0.1 State-of-the-Art

0.1.1 Eargram

Most literature about concatenative sound synthesis is technically minded and focuses on the efficiency and use of the method. Eargram is a Pure Data patch with the main goal of utilizing machine learning to drive synthesis, using concatenative sound synthesis to combine analysed audio units into a coherent musical output (Bernades, Guedes, and Pennycook, 2012). Eargram was created as a larger application with many included tools, such as 4 different recombination methods and several visualization tools, and is architecturally heavily inspired by Skeleton (Jehan, 2005) and CataRT (Schwarz, Cahen, and Britton, 2008). The authors mention that future development is much about the rhythm and correct recombination of separate audio units.

0.1.2 Voice Drummer

Voice Drummer is defined as a "Percussion instrument notation interface" (Nakano, Goto, Ogata, and Hiraga, 2005). It is a software-application developed to aid those without substantial musical knowledge to notate music. The program is limited to transcribing bass drum and snare drum, and these are differentiated between by their onomatopoeic expressions and the rhythm-pattern timing.

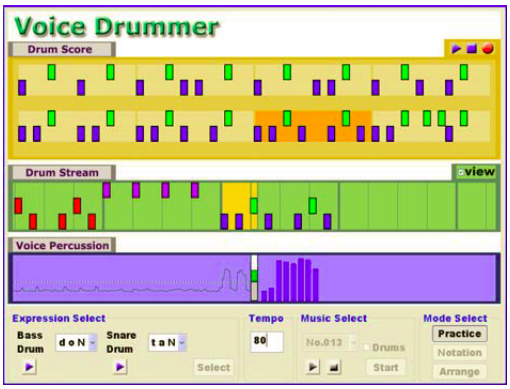

Figure 1: An example voice drummers practice adaption-mode

Onomatopoeic refers to words that phonetically imitates, or resembles the source of the described sound e.g. classifying 'Meow' as a sound a cat would

make. Thus, making it easy for a layman to understand, and use Voice Drummer by uttering don-don to generate a Bass Drum, and ta-ta to generate a snare drum. These are transcribed into phonemic representations through a pronunciation dictionary which will recognize and output the correct corresponding output in real-time.

0.1.3 MATConcat

In "MATCONCAT: AN APPLICATION FOR EXPLORING CONCATENATIVE SOUND SYNTHESIS USING MATLAB By Bob L. Sturm (Sturm, 2004)" such an application has been made for Matlab. Here the method feature Concatenative sound synthesis is used, this methods will approximate an 'target' sound / input sound from a 'corpus' sound / database of sounds, the sound to use from the database is chosen from analysis of corpus sound and target to where they are matching most to some degree that can be set in the program (Sturm, 2004). The matching makes use of different parameters to make the choice of what is matching most. What degree of matching that the program will accept can be set in the interface and also what to do when the sound falls outside that degree (Sturm, 2004).

0.1.4 CataRT

Another project called cataRT is using a similar method but in this project you can explore the analyzed data as they are shown in a space and as one navigates around in the space (cataRT, 2012). The space that one can navigate around in is comparable to what is described as the corpus or database in the MATConcat example, see figure0.1 for an better idea of how the interface looks like, this project is also implemented using a different platform that of maxMSP instead of MatLab. These methods could be using to make a voice imitate an instrument if a database was to be made with a given instrument, then by using this method it will be possible to analyze the input voice and find the closets sound from the database of the instrument sounds. Some analyze criteria would be need to find the right database sound such as the pitch.

0.1.5 Voice Band

Voice Band is an iPhone application by WaveMachine Labs which alters your voice into sampled instruments in real-time. It comes with a set of instruments to choose from including rhythm guitar, lead guitar, bass, sax, synthesizers, organ, drums and microphone, but there are other instruments

Figure 2: cataRT

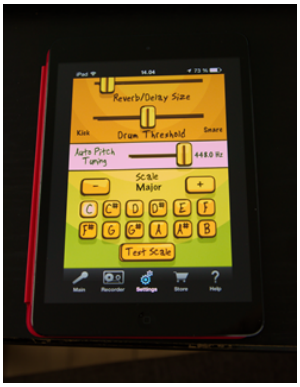

Figure 3: Voice Band

and expansions available for purchase in the Voice Band store, e.g. other drum kits or strings. The target users are, according to WaveMachine Labs' website and demo videos, songwriters who want an idea or arrangement down quickly; they call it a "portable musical scratch board". However, it might also appeal and inspire anyone else interested in music, who wants to have fun.

