These results are calculated using backpropagation Arti£cial Neural Network (ANN) using leave-one-out cross-validation. The best single dimensional feature was number of ZeroCrossings (zcr) and the best multi-dimensional feature set were the Linear Prediction Coef£cients (lpc). The fact that a single feature is so good at discriminating these sounds enables ef£cient realtime implementation for the applications described in section 5. We also experimented with a variety of other features and parameters but the results are not signi£cantly different.

One of the common ways to test the effectiveness of a feature set for describing musical content is style/genre classi£cation experiments. Although, ultimately our goal is to have a feature representation that is useful for drum loop retrieval, evaluating such a feature set directly requires extensive user studies to obtain relevance values. On the other hand ground truth for style classi£cation (although fuzzy even for humans) can be obtained easily. To make sure that the results are based on beat patterns rather than tempo information all the drum loops were generated at 120 beats-per-minute (bpm). Although this constraint probably underestimates the true classi£cation accuracy as tempo information can be an important cue, we wanted to make sure the results were purely based on the drum pattern characteristics.

Table 2 shows the classi£cation accuracy percentage for style identi£cation using drum loops at the same tempo. Four styles were considered: *Dub, Drum & Bass, House, and Rhythm & Blues*. The following classi£ers were compared: a Naive Bayes classi£er (BAYES), a backpropagation Arti£cial Neural Network (ANN), a Support Vector Machine (SMO) and a nearest neighbor classi£er (NN). More details about these classi£ers can be found in [19, 20]. All the results were calculated using 10-fold cross-validation to ensure that the accuracy is not in¤uenced by any particular partitioning of the labeled data into training and testing.

	RND	BAYES	ANN	SMO	NN	HUM
4st	25	44	49	55	44	70
3st	33	65	71	71	65	-

Table 2. Percentages of style classi£cation accuracy for drum loops (Dub, Drum & Bass, House, Rhythm & Blues), st is styles

	DUB	DNB	HSE	RNB
DUB	21	2	2	0
DNB	3	20	0	2
HSE	7	5	13	0
RNB	8	14	2	1

Table 3. Confusion matrix for SMO classi£er

In order to put these results into context an informal user study on style classi£cation was conducted. Two subjects listened to randomly chosen drum loops and had to identify the style. Both subjects were musically trained and one had more experience with dance music and drum loops. Both subjects achieved 70% classi£cation accuracy. As can be seen the automatic results are signi£cantly better than random classi£cation but still fall short of the human classi£cation so there is room for improvement.

It was observed that most errors for both human and computer were related to *Rhythm & Blues* drum loops. This can also be observed in the confusion matrix shown on Table 3. The diagonal of the confusion matrix shows the correct style identi£cation. For example the interpretation of the £rst row is that 21 out of 25 *Dub* (DUB) drum loops were correctly classi£ed, 2 were misclassi£ed as *Drum & Bass* (DNB) and 2 were misclassi£ed as *House* (HSE). Therefore on Table 2 we also show the results of removing *Rhythm & Blues* (RNB) drum loops from the dataset (3st). Both the automatic and informal user study results were done using drum loops at the same tempo (120 BPM).

Tempo information turns out to be an important problem in the classi£cation of drum loops. On the one hand, analysis algorithms have to be tempo invariant, on the other hand the main identifying characteristic of certain styles is their difference in average tempo. For example, Dub drum loops are below 100 bpm whereas Dnb loops are faster (140-150 bpm). We believe that addressing this tradeoff is critical but we haven't yet found a satisfactory way do so. In order to have tempo invariance and also include tempo information the only way we have tried is to include the tempo in the feature set. Unfortunately this approach doesn't work as well as we would like.

Another problem that the designer of audio analysis algorithms for *BeatBoxing* and drum loops has to deal with is the diffculty of evaluation. For example in order to evaluate *BeatBoxing* transcription or drum loop analysis extensive user annotations need to be provided as ground truth. In some cases these annotations can be extremely time consuming and therefore it is faster and more useful to just use the ear for qualitative evaluations. In this work we choose a combination of both approaches: whenever it was possible we conducted experiments and generated numbers but in many cases extensive parameter tuning and investigation of different features was done experimentally and subjectively.

5. IMPLEMENTATION-APPLICATIONS

A large variety of great software tools were used for this work. The feature extraction and classification were performed using Marsyas [4] a free software framework for audio analysis as well as Matlab. The Audacity [5] audio editor was also used. For some of the classification experiments the Weka [20] machine learning toolbox was utilized.

