CONCATENATIVE SYNTHESIS FOR NOVEL TIMBRAL CREATION

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

James Bilous

June 2016

COMMITTEE MEMBERSHIP

TITLE:	Concatenative Synthesis for Novel Timbral Creation
AUTHOR:	James Bilous
DATE SUBMITTED:	June 2016
COMMITTEE CHAIR:	John Clements, Ph.D. Assistant Professor of Computer Science
COMMITTEE MEMBER:	Chris Lupo, Ph.D. Associate Professor of Computer Science
COMMITTEE MEMBER:	Franz Kurfess, Ph.D. Professor of Computer Science

ABSTRACT

Concatenative Synthesis for Novel Timbral Creation

James Bilous

Modern day musicians rely on a variety of instruments for musical expression. Tones produced from electronic instruments have become almost as commonplace as those produced by traditional ones as evidenced by the plethora of artists who can be found composing and performing with nothing more than a personal computer. This desire to embrace technical innovation as a means to augment performance art has created a budding field in computer science that explores the creation and manipulation of sound for artistic purposes. One facet of this new frontier concerns timbral creation, or the development of new sounds with unique characteristics that can be wielded by the musician as a virtual instrument.

This thesis presents Timcat, a software system that can be used to create novel timbres from prerecorded audio. Various techniques for timbral feature extraction from short audio clips, or grains, are evaluated for use in timbral feature spaces. Clustering is performed on feature vectors in these spaces and groupings are recombined using concatenative synthesis techniques in order to form new instrument patches.

The results reveal that interesting timbres can be created using features extracted by both newly developed and existing signal analysis techniques, many common in other fields though not often applied to music audio signals. Several of the features employed also show high accuracy for instrument separation in randomly mixed tracks. Survey results demonstrate positive feedback concerning the timbres created by Timcat from electronic music composers, musicians, and music lovers alike.

ACKNOWLEDGMENTS

Thank you to my family, friends, and advisers who have provided strength, kindness and guidance when it was needed most.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The increase in popularity of personal computing has brought with it a new type of musician; one who relies on software to perform. These electronic artists use digital audio workstations (DAW) to write scores, play sampled instruments from external controllers, and even construct instrument sounds from scratch. This creation of new instrument sounds, or novel timbral creation, has been made possible by to the packaging of digital signal processing (DSP) techniques by talented software and audio engineers into plug-ins that offer intuitive interfaces. The desire for new types of plug-ins and methods to generate interesting timbres for use by electronic artists is likely fueled by a booming electronic music industry which represents a $7.1 billion market at the time of this writing, up 3.5% from the year before [54].

This thesis presents the software application Timcat, a collection of scripts written in the Python programming language that generate novel timbres from prerecorded audio for use as virtual instruments. Timcat approaches the problem of generating new types of sounds by analyzing existing audio on the microsound scale, a method inspired by the field of granular synthesis. The small audio segments are strategically grouped and faded together to produce the final output signals in the spirit of concatenative synthesis. In this work I focus on the evaluation of a handful of readily available DSP techniques, some with slight modifications, for use as timbral descriptors.

Chapter 2

DOMAIN SPECIFIC BACKGROUND

The following chapter first describes several important mechanics of sound and its properties. Then, a brief overview of signal processing is given before discussing spectral analysis techniques that reveal many useful aspects of the components of sound. Section 2.4 goes on to discuss aspects of the perceived qualities of sound called "timbre" that can be exploited for the purpose of analysis and comparison. The descriptors used in this work to represent various aspects of timbre are described in detail in 2.5. Finally, a machine learning algorithm used for vector quantization employed in this paper is covered in Section 2.6.

2.1 Sound Mechanics and Models

Sound is the sensation that arises in a perceiver due to a change in air pressure in their ear canal over time [51]. These changes in pressure propagate from a vibrating source via a medium such as air or water to a listener who *receives* the sound which is in turn *perceived* by their brain. In order to study the phenomenon that is sound it is common to start by constructing a model that facilitates its observation [67]. One simple model can be created by simply recording a sound using an instrument that detects pressure changes, such as an induction microphone, and stores a digital or analog representation of the signal[67]. Using signal processing techniques on such models allows the extraction of descriptors which represent properties of the sound which can be used for comparison with other sound descriptors, identification of sounds, or even reproduction of the pressure differentials that comprised the original signal.

Digital representations of sound are particularly useful models due to the speed with which signal processing can be performed on them by computers. Creating digital representations of sound is accomplished using a technique called pulse code modulation (PCM) which was originally developed by Bell Telephone Labs in the 50s and 60s for telephony technologies [53]. By *sampling* an audio signal at a given interval, voltage values are obtained which are then encoded as digital data and stored for further use or processing [53].

There are several decisions that must be made when digitally sampling an analog signal. First, a sampling period must be selected which involves a trade off between fidelity and storage requirements. An analog signal sampled with a higher sampling rate will better represent the original signal but will require more bytes to represent on disk. On the other hand, a sampling rate that is too low will miss changes in the signal that are caused by higher frequency components. The Nyquist Theorem states that the sampling rate of a signal must be at least twice the frequency of the highest frequency component of the target signal in order to properly represent it without loss of information. Because the range of human hearing is between 20 Hz and 20 kHz on average, sampling rates of above 40 kHz are often used [67]. Sampling rates for compact disks, for example, are usually 44.1 kHz which is adequate for the purposes of reproducing a sound meant for human perception [77].

2.2 Digital Signal Processing

Naturally occurring audio signals are produced by a *system* responding to a stimulus. For example, the drawing of a bow over a violin string causes the violin to respond by vibrating and reverberating in such a way as to produce its characteristic musical note. Similarly, vocal cords and the vocal tract are stimulated by air to produce a speech signal [45]. I

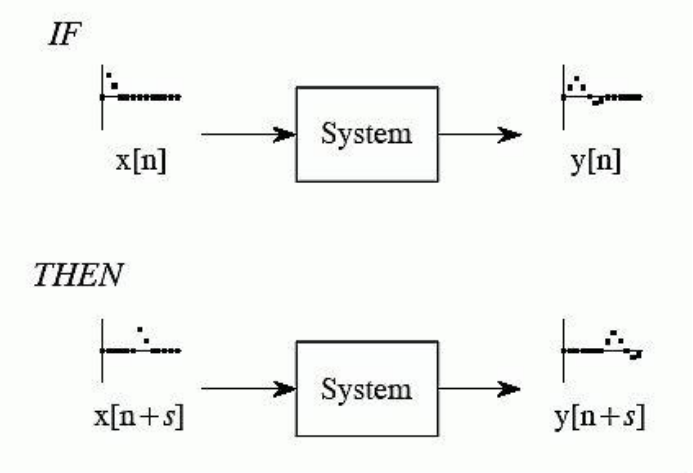


Figure 2.1: In a time or shift invariant system, shifting an input signal results in an identical shift in the output signal [69].

Systems may also be physical or software devices that perform operations on a signal [45]. Such a system could produce an output signal similar to the input but with reduced noise, or even with certain component frequencies of the original signal attenuated.

Systems that satisfy the additivity property, expressed in Equation **??**, as well as the homogeneity property, expressed in Equation **??** are said to be linear. An example of an operation by such a system is shown in Figure 2.2. Likewise, if a time delayed input to a system produces the same output as an undelayed input but shifted in time, then the system is considered *time invariant*, as shown in Figure 2.1.

$$F(x_1 + x_2) = F(x_1) + F(x_2) \tag{2.1}$$

$$F(ax) = aF(x) \tag{2.2}$$

The benefit of working with a system that is linear and time invariant (LTI) is that it can be decomposed into a weighted sum of unit responses to the system from which it originates. Most sound signals are no exception since they are comprised of

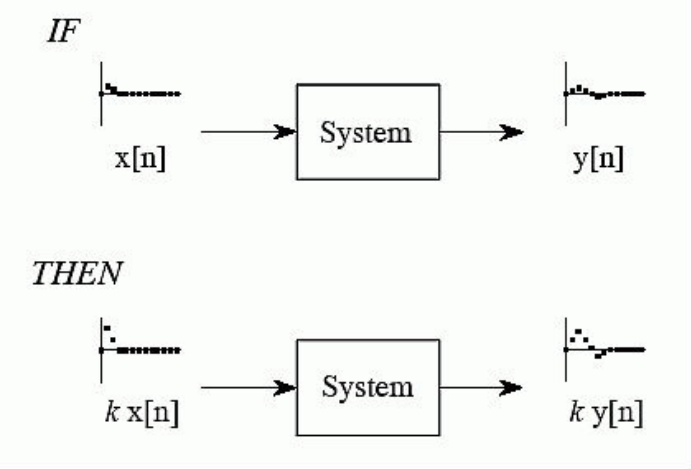


Figure 2.2: In a linear system, an amplitude change of the input signal results in an identical amplitude change in the output signal [69].

periodic perturbations of an LTI system. Performing operations on such systems is considered digital signal processing.

2.2.1 Discrete Time Fourier Transform

One mathematical tool in particular called the Fourier transform is exceptionally useful for decomposing any LTI system into its periodic components. Given an integrable function of time $f(t)$, its Fourier transform is defined by:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)\ e^{-2\pi i t \xi}\, dt, \tag{2.3}$$

However, digital signal processing deals with uniformly spaced discrete samples of a signal which are not suitable for Equation **??**. Instead, the discrete time Fourier transform is used, which transforms a set of N numbers x into a Fourier series of periodic functions that are a function of frequency given by Equation **??**. Equation **??** assumes an ω has units of radians per sample with a period of 2π.

$$X(\omega) = \sum_{n=0}^{N} x[n]\, e^{-i\omega n}. \tag{2.4}$$

2.3 Spectral Analysis

Spectral analysis in the context of sound involves the study of spectra obtained from short time segments of an audio signal. Julius Smith describes the motivation for analyzing short segments of a signal rather than the signal as a whole:

> In spectrum analysis of naturally occurring audio signals, we nearly always analyze a short segment of a signal, rather than the whole signal. This is the case for a variety of reasons. Perhaps most fundamentally, the *ear* similarly Fourier analyzes only a short segment of audio signals at a time (on the order of 10-20 ms worth). Therefore, to perform a spectrum analysis having time- and frequency-resolution comparable to human hearing, we must limit the time-window accordingly. [68]

The Fourier transform assumes a continuous, repeating signal is given as input which is often inconsistent with data samples extracted from a particular time window. Consider the signal segment from time $t = 4$ to $t = 8$ shown in Figure 2.3. When represented as a discrete time Fourier transform the signal segment will be interpreted as a single period of a signal that extends infinitely in time as shown in Figure 2.4.

The discontinuities in Figure 2.4 at $t = 0$, $t = 4$ and $t = 8$ appear in the output of the transform as high frequency components that were not present in the original signal. In order to remove these artifacts it is common to apply a windowing function to the segment before it is analyzed. This comes at the cost of some loss of information at the edge of the window, but the cost to benefit ratio can be negotiated based on the

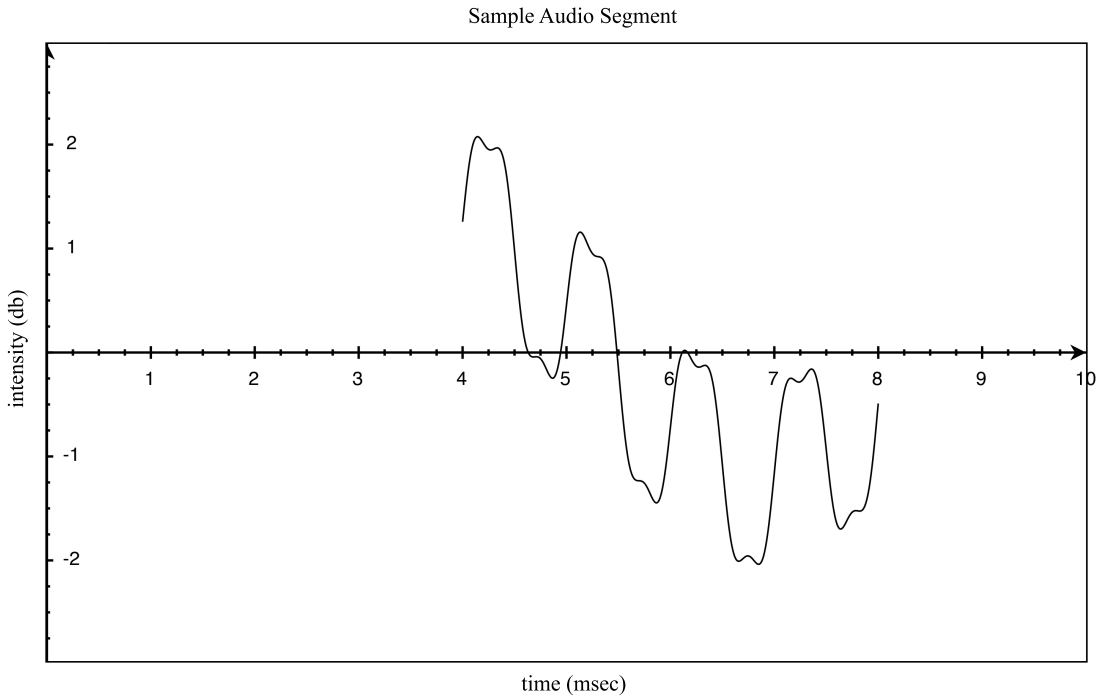


Figure 2.3: A signal segment.

window size. It is important to note that due to the nature of the Fourier transform, smaller window sizes reduce frequency resolution.

Windowing functions come in several forms and are employed based on the desired use of the resulting spectra of the Fourier transform. The windowing function used in this paper and popular in similar work is called the Hanning window and is given by the following Equation:

$$w(n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \tag{2.5}$$

The Hanning window can be "seen as one period of a cosine "raised" so that its negative peaks just touch zero" which causes the artifacts or "side lobes" to "roll-off approximately 18 dB per octave" [68]. The Hanning window equation has a form that can be tuned to cancel out the desired side lobes [68]. Figure 2.5 shows a typical

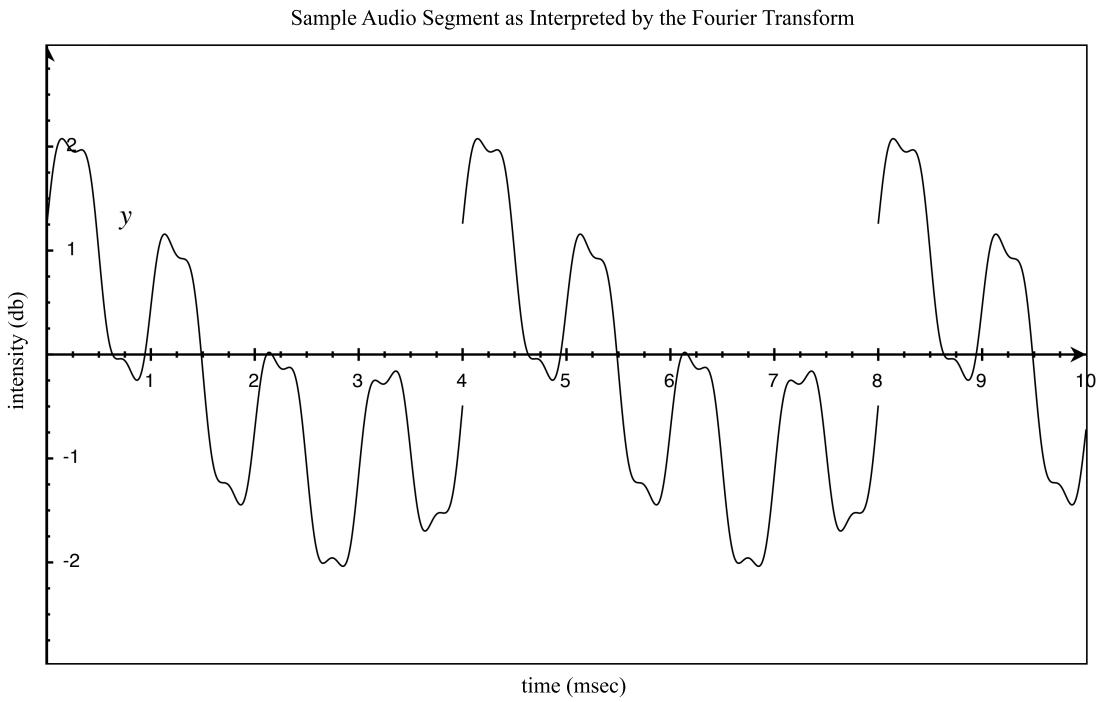


Figure 2.4: A repeating signal segment, as interpreted by the discrete time Fourier transform.

Hanning window used over a signal with 882 samples.

