Beat Detection

COM-415 Final Project Report

Nicola Figari, Dohun Jeong

0. Introduction

Rhythm, along with harmony, and melody, are the three traits that are immediately recognizable to any human who listens to music. Rhythm can be expressed in many different ways. We tap our feet to music, bobble our heads, and synchronize visual effects with the rhythm of a song, in a movie and in live shows. It is the backbone of ensemble music playing

Despite our ability to feel beats naturally, it is hard to describe it mathematically from a wave file. What we do know is that change in timbre, pitch, and loudness may indicate a beat. But merely detecting these changes won't tell us the regular beat pattern, for there are off-beat notes in reggae music, or changes in loudness regardless of beat pattern in classical music.

In this project, we explore different algorithms for beat detection, to find out which is the most robust method. We attack this problem in two different parts: Onset detection, to find the exact moment of a beat, and tempo analysis, to fit these beat candidates into a periodic pattern.

1. Onset Detection

Onset is the beginning of a musical note or other sound (Figure 1). During an onset, there are three things that can change: energy or loudness, pitch or harmony, and timbre.

We implemented two energy-discriminating onset detection method, and one pitch-discriminating to find out which algorithm will provide us with the most accurate results. Identifying these onsets will then allow us to identify where the beats may be, and produce a novelty curve that will be tempo analyzed (as will be discussed in chapter 2).

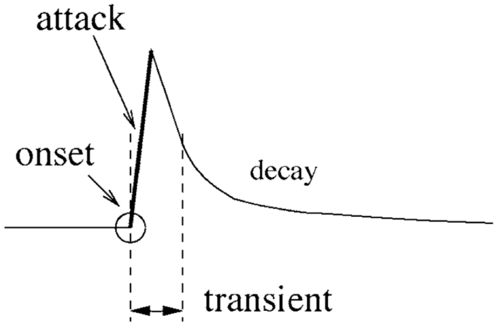


Figure 1.1 (Bello et al, 2005)

1.1 Filterbank

Filterbank method proposed by Scheirer 1998 takes a variety of factors into account. To mimic psychoacoustic model, Scheirer divided the audio into six different frequency sub-bands, and analyzed the signals to get frequency in each sub-band individually.

Analysis of each band included an enveloping through Hann window, differentiating to get positive value at an onset point, and half-wave rectifying to get rid of the decay component. The six curves produced by the filterbank will then be tempo analyzed to produce the final result.

