MUSED: NAVIGATING THE PERSONAL SAMPLE LIBRARY

Graham Coleman
Atlanta, GA
ravelite@gmail.com

ABSTRACT

This work outlines the concept of *personal sample library*, a large database of event-synchronous audio segments extracted from a user's digital music collection. It identifies tasks involved in making sample-based music and audio collage, and shows how interfaces to the personal sample library can aid these tasks and simplify the music development cycle. In this experiment, we implement an interactive scatter plot with *dynamic queries* to browse, discover, and select material from our database of samples.

1. INTRODUCTION

Since the advent of modern recording and reproduction techniques in the early twentieth century, the practice of *sampling*—including actual sounds extracted from previous recordings in new compositions—has become an important tool in music composition. Starting with *musique concrete* in the late 1940s by Pierre Schaeffer, magnetic tapes were cut, pasted, and spliced to compose sounds from disparate settings into a new context. Sampling has since become an integral part of many genres, including hip-hop, popular, and computer music.

Sample-based music composition follows a different pattern than traditional music composition, though most likely current working processes are a hybrid of these patterns. While a traditional composer must invent a melody, progression, lyrics, and so on, the collage composer primarily performs operations on the material available. This process can involve potentially complex music theory, but conservatory training is not strictly necessary—there is a distinct skill set of which sample-composers, DJs, mash-up artists, and remix artists make use. We suggest a task taxonomy for working with samples in the composition process.

The composer first *selects* material to sample. Then she must *prepare* it by choosing where to begin and end the sample. Certain beginning and ending locations are better than others, for both technical and aesthetic reasons. Additional preparation may include aligning the sample with a tempo grid. An optional step is to *store* the sample for future use, which may mean saving it to a certain folder, or a more complex organizational scheme like tagging. She can choose to *transform* the sample by applying effects, for example time shift, pitch shift, or filters. Finally, she *composes* the sample into the context of the composition, by arranging it with respect to traditional musical material or the other samples in the piece. In a non-destructive audio editor, the latter steps of this process can proceed in any order with iterated and continued refinement.

We focus on the first two steps of this process. Current methods of *selection* and *preparation* involve equal measure of search and manual labor. To select material, the composer decides which piece to sample and identifies sections that she may want to use. As this requires a conceptual leap to choose between a large number of potential samples (which are not yet realized), this step may present a paradox of choice, or even a touch of existential angst. A basic strategy involves listening to source tracks in real time, or scrubbing through linearly in search of interesting sounds. Upon selection the composer loads the sample into a traditional destructive audio editor, isolates the material, and either cuts or preprocesses the material, usually taking care to avoid discontinuities that would results in clicks or pops. Optionally, she can apply a *transformation* to bake into the sample, and *stores* it for future retrieval.

One possible measure of the complexity of a collage piece is the number of different samples to compose it, the *sample count*. Given the time and work to select and prepare each sample, a complex piece with a high sample count can be arduous to compose. A collage piece with samples from a variety of source material, with a high *source variety*, would also require a long time selecting material from each source one at a time. While we believe works of art take time and effort, by reducing the cost of these primitive operations we can bring up the quality and quantity of collage music in the world. In this work, we focus on improving the selection process, making use of previous work in automating preparation.

MIR, or music information retrieval, is a recent interdisciplinary research field which deals with extracting information out of music (key, pitch, mood, similarities between artist, song, or album) and organizing it into a representation that supports needed tasks (searching, browsing, details) from its users, be they music creators, librarians, or consumers. Of particular interest from MIR and digital signal processing are *onset detection* methods that determine where salient events in a piece of music occur.