2.4 Timbre and Timbral Spaces

In music, timbre can be intuitively understood as the portions of an audio sensation which allow a listener to distinguish between two different instruments playing the same note at the same pitch and loudness. Pitch, one of the most recognizable attributes of a tone, represents the frequency of a pure tone and the fundamental frequency of a more complex one, both of which can be measured in one of several scales such as the mel scale, the musical pitch scale, or the physical frequency scale [12]. In other words, pitch is simply the subjective "highness" or "lowness" of a sound and makes the most sense when discussed in the context of other tones. Loudness, on the other hand, describes the physical intensity of a tone and is usually expressed

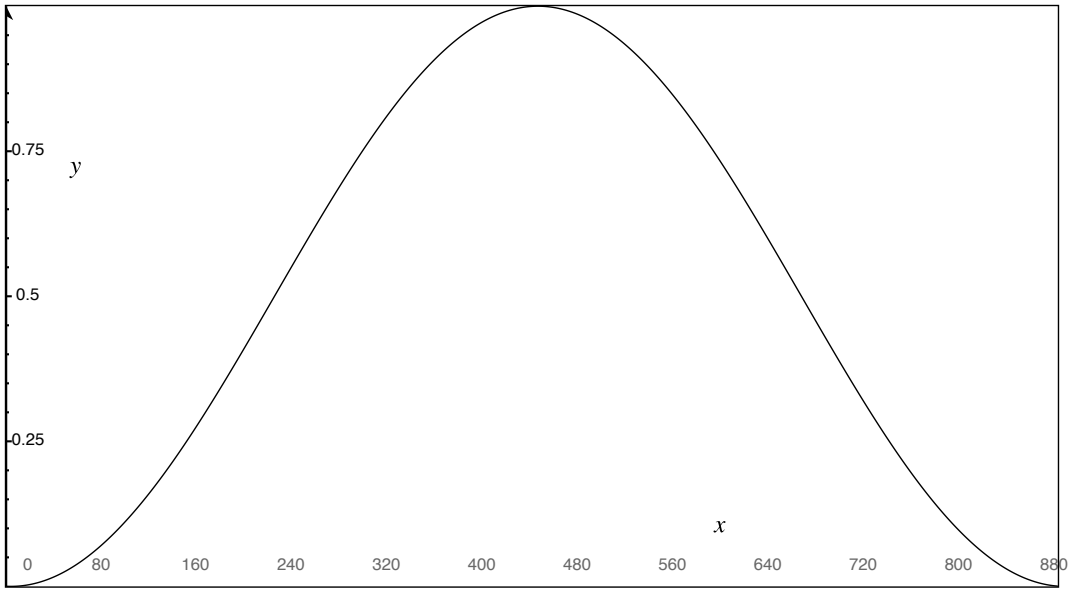


Figure 2.5: A Hanning window over 882 samples.

in decibels, a measure of sound pressure [12].

Timbre, then, encompasses all the descriptors one can use beyond the aforementioned to discuss a sound and is inherently subjective. Due to its broad definition, it is difficult to discuss timbre in terms of a single unit unlike loudness which can be summarized with the logarithmically scaled decibel or the frequency based hertz. In fact, timbre is best described with a slew of features of various units and scales. Analysis of timbre is therefore the analysis of a point or set of points in a multidimensional space. Deciding which features are most useful as axes in such a space depends on the desired results and is an open research question explored in this paper and has lead to many interesting suggestions and discoveries as described in Section 4.2.

One such timbral space is depicted in figure 2.6 which extends into three dimensions though these spaces can and often do extend into a much higher dimension. As perceptual and cognitive psychologist Diana Deutsch mentions in her book *Psychology of Music*, "[t]imbre is a *multidimensional* attribute of the perception of sounds.

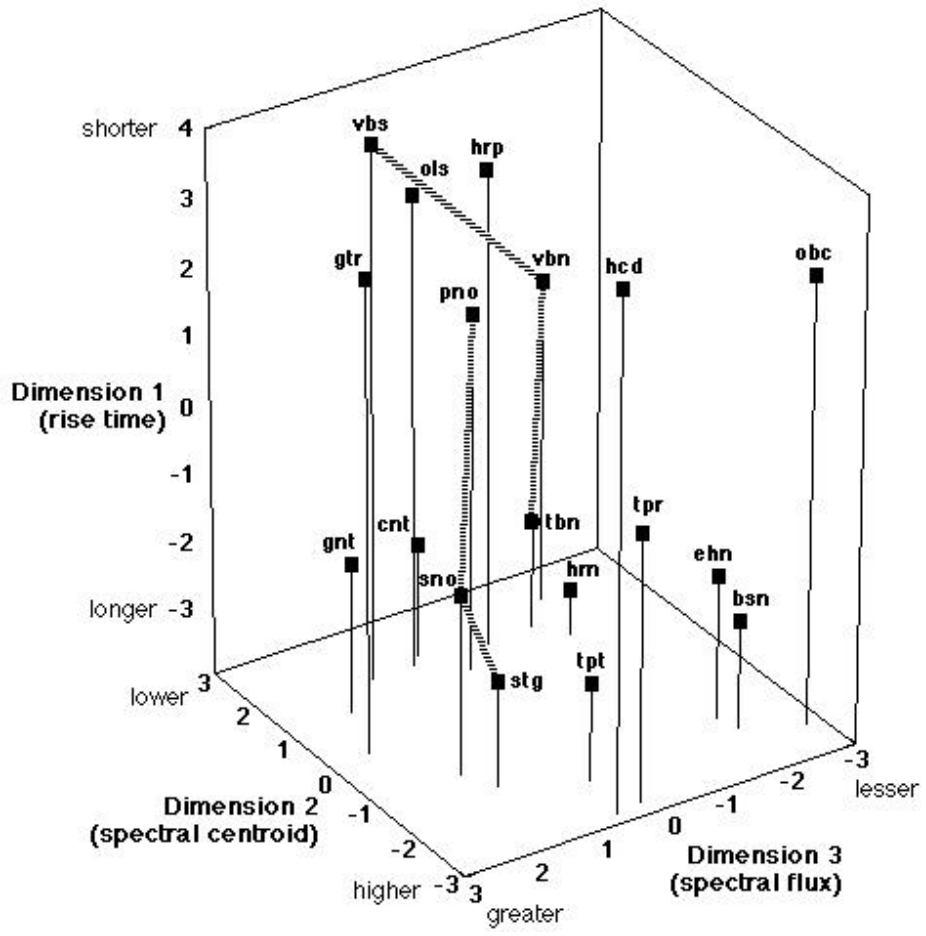


Figure 2.6: A three dimensional timbral space [31].

Dimensional research is highly time-consuming and is therefore always done with a restricted set of sound stimuli" [12]. Thankfully, as the cost of computing power shrinks so does the "time-cost" of exploring timbral spaces of higher dimensionality and gathering the data points to fill them.

2.5 Timbral Features

The following Section describes a selection of features that can be extracted from audio signals that were employed in the enclosed work as part of constructed timbral spaces. This is far from an exhaustive list and many of the following features have

not been explored in the context of timbre and polyphonic audio signals.

2.5.1 Spectral Shape Statistics

Spectral shape statistics are measures used to characterize a spectrum. They are computed by interpreting the spectrum as a distribution of frequencies whose probabilities of observation are given by a normalized amplitude [43]. The most popular spectral shape statistic in audio analysis is the spectral centroid which has been shown to be an excellent indicator for perceived brightness [16, 59]. It is given by the following equation where $x(n)$ is the magnitude of a frequency at bin n, $f(n)$ is the center frequency of bin n, and N is the number of bins for which frequency-magnitude data is available [59]:

$$\mu = \frac{\sum_{n=0}^{N-1} f\left(n\right) x\left(n\right)}{\sum_{n=0}^{N-1} x\left(n\right)} \tag{2.6}$$

Similarly, the spectral spread can be computed which represents how spread out the spectrum is around its spectral centroid [59]. It is given by the following formula where μ_1 is one standard deviation from the spectral centroid:

$$\sigma = \sqrt{\mu - \mu_1^2} \tag{2.7}$$

Skewness characterizes the asymmetry of the spectrum about its centroid [59]. A skewness value of 0 means that the distribution is entirely symmetric while a value less than zero indicates more energy to the left of the centroid and, conversely, a value greater than zero indicates more energy to the right [59]. It is computed using the following formula where μ_2 is two standard deviations from the spectral centroid [16]:

$$\gamma = \frac{2\mu^3 - 3\mu\mu_1 + \mu_2}{\sigma^3} \tag{2.8}$$

2.5.2 Spectral Rolloff

The spectral rolloff feature describes the frequency at which 99% of the energy in the signal is contained in lower frequencies. The measure is similar to "skewness" which is captured by the features mentioned in Section 2.5.1 but was included for comparison purposes based on inspiration from recent work on a music discriminator by Scheirer and Slaney [56].

2.5.3 Mel-frequency Cepstral Coefficients

The mel-frequency cepstral coefficients or MFCC's are an important feature in audio analysis, specifically in speech signal processing where they are used to augment the analysis of the spectral envelope and spectral details by also considering the perceptual effects of human hearing. Obtaining MFCC's from a signal is a multi-step process where each step has its own motivation and importance. The various components will be explained in this Section followed by a summary that describes how they combine to produce one of the most popular set of features used to model human hearing.

2.5.4 The Mel Scale

The mel scale is a "subjective scale for the measurement of pitch" that was proposed by Stevens, Volkmann and Newman in a 1937 journal article as a way to reconcile two different common definitions of the term [74]. They note that to a musician, pitch "has meant the aspect of tones in terms of which he arranges them on a musical scale". They go on to discuss that a musician will "divide the range of audible frequencies into octaves, which in turn are divided into tones, semi-tones, etc." and will then consider two sequential semi-tones as equal intervals in pitch. They consider this to be a perceptual definition.

However, they also cite a textbook which represents the more rigorous scientific definition of pitch as a period of vibration. The scientific and perceptual definitions do not agree, they argue, since it has been shown that raising the intensity of a tone of high frequency will raise the perceived pitch while increasing the intensity of a lower tone lowers its perceived pitch [73]. To resolve this discrepancy, they present the mel scale which is a linear mapping of hertz to the mel unit which "take[s] into account the loudness of tone".

$$m = 2595 log_{10}(1 + \frac{f}{700}) \tag{2.9}$$

Because the mel scale conversion was created by fitting a curve to a plot of actual frequency versus an average of five observer's perceived frequencies there is no official frequency to mel conversion formula [74]. Several formulas have been proposed and one of the more popular ones is described in Equation **??** and shown graphed in Figure 2.7 [40]. Most cover the frequency range of 0 Hz to 20050 Hz [79].
[xlabel = frequency (Hz), xtick=0,1000,...,10000, ylabel = mels, ytick=0,400,...,3200, width=.9samples=100, height=5cm, scaled x ticks=false, scale only axis] [domain=0:10000] 2595*log10(1 + x/700);

Figure 2.7: The mel scale graphed as a function of hertz.

2.5.5 Cepstral Analysis

Cepstral analysis is a powerful spectral analysis tool that can reveal periodic elements of a signal not readily available with standard spectral analysis techniques. In other words, the cepstrum allows "the separation (deconvolution) of source effects from transmission path or transfer function effects" [37]. The power cepstrum is a function that results from first obtaining the square of the power spectral density of a signal

segment obtained using the Fourier transform, then taking the log of the result, and finally taking the output and squaring its inverse Fourier transform. This process is expressed in Equation **??**.

$$cepstrum = \left|\mathcal{F}^{-1}\left\{\log(|\mathcal{F}\{f(t)\}|^2)\right\}\right|^2 \tag{2.10}$$

The resulting cepstrum is a function of τ called the quefrequency [39]. A spike in the cepstrum represents a periodic component of the original signal. The frequency of this component can be determined by dividing the sampling rate of the original frequency by the quefrequency of incident. Many sounds can be partially characterized by their periodic elements, such as the harmonics of an instrument, which makes the cepstrum a useful source of information in audio analysis.

2.5.6 MFCC's

Mel-frequency cepstral coefficients are an attempt to marry the ideas of the mel scale and cepstral analysis using a form of principle component analysis. Sahidullah outlines the steps required for computing MFCC computation in a paper on speaker recognition [55]:

1. First, apply a window to the signal.
2. Compute the power spectrum of the windowed signal using the discrete time Fourier transform.
3. Pass the power spectrum through a triangular filter bank which contains a preselected number of triangular filters spaced according to the Mel scale. An example of such a filter bank is shown in Figure 2.8.
4. Take the logs of the resulting powers, of which there should be as many as the number of filters used in the previous step.

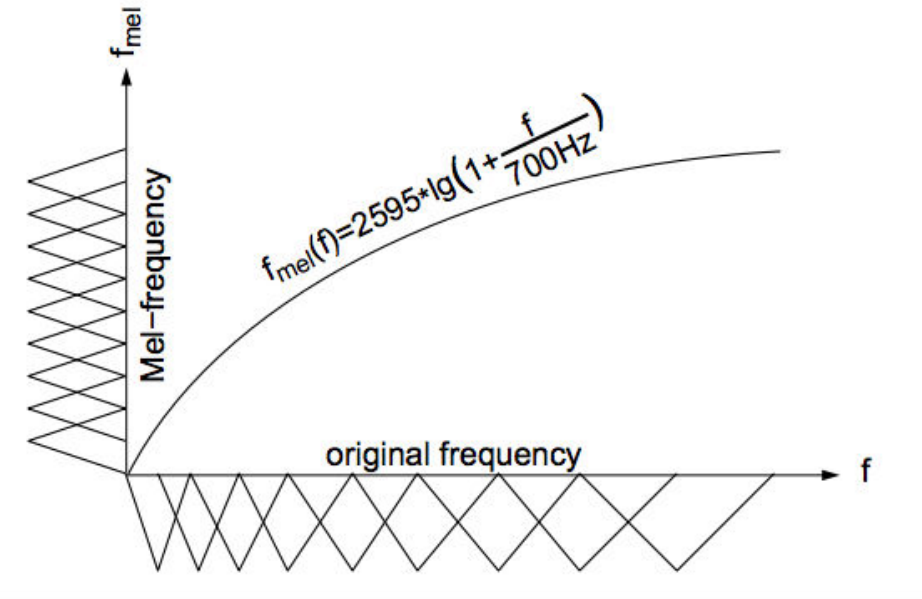


Figure 2.8: Mel-frequency filter bank with 9 filters [33].

5. Compute the discrete cosine transform (DCT) of the filtered power spectrum. The amplitudes of the output spectrum are considered the MFCCs.

The discrete cosine transform is used as a form of principal component analysis in order to decorrelate components of the mel spectra obtained via the use of filter banks [28]. The first coefficient is usually discarded as it represents the *"dc-coefficient"* of the signal [55].

2.5.7 Binergy

Since mel-frequency cepstral coefficients were not studied for use as features for timbral spaces several other binning techniques were used for comparison. The first of these was dubbed "binergy" and simply involved filtering the FFT with 20 evenly spaced, non overlapping triangle filters, and summing the energy under each filter to produce 20 different values. This differs from MFCCs in the following ways:

1. The filters are not logarithmically spaced.

2. The filters have no overlap.

3. The logarithm is not taken of the filter energies.

4. The DCT is not taken of the filtered power spectrum.

5. The filters do not cover progressively larger frequency ranges as frequency increases.

6. 20 Bins are used.

The binergy filter was created by first computing the periodogram of a signal multiplied by a Hanning window. The bin width was determined by $binwidth = \frac{\text{number of bins in periodogram}}{20}$. The bins from the periodogram were then multiplied be each filter and accumulated into an array.

2.5.8 Log Binergy

Log binergies were created in a manner similar to binergy discussed in Section 2.5.7 except for having 13 filters spaced logarithmically instead of being spaced with centers according to the mel scale. The logarithmically spaced filters are very similar to mel spaced ones but, again, neither the log nor discrete cosine transform are are taken of the resulting powers. The filters do, however overlap, much like the mel spaced filters. This was done for comparison to MFCCs in order to see if these two aspects of the feature computation along with the mel spacing made a discernible difference.

2.5.9 X Bins

As the final variant of filter banks for features, "x bins" were calculated which were simply the energies from a configurable number of logarithmically spaced filters which overlapped in a way similar to the mel filter bank filters. This was done to see whether

13 really was some sort of optimum filter number, or if gathering more features over more filters was helpful. Extremely large values for the number of filters causes clustering to become prohibitively slow given the poor performance of k-means in higher dimensions as described in 7.4 so 100 of these filters were used.

2.5.10 Energy

The root mean squared energy over an audio frame was computed using Equation **??** over each grain where N is the number of frames in the grain and $x(i)$ is the amplitude of the signal at frame i.

$$\sqrt{\frac{\sum_{i=0}^{N-1} x(i)^2}{N}} \tag{2.11}$$

2.5.11 Zero Crossing Rate

The zero crossing rate is a measure of the amount of times a signal changes from positive to negative or from negative to positive. An example of a zero crossing is given in Figure 2.9.

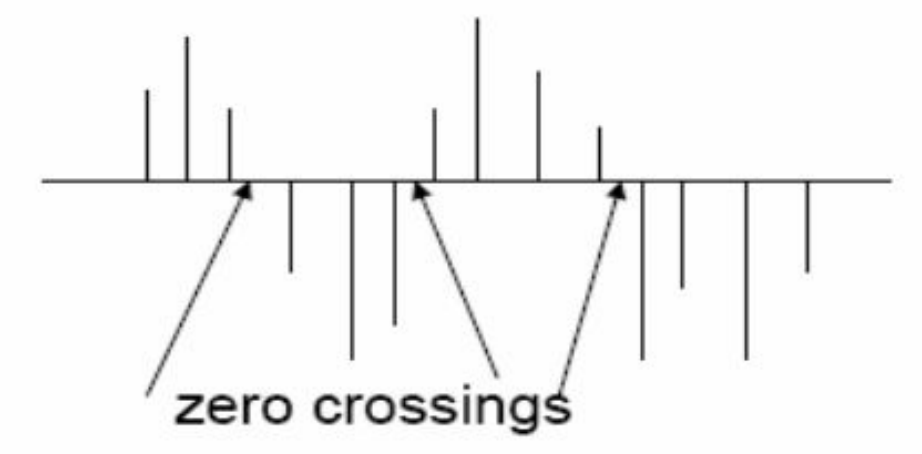


Figure 2.9: An example of zero crossings in a signal [63]

The zero crossing rate can be calculated using Equation **??**. It represents the frequency content for signals of a narrow frequency band, or broad frequency band signals over a very short amount of time [38].

$$Zn = \sum_{m=-\infty}^{\infty} |sgn[x(m)] - sgn[x(m-1)]|$$
$$\text{where } sgn[x(n)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) < 0 \end{cases} \tag{2.12}$$

2.6 K-means Clustering

Creating timbral spaces populated with data points provides a model over which further analysis can be performed. In the enclosed work, the goal of the analysis step is to uncover data points with similar timbres and form them into groups. This requires an algorithm that will take as input a vector of data points in N-dimensional space and output labels for the data points which represent groupings of similar inputs. Clustering algorithms, which attempt to group similar data points based on a given distance measure, are excellent candidates for such analysis.