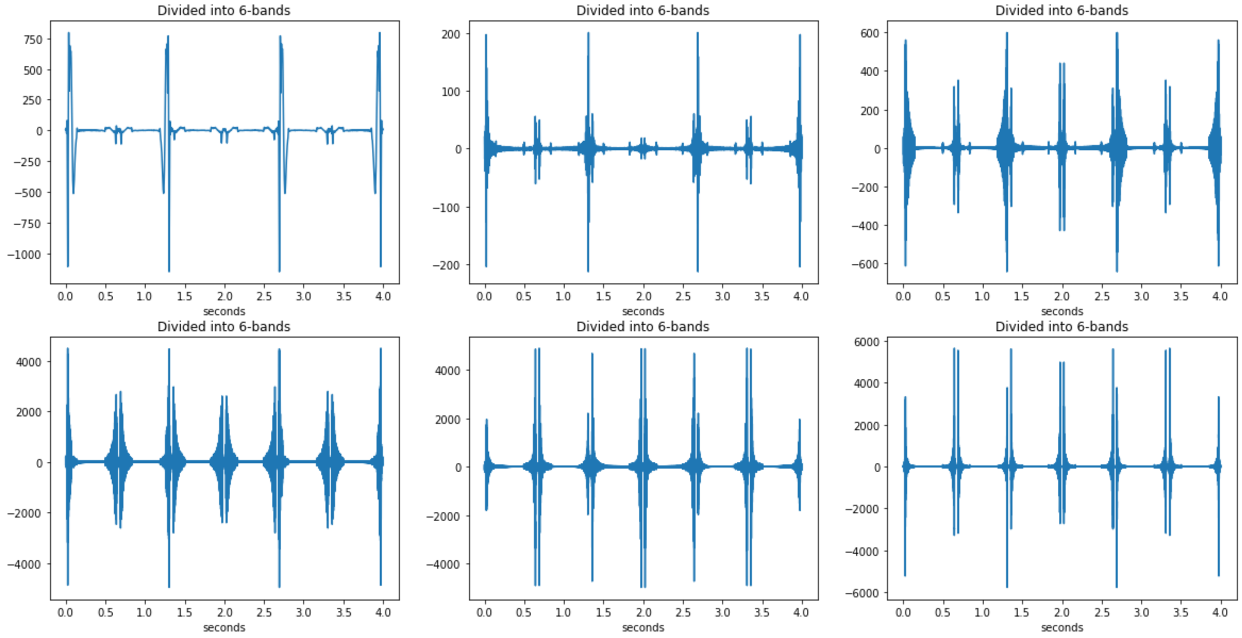


Figure 1.2 Six different frequency bands of bottle_90bpm.wav

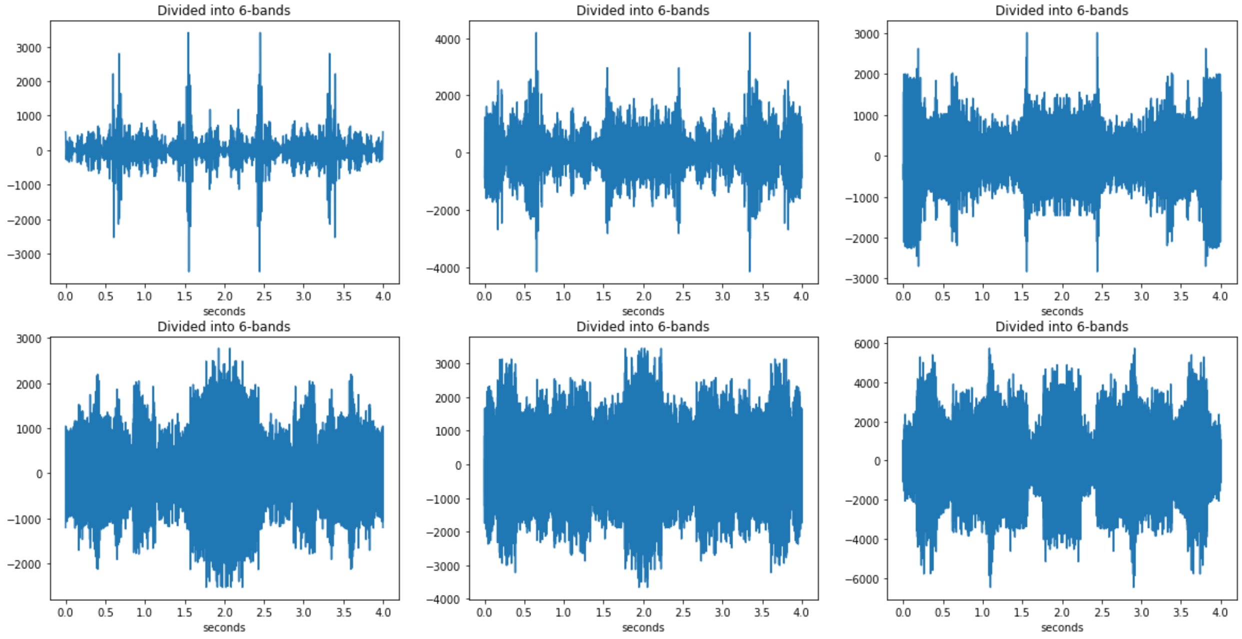


Figure 1.3. Six different frequency bands of X-Kid.wav

