SIEVE: A PLUGIN FOR THE AUTOMATIC CLASSIFICATION AND INTELLIGENT BROWSING OF KICK AND SNARE SAMPLES

Jordie Shier, Kirk McNally and George Tzanetakis

University of Victoria
Victoria, British Columbia
{jshier,kmcnally,gtzan}@uvic.ca

ABSTRACT

The use of electronic drum samples is widespread in contemporary music productions, with music producers having an unprecedented number of samples available to them. To be efficient, users of these large collections require new tools to assist them in sorting, selection and auditioning tasks. This paper presents a new plugin for working with a large collection of kick and snare samples within a music production context. A database of 4230 kick and snare samples, representing 250 individual electronic drum machines are analyzed by segmenting the audio samples into different sample lengths and characterizing these segments using audio feature analysis. The resulting multidimensional feature space is reduced using principle component analysis (PCA). Samples are mapped to a 2D grid interface within an audio plug-in built using the JUCE software framework.

1. INTRODUCTION

The problem of automatic sorting, selection and auditioning for large sample library collections within a music production context has not been fully studied. The concept of visualization in music information retrieval (MIR) is comprehensively reviewed in [1], and updated to represent contemporary work in [2]. Studies relating to the specific case examined in this paper include [3], where the use of metric learning and kernelized sorting for arranging audio samples in two dimensions is explored using the Audio-quilt interface. In [4], the creative potential of similarity based, two dimensional visual browsing tools for drum samples is reported. Experimental 3D graphic tools for the exploration of mixed source sound collections are examined in [5]. Mixploration [6] is an example of an application using an experimental interface within a music production context. The application uses a self-organizing map algorithm for the visualization and manipulation of multiple audio mixing parameters using a 2D interface. Related analysis studies include [7], where unpitched percussive sounds, including a mix of acoustic and synthetic kick and snare drum sounds are analyzed. Acoustic kick and snare drum sounds are explored in [8], and sound similarity between electronic samples is a component of the experiments in [9]. In [10] the drum track is identified as being of particular importance in electronic dance music and an experimental system to aid users in the creation of this musical element is studied in this work.

The methodology proposed here utilizes various time segmentation methods as a pre-processing step to audio feature extraction. Principle component analysis and audio classification tasks are used to explore and evaluate the effect and efficacy of this approach. Using the results of this analysis, an audio plugin for sample browsing was created using the JUCE[1] platform. The plugin uses a 2D grid interface, which has been shown to be an effective design for this task [4] providing users with a visual representation of sample organization based on sound similarity. This application is intended to streamline the sample selection process and enhance studio creativity by allowing users to audition samples using grid-based music production tools such as the Ableton Push controller.

2. ANALYSIS

All audio samples are down-mixed to mono and resampled to a rate of 44.1kHz if required. A normalization step includes applying a ReplayGain of -6dB and an equal loudness filter. Time segmentation choices for this work are derived from [11], where a 23ms window is used to segment snare drum samples. A 50ms time window is used in [12] and it is reported in [8] that certain audio features can better characterize percussive sounds when using different sample lengths. In this work time segments of 25ms, 100ms and 250ms are used in the analysis. 500ms and full sample durations are also included for completeness. The starting position of a time segment is determined in relation to the signal envelope as the point in time when the signal reaches a threshold of its maximum value. Choices for this threshold are derived from the Essentia documentation[2] and include 20, 50, and 90 percent.

Audio feature extraction was performed using the Essentia library [13]. This library was selected based on the findings in [14], where it was found to be the most comprehensive library with regards to feature coverage. The features selected for use are from those defined both within the MPEG-7 format and prior work into the classification of percussion sounds. The features we used include Bark

[1] https://www.juce.com/
[2] www.essentia.upf.edu/documentation

bands, MFCCs, HFC, spectral and temporal features, which together constitutes a 133-dimensional feature-space. A 2048 sample Hann window using a hop-size of 1/8, derived from [8], is used for the STFT required for computation of all spectral features. Calculations for each feature using the 2048 sample window are summarized over time using mean and standard deviation.

Principal Component Analysis (PCA) is a technique that has been found useful in similar work [5], and is used here to reduce the dimensionality of the original feature space. In [15], PCA is used to help characterize multitrack music mixes and explore the most relevant features in terms of the variance. We use PCA here to characterize kick and snare samples. The principal components that result from PCA are a set of new axes that maximize the variance of the dataset such that the first axis contains the most variance. Using this, the 133-dimensional feature-space can be reduced down to two dimensions for visual plotting in a way that retains as much variance as possible. PCA was conducted independently on kick and snare drum samples for each time segmentation method. For each individual PCA, the null hypothesis was rejected using Bartlett's test of sphericity, calculated using the NumPy package [16]. All feature variables were standardized prior to PCA to have a mean equal to zero and unit standard deviation.