In k-means clustering specifically, n data points in a d-dimensional space $\mathbb{R}^d$ are provided along with an integer k [24]. The goal of the algorithm is to produce k points or "centroids" in $\mathbb{R}^d$ such that a provided distance function is minimized over the distance between all points and their closest centroid. When finished, the algorithm assigns labels to data points which represent assignments to computed clusters.

A basic form of the algorithm works by first randomly initializing all centroids in $\mathbb{R}^d$ and then assigning each point to a cluster containing its closest centroid. For each new cluster, the actual center is calculated and the centroid is updated to be at that point. The first step of assigning all other points to a cluster with their closest

centroid is repeated, and the centroids are updated in turn. This cycle continues either until convergence i.e. centroids no longer move or until a predefined iteration count is reached.

K-means clustering is a comparatively simple unsupervised machine learning technique that scales well with the number of data points but has some well known issues. Choosing a k must be done before running the algorithm and can be difficult to do without being able to observe the data which is extremely difficult for high dimensional data sets [65]. K-means is also particularly sensitive to noise and outliers and will often terminate at a "local, possibly suboptimal, minimum" [36].

Chapter 3

IMPLEMENTATION DETAILS

This section details the software suite called Timcat which was created for the purpose of novel timbre creation via signal analysis, clustering, and concatenative synthesis techniques. The framework is comprised of three parts: a granulizer, an analyzer, and a synthesizer. An audio signal is fed as input into the granulizer which divides the signal into segments called *grains* based on a provided time interval. The analyzer performs signal analysis on the grains, saving the data points in a database keyed on file name for later use. The synthesizer then performs clustering and concatenates the audio segments based on the output of the analyzer, ultimately outputting audio files that represent new timbral patches. The flow of the framework is represented in Figure 3.1.

The code for Timcat can be found on github at `https://github.com/neobonzi/ConcatenativeSynthesisThesis`

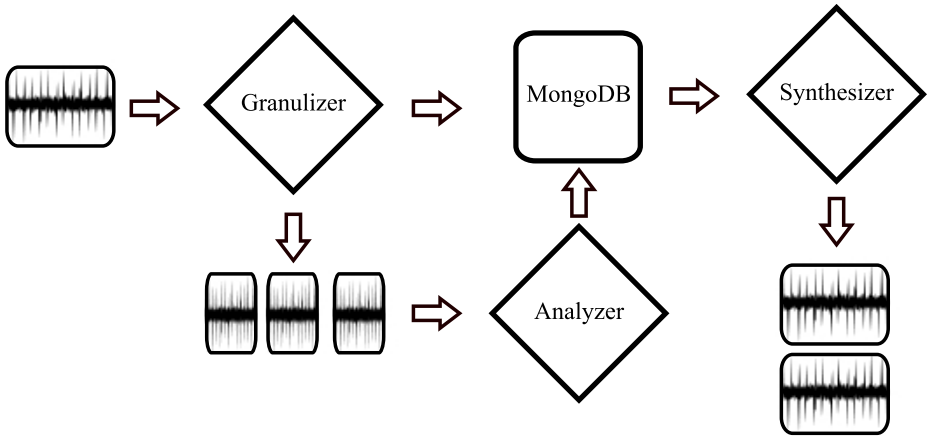


Figure 3.1: Diagram of the flow of the Timcat framework.

3.1 Granulizer

The granulizer's task is to divide an audio signal into small signal segments called "grains" for use as input into the analyzer. It requires as input a monophonic mp3 file as the system does not currently handle multi track audio files. Because of this, preprocessing usually entails merging multi channel audio files into a a single channel. The granulizer is called using the source file, a destination folder where the grains will be placed, and the grain size in milliseconds as indicated in 3.2.

```
./granulizer.py source destination grainSize
```

Figure 3.2: Synopsis of call signature for the granulizer script.

The input file is converted to an AudioSegment object using the eyeD3 library [64] which allows for extraction of the metadata contained in the MP3 file's ID3 tag [30]. If the ID3 tag contains a title, it is combined with numbers representing the start and end frame numbers of the original audio file using an underscore delimiter to create the grain filename. The audio segment is then exported to the provided destination folder in the Waveform Audio File (WAV) format with the same sampling rate as the original file.

Finally, the granulizer creates a new entry for the grain in a MongoDB instance. MongoDB is a high performance NoSQL document store that was used as the data persistence layer for this project [34]. It was chosen over other database solutions since it is performant when working with the type of"flat" records that needed to be stored which are looked up only by primary key. The initial entry as created by the granulizer contains the relative file path to the grain, the artist and title from the ID3 tag, the sample rate of the audio, and its length in frames.

3.2 Analyzer

The analyzer performs the bulk of the work in Timcat by performing signal analysis on the grains created by the granulizer. Many of the features were computed using a Python library called Yaafe which stands for "Yet Another Audio Feature Extractor" and which its creators describe as an "audio features extraction toolbox" [62]. Several features were extracted using custom tools while mel-frequency cepstral coefficients were calculated using the Aubio library [9].

```
./analyzer.py [--clear] [--mfcc] [--pitch] [--energy]
[--shape] [--rolloff] [--all] [--zcr] [--xbins] [--binergy]
[--logbinergy] [--ratios]
```

Figure 3.3: Synopsis of call signature for the analyzer script.

The analyzer is called with zero or more parameters which indicate which features should be computed as indicated in Figure 3.3. When the analyzer is called with no arguments, it computes all the features available for all of the grains which currently do not have the features computed. When called with the "clear" argument, the analyzer will delete all data store entries as well as any grain files. If one or more feature arguments are present, the analyzer will compute only those features. When a feature or group of features is computed the entry for the grain in the data store is updated with the new data so it can be retrieved and used for clustering in the next phase.

3.2.1 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients, described in detail in 2.5.3, are computed using the Aubio Python library [9]. Aubio's implementation of the MFCC computation is

a Python rewrite of a tool written for Matlab by Malcolm Slaney for the Auditory Toolbox [66]. The library handles the creation and application of a Hanning window to a signal segment, which it slides across the whole signal based on a given hop size. Because the grains are so short in length, the hop size is given to be the same as the window size which in turn is simply the number of samples in a grain. This causes the algorithm to exit after a single iteration and returns only one set of coefficients.

3.2.2 Log Binergies

Mel-frequency cepstral coefficients have not been widely used in the analysis of music and have been noted to be, at worst, at least not harmful for use as a feature [28]. In Logan's 2000 research paper in which she analyzed the use of MFCCs for music modeling, she mentions that "[f]uture work should focus on a more thorough examination of the spectral parameters to use such as the sampling rate of the signal, the frequency scaling (Mel or otherwise) and the number of bins to use when smoothing" [28]. This lead to the inclusion of a tool for computing the energy contained in an arbitrary number of logarithmically spaced bins in Timcat called "log binergies".

To compute the log binergies, first a periodogram is computed which utilizes the discrete time Fourier transform as described in **??** to construct a histogram of energy distribution in frequency bins. Due to the inner workings of the DTFT, the resolution of the frequency bins is given by the following formula, where f_s represents the input signals sampling rate:

$$I = \frac{f_s}{N} \tag{3.1}$$

Unfortunately, because the grains are very short in length the resolution of the frequency bins is consequentially somewhat low. For example, a CD quality audio signal is sampled at 44100 Hz. With grains that are approximately $20ms$ in length,

the frequency bin resolution is $resolution = \frac{44100 \text{ Hz}}{(.02s)(44100 \text{ Hz})} = 50Hz$. The Nyquist-Shannon sampling theorem described in Section 2.1 indicates that the maximum frequency contained in such a signal is 22050 Hz which leaves $\frac{22050 \text{ Hz}}{\frac{50 \text{ Hz}}{bin}} = 441$ bins to work with.

In light of this, it is clear that a logarithmically spaced set of filters will have a margin of error in terms of the energies that they cover. The best way to alleviate this issue would be to increase the length of the grains, thereby increasing the number of samples. This, however, would come at a cost since the longer the signal segment the less the assumption that the segment represents an unchanging signal over its duration holds.

3.2.3 Zero Crossing Rate

The zero crossing rate (ZCR) was easily obtained using Yaafe which provides an efficient implementation of the Equation shown in **??**. Creating a feature plan with a step size and block size over the length of the grain allows the return of the ZCR in several milliseconds.

3.2.4 Spectral Features

Again, Yaafe provides a robust and fast implementation of the spectral centroid, spread, skewness and kurtosis as they are detailed in [16]. The grain is multiplied by a Hanning window before the FFT is taken and only one frame is used. The output of Yaafe is a tuple with all statistics concerning the spectral shape included.

3.2.5 Harmonic Ratios

The goal of computing harmonic ratios was to capture some sort of aspect of the signal that compared how much energy was in each of the harmonics, if there were

any at all. Recall from Section 2.5.5 that sounds can be partially characterized by their repeating elements. This is especially true of most instruments which are simply acoustic oscillators that produce pressure fluctuations at integer multiples of the fundamental frequency in order to produce a characteristic tone.

As an example, Figure 3.4 shows a plot of the energy at each harmonic of an F4 fundamental as played by a flute. If a piano played the same note in the same room as the flute, the energy levels at each harmonic would be different which would be indicative of the pianos difference in timbre when compared to the flute. By determining the ratios of one of these harmonics to another, it was hoped that a novel feature could be obtained that correlated with timbre.

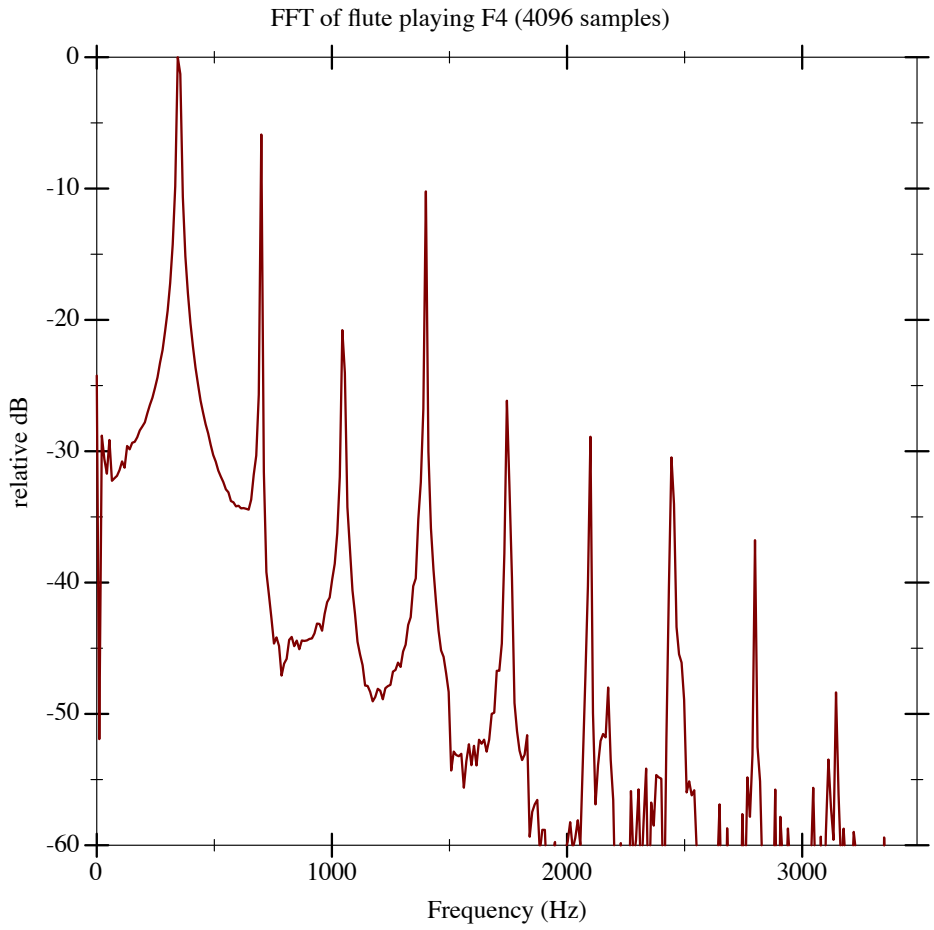


Figure 3.4: Plot of the fast Fourier transform of a flute playing F4.

The first obstacle in computing these ratios was to find the fundamental frequency. For a monophonic sound, this is much easier to accomplish than doing so from a sample obtained from ambient noise or polyphonic music. It is entirely possible for a grain to not have a fundamental frequency at all.

There are many ways to find the fundamental frequency, each with its own drawbacks. The simplest and the one first attempted was to simply take the FFT of the grain and locate the frequency bin which contained the most energy. In Figure 3.4, this method would indicate a fundamental of 350 Hz which is in line with what we would expect from a flute tuned to a 440 Hz A4. There is no guarantee that this would correspond with the fundamental, but it was hypothesized that if the fundamental did exist that this bin would at least correspond to one of the low harmonics which would still be useful for comparisons.

Figure 3.5 shows a power spectrum density plot for a single grain from Hey Jude by The Beatles with the detected fundamental and 6 subsequent harmonics marked with dashed red lines as detected by the "most energy" technique. The harmonics were detected by first finding the maximum energy bin from the windowed FFT of the grain, which turned out to be at 300 Hz. Subsequent harmonics were determined by taking integer multiples of the fundamental (600 Hz, 900 Hz, and 1200 Hz, etc.).

An important caveat to this method of fundamental detection is that it's highly dependent on the resolution of the FFT and, therefore, the sampling rate of the original audio file. The grain above was captured from an audio file sampled at CD quality sampling rate: 44100 Hz. According to Equation **??** the bins of the resulting FFT represent 50 Hz worth of energy given a grain size of 20 mS which results in 882 samples per grain. Low resolution can lead to large discrepancies in fundamental and harmonic detection especially at high frequencies where not only is the human ear much more discerning between pitch differences but where the accuracy of the chosen

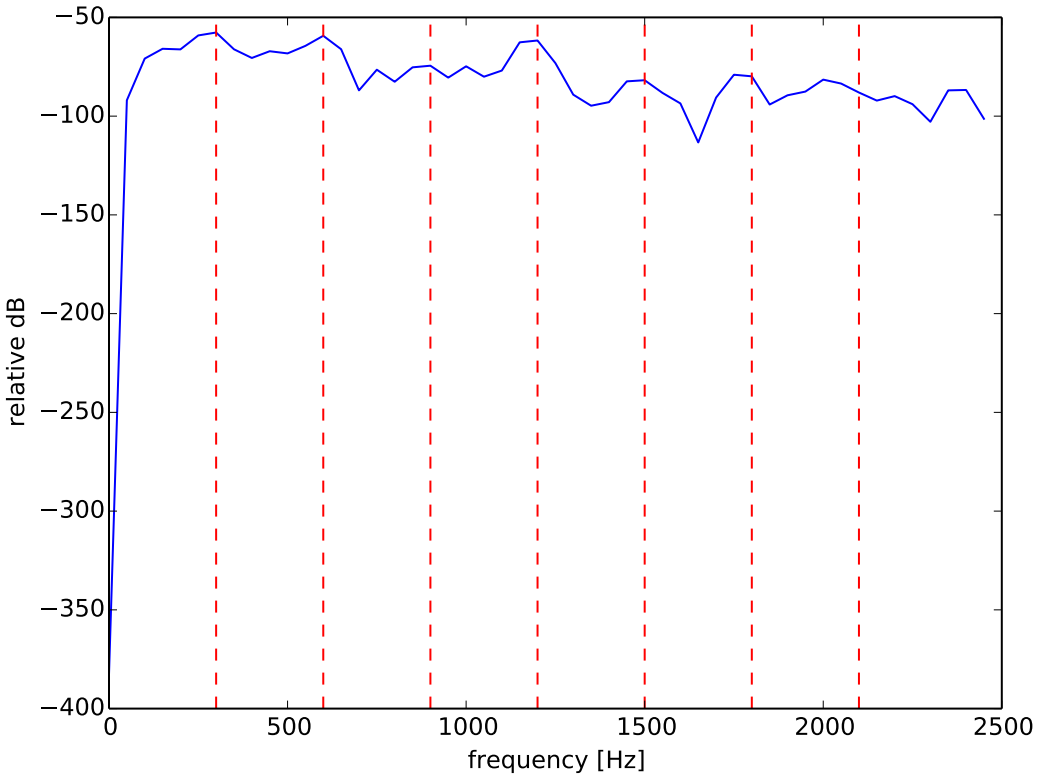


Figure 3.5: Periodogram of a single grain extracted from Hey Jude by The Beatles.

harmonic is less accurate due to any error in fundamental frequency detection being magnified by the multiplication technique for discovery of subsequent harmonics.