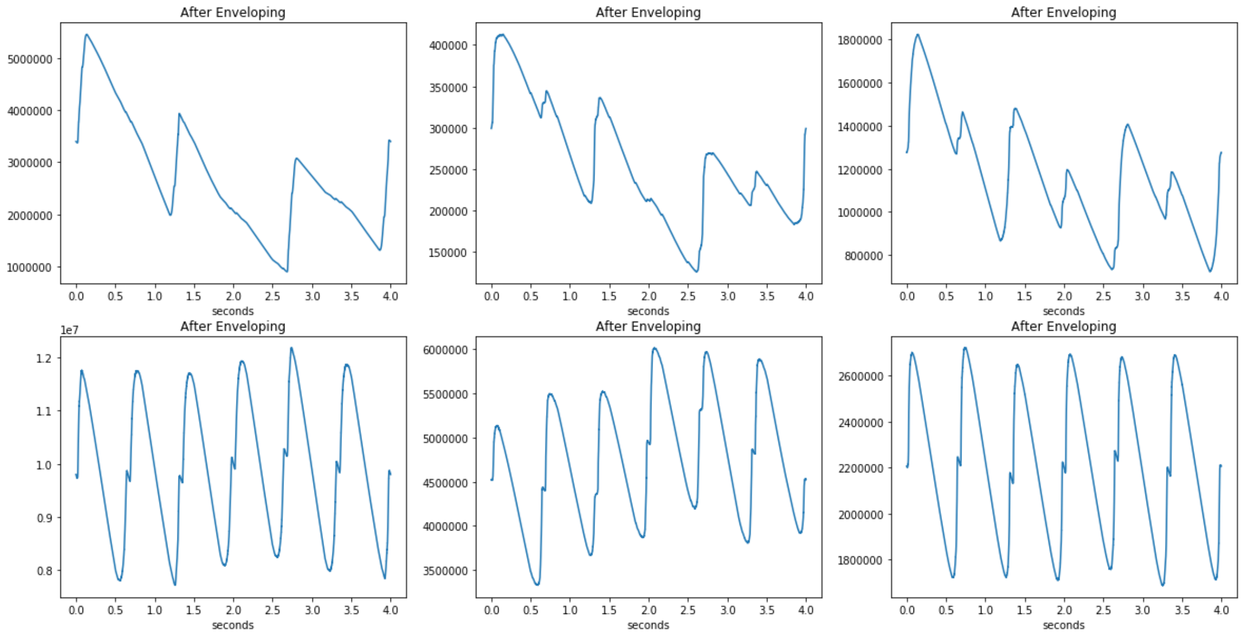


Figure 1.4. Six different frequency bands after enveloping bottle_90bpm.wav

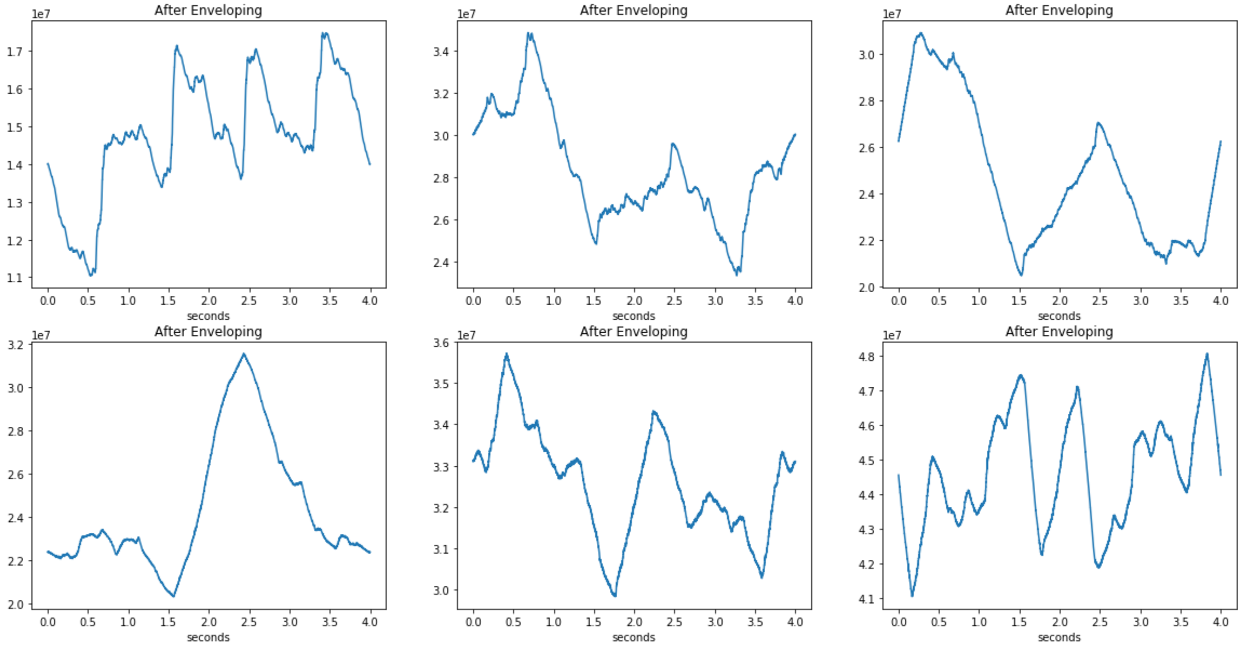