We propose a new MIR representation which will be used directly in the music making process. By employing onset detection methods to automatically segment songs, we can induce a *personal sample library* over a collection of recorded music tracks. Since each track contains thousands of salient (and potentially sample-able) musical

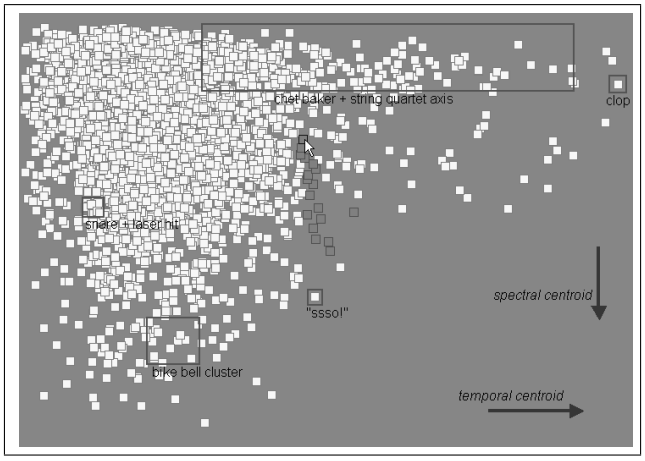

Figure 1. Browsing a sample library of about 4400 segments extracted from 5 songs. Segments are plotted by spectral and temporal centroid.

events, this database is an order of magnitude larger than its parent database, the *personal music library*. Figure 1 gives a glimpse of a small portion of such a library.

Information visualization is the discipline of making interfaces that make allow exploration and intuitive understanding of information, particularly large databases. In this work, we employ techniques from this field, *direct manipulation* and *dynamic queries*, to allow a musician to browse and navigate the personal sample library.

2. RELATED WORK

Previous systems that address the problem of browsing large music databases include Fernstrom [6] and Tzanetakis [14].

Several previous works algorithmically compose bits of sound into new structures. Musical Mosaicing [16] employs a system of high-level constraints for sequencing uniform overlapping segments from a database to match a given target, much like image and photo mosaics. Jehan [8] suggests event-synchronous segmentation and provides an automatic method of sample segmentation and sample preparation, which is used by this work. In his thesis *Creating Music by Listening* [9], he demonstrates a number of algorithmic music processes that use the representation, including cross-synthesis, compression by time-redundancy, and generation of new sample compositions by imitating segment transition statistics.

Schwarz [13] surveys the early work, much of which focuses on autonomous and algorithmic music. However, if we wish to augment music makers or even just empower more casual music users, we must create interfaces to collage where the user is at the center.

For live and interactive mosaicing, Lazier and Cook [11] and Aucouturier and Pachet [3] provide strategies for selection in real time collage. This work can potentially provide a UI counterpart to this domain.

Existing interfaces to user centered collage are being developed and refined. Giving casual users more control over the composition, Freeman's iTunes Signature Generator [7] mines a user's iTunes library to select preferred songs, based on play counts or ratings. By comparing segment features it finds sections that blend well together. Finally, it organizes sections into a short composition the user can download and play for their friends.

Zadel and Scavone [15] provide both a survey of current tools in *laptop music*, and a novel visual tool for making live sample-based music.

Tapestrea [12], a recent tool designed for sample-based composition of *soundscapes*, supports the task of *preparation* by allowing the user to decompose sounds into sinusoids, transients, and stochastic noise, as well as isolate sounds in frequency and time. It also aids the *composition* task by providing interface elements for triggering material by chance, grouping, and timing operations.

Still, composers and music lovers could benefit further from tools that facilitate the selection process and transition seamlessly into organizing the source material.

3. FEATURE EXTRACTION

3.1. Onset Detection as Segmentation

Onset detection methods are designed to find significant events in an audio stream. We can then segment the events into samples with some additional processing.

Several good methods for onset detection exist. Bello et al [4] provide an in-depth tutorial to a few approaches. Interesting approaches include Jehan [8], which defines an *auditory detection function*, which warps the magnitude spectrum into the Bark scale and accounts for both temporal and spectral masking. Duxbury et al [5] provide a measure of complex spectral flux which uses a linear predictor for phase to differentiate between transients and steady state regions.

Onset detection typically includes some level of error, such as spurious onsets, or an unbroken sequence of several human- percieved onsets. Even a perfect temporal segmentation will contain the tails of overlapping events, given the yet unsolved problem of source separation of polyphonic audio. Hopefully a good interface can help mitigate these errors, by allowing the user to manually subdivide segments, or to extend a segment with trailing segments. The latter would be essential for using longer segments in the *composition* task.

For our prototype we choose a deliberately simple onset detection method, which uses a detection function derived from the amplitude envelope. Implementors will prefer to use the current state of the art.