3. EXPERIMENTAL RESULTS

3.1. Audio Classification

Audio classification was used here to evaluate and compare the effect of the time segmentation choices. The classification tasks performed are sample type, drum machine and manufacturer classification. Three different classification algorithms implemented in Scikit-learn [17] are used: Support Vector Machine, Perceptron, and Random Forest. 10-fold cross-validation was used for each classification task and accuracy scores are calculated as an average between the three algorithms. ZeroR is used to determine the baseline accuracy for each task.

Sample type classification seeks to distinguish between kick and snare samples. All of the samples from the dataset were used and the baseline accuracy score was calculated to be 52.11%. The highest accuracy for kick and snare classification was 97.76 % and was achieved using a 250ms time segment positioned at 90% of the signal envelope.

Drum machine and manufacturer classification tasks performed within each sample class based on [7] are used to evaluate the ability of this analysis technique to characterize percussion samples of the same type.

For drum machine classification, machines were selected for kicks and snares separately such that each drum machine would have at least 50 samples for each type. Six distinct classes were used for kick drums which resulted in 464 kick samples in total and a baseline accuracy of 22.20%. Nine distinct classes were used for snare drums which resulted in 726 snare samples and a baseline accuracy of 14.88%. Classification performed best using mixed time segments for both kicks and snares, reporting 86% and 69% accuracy respectively.

Manufacturers were selected such that each manufacturer was represented by at least 100 samples of each type. The same six manufactures were used for the kicks and snares and are Alesis, Boss, E-MU, Korg, Roland, and Yamaha. For kick drums a total of 1329 samples were included reporting a baseline accuracy score of 39%. For snare drums a total of 1556 samples were used reporting a baseline accuracy of 33.10%. Results show that manufacturer classification was a more difficult task than the previous two tasks; kick drum classification reported 45% accuracy using a 500ms time segment at 20% of the signal envelope, and snares reported 46% accuracy using the full sample length.

3.2. Principal Component Analysis

Results of PCA give insight into how time segmentation effects variance and which features are most useful for characterizing kick and snare drum samples. Variance is maximized in the first two dimensions when using a 100ms window starting at 20% of the attack for kick sounds, and a 250ms window starting at 90% of the attack for snare sounds. The first two dimensions explain 31.65% and 32.79% of the variance for kick and snare drums respectively. Table 1 summarizes results for kick and snare drum PCA, and highlights the time segmentation method and corresponding contributing features that results in maximum variance in the first two dimensions. Dimensions three and four are also included to show how time segmentation effects the higher dimensions resulting from PCA and the variance that is lost during 2D plotting.

4. IMPLEMENTATION

The principle goal of the plugin is to provide users with a more efficient and intuitive method to search and audition drum samples within a music production environment. This is achieved by automatically organizing large sample collections based on sound similarity. A further goal was to have the plugin work in any of the mainstream music production environments and with standard hardware music production tools, such as the Ableton Push controller. A screenshot of the plugin user interface (UI) is shown in Figure 1.

4.1. Overview

Before drum samples can be auditioned, they must be selected from the users' existing sample library and analyzed. Each sample is automatically tagged as being a kick or snare based on the name of its source folder. The samples are then

Table 1: Principle Component Analysis: Variance Ratios

Length	Start	Kick					Snare				
		Dim 1	**Dim 2**	**Dim 3**	**Dim 4**	**1+2**	**Dim 1**	**Dim 2**	**Dim 3**	**Dim 4**	**1+2**
$25ms$	20%	17.95%	12.46%	9.44%	7.91%	30.41%	17.11%	13.85%[5]	9.51%	5.48%	30.97%
$25ms$	50%	17.64%	11.93%	9.52%	7.65%	29.57%	18.08%	13.73%	9.39%	5.37%	31.81%
$25ms$	90%	15.69%	11.53%	9.60%	7.76%	27.21%	19.38%	12.81%	8.79%	5.36%	32.18%
$100ms$	20%	17.30%	14.35%	9.43%	7.80%	31.65%[3]	20.16%	11.03%	9.85%	5.93%	31.19%
$100ms$	50%	16.72%	13.65%	9.41%	8.22%	30.37%	20.75%	10.75%	9.77%	5.95%	31.32%
$100ms$	90%	15.62%	12.45%	10.57%	8.70%	28.08%	21.22%	10.5%	9.37%	6.26%	31.37%
$250ms$	20%	17.01%	14.40%[2]	8.95%	8.18%	31.41%	21.22%	10.73%	9.22%	6.52%	31.95%
$250ms$	50%	16.48%	13.83%	9.16%	8.16%	30.30%	21.70%	10.54%	9.08%	6.52%	32.24%
$250ms$	90%	15.31%	12.92%	9.91%	8.79%	28.23%	22.75%[4]	10.04%	8.89%	6.63%	32.79%[6]
$500ms$	20%	17.16%	13.52%	8.87%	7.83%	30.68%	21.43%	10.19%	9.15%	6.94%	31.62%
$500ms$	50%	16.52%	13.10%	8.92%	8.06%	29.63%	21.86%	10.01%	9.03%	6.97%	31.87%
$500ms$	90%	15.16%	12.70%	9.47%	8.71%	27.86%	22.70%	9.69%	8.71%	7.01%	32.39%
Full	0%	18.03%[1]	13.44%	9.07%	7.37%	31.47%	21.01%	10.38%	9.15%	7.08%	31.39%