An attempt was made to more accurately determine the fundamental frequency of grains using a technique called autocorrelation which presents a measure of "how similar a signal is to itself" [70]. The equation for computing autocorrelation, or the cross correlation of a discrete time signal with itself, is shown in Equation **??**.

$$C_s(m) \equiv \sum_{n=-\infty}^{\infty} s_n s_{n-m} \tag{3.2}$$

An intuitive understanding of autocorrelation can be gained by imagining multiplying a discrete time signal at all points in time by itself at all other points in

time. The more the two signals match up, the greater the number the function in Equation **??** will yield at any given sample m, where m is also called the *lag* [70]. If the signal is periodic, the autocorrelation function "shows peaks at multiples of the period" [11]. The autocorrelation method for pitch detection, then "chooses the highest non-zero-lag peak by exhaustive search within a range of lags" and assumes that this is the fundamental. An example of the process is shown in Figure 3.6.

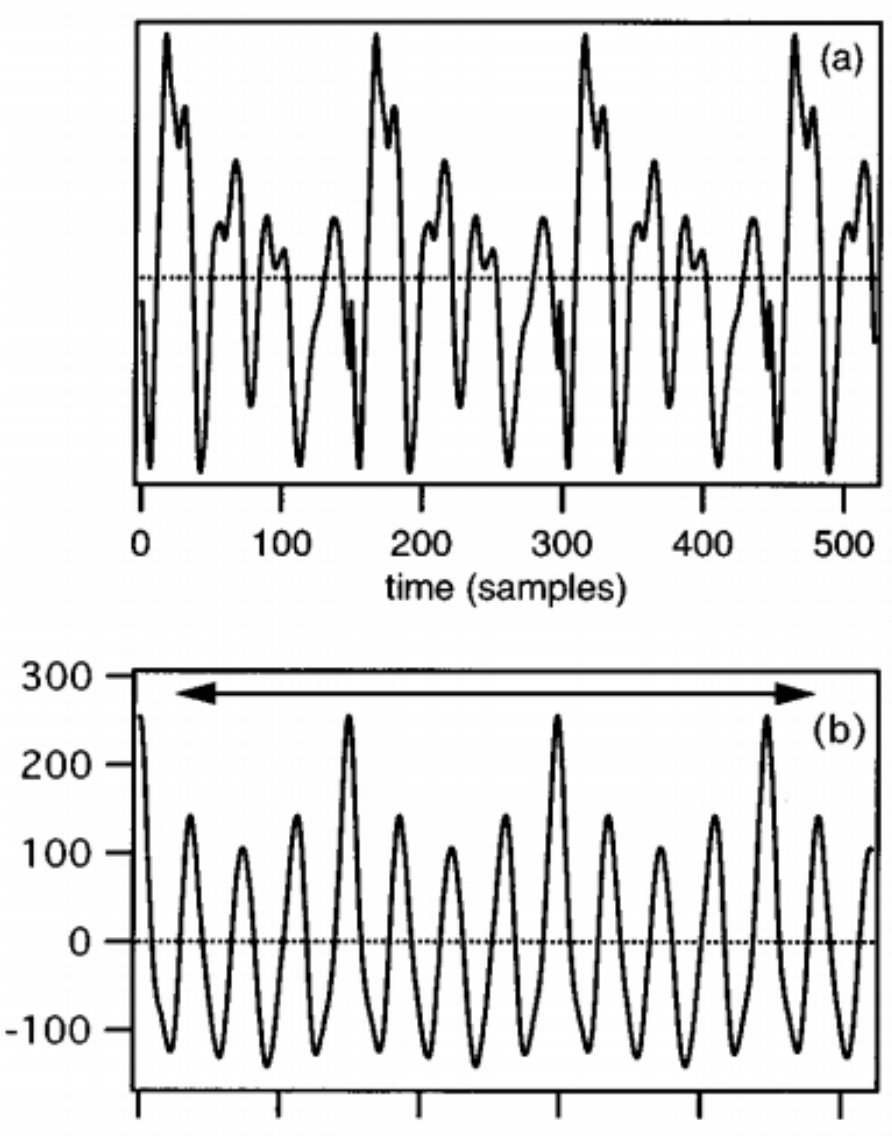


Figure 3.6: Autocorrelation (b) of signal (a) where the arrows represent the search range of lags for fundamental [11].

The autocorrelation method has many shortcomings which have lead to many modifications in order to correct for errors. One of the most successful modifications comes in the form of the Yin algorithm which, at the time of its release, had "error rates [sic] about three times lower than the best competing methods, as evaluated over a database of speech recorded together with a laryngograph signal" [11].

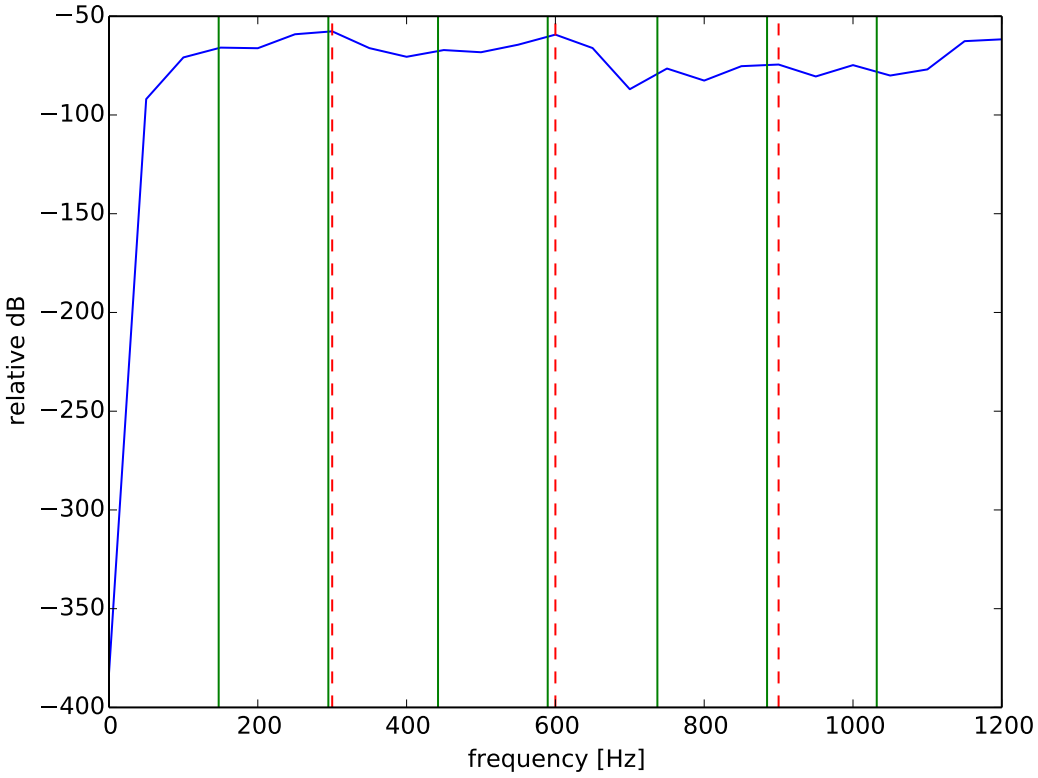


Figure 3.7: Fundamental and harmonics overlayed on a periodogram as detected by the "most energy" method (dashed red) versus the Yin method (solid green).

In Figure 3.7 we see the benefits of a more precise fundamental pitch detection method. The Yin algorithm detected a fundamental at 147.415 Hz while the "most energy" method provided the fundamental at the bin centered at 300 Hz. In this case, the fundamental was missed completely and the "most energy" method instead discovered the second harmonic. Using the old technique for fundamental detection,

every other harmonic would have been missed since integer multiples of the fundamental are used to find them. Even if the "most energy" technique was close in finding the fundamental by providing the bin centered at 150 Hz, the small error of 2.585 Hz (assuming the Yin algorithm is accurate) is doubled with every subsequent harmonic found. This growth in error exacerbated by the fact that the human ear is much more discerning at higher frequencies. Clearly, the Yin algorithm was the more appropriate choice.

With fundamental detection in place, the next step was to calculate the energy in each of the harmonic bins and compare them. To encapsulate this into a feature set, the energy of each harmonic is divided by the energy in the fundamental harmonic and the ratio is saved. The number of features available for extraction in this manner is highly dependent on where the fundamental lies. If it is too high, multiples of harmonic frequencies will quickly go "off the edge" of the available frequency range. A decision was made at this point that a fundamental will only be considered if it has a fourth harmonic within the frequency range of an FFT of a signal sampled at CD quality 44100 $\frac{samples}{sec}$. Thus, the max allowable fundamental by the system is:

$$\text{maximum fundamental} = \left\lfloor \frac{22050 - 1}{5} \right\rfloor = 4409 \text{ Hz} \tag{3.3}$$

This technique also helped filtering grains that contained silence, which were very common at the beginning and end of audio tracks.

3.3 Synthesizer

The synthesizer performs clustering on the feature vectors made available from the analysis phase covered in Section 3.2. It is called using the signature shown in Figure 3.8 with options that inform the synthesizer how many clusters to use for creating

new groupings and which and how many of each feature to use for clustering.

```
./synthesizer.py [-h] [-numClusters [NUMCLUSTERS]]
  [-numXBins [NUMXBINS]]
  [-numBinergies [NUMBINERGIES]]
  [-numLogBinergies [NUMLOGBINERGIES]]
  [-numMFCCs [NUMMFCCS]] [-numRatios [NUMRATIOS]]
  [--rolloff] [--energy] [--zcr] [--centroid] [--spread]
  [--skewness] [--kurtosis]
```

Figure 3.8: Synopsis of call signature for the synthesizer script.

The clustering algorithm used is an implementation of Lloyd's algorithm by the Scikit Learn Python library [27, 42]. The method used for selecting initial clusters centroids is called k-means++ originally proposed by David Arthur and Sergei Vassilvitskii in 2007 and can intuitively be understood as an attempt to spread out cluster centroids as much as possible [3]. This has been shown to cost some initial time up front but speeds up convergence of the clustering algorithm significantly which is where a bulk of the computation occurs [3]. The algorithm is forced to finish after 300 iterations to avoid excessively long run times and is run 20 times in parallel on as many CPUs as are available on the host machine. The best run of the 20, as determined by the least change in object to cluster assignment on the final step, is kept.

As mentioned in Sections 2.6 and 7.4, the appropriate number of clusters to use over any given set of data is difficult to determine *a priori* and, as such, is left to the user to provide as a parameter to the analyzer as a matter of preference. There are several techniques that can be used to estimate the cluster count before running the algorithm as described in [44] but implementation of these methods are left for future

work.

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{if } a(i) < b(i) \\ 0, & a(i) = b(i) \\ \frac{a(i)}{b(i)} - 1, & a(i) > b(i) \end{cases} \tag{3.4}$$

To provide some measure of performance for the clustering algorithm, the synthesizer will also output a statistic called the silhouette score for each k-means output. Silhouettes, first described by Rousseeuw in 1986, provide a measure of how similar objects are to other objects in the same cluster [52]. A silhouette value close to 1 for a labeled object represents large intra cluster similarity when contrasted with how dissimilar it is to the closest cluster. A value close to -1 represents a poor labeling job done by the clustering algorithm. By taking the average of all silhouettes, a single number can be obtained that represents the silhouette score across all assigned objects and clusters. This is the value provided by the analyzer.

The equation for calculating the silhouette of an object i is given in Equation **??** where $a(i)$ is the mean intra cluster distance of i and $b(i)$ is the least average dissimilarity of i to any other cluster. An example of the graphical representation of silhouettes for a cluster first demonstrated by Rousseuw is shown in Figure 3.9.

The final task of the synthesizer is to perform concatenative synthesis on each cluster of audio samples that have been labeled by the clustering algorithm. For each cluster, member data points are cross referenced with an array of ids which are in turn used to look up their corresponding audio grain files in the database. These grains are then "concatenated" together at random using a fade that is 50% by length as shown in Figure 3.10. Maximizing the amount of fade without overlapping three grains at any point was experimentally determined to cause the least abrupt discontinuities between audio signals thereby reducing "clicking" noise artifacts in the final virtual

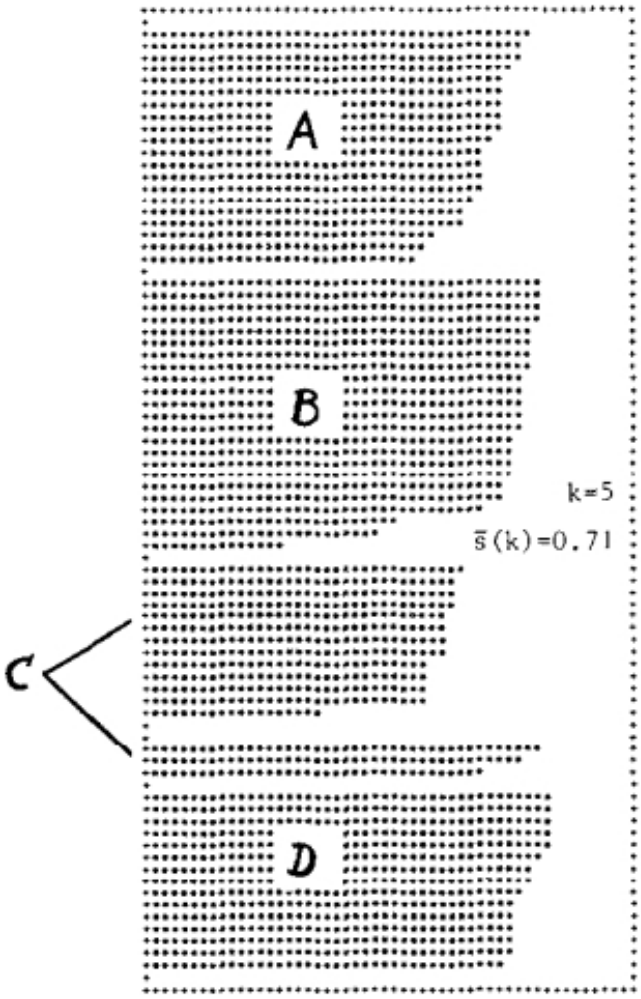


Figure 3.9: Example silhouette graphical representation for 5 clusters with actual classifications labeled A, B, C, and D. The average silhouette score is 0.71 [52].

instrument patch.

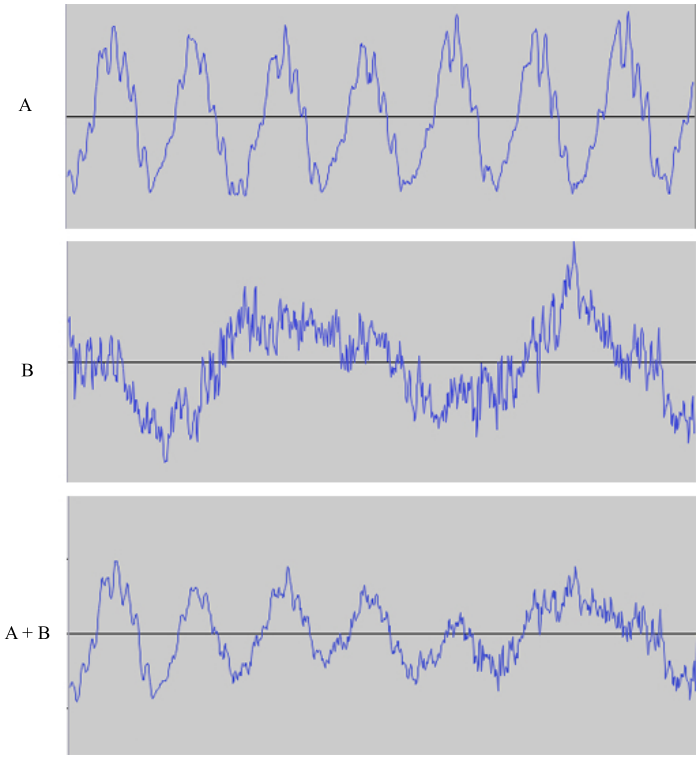


Figure 3.10: 20 mS grains A and B crossfaded by 50% (10mS).

Chapter 4

RELATED WORK

Research in the field of concatenative synthesis explores methods of combining pieces of audio to best match a target based on pre-analyzed descriptors. Timbral analysis focuses primarily on extracting descriptors of sound for use in various frameworks and fields of study, such as concatenative synthesis. This chapter presents similar work in both fields in the context of music.

4.1 Concatenative Synthesis

Contemporary concatenative synthesis for music became prevalent around 2001 when Zils and Patchet presented "musaicing", a method to recall audio samples from a large database based on a set of provided constraints [82]. The authors created a cost function that expressed differences between constraint values of a target audio segment and the audio segments in their corpus. By selecting the audio segments which minimized this cost function over many targets and concatenating them together the authors effectively created a method to automate the process of sample selection for music composition based on high level descriptors. The segment descriptors they used were gathered using techniques popular for audio analysis as described in Section 4.2 and represented pitch, loudness, percussivity and global timbre, each calculated from properties of the spectral and temporal envelopes of the segments in their corpus.

Diemo Schwarz also spent considerable effort looking at the task of concatenative synthesis as a sequence alignment problem in his 2004 PhD thesis [61]. Using hidden Markov models and dynamic time warping techniques he presented a framework by which one could perform automatic music alignment, or the automatic association

of musical events in a score with time events of an audio signal based on segment features similar to those extracted in Zil's and Patchet's work [82]. Schwarz presented his results in a software system called Caterpillar which allowed for interactive instrument synthesis based on a given audio sample target. He later extended the idea of interactive concatenative synthesis to a more modern implementation using the Max/MSP visual programming framework which allowed a composer to visually explore audio segments based on spectral and temporal features in a 2D descriptor space [60].

More recently, Maestre et al. used concatenative synthesis techniques aided by an expressive performance model to "generate an expressive audio sequence from the analysis of an arbitrary input score" [29]. Inspired by previous work by Schwarz, the authors gleaned segment information not only from analysis of the signal of an instrument note, but from the context of the score during the time that the note was played. Knowing the next or previous note's pitch, duration and strength proved to be valuable information in their attempts to reproduce a note for an arbitrary score with not only the correct frequency and loudness, but with similar expressivity as well.