Figure 1.5. Six different frequency bands after enveloping X-Kid.wav

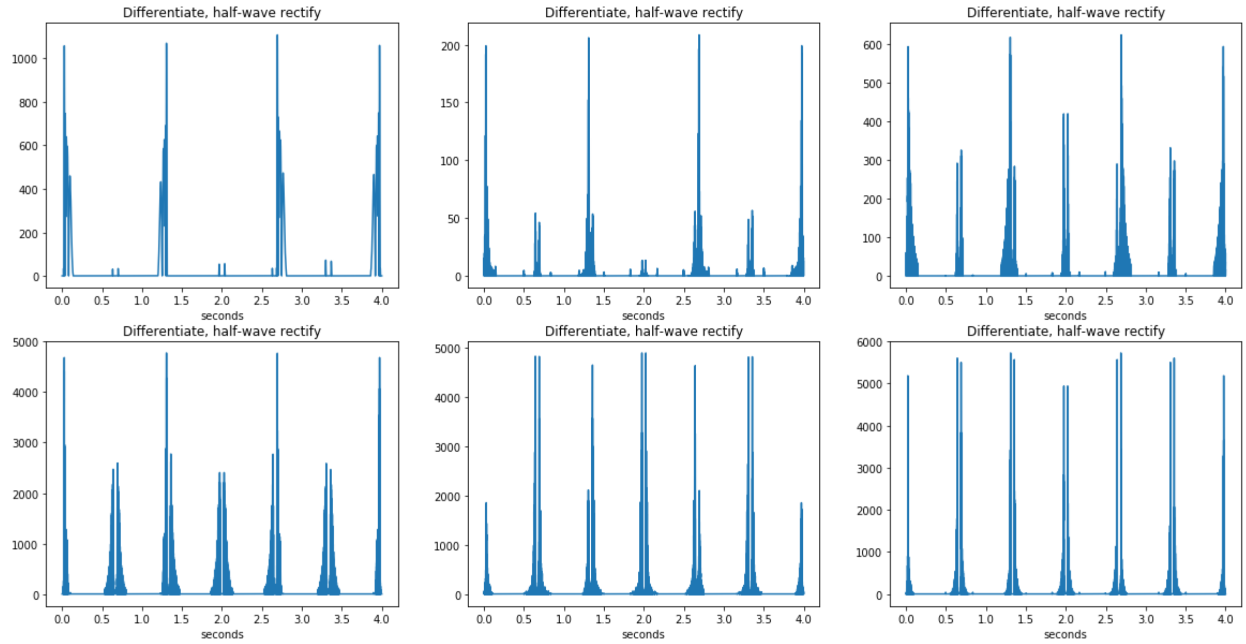


Figure 1.6. Six different frequency bands after differentiating and half-wave rectifying bottle_90bpm.wav