In addition, two prototype applications were developed to demonstrate the potential of Query-by-BeatBoxing. The *Bionic BeatBoxing Voice Processor* is the front-end to recording and analyzing vocal percussion. The analyzed signal can then be used to initialize *Musescape* which is a direct sonification tool for browsing music.

5.1. Bionic BeatBoxing Voice Processor

The *Bionic Beatbox Voice Processor* (BBVP) is a custom built GUI interface in MATLAB (shown in Figure 7) which allows a user to *BeatBox* into a microphone and use the interface to transform the voiced beat into a professional high quality drum loop using existing prerecorded audio samples. The voiced beat is parsed into individual vocal hits and compared to a user-specifc training set of data. Each vocal burst is classifed and the appropriate real drum sound is transplanted into the loop. The user has the ability to map any vocal sound to any WAV file sample which enables a variety of creative possibilities. This way we can alternate between *BeatBoxing* and drum loops easily. In addition, the interface can be used to evaluate the performance of different features for classification in a qualitative rather than quantitative way.

When 'record' is clicked, the software starts acquiring the audio input from the soundcard. The sampling rate of the data acquisition is fixed at 44100 Hz. To help the user stay in tempo, a click track can be generated.

The 'Process Beat' button trigger the transformation of the voice input into a real drum loop. First the time-domain signals are analyzed to find the start and end points of each individual *beatbox* sound burst. These points are used later to determine where to place the drum samples.

[4] `http://marsyas.sourceforge.net`
[5] `http://audacity.sourceforge.net`

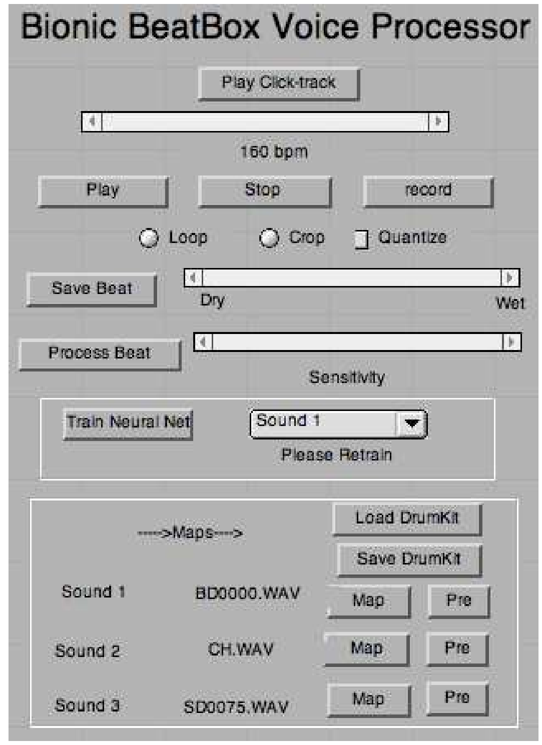

Figure 7. Bionic BeatBox Voice Processor Matlab GUI interface

The sound file is de-noised and thresholding is used to locate the voice bursts. The threshold can be adjusted using the "Sensitivity" slider to accommodate differences in background noise and magnitude of the *BeatBoxing* sounds.

Once the beat is parsed into bursts, a classification algorithm is used in order to identify each type of vocal hit. A back-propagation neural network based on a ZeroCrossings feature is used. The choice of this feature was based on the experiments described in section 4. Using a single feature allows quick results for this real time application. The user must "train" the neural net with 4 sounds for each type of vocal hit. Each sound must be performed five times, creating the necessary training data. After the vocal hits are identified, appropriate mappings can be made based on selected sound files containing individual drum samples.

After the beat is processed, and the appropriate identifed beats are mapped accordingly, the new enhanced beat is ready to be played. The user has a dry/wet mix option to hear the processed loop. If the slider is all the way dry when the 'Play' button is pressed, only the original voiced beatbox will be heard. If the slider is all the way wet, only the transformed beat will be played. The playback can also be infinitely looped with the 'loop' button. The analyzed information (tempo, features, individual drum sounds) can be saved for later use with other applications, such as *MuseScape* and the transformed query with the "real" drum sounds can be saved as a new audio file.