4.2 Feature Extraction for Timbral Analysis

Quantifying qualities of timbre can prove to be a difficult task given that even the most official and rigorous definition of the term somewhat cryptically refers to its subjective nature [48]. Indeed, authors have written about their frustration with a term that, although extensively studied, is grounded in perception and is therefore often interpreted differently in various contexts. For example, in a critique of the ANSI definition for timbre by Institute of Perception Research's A.J.M. Houtsma, concerns were raised as to "whether timbre recognition is synonymous with the recognition

of a sound source [such as a] particular musical instrument" or whether it is the recognition of a "musical object" from a "perceptual space" [22].

The variety of methods used to evaluate and characterize timbre is a testament to the subjectiveness of the word. In a 1979 paper submitted to the *Computer Music Journal*, Stephen McAdams and Albert Bregman described timbre as a "psychoacoustician's multidimensional waste-basket category for everything that cannot be labeled pitch or loudness" [72]. Schouten is often cited for casting aside of the typical definition of timbre as the "overtone structure or the envelope of the spectrum of the physical sound" in favor of his summary based on "at least five major parameters" [57]. He goes on to define these as the tone to noise ratio, the spectral envelope, the rise, duration, and decay of the time envelope, changes in the spectral envelope and fundamental frequency, and differences between the onset of a sound and when it is sustained.

The spectral envelope, or curves that represent the magnitudes of spectra in the frequency domain, emerged as one of the most frequently used tools for quantifying timbre. One such envelope is shown in Figure 4.1. As early as 1977, researchers J. Grey and J. Gordon attempted to analyze and quantify changes in perception of trumpet tones by tweaking the spectral envelopes of audio played for test subjects [17]. In a recent project, Burred et al. developed a model of various instruments by measuring the spectral envelope which they then successfully used to both classify instrument samples and detect the presence of instruments in polyphonic music [10]. In similar work, Ron Yorita showed some correlation between tone quality descriptors by flutists and harmonic spectra [81]. The spectral centroid alone has been proven time and again to be a very useful aspect of the envelope which has been shown to map effectively to perceived brightness [1, 58, 78].

Data points gleaned from interpretations of spectral energies mapped to the psy-

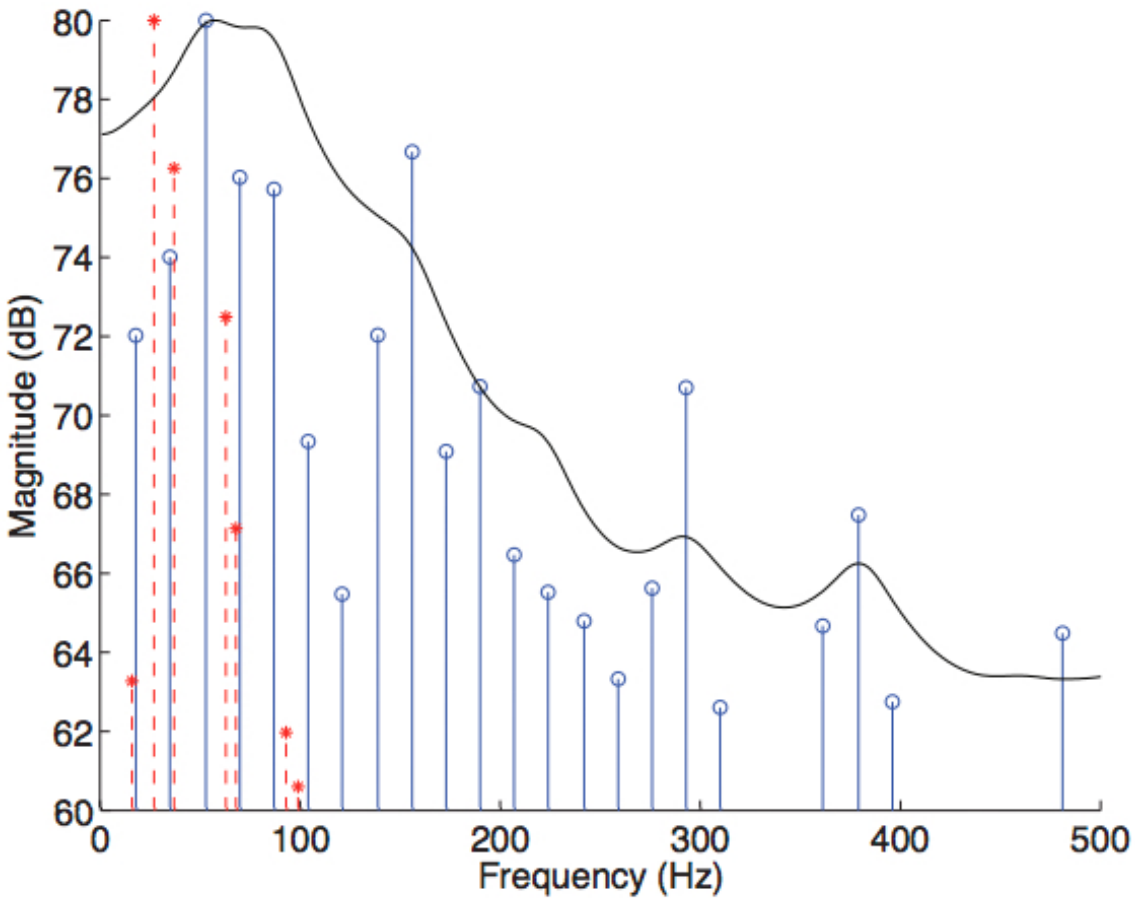


Figure 4.1: Example of a spectral envelope of a double bass tone (solid line), spectral peaks of a different sound from the same double bass (solid lines) and spectral peaks of a Bassoon (dashed lines) [26].

choacoustic Mel frequency scale are increasingly prevalent in feature sets for use in music analysis. The Mel scale is a subjective pitch measurement proposed by Stevens and Volkmann that accounts for discrepancies in perceived pitch intervals in lower frequency tones [74]. Thomas Gill used various processing techniques of the spectral energy in the Mel frequency scale in order to uncover semantic descriptors for audio "textures" that were generally agreed upon in human trials [18]. Aucouturier and Pachet from the Sony Computer Science Lab used Mel frequency cepstral coefficients or MFCC's extracted from a similar scale as a primary feature in their timbral similarity framework. They describe the features as a "good and compact representation

of the 'local timbre' of the [audio signal] frame" that they were analyzing [4]. Aucouturier would further analyze the use of MFCC's as a feature for timbral similarity algorithms in a later paper which revealed mixed results [41]. This reaffirms previous work done by Beth Logan from the Cambridge Research Laboratory on the usefulness of the Mel scale in which she finds its use for "speech [and] music discrimination" to be "at least not harmful for this problem" [28].

The aforementioned research deals primarily with the Mel scale as a measure of timbral similarity in the context of music information retrieval systems. Unfortunately, fewer studies have been done concerning the psychoacoustic scale or cepstrum extracted from it in other contexts, though it has appeared in some feature sets for work in granular synthesis. Judy Franklin made creative use of as many as 51 Mel frequency cepstrum coefficients as a feature in her reinforcement learning based granular synthesis engine that attempted to generate a tone that matched a recording [15]. In her future work section, she mentioned that exclusion of the Mel scale in favor of another psychoacoustical scale called the Bark scale could yield better results since it had been shown in at least in one case to improve accuracies for percussive instrument classification [8].

4.3 Polyphonic Timbre

Most studies concerning timbre in the context of audio analysis have dealt with monophonic sound sources, or sound generated from a single instrument or singer. However, several interesting attempts have been made to extract meaningful components from polyphonic sources. Kendall and Carterette compared and contrasted perceptual similarities between layered timbres using a technique called multidimensional scaling but made no use of temporal or spectral features, instead relying on human trials in which participants rate similarities between combinations of various instruments

as the source for their data [25] .

Alluri and Toiviainen were among the first to take a close look at polyphonic audio and the mapping of "semantic associations of listeners to polyphonic timbre, and to determine the most salient features of polyphonic timbre perception" [1]. Their motivations were similarly novel, noting in the same paper that the "deviati[on] from the well-known theories of Western melodic, harmonic, and rhythmic progressions" and movement towards "creating new sounds and textures by focusing on the blending of varied timbres" as a chief motivator for their work. An interesting result of the two experiments performed by the authors was an ability to map ambiguous semantic descriptors to certain empirical features. For example, the term "activity" was found to correlate with the energy in the spectral band between 1600 and 3200 Hz, "fullness" with the energy in the spectral band between 50 and 100 Hz, and the zero crossing rate of a signal was highly correlated with perceived "brightness". The results seemed contradictory to the claims of McAdams and Bregman many years before, who claimed that instead of being the result of a waveform, timbre was instead a "perceived property of a stream organization" [72]. In other words, they believed that a discussion of the timbre of a sound is impossible simply by viewing its waveform; instead it must be considered in the context of the tones played both before and after it. Alluri and Toivaine, however, showed a statistically significant correlation between several waveform properties and perceived qualities of sound.

Chapter 5

RESULTS

This section describes general results based on observations of different virtual instrument patches made using various feature combinations and k-means parameter configurations. The results of a survey given to the public concerning some hand picked instrument patches are also given, along with results of an experiment in timbral segmentation using Timcat.

5.1 General Findings

After iterating through many combinations of parameter configurations for the number and types of features over a wide variety of audio signal inputs, some general findings surfaced. Spectral features and zero crossing rate did not contribute positively to results in most cases. Instead, introduction of these features seemed to correlate with "clicking" noise artifacts and lower silhouette scores.

Setting the analyzer to cluster based on a low number of groupings, around 10 to 20, created groupings in which the grains did not sound very alike. Setting the number of clusters too high, around 100 or greater, resulted in redundant groupings, or groupings that sounded very similar. Finding the correct number of clusters is a difficult problem to determine *a priori* as discussed in 3.3. Instead of honing the number of clusters parameter, it was better to simply overestimate it and have shorter, redundant patches since they could simply be looped in the sampler.

Binergies as features produced very choppy sounding results which seemed to indicate that they were not useful as indication of timbre. The results did not seem completely random but were nonetheless very unpleasant to listen to. This was in

line with my hypothesis that a binning technique would be useful but the bins would have to be more carefully selected. Indeed, log binergies sounded much better in all cases. Sound groupings sounded consistent across their full play length which indicated that the grains were similar in timbre. The XBins feature however, which simply represented energy in 100 logarithmically spaced bins over the FFT, was much noisier than only 13 logarithmically spaced bins. Finally, MFCCs were the clear winner regardless of the input signal, creating consistent sounding patches that had the fewest noise and clicking artifacts.

Harmonic ratios also produced interesting, but noisy patches. The samples produced using only these four features were surprisingly consistent and different than the binning techniques, but noisy when compared to the MFCCs. When added together, the output was often very interesting and easy to listen to, indicating that the two types of features worked well together. Finally, adding in the root mean square energy to the MFCCs and the harmonic ratios produced smoother sounding results in most cases. Less chaotic patches were observed with fewer artifacts than with either of these two features alone. In the end, it was clear that grains grouped in 100 clusters based on a timbral space comprised of 13 MFCCs, 4 harmonic ratios, and the average root mean square energy were clear winners when attempting to make interesting, listenable timbres.

5.2 General Survey

In order to gather some feedback concerning some of the virtual instrument patches produced using Timcat, a survey was made available to the general public that asked for short descriptions or opinions about a selection of results. The survey given to participants can be seen in Appendix G. Four groups of instrument patches were provided to participants prepared using the most promising techniques that were

found as described in Section 5.1. Specifically, 13 MFCCs, 4 harmonic ratios, and the root mean squared energy of 20 mS grains were used as features for the synthesizer. The synthesizer performed k-means clustering over 100 clusters. Six to seven patches were hand picked from the hundred ensuring that they weren't too quiet or noisy and had some unique or interesting qualities.

The patches were then loaded into the Kontakt sampler by Native Instruments [50]. Kontakt allows a sound file to be used as a virtual instrument by resampling it and mapping it to a virtual keyboard. Kontakt also contains a rich set of features that allows modifying the instrument to better suit a performers needs. For the survey sounds, a basic filter was added with cutoff frequencies that resembled an AHDSR (attack , hold, decay, sustain, release) envelope. An example of the configuration of envelope for the sampler is shown in Figure 5.1.

The survey was distributed to the public through many channels including the Cal Poly Computer Science Department weekly mailer, the electronic music producer subreddit on the website Reddit.com, and the KVR digital signal processing and plugin creation forum at kvr.com. Participants were asked whether they were musician's, either electronic or other, to see whether having a music background effected their opinion. Sounds were linked within the online survey via Soundcloud.com for ease of access.

Figure 5.1: Kontakt ADHSR envelope configuration for virtual instruments used for the general survey.

It seemed important to show a range of notes for each virtual instrument without drawing the focus off of the timbre and onto the composition, so a very simple set of MIDI notes was played for each instrument patch. A basic MIDI track was created with a C2 on the piano keyboard played for 4 beats before playing an ascending C major scale over 16 beats at 120 beats per minute in a 4/4 time signature. An example of the MIDI track as shown on the piano roll can be seen in Figure 5.2. Because pitch normalization was left for future work as indicated in Section 7.3, the scales were not actually in the key of C, but rather in the key of the original patch's fundamental frequency.

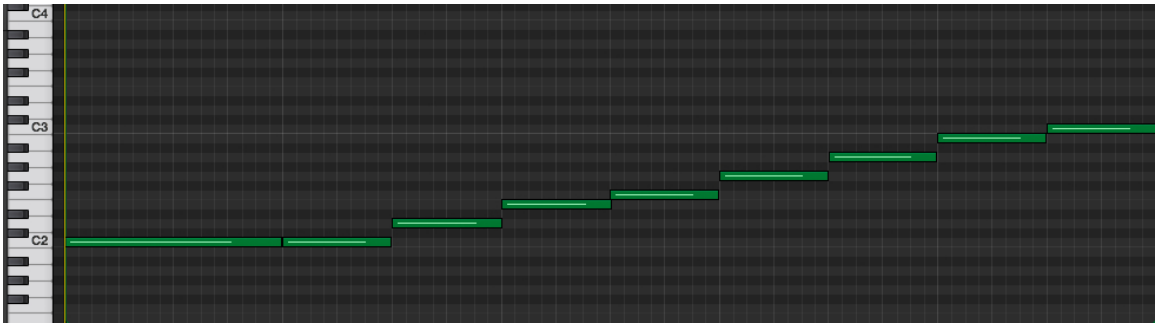


Figure 5.2: Scale played by the virtual instruments used for the general survey.

Four audio signals were used as input to Timcat. The song "The Earth is not a Cold Dead Place" by Explosions in the Sky was used because it was a polyphonic audio signal with a range of intensity that did not include any human voices [49]. A recording of a choir performing "Sleep" by Eric Whitacre was also used to demonstrate the systems response to a cascade of voices over a wide range of frequencies [14]. For similar reasons and because audio tracks of its kind tend to contain a large amount of overtone energy, a recording of throat singers was used [23]. Finally, an amateur recording of rainforest ambiance noises was used [80].

5.2.1 Survey Results

Responses to the four groups of audio files were varied with some common themes and can be found in Appendices A.0, B.0, C.0, D.0, E.0 and F.1. Many descriptions contained a reference to "wind" or "water" and "bubbling". The word "noisy" was mentioned many times which seems to be a characteristic of the audio that Timcat produces.

Interestingly, the fourth group of audio files made from ambient rainforest noise saw the most polarized responses. Some survey participants used words such as "pleasing", "interesting textures", and "pleasant to the ears" while others found some of the patches "earsplittingly squeaky" with "clicking" noises that "kind of ruined the sound" D.0.

In general, many survey participants found that Timcat produced a specific "type" of sound. One participant remarked that the sounds were all "fascinating in that they're not the type of sounds [he] could associate with any other production method" E.0. Several respondents felt that the audio produced by Timcat would be better suited for creating "soundscapes" or sound effects rather than virtual instrument patches and would benefit from post processing to remove some of the more displeasing artifacts E.0.

