Sampling

Trevor Henderson

August 4, 2016

Contents

1 Introduction

Sample based music has an Samples of Motown backing bands, pop stars, and jazz legends can be stitched together to create a collage of sound combining all of their talents. Sampling has the potential to make music that was never possible before. Out of a musician's life work, the sampler can choose a 10 second snippet that perfectly Condensing hours and hours of recording time, studios across the globe and talents of all sorts.

While some artists have made sampled based masterpieces, they take considerable ammount of time. The issue The goal of this project is to reduce almost all of the manual labor associated with sampling. The process should ideally be simply listening to music, choosing what sounds good, and then In reality lots of time is spent tuning samples, matching beats, and matching chords.

Phase vocording works to solve this problem, it nessesarily introduces digital artifacts. Although subtle, these artifacts can break the suspension of disbelief holding together the dream that Madonna and James Brown really did record together. Rather than using computers to directly manipulate audio, this work intends to do an intense analysis on a sample library, beyond the scope of humans, to find pairings for samples. While more digital processing has been done to the audio than probably any other production, the actual audio signal path could be completely analog.

2 Tuning

In order to get the sample ready to be heard by our machine learner, the first thing we do is tune the sample so that the frequencies will fit nicely into semitone bins. This is the easiest step, but it introduces some techniques we will be using later on.

We will assume, very reasonably, that the song being taken as the input is not long enough that the instruments began to go out of tune, and was recorded and transcoded with good technology - a turntable with a stable belt, etc. Therefore the tuning will be constant which makes our job easier.

What we will do is take the fourier transform and then lump frequencies together into bins. This is fairly good for large samples, but for small samples, sometimes the frequency resolution isn't fine enough to discriminate between semitones, so we will use frequency estimation:

$$\kappa(x) =$$

3 Tempo Detection

Tempo detection Approximating the tempo Onset detection Autocorrelation Then use wavelets to detect self similarity

Tempo detection happens in three steps. First we convert the audio signal into an "onset" signal, which is a positive signal whose amplitude correlates to to how much the signal is changing at that point. This signal is high when the amplitude or pitch of the signal suddenly changes, which would be a "beat." Next we approximate the tempo by looking for the frequency of self simiarity in this onset signal. Finally, we fine tune this approximation to get function for the tempo that minimizes the ammount of onset information that occurs off-beat.

Onset signal

Using an analysis by [], we are going to We window the audio signal into frames. We use the previous two frames to linearly predict what the third frame should be. If the signal is changing this third frame will be different than the previous two. So we simply calculate the euclidean distance between our predicted frame and the actual frame.

Approximating the signal

A tempo in music is the frequency at which the music exhibits some sort of self similarity. Self similariy on music exists on many levels. There is repeating verse structure - 16 bars, repeating chord progressions - 4 bars, repeating drum beats - 1 bar, and the beat itself - 1 quarter note. In order to divide up a song as finely as to display its harmonic and rythmic features want to find the minimum frequency at which the song exhibits self similarity.

Fine tuning the approximation

Many modern music recordings are

While in most cases playing to a click or quantized electronic music has a perfectly consistent tempo, we would like to include music played live, where the tempo can slowly shift or change Series approximation with wavelets? Fourier series? Polynomial approximation Approximation $b(t) = a_0 + a_1 t + ...$ $b(t) = a_0 + a_1 cos($ Find coeffiecient, then next coefficient must have average of first ... Most songs will have 0 higher order coefficients But those that fluctuate will not.

4 Harmonic Analysis

Using the tempo and the one as a guide, divide the song according to beat positions

Equal loudness contour Normalizing energy per beat. Wrap to an octave (hard quantize, filter out very low signals)

5 Machine Learning

ongs that work together Scale polynomials (tune) and find starting point, so that fluctation of beat over length is negligible

Recurrent neural nets and PU Learning Train the NN to learn the difference between positive examples (real songs) and songs that have been built like above The samples should be Punish false positives less than false negatives. So that all positive examples are learned, and some negative ones will bleed through Search for ones that do indeed bleed through.