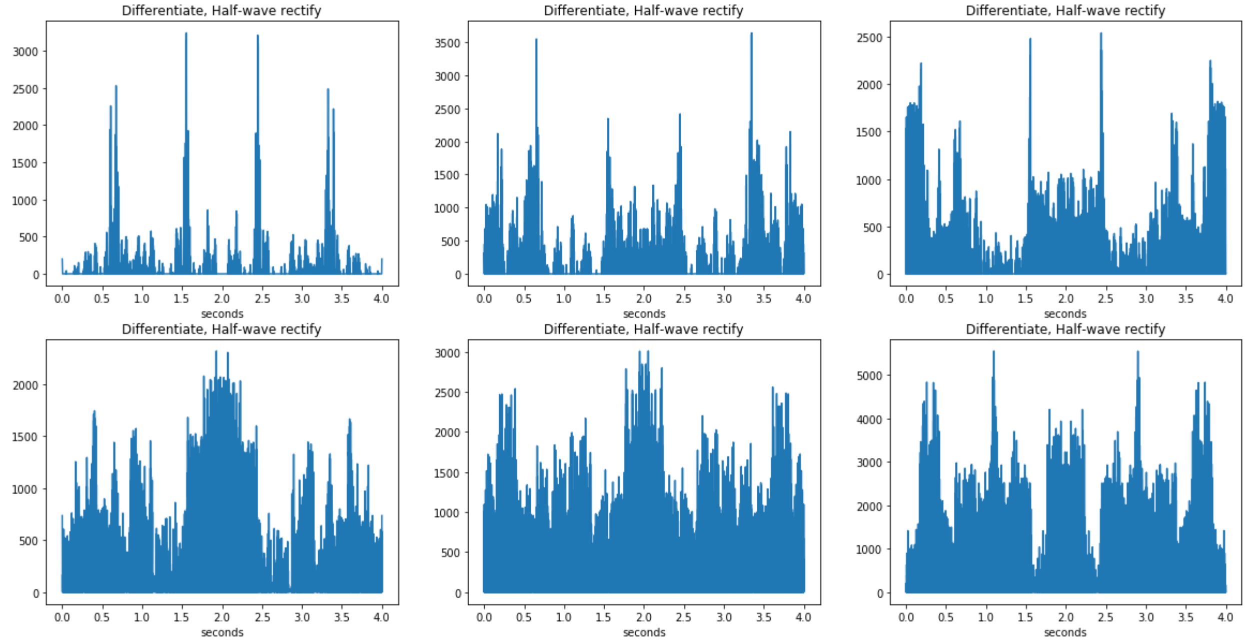


Figure 1.7. Six different frequency bands after differentiating and half-wave rectifying X-Kid.wav

bottle_90bpm.wav (90bpm) X-Kid.wav (132bpm)

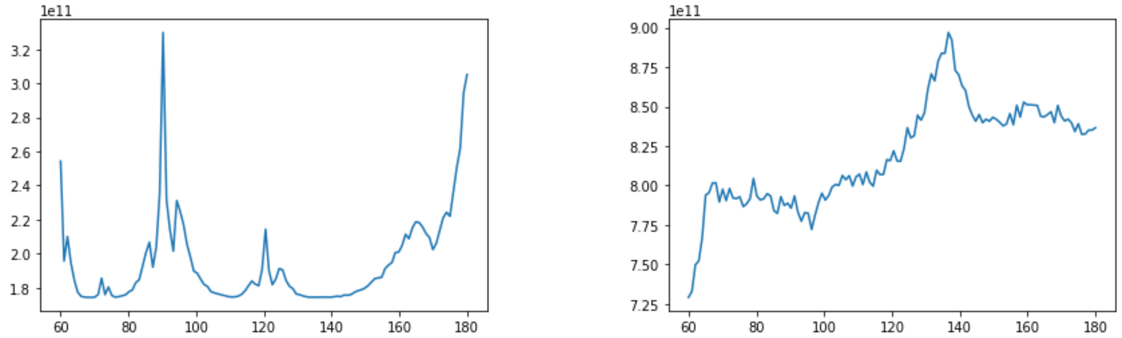


Figure 1.8. BPM likelihood curve

Cascading Discrete Wavelet Transform (DWT) was soon suggested by Tzanetakis et al, 2001 to further mimic the psychoacoustics (increasing time resolution and decreasing frequency resolution for higher frequency content). However, this method was not fully implemented as it is beyond the scope of our class, and also because it failed to yield significantly better results. However, we discuss a different time-frequency analysis in a later method (Chapter 1.3).