Main contributing features resulting in maximum variance in the highlighted dimension:
[1] **Dim 1:** HFC, HFC Std Dev, Mid-High Spectral Energyband
[2] **Dim 2:** Spectral Flatness dB, Spectral Centroid, Spectral Kurtosis
[3] **Dim 1:** HFC, HFC Std Dev, High Spectral Energyband; **Dim 2:** MFCC Band 2, Mid-Low Spectral Energyband Std Dev, Mid Low Spectral Energyband
[4] **Dim 1:** Spectral Energy, Bark Band 18 and 19
[5] **Dim 2:** Spectral Decrease, Spectral Decrease Std Dev, Spectral RMS
[6] **Dim 1**:Spectral Energy, Bark Band 18 and 19 **Dim 2**: Bark Spread Std Dev, Zero Crossing Rate Std Dev, MFCC Band 5

analyzed using the results from the analysis in Section 3. Different time segmentation choices are used for kick and snare drums to ensure that variance is maximized in the first two dimensions after PCA. Once the samples are loaded and analyzed, the task of browsing and auditioning can be performed using the grid interface.

The 2D interface uses an 8x8 grid of squares, which is a common configuration found on music production hardware interfaces. Samples are arranged on the grid using the first two dimensions resulting from PCA. Each grid square contains one sample and auditioning of that sample is performed by clicking on the UI component for that grid square, or by triggering a MIDI note from a connected hardware interface. In the situation where the selected library of samples exceeds the maximum number of grid squares available, each grid square will then represent a subset of drum samples evaluated as being perceptually close to one another. The number of samples in a subset corresponds to the total number of samples divided by number of grid squares. The sample used to represent the subset, and loaded to the grid square for auditioning purposes, is found by calculating the mean of dimension 1 and 2 from PCA for the given subset.

To explore an area of the sample library and corresponding 2D feature space, a user auditions a sample on the UI grid and can then select "zoom-in". This moves the last selected sample to the centre of a new grid layer, where samples from the previously described subsets are redistributed and displayed. In this way the feature-space can be explored at a high-level initially and then iteratively by zooming in until a drum sample that meets the users satisfaction has been found. Once a drum sample has been found, a click-and-drag operation can be used to move that sample into the desired application for use. The overview of the plugin browsing and navigation is shown in Figure 2.

4.2. Architecture

The JUCE C++ software framework was used to develop the plugin. JUCE takes care of many of the technical considerations required when developing an audio plugin and allows for developers to quickly start writing audio processing code. JUCE also provides a number of graphical user interface component classes which were used for developing the 2D interface. A SQLite[3] database was implemented to manage the samples that had been loaded, the results of the feature-extraction and PCA. Essentia is used for the feature extraction and computation of principal component analysis.

4.3. Reproducibility

The authors welcome any feedback and contributions on the GitHub page [4] in accordance with the recommendations for open access and reproducibility in signal processing research presented in [18]. The dataset is also available upon request.

5. CONCLUSION

The Sieve plugin is a unique new tool for music producers working with large libraries of kick and snare samples

[3] `https://www.sqlite.org/`
[4] `https://github.com/jorshi/sieve`

Figure 1: *Sieve Plugin UI*. Sample type is selected using the drop-down menu and "load samples" button. Samples are auditioned by clicking on the pads of the 2D grid. Areas of the grid can be further explored using the "zoom in" button.

within contemporary music software environments. The plugin automatically organizes samples according to similarity based on audio analysis and offers an efficient method for the browsing and auditioning of samples. Through the development of the plugin framework, it was found that using shorter time segments of audio for analysis resulted in an improvement in a majority of the kick and snare classification tasks tested. Future work includes perceptual testing to determine whether the time segmentation methods used as a pre-processing step lead to more perceptually relevant characterization of audio samples. Exploring alternative methods to dimensionality reduction may also serve as potential future work. The self organizing map used in [6] may be beneficial for further development of this work.