5.3 Timbral Segmentation Evaluation

It was clear that certain features were better suited as timbral indicators than others based on general observations of virtual instrument patches produced in various timbral spaces. In order to better quantify how good a feature was as an indicator of timbre, Timcat was provided two programmatically created audio files that were crafted by intermingling two different recordings of two different instruments playing

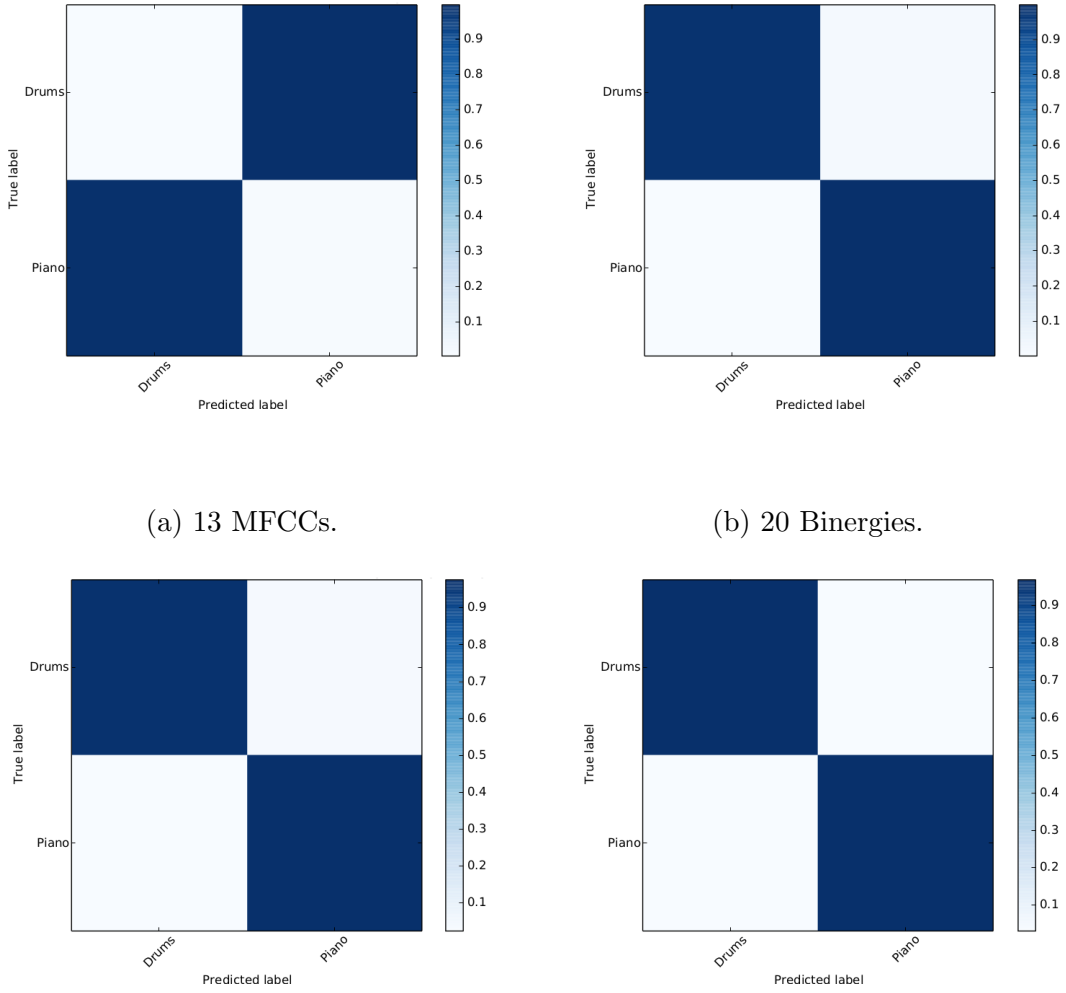
alone. More specifically, the two files were labeled either 0 or 1 and mixed randomly in 20 millisecond increments.

Each time 20 milliseconds of one file was added to the mixed track, its label was appended to an array such that by the end the order in which the files were mixed was accurately represented by the array. Timcat's granulizer component was then modified to include the label of each grain it produced along with its standard entry into the database. In this way, labeled data was created that could be cross referenced against the output of Timcat's synthesizer which was configured to cluster grains into two groupings. Because the clustering performed by the synthesizer labels groupings arbitrarily, the results of comparing known labels to those provided by the k-means algorithm are represented as confusion matrices for analysis.

5.3.1 Piano and Drums

The first experiment using the described evaluation method involved a drum and piano track playing constantly over 1 minute that, when mixed, resulted in an audio file that was 2 minutes long. The drum track was taken from a warm-up and solo performance by Travis Barker [35] while the piano track was taken from a performance by Jarrod Radnich [46]. The result of using features based on various filtering methods such as MFCCs and log binergies are shown in Figure 5.3. Binning techniques performed very well with accuracies in the high 99 percentiles. The high values in the off diagonal of the MFCC confusion matrix are likely due to the swapping of labels by the clustering algorithm.

The other features did not perform nearly as well as the filtering techniques as shown in Figure 5.4. The lack of a strong diagonal in most cases suggests that the clustering algorithm assigned too many grains to a single cluster. ZCR, spectral rolloff and all spectral features performed moderately well as features for segmentation pur-



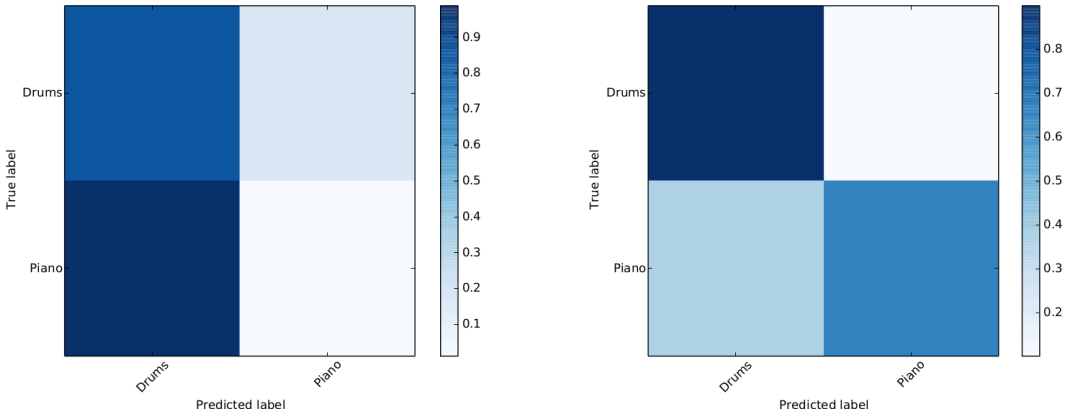
(a) 13 MFCCs.

(b) 20 Binergies.

(c) 13 Log Binergies.

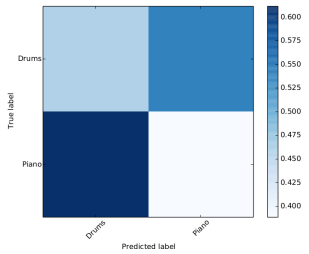
(d) 100 XBins.

Figure 5.3: Confusion matrices for the results of Timcat labeling piano and drum grains using filter bin energy based features.

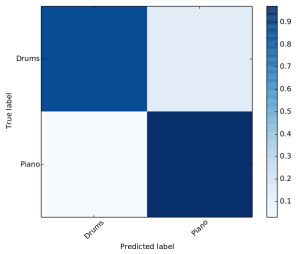


(a) RMS Energy.

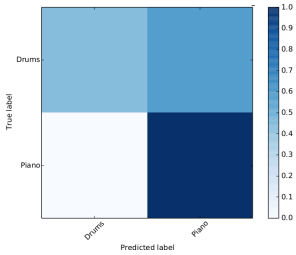
(b) Spectral Features.



(c) 4 Harmonic Ratios.



(d) Spectral Rolloff.



(e) ZCR.

Figure 5.4: Confusion matrices for the results of Timcat labeling piano and drum grains using RMS energy (a), all spectral features (b), 4 harmonic ratios (c), spectral rolloff (d), and zero crossing rate (e).

poses at around 75% accuracy, while harmonic ratios and RMS energy demonstrated poor accuracy.

The average silhouette scores and accuracy of clustering using each feature is show in Table 5.1.

5.3.2 Piano and Trumpet

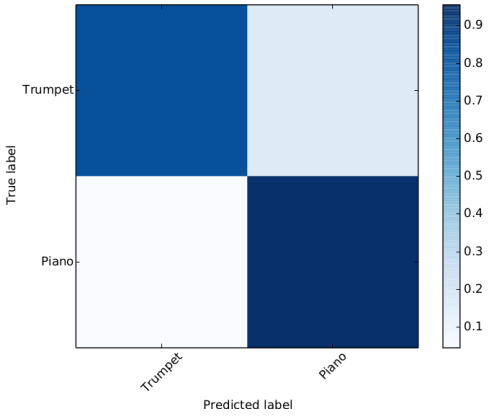
Next, an audio track was made in a manner similar to the one created in section 5.3.1 with a trumpet solo recording substituted in place of the drum recording [76]. Again,

	Silhouette Score	Accuracy
Spectral Rolloff	.76	74.00%
13 MFCCs	.59	100%
Spectral Features	.42	75.58%
Zero Crossing Rate	.76	74.00%
13 Log Binergies	.52	99.97%
20 Binergies	.46	99.28%
100 X Bins	.44	99.97%
4 Harmonic Ratios	.36	57.86%
RMS Energy	.74	60.89%

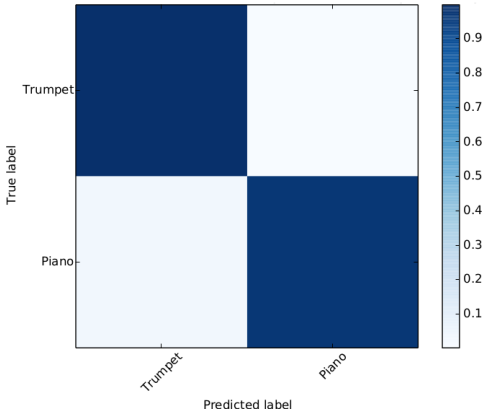
Table 5.1: Average silhouette scores and accuracy for clusters created by Timcat when analyzing the piano and drum track.

the filter bin energy techniques were very accurate as shown in Figure 5.5 except for the binergies feature. Again, since the off diagonal demonstrated high values in the confusion matrix it is probable that the clustering algorithm again swapped labellings. Correcting for this mislabeling revealed that the binergy feature produced the highest accuracy clustering by the synthesizer at 98.39%.

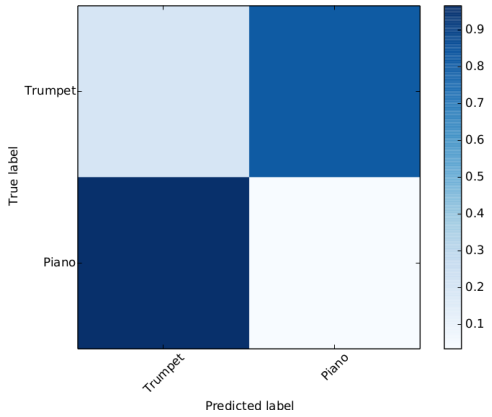
Similar to the previous experiment, the other features were not nearly as good as acting as indicators of timbre as the filter features as shown in the figures in Figure 5.6. The exception was the zero crossing rate which correctly labeled grains with 90.74% accuracy. The average silhouette scores and accuracies of clustering using the various features over the piano and trumpet tracks are shown in Table 5.2.



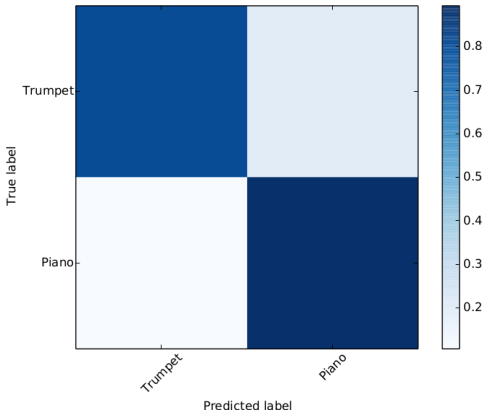
(a) 13 MFCCs.



(b) 20 Binergies.

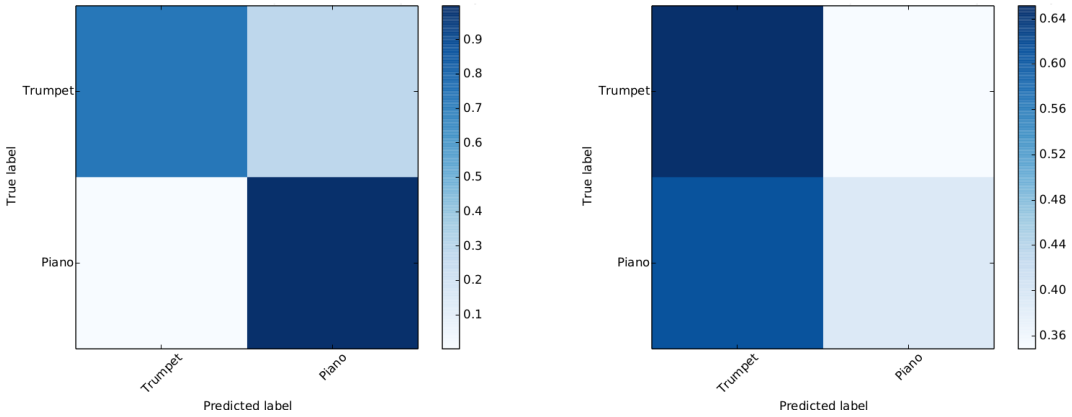


(c) 13 Log Binergies.



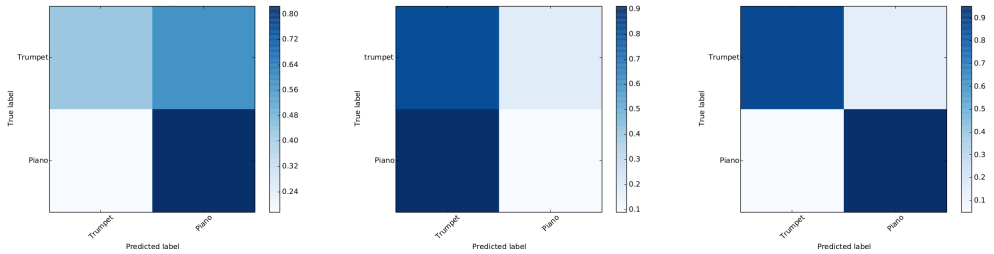
(d) 100 XBins.

Figure 5.5: Confusion matrices for the results of Timcat labeling piano and trumpet grains using filter bin energy based features.



(a) RMS Energy.

(b) Spectral Features.



(c) 4 Harmonic Ratios.

(d) Spectral Rolloff.

(e) ZCR.

Figure 5.6: Confusion matrices for the results of Timcat labeling piano and trumpet grains using filter bin energy based features.

	Silhouette Score	Accuracy
Spectral Rolloff	.70	54.4%
13 MFCCs	.42	89.82%
Spectral Features	.53	52.08%
Zero Crossing Rate	.66	90.74%
13 Log Binergies	.53	88.74%
20 Binergies	.20	98.39%
100 X Bins	.40	85.02%
4 Harmonic Ratios	.36	62.17%
RMS Energy	.76	86.07%

Table 5.2: Average silhouette scores and accuracies for clusters created by Timcat when analyzing the trumpet and drum track.

Chapter 6

CONCLUSIONS

In this thesis a software system for discovering novel timbres in prerecorded audio tracks was presented. Many features were evaluated for use in timbral spaces over which clustering was performed using various configurations of the k-means algorithm. Finally, concatenative synthesis techniques were used to generate sources for new virtual instruments.

Survey participants were asked to evaluate some of the sounds created by Timcat and provided mixed to generally favorable responses. While some found the audio produced by the software to be noisy and unpleasant, others enjoyed the instrument patches and saw the potential for the use of Timcat for making electronic music. It is this author's opinion that the sound files produced by Timcat would benefit greatly from further processing or manipulation as opposed to using the files "as is". Several survey participants felt this to be the case as well while others expressed that the sound files would serve much better as sound effects or ambient noise than sources for virtual instruments.

In addition to evaluating Timcat via survey, a basic test was presented to discover how Timcat performed when separating instruments in audio files based on several timbral features. Audio files were mixed and Timcat was made to separate them using its grain feature extraction methods. The use of filters over the frequency domain representation of grains were found to provide the most accurate features for segmentation. Zero crossing rate and some spectral distribution features was also found to perform moderately well for this purpose.

Timcat could prove to be a useful tool for musicians wishing to discover interesting

timbres for use in their electronic music. Requiring only a single monophonic audio file, the source material that Timcat uses to produce instrument patches abounds. Providing Timcat as a plug-in for modern digital audio workstations in VST or AU format would allow electronic artists to use Timcat from the comfort of their most familiar digital audio work environment and was one of the most requested features in forum responses concerning the project. Making Timcat more accessible and easily configurable in this way could allow for artists of all types to benefit from a software framework that produces interesting, novel timbres from their favorite audio sources, making it a promising addition to the electronic musicians toolkit.

Chapter 7

FUTURE WORK

7.1 Alternate Psychoacoustic Scales

Observing that features such as mel-frequency cepstral coefficients worked quite well for timbral recognition begged the question of whether other psychoacoustic scales would perform even better. Judy Franklin, for example, recommended the Bark scale, proposed by Eberhard Zwicker in 1961, as a possibly better tuned perceptual scale for audio segmentation and analysis [8]. Similar to the mel scale, the Bark scale maps frequency to Barks based on critical bands that "have been directly measured in experiments on the threshold for complex sounds, on masking, on the perception of phase, and most often on the loudness of complex sounds" [83]. Table 7.1 shows the critical bands as recommended by Zwicker. For speech recognition, this scale had been seen to perform worse than the mel scale but little could be found on its performance for music signal analysis [19].

Since MFCCs and BFCCs were proposed, perceptual linear cepstral coefficients or LPCCs have also been recommended for speech analysis. However, similar to BFCCs, their use for music signal segmentation and analysis remain unexplored. Hermansky presented the new analysis technique in 1989 which used "three concepts from the psychophysics of hearing to derive an estimate of the auditory spectrum: (1) the critical-band spectral resolution, (2) the equal-loudness curve, and (3) the intensity-loudness power law" [21]. However, in the same paper in which BFCCs were shown to be worse than MFCCs, Gulzar et al. show that LPCCs were not as performant as MFCCs for word recognition though it did perform better than BFCCs in all cases. Use of LPCCs as features for music signal analysis has not been explored in any depth

Number	Center frequencies Hz	Cut-off frequencies Hz	Bandwidth Hz
1	50	100	80
2	150	200	100
3	250	300	100
4	350	400	100
5	450	510	110
6	570	630	120
7	700	770	140
8	840	920	150
9	1000	1080	160
10	1170	1270	190
11	1370	1480	210
12	1600	1720	240
13	1850	2000	280
14	2150	2320	320
15	2500	2700	380
16	2900	3150	450
17	3400	3700	550
18	4000	4400	700
19	4800	5300	900
20	5800	6400	1100
21	7000	7700	1300
22	8500	9500	1800
23	10500	12000	2500
24	13500	15500	3500

Table 7.1: Critical bands of the Bark scale [83].

but may prove useful.