1.2 Average and Instantaneous Energy Thresholding [4]

The algorithm divides the data into blocks of 1000 samples and compares the energy of a block E_j with the energy of a window of blocks E_{avg} (48 blocks) to which the block itself belongs. The energy of a block is used to detect a beat. If the energy E_j of a block is above a certain threshold then the block is considered to be a beat.

$$E_j = \sum_{i=0}^{1000} f(i)$$

$$E_{avg} = \frac{1}{48} \sum_{j=1}^{48} E_j$$

The threshold can be defined in two different ways:

1. We consider the average energy of a window of blocks as the threshold

$$E_j > E_{avg}$$

2. We take the average energy of a windows and weight it by a factor c

$$E_j > cE_{avg}$$

The factor c is defined as follows:

$$c = -0.0025714 var(E) + 1.5142857$$

$$var(E) = \frac{1}{48} \sum_{j=0}^{48} (E_{avg} - E_j)^2$$

c depends on the energy variance of the window of blocks and quantities how marked the beats of the song are. The bigger the variance, the smaller the weight.

Once we detected the blocks which correspond to beats and the ones which don't, we assign 1 to the beat blocks and 0 to the non beat blocks. We then pass the resulting signal through a comb filter that gives us the BPM.

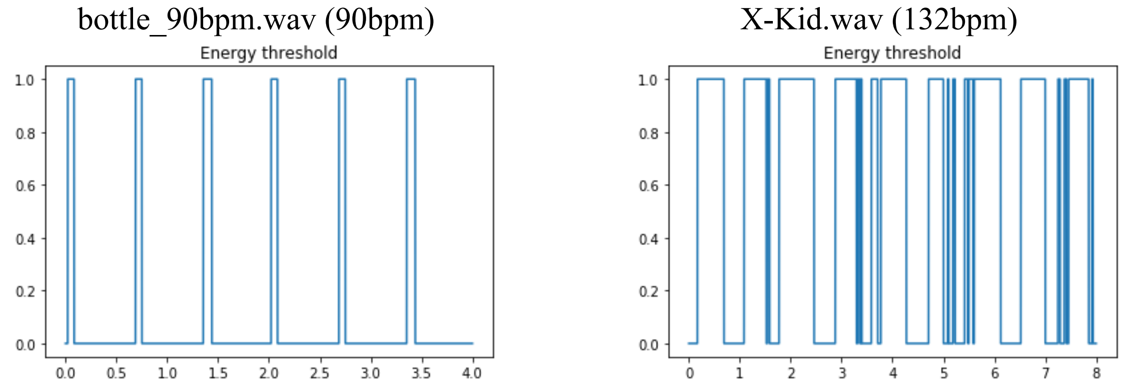


Figure 1.9. Energy threshold

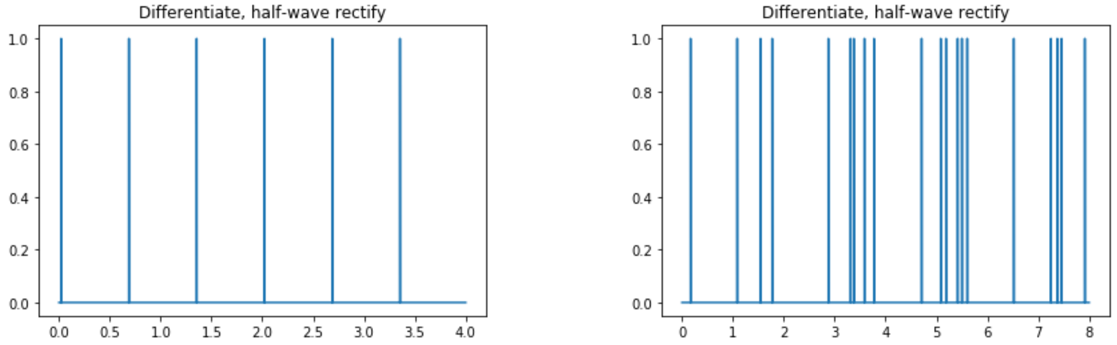
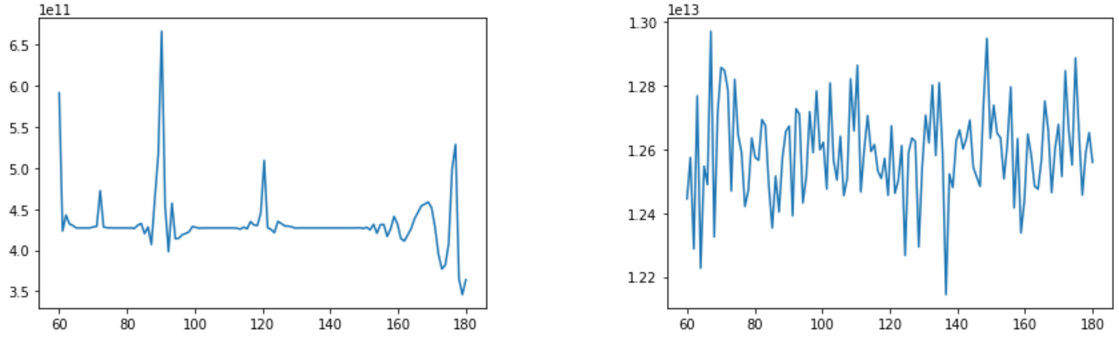


Figure 1.10. Differentiated, and half-wave rectified



*Resulting 67BPM ~= 132/2

Figure 1.11. BPM Histogram

1.3 STFT Change of Harmony Method

Short-Time Fourier Transform (STFT) is used for time-frequency representation of a signal. We employ the time-frequency representation to look for note/harmony changes in the music, in case the music lacks percussive instruments.

STFT is performed on the audio signal, and then scaled logarithmically with the following formula:

$$Y = log(1 + 1000|X|)$$

Then the signal is differentiated, to indicate changes in frequency content at each time. These frequency content change is accumulated back to 1-dimensional time-axis to produce spikes wherever frequency change exists. These spikes are normalized.

When we switch back from time-frequency representation, the signal time-scale was adjusted to window length, rather than sampling period. Fourier upsampling is used to space the spikes back into the same timescale as the original audio signal. These curves are ready to be tempo analyzed.