6. REFERENCES

[1] M. Cooper, J. Foote, E. Pampalk, and G. Tzanetakis, "Visualization in audio-based music information retrieval," *Computer Music Journal*, vol. 30, no. 2, pp. 42–62, 2006.

[2] M. Schedl, E. Gómez, J. Urbano, *et al.*, "Music information retrieval: Recent developments and applications," *Foundations and Trends® in Information Retrieval*, vol. 8, no. 2–3, pp. 127–261, 2014.

[3] O. Fried, Z. Jin, R. Oda, and A. Finkelstein, "Audioquilt: 2D arrangements of audio samples using metric learning and kernelized sorting.," in *New Interfaces for Musical Expression*, pp. 281–286, 2014.

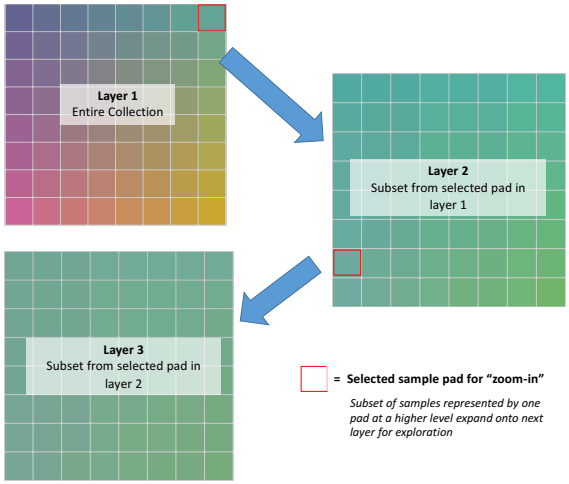

Figure 2: *Sieve Browsing Overview*. When there are more samples of a particular type than grid squares, the entire sample collection is represented using multiple levels. The highest level (Layer 1) uses a single square to represent a subset of the sample collection that is similar. This subset can be explored by iteratively "zooming" into that square.

[4] C. Turquois, M. Hermant, D. Gómez-Marín, and S. Jordà, "Exploring the benefits of 2D visualizations for drum samples retrieval," in *2016 ACM on Conference on Human Information Interaction and Retrieval*, pp. 329–332, ACM, 2016.

[5] G. Tzanetakis and P. Cook, "3D graphics tools for sound collections," in *Conference on Digital Audio Effects, DAFX*, 2000.

[6] M. Cartwright, B. Pardo, and J. Reiss, "Mixploration: Rethinking the audio mixer interface," in *The 19th international conference on Intelligent User Interfaces*, pp. 365–370, ACM, 2014.

[7] P. Herrera, A. Dehamel, and F. Gouyon, "Automatic labeling of unpitched percussion sounds," in *Audio Engineering Society Convention 114*, Audio Engineering Society, 2003.

[8] E. Pampalk, P. Herrera, and M. Goto, "Computational models of similarity for drum samples," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 408–423, 2008.

[9] F. Tutzer, *Drum rhythm retrieval based on rhythm and sound similarity*. Universitat Pompeu Fabra, Barcelona, 2011.

[10] R. Vogl, M. Leimeister, C. Ó. Nuanáin, S. Jorda, M. Hlatky, and P. Knees, "An intelligent interface for drum pattern variation and comparative evaluation of algorithms," *Journal of the Audio Engineering Society*, vol. 64, no. 7/8, pp. 503–513, 2016.

[11] A. Danielsen, C. H. Waadeland, H. G. Sundt, and M. A. Witek, "Effects of instructed timing and tempo on snare drum sound in drum kit performance," *Journal of the Acoustical Society of America*, vol. 138, no. 4, pp. 2301–2316, 2015.

[12] J. P. Bello *et al.*, "A tutorial on onset detection in music signals," *IEEE Transactions on speech and audio processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[13] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: an open-source library for sound and music analysis," in *21st ACM International Conference on Multimedia*, pp. 855–858, ACM, 2013.

[14] D. Moffat, D. Ronan, J. D. Reiss, *et al.*, "An evaluation of audio feature extraction toolboxes," in *18th International Conference on Digital Audio Effects (DAFx-15), Trondheim, Norway*, 2015.

[15] A. Wilson and B. Fazenda, "Variation in multitrack mixes: analysis of low-level audio signal features," *Journal of the Audio Engineering Society*, vol. 64, no. 7/8, pp. 466–473, 2016.

[16] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[18] P. Vandewalle, J. Kovacevic, and M. Vetterli, "Reproducible research in signal processing," *IEEE Signal Processing Magazine*, vol. 26, no. 3, 2009.