For every instrument, excluding the drums and cymbals, the system detects the pitch of the voice, and changes that into an instrumental output. To avoid distortion or unwanted noise, one will get the best results by using a pair of headphones, and staying in a quiet environment, because the system is very sensitive sound. It is also recommended to use solid, pure tones with no vibrato to reach a more accurate output, and to sing with hard consonants such as ba or da, because Voice Band recognizes the start of a sound, and it

is therefore easier to detect.

As for the drums, they are divided into kick/snare (two-in-one), hi-hat and crash. The kick and snare drums are together, where quiet will output kick, and loud notes will output snare. However there is the possibility to adjust the sensibility on a trigger in settings, such that only kick or only snare will be outputted, not depending on the loudness of the note. This way it is possible to record kick and snare separately as well. The hi-hat system works similarly in which soft notes produces a closed hi-hat and loud notes produces an open hi-hat. Crash is for itself. For all drums, and most of the other instruments, one can adjust reverberation and volume. All instruments and singing will have to be recorded separately, but on the same track. After having recorded an instrument, it will play back the previous recorded sounds while recording another one. It is possible to undo the last take without having to delete everything, but if you want to edit some of the first recordings this is not possible. You cannot go back and change or edit the whole arrangement. It converts the samples into a MP3 file, which can be sent to an e-mail.

0.2 Collection of dataset

This chapter will go through how a dataset was collected containing only some specific beat boxing sounds. For collection of the dataset containing beat boxing, only three type of different beat boxing sounds there was made use of a recorder, a microphone, a headset, some boards (to limit the noise from surroundings as much as possible) and a computer to take notes of when the different people did record their beat boxing, The setup of the place where the record took place and be seen on figure 4. For choosing the people to help us make the beatboxing random convenience method was used. When the participants sat in the stall they was asked to practice a few different beat boxing sounds, when they had learned one they would be recorded making that sound this was repeated with three sounds, a kick drum beat boxing sound, a snare drum beat boxing sound and a hi-hat drum beat boxing sound. After they had made the three sounds they were also asked to improvise a short mix of the three beat boxing sounds that they had learned.

Figure 4: data-collection-pic

Chapter 1

Theory

1.1 MFCC

this chapter will explain the Mel Frequency Cepstral Coefficients or MFCC, which is a feature that could be used to make a transcription system.
the MFCC feature is a compact description of the spectral envelope. the MFCC is often used in speach recognition and have been useful in musical processing as well. In audio signal classification a small subset of the resulting MFCCs will already contain the principle information, in most cases between 4-20 MFCC is used. the way that the MFCC is calculated is similar to the way human perceive sound, instead of a linear frequency scale it uses a non-linear frequency scale that model the human perception also the DCT is used instead of DFT. The jth coefficient vj MFCC (n) can be calculated like this:

$$v^{j}_{MFCC}(n) = \sum_{\kappa'=1}^{K'} log(|X'(\kappa', n)|) * \cos(j * (\kappa' - \frac{1}{2})\frac{\pi}{K}) \tag{1.1}$$

there are different way to implement the MFCC feature the main difference between the different implementations are the way that the spectrum is calculated, there is the originals being David and Mermelstein DM, HTK an implementation found in the HMM tool kit software and the implementation found in Slaney's Audiotory Toolbox SAT.
for our project the MFCC can be considered as one of the features to describe the audio. one point is that it is a compact description of the spectral envelope another is that is follow the human perception to some degree.