7.2 Alternate Non-Cepstral Features

This paper explores a small subset of features that have been shown or hypothesized to be good characterizations of timbre. There are, however, a slew of other features that could prove useful in timbral spaces for use in the clustering methods contained herein.

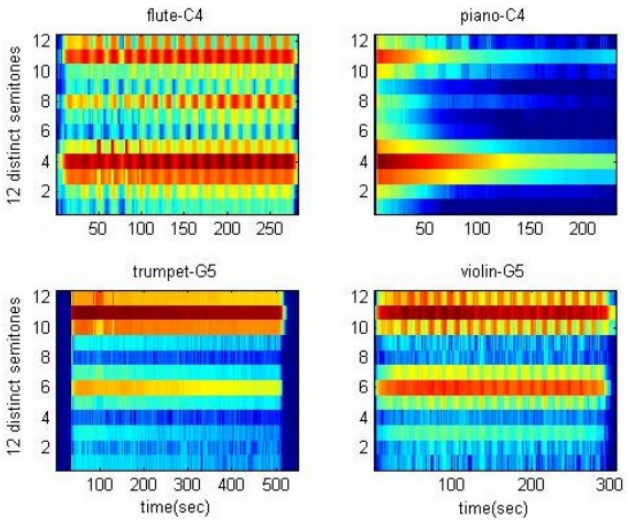


Figure 7.1: Chromagrams of four instruments [20].

In musical information retrieval, for example, chroma contours have been shown to be promising features for instrument characterization [20]. Use of chroma usually involves the construction of a "chromagram" which is defined as "the whole spectral audio information mapped into one octave" which is then divided into 12 bins representing semitones [20]. Determining energy levels in each of the bins allows the creation of features which could be used in timbral spaces. An example of several chromagrams is shown in Figure 7.1.

One of the most recent features which is of extreme interest due to showing great

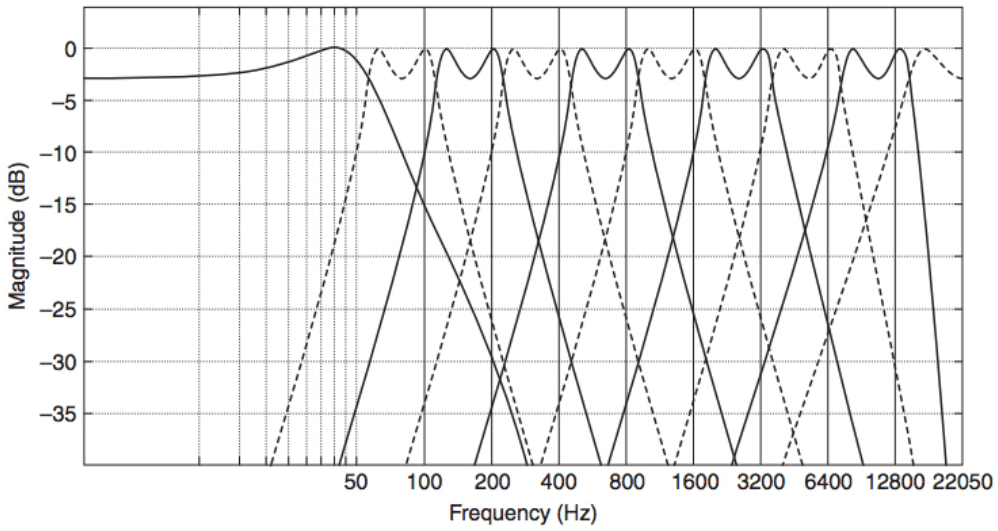


Figure 7.2: Frequency response for the 10-channel filterbank used to obtain SBFs [1].

promise in characterizing *polyphonic* timbre is the Sub-Band Flux or SBF which represents the "fluctuation of frequency content in ten octave-scaled bands of the spectrum" [1]. By binning FFT energies based on octave-scaled filterbanks and computing the Euclidean distance between successive bins, Alluri and Toiviainen showed that SBF was highly correlated with perceived "activity" of a sound [1]. The frequency response of the 10-channel filterbank used to obtain the SBF of a windowed signal is shown in Figure 7.2.

7.3 Pitch Normalization

Using features based on energies calculated from logarithmically spaced bins naturally captures some aspects of pitch. However, varying the number of these bins, their spacing, and adding new features into the timbral space eschews any pitch data that may have been gleaned during clustering. It is therefore incorrect to assume that grains clustered together with high dimensional timbral spaces are of the same pitch.

Because the end goal of concatenative synthesis of timbral grain groupings is to create a new virtual instrument patch, it would be intelligent to pitch-adjust the grains before concatenation so they are in the same key. For use in something such as a sampler which allows the virtual instrument patch to be played in any key, Ensuring that the instrument patch is in the key of C4 would help to make the patches more viable for use in a sampler such as Kontakt [50]. An example of resampling and applying gain for pitch adjustment in concatenative synthesis are described by Einbond, Trapani and Schwarz in a 2012 paper concerning their 2006 project CataRT [13, 60]. Further efforts should be made to detect the primary pitch of the grains and adjust them before the concatenation phase.

7.4 Clustering Techniques

K-means was used as the clustering algorithm of choice for grouping grains due to its simplicity and speed. There are, of course, many other clustering algorithms that could prove to be much better for the purposes of novel timbral creation. Clustering "on the fly", for example, would not be practical using K-means on a high dimensional timbral space due to its time complexity inefficiently scaling as axes are added to the input vectors. In its worst case, K-means has been shown to be as complex as $O(n^{k+\frac{2}{p}})$, where k is the number of clusters, n is the number of data points, and p is the number of features per vector [2].

This problem, called the "curse of dimensionality" by Richard Bellman, plagues all types of data analysis and clustering techniques that require optimization based on a large number of variables [5]. Recent efforts have revealed promising techniques for dispelling this curse and may be well suited for clustering in timbral spaces. Steinbach et al. recommend many techniques such as hypergraph partitioning, grid based clustering, CLIQUE, and Merging Adaptive Finite Intervals [71].

At the very least, several improvements to standard K-means which is usually a specific implementation of Lloyd's algorithm should be explored. Some new techniques allow for the weighting of certain features in the feature space based on unsupervised learning [32]. This could possibly produce better groupings without requiring the meticulous and subjective observation and decision making process that is determining which features are helpful or harmful. Similar approaches simply eliminate redundant features altogether [75].

BIBLIOGRAPHY

[1] V. Alluri and P. Toiviainen. Exploring perceptual and acoustical correlates of polyphonic timbre. *Music Perception: An Interdisciplinary Journal*, 27(3):223–242, 2010.

[2] D. Arthur and S. Vassilvitskii. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153. ACM, 2006.

[3] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[4] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR*, 2002.

[5] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[6] A. Benade and S. Kouzoupis. The clarinet spectrum: Theory and experiment. *The Journal of the Acoustical Society of America*, 83(1):292–304, 1988.

[7] J. W. Beuchamp. Synthesis by spectral amplitude and" brightness" matching of analyzed musical instrument tones. *Journal of the Audio Engineering Society*, 30(6):396–406, 1982.

[8] W. Brent. Perceptually based pitch scales in cepstral techniques for percussive timbre identification. In *ICMC Proceedings*, 2009.

[9] P. M. Brossier. Aubio, a library for audio labelling. http://aubio.piem.org, 2003.

[10] J. J. Burred, A. Röbel, and T. Sikora. Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(3):663–674, 2010.

[11] A. De Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[12] D. Deutsch. *Psychology of music*. Elsevier, 2013.

[13] A. Einbond, C. Trapani, and D. Schwarz. Precise pitch control in real time corpus-based concatenative synthesis. In *International Computer Music Conference Proceedings*, volume 2012, pages 584–588. International Computer Music Association, 2012.

[14] V. E. C. Eric Whitacre and E. Singers. Sleep. https://www.youtube.com/watch?v=9shXm0cIeEY, May 2016.

[15] J. A. Franklin. Generating soundwaves via granular synthesis and reinforcement learning. Smith College, preprint on webpage at http://www.cs.smith.edu/ jfrankli/papers/index.html.

[16] O. Gillet and G. Richard. Automatic transcription of drum loops. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 4, pages iv–269. IEEE, 2004.

[17] J. M. Grey and J. W. Gordon. Perceptual effects of spectral modifications on musical timbres. *The Jouranl of the Acoustical Society of America*, 63:1493–1500, 1978.

[18] T. Grill. Constructing high-level perceptual audio descriptors for textural sounds. In *Proceedings of the 9th Sound and Music Computing Conference (SMC 2012), Copenhagen, Denmark*, pages 486–493, 2012.

[19] T. Gulzar, A. Singh, and S. Sharma. Comparative analysis of LPCC, MFCC and BFCC for the recognition of hindi words using artificial neural networks. *International Journal of Computer Applications*, 101(12):22–27, 2014.

[20] G. E. Hall, H. Ezzaidi, and M. Bahoura. Instrument timbre chroma contours and psycho-visual human analysis. In *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*, pages 327–330. IEEE, 2014.

[21] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.

[22] A. Houtsma. Pitch and timbre: Definition, meaning and use. *Journal of New Music Research*, 26(2):104–115, June 1997.

[23] Huun-Huur-Tu. Huun-huur-tu - live. https://www.youtube.com/watch?v=i0djHJBAP3U, May 2016.

[24] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

[25] R. A. Kendall and E. C. Carterette. Perceptual scaling of simultaneous wind instrument timbres. *Music Perception: An Interdisciplinary Journal*, 8(4):369–404, 1991.

[26] M. Lagrange, R. Badeau, and G. Richard. Robust similarity metrics between audio signals based on asymmetrical spectral envelope matching. In *Acoustics*

Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 405–408. IEEE, 2010.

[27] S. P. Lloyd. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[28] B. Logan et al. Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information*, 2000.

[29] E. Maestre, R. Ramírez, S. Kersten, and X. Serra. Expressive concatenative synthesis by reusing samples from real performance recordings. *Computer Music Journal*, 33(4):23–42, 2009.

[30] M. M. Martin Nilsson. *ID3v2*, June 2016.

[31] S. McAdams, S. Winsberg, S. Donnadieu, G. De Soete, and J. Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological research*, 58(3):177–192, 1995.

[32] P. Mitra, C. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):301–312, 2002.

[33] S. Molau, M. Pitz, R. Schluter, and H. Ney. Computing mel-frequency cepstral coefficients on the power spectrum. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, volume 1, pages 73–76. IEEE, 2001.

[34] MongoDB, Inc. *Mongo DB*, 6 2016.

[35] MrMegaDrummer. Travis barker - drum solo & warm up. https://www.youtube.com/watch?v=Aw7w98hI-ic, 2011.

[36] V. Nikulin and G. McLachlan. Regularised k-means clustering for dimension reduction applied to supervised classification. In *CIBB Conference, Genova, Italy*, 2009.

[37] M. Norton and D. Karczub. *Fundamentals of Noise and Vibration Analysis for Engineers*. Cambridge University Press, 2003.

[38] A. H. Omar. *Audio segmentation and classification*. PhD thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2005.

[39] A. V. Oppenheim and R. W. Schafer. From frequency to quefrency: A history of the cepstrum. *Signal Processing Magazine, IEEE*, 21(5):95–106, 2004.

[40] D. O'Shaughnessy. *Speech communications: human and machine*. Institute of Electrical and Electronics Engineers, 2000.

[41] F. Pachet and J.-J. Aucouturier. Improving timbre similarity: How high is the sky. *Journal of negative results in speech and audio sciences*, 1(1):1–13, 2004.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. preprint on webpage at http://recherche.ircam.fr/, 2004.

[44] D. T. Pham, S. S. Dimov, and C. Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.

[45] G. Proakis John and G. Manolakis Dimitris. Digital signal processing: principles, algorithms, and applications. *Pentice Hall*, 1996.

[46] J. Radnich. Star wars - fantasy suite, movement #2 - jarrod radnich virtuosic piano solo 4k. https://www.youtube.com/watch?v=IOL9OvxWqh0, 2015.

[47] C. Rauscher. *Fundamentals of spectrum analysis*.

[48] American National Standards Institute and M. Sonn. *American National Standard: Acoustical Terminology*. American National Standards Institute, 1973.

[49] Explosions in the Sky. The earth is not a cold dead place. https://www.youtube.com/watch?v=Ziw4yd5R0QI, May 2016.

[50] Native Instruments. Kontakt 5. http://www.native-instruments.com/en/products/komplete/samplers/kontakt-5/, 2016.

[51] D. Rocchesso. *Introduction to sound processing*. Mondo estremo, 2003.

[52] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[53] M. Russ. *Sound synthesis and sampling*. Taylor & Francis, 2004.

[54] D. Rys. Global Electronic Music Industry, Worth $7.1 Billion Last Year, Sees Growth Slow. http://www.billboard.com/articles/business/7385168/global-electronic-music-industry-growth-slows-still-worth-billions, 2016. [Online; accessed 30-May-2016].

[55] M. Sahidullah and G. Saha. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition. *Speech Communication*, 54(4):543–565, 2012.

[56] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1331–1334. IEEE, 1997.

[57] J. F. Schouten. The perception of timbre. In *Reports of the 6th International Congress on Acoustics*, volume 76, August 1968.

[58] E. Schubert and J. Wolfe. Does timbral brightness scale with frequency and spectral centroid? *Acta acustica united with acustica*, 92(5):820–825, 2006.

[59] E. Schubert, J. Wolfe, and A. Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proc. 8th Int. Conf. on Music Perception & Cognition (ICMPC), Evanston*, 2004.

[60] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton. Real-time corpus-based concatenative synthesis with CataRT. In *9th International Conference on Digital Audio Effects (DAFx)*, pages 279–282, 2006.

[61] D. Schwarz et al. *Data-driven concatenative sound synthesis*. PhD thesis, University Pierre and Marie CURIE, 2004.

[62] J. G. S.Essid, T.Fillon. Yaafe, an easy to use and efficient audio feature extraction software. In *Proceedings of the 11th ISMIR conference*, Utrecht, Netherlands, 2010, 6 2010. Telecom Paristech, AAO Team.

[63] D. Shete, S. Patil, and P. Patil. Zero crossing rate and energy of the speech signal of devanagari script. *IOSR-JVSP*, 4(1):1–5, 2014.

[64] T. Shirk. eyed3. http://eyed3.nicfit.net/, April 2016.

[65] K. Singh, D. Malik, and N. Sharma. Evolving limitations in k-means algorithm in data mining and their removal. *International Journal of Computational Engineering & Management*, 12:105–109, 2011.

[66] M. Slaney. Auditory toolbox. *Interval Research Corporation, Tech. Rep*, 10:1998, 1998.

[67] J. O. Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K Publishing, 2010.

[68] J. O. Smith. *Spectral audio signal processing*. W3K, 2011.

[69] S. W. Smith et al. *The scientist and engineer's guide to digital signal processing*. California Technical Pub. San Diego, 1997.

[70] J. Y. Stein. *Digital signal processing: a computer science perspective*. John Wiley & Sons, Inc., 2000.

[71] M. Steinbach, L. Ertöz, and V. Kumar. The challenges of clustering high dimensional data. In *New Directions in Statistical Physics*, pages 273–309. Springer, 2004.

[72] A. B. Stephen McAdams. Hearing music streams. *Computer Music Journal*, 3(4):26–43, Dec. 1979.

[73] S. S. Stevens. The relation of pitch to intensity. *The Journal of the Acoustical Society of America*, 6(3):150–154, 1935.

[74] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[75] W. Sun, J. Wang, Y. Fang, et al. Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, 6:148–167, 2012.

[76] H. Wahab. Solo trumpet performance by aiman. https://www.youtube.com/watch?v=yfr-awXCxe8, 2012.

[77] J. Watkinson. *The art of digital audio*. Taylor & Francis, 2001.

[78] S. Wun, A. Horner, and B. Wu. Effect of spectral centroid manipulation on discrimination and identification of instrument timbres. *Journal of the Audio Engineering Society*, 62(9):575–583, 2014.

[79] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian. HMM-based audio keyword generation. In *Advances in Multimedia Information Processing-PCM 2004*, pages 566–574. Springer, 2004.

[80] S. M. . Y. YogaYak. Rainforest sound 11 hours. https://www.youtube.com/watch?v=-IAfgIy7n8, May 2016.

[81] R. Yorita and J. Clements. Using spectral analysis to evaluate flute tone quality. In *Proceedings of Meetings on Acoustics*, volume 23, page 035001. Acoustical Society of America, 2015.

[82] A. Zils and F. Pachet. Musical mosaicing. In *Digital Audio Effects (DAFx)*, volume 2, page 135, 2001.

[83] E. Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.

APPENDICES

Appendix A

SURVEY RESPONSES, GROUP 1

Responses
Feels like I'm in a Stanley Kubrick film. It's sounds like chaos and horror disguised by major key overtones.
wobbly, holllow, detuned, windy, unstable, with background "heartbeat"
The higher pitched notes in this sample tend to have an almost hazy quality around the sound but the effects ranged from spacey to underwater, to sci-fi.,Also, because the segments were chopped together, there is a second layer of notes that follows the same scale, which is interesting.,The sound most reminds me of a synth from the 80s. Samples 4 and 6 sounded the most natural of the set. This sound could be used as a synth background in music.
Many of these have a wavering, under water feel (especially Sample 2), Sample 3 has an outer space feel to it and the slight hiss makes it feel almost industrial (thinking Total Recall or Dead Space). Sample 5 has a heavy slowness to it and Sample 8,sounds ethereal with a metallic tinge to it. Overall these have an atmospheric quality that I imagine would work well in the back of a mix - a kind of rumbling foundation for a song to play out over.
The samples had an overall haunting quality to them.