As in Jehan [8], after finding peaks with the onset detection method, we search backwards in each sample for a trough in the loudness function. Once we find that, we search backwards to find the nearest zero crossing. These steps prevent edit click transients.

To filter out spurious onsets, we *threshold* by storing the value of the filtered detection function at each peak, discarding low values. We create a new segment for each allowed onset and end the segment at the next onset.

3.2. Features for Short Segments

For navigating the space of samples, we wish to have perceptually relevant features that we can use to distinguish and select target samples.

Perhaps the most important classes of features to include for this purpose are timbre and pitch. For timbre features, we have implemented spectral and temporal centroid, as well as a loudness measure (avg. power). Timbre features we might add include: temporal moments of the amplitude envelope, spectral moments such as standard deviation, skew, kurtosis, energy band ratios, moments of the log-spectrum, measures of noisiness and inharmonicity, and mel-frequency cepstral coefficients(MFCCs). For pitch, we have implemented max(chroma) and could add f0 estimation.

Besides the features actually contained in short segments, we can also add *context features* from larger-time structures such as beat and tempo. For example, after estimating the tempo and beat phase for a song, we can add a *beat number* feature which identifies in which beat a segment falls. Or after identifying chorus sections from a song, we can add a feature positively identifying segments from the chorus.

Finally, we could add tags to allow the user to curate their sample library.

4. USER INTERFACE

Our *personal sample database* contains thousands of segments from each song that we include. For example, Paul Lamere, an avid music enthusiast and MIR researcher, has about 9,000 tracks in his digital music listening library [10]. That makes the resulting sample library on the order of 9 million segments, which may be unwieldy to make sense of using naive techniques.

So that we may create a useful interface for browsing this sample database, we will borrow results from information visualization, a UI discipline that specializes in making sense of large amounts of information. In our first prototype, we apply the philosophy of *direct manipulation* and the technique of *dynamic queries*. Direct manipulation interfaces support (as stated in [2]):

- continuous visual representation of objects and actions of interest
- physical actions or labeled button presses instead of complex syntax
- rapid, incremental, reversible operations whose impact on the object of interest is immediately visible *or audible*
- layered or spiraled approach to learning that permits usage with minimal knowledge

By encoding objects and actions visually, the user need not possess any special knowledge to start interacting with them. By providing graphic representations instead of artificial language representations, the user can quickly form an intuitive grasp of the domain. By giving *immediate feedback*, the user can understand the consequences of each small action in rapid succession. By allowing use with minimal knowledge, the user faces a lower barrier to entry and is thus encouraged to learn.

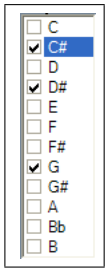

Figure 2. A filter for selecting on Pitch Class Profile

To facilitate sound selection, we provide an interactive scatter plot in which segments are visually represented in a location according to their attributes. For example, we plot segments by spectral and temporal centroid, so the low bassy sounds go to one side of the screen and the high breathy sounds to the other, while the quick sudden sounds go to one side and the sustained sounds to the other. We can also ornament the plot points with information from additional attributes, by varying the color, size, or shape of the points. The interactive scatter plot is additionally inspired by Spotfire [1].

When the user mouses over its scatter point, we play the complete audio of the segment. This provides immediate feedback to the user in the most accurate and most rapid way possible. It avoids the ambiguities inherent in visual representations of sound. Allowing several samples to be triggered in quick succession gives the user a sense of the segments faster than real-time, meaning in less time it would take to play all of them concatenated end-to-end with no breaks. This is similar to scrubbing.

4.1. Exploring with Dynamic Queries

Simply plotting a large database produces significant clutter. Strategies making sense of these plots can include:

- zooming in on results
- representing aggregates at high-level scale (semantic zoom)
- filtering the result dataset

The technique of dynamic queries [2] applies the direct manipulation strategy to database queries. The interface provides filter widgets for a number of attributes (or features), which specify a range or subset of points in the database. For example, on real-valued attributes such as loudness, we could use *range sliders*, sliders with two heads specifying min and max. For nominal data we could check boxes (as in figure 2), additive lists, or *alpha sliders* for attributes of larger cardinality. The result of a query is the intersection of all the restrictions on its attributes.