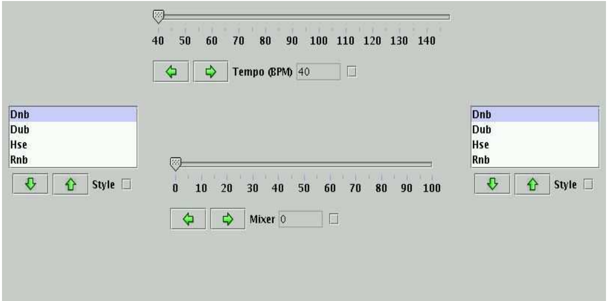

Figure 8. Musescape drum loop browser

5.2. Musescape

Musescape is a direct soni£cation interface for browsing large collections of music. In most existing retrieval software the users £rst adjust the parameters of their query, then click a "Submit" button and a playlist of relevant results is returned. In contrast, the main idea in *Musescape* is to provide continuous aural feedback that corresponds directly to the actions of the user (direct soni£cation). For example, when the user sets the tempo to 120 beats-per-minute (bpm) and selects the Dub style there is immediate feedback about what these values represent by hearing a corresponding drum loop. Sound is always playing and no "Submit/Search" button is used. Figure 8 shows a screenshot of *Musescape* used for the browsing of drum loops and *BeatBoxing* loops. A mixing slider can be used to cross-fade between different loops in a similar fashion to a DJ mixing console. The user can record a *BeatBoxing* loop using the *Bionic BeatBoxing Voice Processor* which is subsequently analyzed for tempo and style information as described above. The extracted tempo/style information is then utilized to initialize *Musescape* to a particular region of the drum loop collection. More information about *Musescape* and an music browsing evaluation user study can be found in [21]. A position paper arguing for the use of alternative interfaces to the typical Query-by-Example paradigm for MIR is [22].

6. DISCUSSION

In this paper, the design and development of a Query-by-BeatBoxing system was presented. More speci£cally we describe techniques for solving the following subtasks: vocal percussion sound identi£cation, drum loop analysis and style classi£cation. Experimental results showing the potential of the proposed algorithms are provided. In addition, two user interfaces for experimentation and prototyping were developed. The *Bionic BeatBoxing Voice Processor* is used to analyze vocal percussion signals and map them to audio drum sounds in order to create drum loops. It can also be used as a front-end to *Musescape* which is a direct soni£cation audio browsing environment. We believe our work, demonstrates the great potential of using rhythm and in particular *BeatBoxing* for music information retrieval.

DJs are a particularly good target user group as they are very knowledgeable about music and are interested in the use of new technologies. In some ways, even before this work, they are prime examples of music information retrieval users. In our opinion, research in music information retrieval has until recently emphasized melodic and timbral aspects. We hope that this paper will inspire more work in exploring rhythm as a retrieval mechanism.

There are numerous directions for future research. One direction is collecting more data from multiple *BeatBoxers* performing more than the 4 styles we explored. Such a study would aid in validating our existing results. User studies of DJs using the system in live performance situations are planned for the future. The initial response of a few DJs we have shown the system has been positive.

We believe that similar techniques can be used for beat retrieval of Indian music, especially tabla *theka's* (cycles) [9] and we are planning to explore that direction. In general, the use of MIR techniques in live performance is of particular interest. The development of domain speci£c query methods and retrieval systems is another goal for the future of MIR which until now has mainly concentrated on western art and popular music.

One of the most unexplored and challenging aspects of this work is the similarity of beat patterns by humans. Although we have some intuitive understanding of the process, more detailed experimentation with human subjects is required. The tradeoff of using tempo information or not is a typical example were our knowledge of how human perception works is incomplete. *Musescape* is a perfect tool to collect relevance and similarity information just by logging user interactions with the system. For example it is easy to explore how long subjects remember a particular rhythm and which rhythms are similar.

Another important direction is the exploration feature extraction based on each seperate subband of the wavelet analysis. We believe that high level structural representations of rhythm patterns are essential for this task and there is a lot of future work to be done in this area. There is a large legacy in rhythm analysis and representations for the analysis of symbolic data [23] which we would like to connect with automatic audio analysis method such as the ones described in this paper.

We hope, that one day MIR techniques will be as indispensable to DJs as records and turntables are today.

Acknowledgments

We would like to thank Joel Fieber for his help in implementing the *Bionic BeatBox Voice Processor* and Andreye Ermolinskyi for his help with *Musescape*. Many thanks to Adam Tindale for his help in implementing Neural Networks and feature extraction in Matlab. Thanks to Andrew Schloss, Peter Driessen and Wu-Sheng Lu for technical support and discussions.