These are all very noisy, which would actually make them excellent for filter sweeps or patterned ambience. Some of them are very bassy, which could make them inarticulate. If I were to use them, I'd probably have to use a high-pass to cut some of this out.
The sound clip sounds like a windy echo with a kind of percussive thumping in the background. Sort of like wind across an open bottle, or the sound of blood rushing past as you listen to a stethoscope or put a seashell up to your ear. Generally, there appear to be two main aspects of these sounds, the wind/blood rushing and the percussion/heartbeat. The tracks, with respect to these two aspects, tend to be somewhere on a scale between these (wind seashell and heartbeat percussion). Sample1 sounds particularly like wind but with a heartbeat in the background. Sample4 sounds especially like the wind across a bottle, but with a children's choir singing mixed in and a bouncy percussion. Sample5 sounds more impactful on the percussion and with a more burbly wind sound, like wind against a microphone. Sample8 sounds especially percussive, almost exactly like a hammer against a brake drum.
Sounds like a time stretched ableton/cubase sample
Sample 4 was very crisp sounding. Sample 5 seemed to have more variance. Sample 8 seemed to have more noise in the background.
Many samples sounded bubbly or rumbly, like being underwater; those qualities make me think of containment or destruction.,It seemed like many of the sounds were pretty mellow yet uneasy or threatening at the same time.
Interesting but not particularly musical. A bit "spectral mush" sounding as well, like musical noise from bit-rate compression. Also the obvious periodicity as the pitch rises sounds a bit naff. Sample 8 was cool tho, quite bell-like
Sounded the most natural of the bunch.

I know that explosions in the sky use 4 guitars, and this sounds nothing like a guitar. This instrument would work great when composing futuristic science fiction beats. I could see it being used on a Deltron song.
Having been primed with these coming from Explosions in the Sky, I can definitely feel the vibe (especially from tracks 1, 4 and 6), pretty cool. I also thought that it was odd that track 4 was so much different than the rest with its higher volume and higher tone. Also the mixing of the wind noises into the notes (which seemed to increase with higher pitch) also made it seem more like Explosions in the Sky.
Having been primed with these coming from Explosions in the Sky, I can definitely feel the vibe (especially from tracks 1, 4 and 6), pretty cool. I also thought that it was odd that track 4 was so much different than the rest with its higher volume and higher tone. Also the mixing of the wind noises into the notes (which seemed to increase with higher pitch) also made it seem more like Explosions in the Sky.
Sounds very "woobly", not very stable. The fourth and eighth sample are exceptions and are the only sounds I would consider usable.

Table A.1: Survey responses to first group of patches.

Appendix B

SURVEY RESPONSES, GROUP 2

Responses
Sounds just like the last one.
pretty much the same as Group 1.
In this set, I could hear an audial ripple in some of the tracks, most notably samples 1 and 5. I could tell by set 3 the most clearly that it was a natural sound, but it sounded the most like a church organ. This set does also have some of the hazy quality, but does not have the same sci-fi feeling as from the first group. This sound could probably be used in a wide range of scenarios since it sounds the most natural.
Sample 1 sounds kind of vocal, especially toward the higher notes. Sample 3 is similar, but also sounds like the low end of an orchestra. Sample 6 is ominous and windy/watery.
Choral performance lends itself to this technique. With the notes rising it feels like a many people crescendoing.
The watery sounding artifacts in the first two samples are very cool. Third one has a really neat hollow-pipe type of attack. Similar to group one, the noise would have to be filtered pretty heavily to make melodic sequences from these samples.

Sample 1: Hellfire. Screaming, trill-like sounds. Wind rushing through a PVC pipe. Sample 2: Impacts like someone smacked my earbuds while they're in my ear for the lower. More like a ringing tambourine for the higher notes. Submarine drone overall. Sample 3: Sounds primarily like a synth chaoir. Specifically like something out of The Legend of Zelda: The Wind Waker. Less dull percussion, more ringy, like a snare drum. Sample 4: More muddled drone. Percussion sounds like that impact setting on cheap keyboards. Sample 5: Like a large, intentionally out of tune choir. As the notes get higher, it's easier to notice that they're a choir. Sample 6: Percussion still sounds like a snare, but maybe more like a cymbal (one of the smaller, rattlier ones). Very muddled drone.
The attack is so punchy
Samples 1 and 2 seemed to be fairly mellow. Sample 3 was much crisper. Samples 4 and 5 back to a blander sound, followed by 6 which was extremely mellow sounding.
The sounds had unsettling intervals that sounded vaguely demonic, especially at higher pitches. Sounds 1, 2, 4 had a certain wind-like nature to them. Sample 5 had a subtle choir voice-like quality about it that made it especially eerie, especially when combined with what sounded like reverb at the end.
Pretty similar to group 1. Probably good for sound fx or moody stuff
Sounded like Group 1 with a chorus effect applied. Didn't like the heart beat like sound in the background.
All the samples sound very frenetic and vibration-filled. Sample 3 is my favorite, and would work great in the composition of a retro video-game sounding song.
As the tracks progressed it seemed like they got more ethereal, with the fifth track sounding almost hollow and the six track being pretty dull (not in a bad way, just dull like a drum without much spring). I could see this instrument being good for expressing anxiety or tension.

They all sound very similar even similar to the sounds from group 1. They all have a plucky start and the airy instable sound.

Table B.1: Survey responses to second group of patches.

Appendix C

SURVEY RESPONSES, GROUP 3

Responses
Sounds just like the last one.
somewhat higher pitched, similar to other groups.
Having heard Mongolian throat singing in person before, sample 6 sounds the most natural of them all, but only sample 5 had a non-natural sound. There is a little less of an audial haze or ripple throughout the set than there were from other sets so far. However, there was almost no identifiable instrument this would sound like since the note progression was the most staccato. This sound could probably be used in a wide range of scenarios as it is mostly natural sounding.
Overall these sound more sharp and metallic. Sample 6 has a more musical feel to it than the others, like it could make up the melody in a Grimes song. The rest of the samples are deeper and more foreboding.
The samples had a human aspect that I wouldn't normally associate with distinct computer generated notes. This was more apparent in the lower, guttural sounding notes.
The quickly decaying brightness on these note hits would make them a bit distracting in a background for anything but sharp transitional atmosphere.

Sample 1: Breathy drone. Twangy percussion, like suddenly stretched metal cables (or in movies when the suspension on a bridge fails) Sample 2: Steel drum percussion with a bit of echo, I think. It sounds like they're hitting the steel drum hard enough to break/dent it. The rest is just kind of a noisy drone. Sample 3: The drone is more noticeable in this one, maybe the higher notes or frequencies. Still a harsh steel-drum-y sort of percussion. Sample 4: Noticing a weird chirpy reverb in the impact. Drone in the background is pretty noisy and a wash. Sample 5: Drone gets warbly as it goes higher in pitch. Percussion is less impactful, still sort of steel drum-y. Sample 6: Choir sound is more noticeable. That low noise is quieter. The percussion is less impactful. Kind of like someone is hitting something with sandy characteristics.
sounds like reverb from hell
Sample 1 was crisp, 2 had more noise, 3 sounded sharper again, 4 had more noise than 2, 5 had the most noise, 6 had a sharp component.
Percussion in sample 1 was like a xylophone mixed with a gong. Sample 3's percussion sounded metallic, like the clinking of tools. Sample 5 has a growling quality about it, like an angry tiger, especially at lower pitches. Sample 6 sounds like being stuck in a windy cave at lower pitches and like walking on a metal platform at higher pitches.
Very similar again.... surprising given the source material of this one
Sounded the most unrealistic, harsh, aliased.
You can definitely hear some scrambled vocals in these samples. Sample6 is probably my favorite, as the white noise provides a good backdrop for the vocals.
These notes seemed to have a trailing garble to them, where the note would get to its highest point and fall out from under itself to be replaced with just some scratchy white noise (for like a split second). I guess it sounded similar to Group 2 for me.
It's interesting how all sounds have the same basic characteristic. They sound thiner, duller brighter, whatever, but they still aren't very distinctive.

Table C.1: Survey responses to third group of patches.

Appendix D

SURVEY RESPONSES, GROUP 4

Responses
Pretty much the same, except it sounds kind of like an underwater choir of robots.
most interesting, a bit of texture here, otherwise similar to other groups.
This set had the most obvious audial ripples, and besides sample 5, the rest all sounded somewhat like wind or tune static from a radio (like in a ghost movie/Paranormal Activity). Again, the note progression was extremely staccato. This sound could probably be used in a horror movie as wind or ghosts.
Sample 1 at first sounds like a car burning out then becomes very insect like. Sample 2 reminds me of chimes for some reason, there's a sort of cluster of sounds with each hit. Sample 3 has a warbly raygun feel to it. The low end of Sample 4 sounds like a communications signal and as it ramps up it sounds like alien communication sound effects from a sci-fi movie. Sample 5 is ethereal and pleasant. Sample 6 sounds like the classic submarine radar sound effect. Sample 7 sounds like a theremin or a keyboard. Overall this set feels the most diverse.
These samples are the most pleasing group. The background noise felt structured similar to what I would expect rain-forest white noise to sound like. Notes arising from it made it seem as if the ambient noise was singing.
This bunch is very cool, really interesting textures. I'm really curious what the first couple and the 6th would sound like pitched down. The warbling in the third sample seems steady enough to actually fit to tempo in a song, if the pitch alterations were held time-steady relative to the song.

Sample 1: Very chirpy, burbly drone sound. Much much less noisy than the previous ones. Doesn't sound like I'm in a submarine anymore. Sounds a little compressed, though. Has that kind of glassy sound that you get when you turn bitrate really low on an mp3. The percussion is less intense than previously. More like hitting a drum with a thin skin on it or something. Sample 2: Sounds like I'm using one of those cheap bird whistles that you fill with water, but there's, like, not enough water in it. The percussion is difficult to parse. Sounds much less like a drum. Sample 3: There's that submarine sound, but with a bubbly mixed in that speeds up as the notes increase. Percussion sounds very distant. Sample 4: Sounds like propellers on a biplane starting up instead of bubbles, including wind against a mic noise. Very high pitched percussion, where the attack lasts a little longer (like your'e breaking 5 panes of glass instead of just 1). Sample 5: Sounds underwater, but not necessarily in a submarine, and with some of that windy breathiness across a bottle. Very light, but still precise percussion. Sample 6: Very high pitched. Bubbly/boily sound. Hurts my ears. Sounds like of like those sorting algorithm sound videos. Thin percussion. Sample 7: Sounds like wind swirling in a metal bottle. A little like there's some singing mixed in. That glassy compression sound is more noticeable. Light smacking for the percussion.
way more highs and mids, transients sound filtered
Sample 1 sounds like rain in the background. Sample 3 has what sounds like wind. This set was more interesting and varied than the other three.

I found sample 1 earsplittingly squeaky. I feel like I could hear distorted water from a bubbling brook along with frogs, though the sound resembled bats at a certain point. Sample 2 sounded like old-school sci-fi "beep boop" style computers. The general trend for samples 36 was that they sounded a little unpleasantly distorted and electronically bubbly, with a sound of wind in the background. I suppose #4 had a splashing sound about it, like dropping a rock into a pool of water. And sample #6 made me think of what a raindrop must sound like to an ant. Sample 7 reminded me of a squeaky wheel, like somebody had to step on a pedal to make a wheel move... Were old sewing machines like that or am I thinking of something else? It's hard to describe.
Pretty cool! Kind of like space probe recordings :)
Sounded industrial, didn't like the clicking of some sort of transient throughout, kind of ruined the sound.
All of the samples had an interesting watery sound that was pleasant to the ears - sample 5 was my favorite, and it was probably the most watery-sounding. The sounds on samples one and two were very high-pitched.
Track 1 sounded like when the wind blows a tree against a window (screechy). Track 2 sounded like you were underwater and some crazy little seal was swimming around making noise. The rest sounded like you were underwater to varying amounts.
The second sample has the instability focused to the point where it sounds a bit like birds chriping

Table D.1: Survey responses to fourth group of patches.

Appendix E

SURVEY RESPONSES, GENERAL THOUGHTS

Responses
No Response
No Response
The only thing that was distracting was the audial ripple heard on each note in the progression in the rainforest and sleep sets. It sounds like there is a progression within each note of the progression, like a small quick wave making up the parts of a larger wave. This is a really cool idea, too.
The sounds had a vintage, atmospheric feel overall, with some more defined samples spread throughout that had an almost orchestral feel to them. They often made me think of those cryptic recordings of "numbers stations" broadcasts.
I like the novel-ness of the technique and how it was applied to a wide range of audio samples. It could be improved on with some post processing to reduce some of the higher frequency noise and add increase some of the mids. I think this type of filter could be used for creating more musical background soundscapes with finer control.
They're all fascinating in that they're not the type of sounds I could associate with any other production method. These would be enormous patches on a modular synth, or really really heavily edited and effected ambience recordings. Most of them are too noisy to make melodic instruments, it'd be really difficult to edit them into memorable patterns. But they'd be amazing sound effects, or an alternative to traditional "noise" like cymbals or dripping-in-reverb synth/guitar sounds.

Sorry if I wrote too much, I was just trying to describe what I could hear. I tried to describe the two aspect of each that I noticed among all of them: the drone sound and the percussion. For the most part, the drone sound sounded like I was in a submarine and the percussive sounds sounded generally like metal.
floating points would be proud
Impressive that these tracks were compiled from grains of music. As mentioned above, I thought group 4 had the most variation which made it the most interesting to me.
I can't say that I found the sounds particularly pleasing or that they alone drew a significant level of emotion out of me; but that said, I'd be very interested to hear them in context of something other than a scale. Many of the sounds I described as electronic or bubbly reminded me of that one sorting algorithm video on YouTube for some reason: https://youtu.be/kPRA0W1kECg.
I love the idea of using some machine learning techniques in synthesis (and would love to see more of this!), but i think it needs work, even if just to get more varied timbres from different source music. I'm thinking in pretty standard terms of music though, and I could see this being useful for more abstract, noisy music or sound fx.
Interesting sounds, though most had artifacts/issues which spoilt them.
I think this project could prove extremely useful in generating new virtual instruments from existing songs. It could facilitate the creation of some very interesting electronic music.
I think that Group 1 and Group 4 were my favorites. I liked Group 1 because it wasn't overly screechy and I imagine it could remind me of Explosions in the Sky if used in a song. I liked Group 4 because it was weird and I haven't really heard anything like that before.

I expected more diversity, but your process of creating these sounds seems to produce a very distinktive sound. They all have this plucky start, this airy noise and this instable wobble.

Table E.1: Survey responses to general thoughts about the Timcat patches.

Appendix F

SURVEY RESPONSES, RESPONDENT CLASSIFICATION

Response
Musician - General
None
Musician - General
Musician - General
None
Musician - General
Musician - General
Musician - Electronic Artist
None
Musician - General
Musician - Electronic Artist
None
None
None
Musician - Electronic Artist

Table F.1: Survey responses to self classification as either Musician - Electronic Artist, Musician - General, or None.

Appendix G

GENERAL SURVEY

Thesis Survey

Thank you for participating! The purpose of this survey is to gain feedback concerning virtual instrument patches created using a novel technique proposed as part of my thesis work. The audio files you will be listening to were created by chopping longer audio tracks into 20 millisecond segments called "grains", performing signal analysis on these grains to obtain various features, then using clustering techniques over the resulting feature vectors in order to group similar grains together. Each group of grains is then concatenated together to form a sample that is in turn used by a program as a virtual instrument.

When evaluating, please be descriptive and consider only the timbre, or quality of the sound you are listening to, instead of the specific notes or composition. All of the audio files are of a simple ascending major scale played using the various virtual instrument patches produced using the method described above.

Feedback is open response and a sentence or two to a paragraph per response will suffice. Feel free to draw comparisons to other instruments or sounds you are familiar with, describe emotions or feelings the instruments do or do not evoke, or discuss situations the instrument might be used in. By submitting this survey, you agree to have your responses included anonymously in my thesis report.

Thank you again for your help!

* Required

1. **Group 1** *
   The following samples were created from The Earth Is Not A Cold Dead Place by Explosions in the Sky. Please listen to them and comment on individual samples, or the group as a whole: https://soundcloud.com/neobonzi/sets/timcat-eits

2. **Group 2** *
   The following samples were created from a choral performance of "Sleep" by Eric Whitacre. Please listen to them and comment on individual samples, or the group as a whole.
   https://soundcloud.com/neobonzi/sets/timcat-eric-whitacre-sleep

3. **Group 3** *

   The following samples were created from a Mongolion throat singing recording. Please listen to them and comment on individual samples, or the group as a whole.
   https://soundcloud.com/neobonzi/sets/timcat-throat-singing

4. **Group 4** *

   The following samples were created from a recording of ambience noise in the rainforest. Please listen to them and comment on individual samples, or the group as a whole.
   https://soundcloud.com/neobonzi/sets/timcat-rainforest

5. **Final Thoughts**

   Please provide any general thoughts concerning what you heard in the previous groups of sounds.

6. **Which of the following best describes you?**

   Mark only one oval.

   - ◯ Musician - General
   - ◯ Musician - Electronic Artist
   - ◯ None