As with direct manipulation, feedback provided from dynamic queries should be immediate. This allows the

user to form queries incrementally for complex and arbitrary subsets of the database, and develop a sense for density in query regions by sight, and by brushing over results, a sense by sound.

5. IMPLEMENTATION

We implement a working prototype of the personal sample library in C# on Windows. Our frontend decodes an MP3 using the Bass library, segments it using simple onset detection, extracts features, making use of fftw, and stores the metadata and features for the segments in a SQLite database. We provide an interactive scatter plot which plots the segments by temporal and spectral centroids, and filters on attributes including max(chroma).

We have prepared a few screencasts of our research prototype in action. These can be found at:

`http://ravelite.org/mused/`

6. FUTURE WORK

A user study would help to evaluate this selection method. Measuring relative filter usage could provide an interesting basis by which to rate musical feature relevance.

Mused, or Music Editor, is a series of prototype interfaces for editing and composing digital music and audio. These interfaces will take into account musical structure that will be informed by MIR techniques, rather than the traditionally flat representation of audio.

7. CONCLUSION

We have presented a preliminary design for an Infovis-inspired browser over the *personal sample library*. We employ previous results in segmentation and feature extraction to populate a database with samples from a music library. To navigate the library, we provide an interactive scatter plot that supports *dynamic queries* to filter samples by musically relevant feature values. This interaction technique can potentially be applied fruitfully to other MIR applications, including non-induced sample or music libraries, and interactive collage.

In this work, we quicken the *selection* process in the creation of sample-based music. Doing so, we intend to facilitate sample-based music of greater quantity and complexity. In general, by reflecting more structure into the audio editing process (as provided by MIR methods), we can create music tools that are more immediate and facile.

8. ACKNOWLEDGEMENTS

Special thanks go to Parag Chordia (Georgia Tech) for critical oversight, and to Emily, Mark, and Alex for proofreading and suggestions.

9. REFERENCES

[1] C. Ahlberg. Spotfire: an information exploration environment. *ACM SIGMOD Record*, 25:25–29, 1996.

[2] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619 – 626, 1992.

[3] J. Aucouturier and F. Pachet. Jamming with plunderphonics: Interactive concatenative synthesis of music. *Journal of New Music Research*, 35:35–50, March 2006.

[4] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13:1035–1047, 2005.

[5] C. Duxbury, J. P. Bello, M. Davies, and M. Sandler. Complex domain onset detection for musical signals. *Proc. of the 6 thInt. Conference on Digital Audio Effects (DAFx-03)*, 2003.

[6] M. Fernstrom and E. Brazil. Sonic browsing: an auditory tool for multimedia asset management. *Proc. Int. Conf. on Auditory Display (ICAD)*, 2001.

[7] J. Freeman. Fast generation of audio signatures to describe iTunes libraries. *Journal of New Music Research*, 35:51–61, March 2006.

[8] T. Jehan. Event-synchronous music analysis/synthesis. *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-04)*, 2004.

[9] T. Jehan. *Creating Music by Listening*. PhD thesis, Massachusetts Institute of Technology, 2005.

[10] P. Lamere. Duke listens! : Weblog. blogs.sun.com/plamere/entry/my_personal_long_tail *accessed May 4, 2007.*

[11] A. Lazier and P. Cook. Mosievius: Feature driven interactive audio mosaicing. *Digital Audio Effects (DAFx)*, 2003.

[12] A. Misra, P. Cook, and G. Wang. Tapestrea: Sound scene modeling by example. *Proceedings of the International Computer Music Conference*, page 177, 2006.

[13] D. Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35:3–22, March 2006.

[14] G. Tzanetakis. Musescape: An interactive content-aware music browser. *Proc. Conference on Digital Audio Effects (DAFX)*, 2003.

[15] M. Zadel and G. Scavone. Laptop performance: Techniques, tools, and a new interface design. *Proceedings of the International Computer Music Conference*, pages 643–648, 2006.

[16] A. Zils and F. Pachet. Musical mosaicing. *Digital Audio Effects (DAFx)*, 2001.