7. REFERENCES

[1] Patten J., B. Recht, and H. Ishii, "Audiopad: A tag-based interface for musical performance," *Proc. New Interfaces for Musical Expression (NIME)*, Dublin, Ireland, 2002.

[2] Newton-Dunn H., H. Nakono, and J. Gibson, "Block jam," *(abstract) SIGGRAPH*, Tuscon, Arizona, 2002.

[3] Anderson T.H., "Mixxx: towards novel DJ interfaces," *Proc. New Interfaces for Musical Expression (NIME)*, Montreal, Canada, 2003.

[4] Goto M. and Y. Muraoka, "Real-time rhythm tracking for drumless audio signals - chord change detection for musical decisions," *Proc. Int. Joint. Conf. in Arti£cial Intelligence: Workshop on Computational Auditory Scene Analysis*, 1997.

[5] Scheirer E., "Tempo and beat analysis of acoustic musical signals," *Journal of the .Acoustical Society of America*, vol. 103, no. 1, pp. 588,601, Jan. 1998.

[6] Dixon S., E. Pampalk, and G. Widmer, "Classi£cation of dance music by periodicity patterns," *Proc. Int. Conf. on Music Informaiton Retrieval (ISMIR)*, Baltimore, USA, 2003, pp. 159–167.

[7] Paulus J. and A. Klapuri, "Measuring the Similarity of Rhythmic Patterns," *Proc. Int. Conf. on Music Information Rertieval (ISMIR)*, Paris, France, 2002, pp. 150–157.

[8] Gouyon F., F. Pachet, and O. Delerue, "On the use of zero-crossing rate for an application of classi£cation of percussive sounds," *Proc. COST-G6 Conf. on Digital Audio Effects (DAFX)*, Verona, Italy, Dec. 2000.

[9] Gillet O. and G. Richard, "Automatic labelling of tabla symbols," *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, Baltimore, USA, 2003, pp. 117–125.

[10] Paws S., "CubyHum: A fully operational QBH system," *Proc. Int. Conf. on Music Information Rertieval (ISMIR)*, Baltimore, USA, 2003, pp. 180–197.

[11] Dannenberg R. and et al., "The MUSART testbed for query-by-humming evaluation," *Proc. Int. Conf. on Music Information Rertieval (ISMIR)*, Baltimore, USA, 2003, pp. 41–51.

[12] Rauber A., E. Pampalk, and D. Merkl, "Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structure of Music by Sound Similarity," *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, Paris, France, Oct. 2002, pp. 71–80.

[13] Cano P., M. Kaltenbrunner, F. Gouyon, and E. Batlle, "On the use of FastMap for audio retrieval and browsing," *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, Paris, France, 2002, pp. 275–276.

[14] Fernstrom M. and E. Brazil, "Sonic Browsing: an auditory tool for multimedia asset management," *Proc. Int. Conf. on Auditory Display (ICAD)*, Espoo, Finland, July 2001.

[15] Davis S. and P. Mermelstein, "Experiments in syllable-based recognition of continuous speech," *IEEE Transcactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357–366, Aug. 1980.

[16] Makhoul J., "Linear prediction: A tutorial overview," *Proceedings of the IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[17] Guohui L. and A. Khokar, "Content-based indexing and retrieval of audio data using wavelets," *Int. Conf. on Multimedia and Expo (II)*. IEEE, 2000, pp. 885–888.

[18] Tzanetakis G. and P. Cook, "Musical Genre Classi£cation of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, July 2002.

[19] Duda R., P. Hart, and D. Stork, *Pattern classi£cation*, John Wiley & Sons, New York, 2000.

[20] Witten I.H and E. Frank, *Data Mining: Practical machine learning tools and techniques with Java Implementations*, Morgan Kaufmann, 1999.

[21] Tzanetakis G., "MUSESCAPE: An interactive content-aware music browser," *Proc. Conference on Digital Audio Effects (DAFX)*, London, UK, 2003.

[22] Tzanetakis G., A. Ermolinskyi, and P. Cook, "Beyond the Query-by-Example Paradigm: New Query Interfaces for Music Information Retrieval," *Proc. Int. Computer Music Conference (ICMC)*. Gothenburg, Sweden, Sept. 2002.

[23] Honing H., "From time to time: the representation of timing and tempo," *Computer Music Journal*, vol. 35, no. 3, pp. 50–61, 2001.