1.2 Spectral Analysis

in this chapter different way of analysing the spectrum will be covered. also how it might be useful in the implementation of the transcription system will be considered

1.2.1 Spectral Centroid

this chapter will show what the feature spectral centroid is.
the spectral centroid feature will calculate the center of gravity(COG) of a spectrum. it is defined by the frequency weight power spectrum normalized by the unweigth sum:

$$\upsilon_{SC}(n) = \frac{\sum_{\kappa=0}^{\frac{K}{2-1}} \kappa |X(\kappa, n)|^2}{\sum_{\kappa=0}^{\frac{K}{2-1}} \kappa |X(\kappa, n)|^2} \tag{1.2}$$

however it can also be calculated using the magnitude spectrum instead of the the power spectrum.
the point found by the spectral centroid feature should correlate with the timbre dimension of how sharp or bright the sound is.
this feature might be used in a transcription system because it should describe how sharp or bright a sound will sound, so this feature could be used to classify as sound.

1.2.2 Spectral Rolloff

this chapter will explain what the spectral rolloff is.
The Spectral Rolloff is defined as the frequency bin at which the magnitude of the STFT reaches a percentage K of the overall sum of magnitudes, can be calculated like:

$$\upsilon_{SR}(n) = i| \sum_{\kappa=0}^{i} |X(\kappa, n)| = \kappa * \sum_{\kappa=0}^{\frac{K}{2-1}} |X(\kappa, n)| \tag{1.3}$$

normal the value for K is around 0.85 or 0.95. Low results indicates insufficient magnitudes components at high frequencies and the a low audio bandwidth.

different ways to compute the spectral rolloff can be that only parts of the spectral energy is taken into considerations that is done by use an fmin and a fmax:

$$\upsilon_{SR}(n) = i| \sum_{\kappa=\kappa(f_m in)}^{i} |X(\kappa,n)| = \kappa * \sum_{\kappa=\kappa_(f_m in)}^{\kappa-(f_m ax)} |X(\kappa,n)| \tag{1.4}$$

it can also be common to use the power spectrum

$$\upsilon_{SR}(n) = i| \sum_{\kappa=0}^{i} |X(\kappa,n)|^2 = \kappa * \sum_{\kappa=0}^{\frac{K}{2-1}} |X(\kappa,n)|^2 \tag{1.5}$$

for our project the spectral rolloff could be used to determine the sound in the classification.

1.2.3 Spectral Flux

this chapter will explain the feature Spectral Flux and its uses.
The spectral Flux is how much the spectrum shape change between the different frames it can be defined as:

$$\upsilon_{SF}(n) = \frac{\sqrt{\sum_{\kappa=0}^{K/2-1} (|X(\kappa,n| - |X(\kappa,n-1)|)^2}}{K/2} \tag{1.6}$$

the spectral flux feature can be described as a representation of the roughness of a sound. the result that one will get from a spectral flux feature is in the range from 0 to A where A is the maximum magnitude possible in the spectrum. when looking at spectral flux in a signal it will be flat at silence and spike at pitch changes. for use in note onset detection (finding the start of the note) only an increase in the spectral energy is wanted an one can consider using a different way to calculate the spectral flux:

$$\upsilon_{SF}(n) = |X(\kappa,n)| - |X(\kappa,n-1)| \tag{1.7}$$

when doing this all negative values has to be set to zero so that only increase is detected.
for a transcription system the spectral flux as described can be considered to do the segmentation of a sound signal because it can detect an onset.

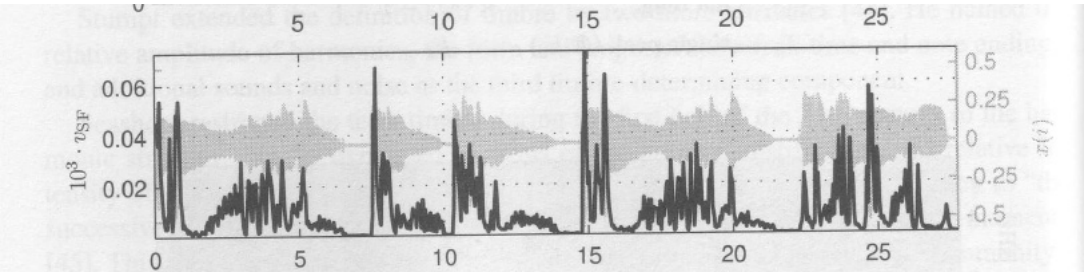

Figure 1.1: Spectral Flux black = the Spectral Flux Gray = the sound

1.2.4 Spectral Slope

This chapter will be on explaining what the spectral slop feature is.

Bibliography