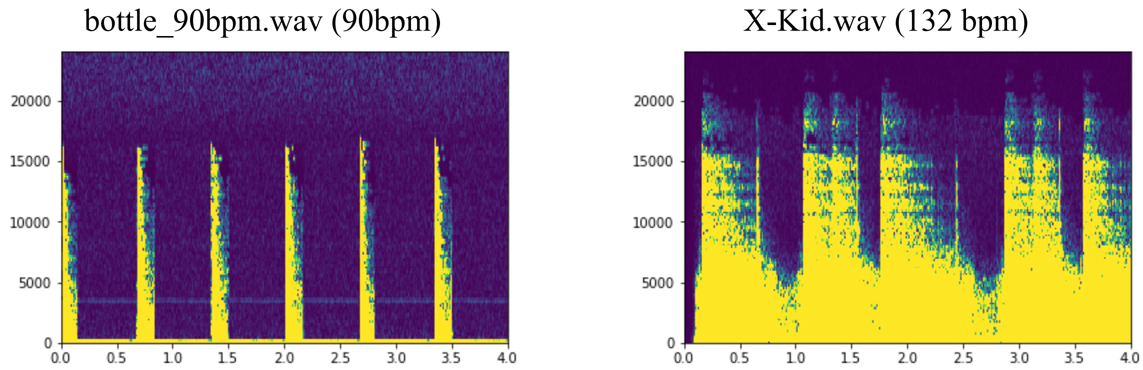
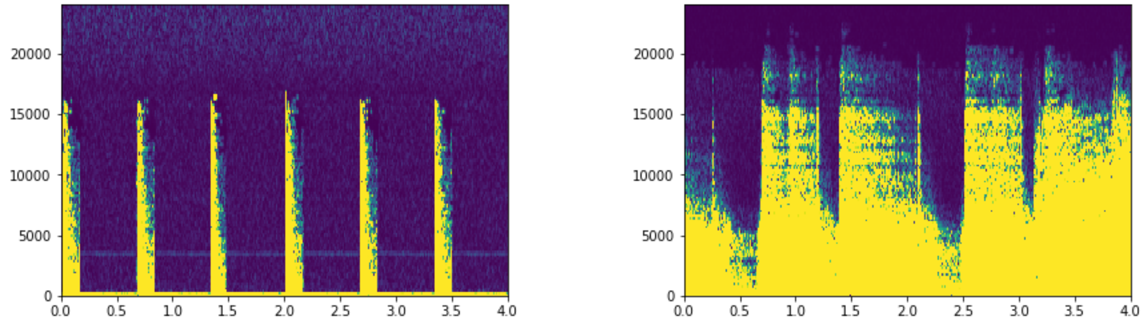
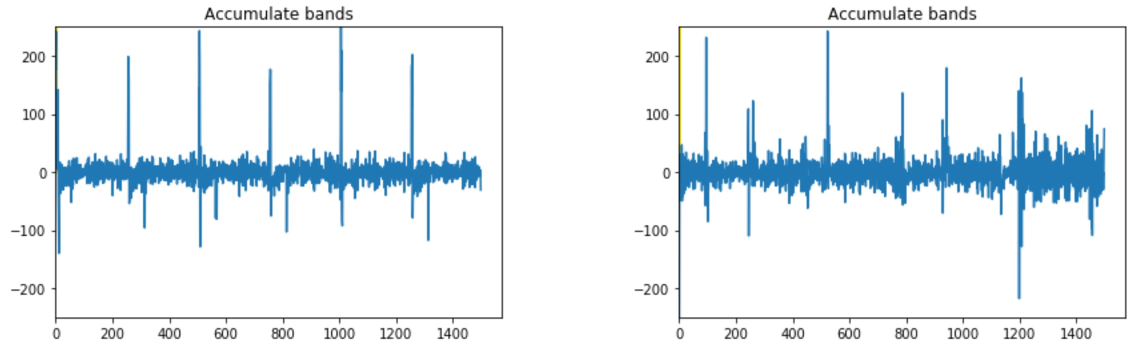


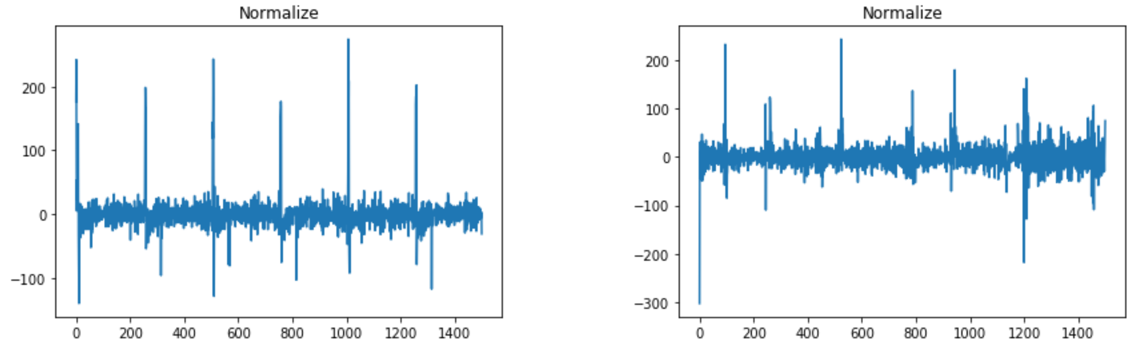
Figure 1.12. STFT of original signal



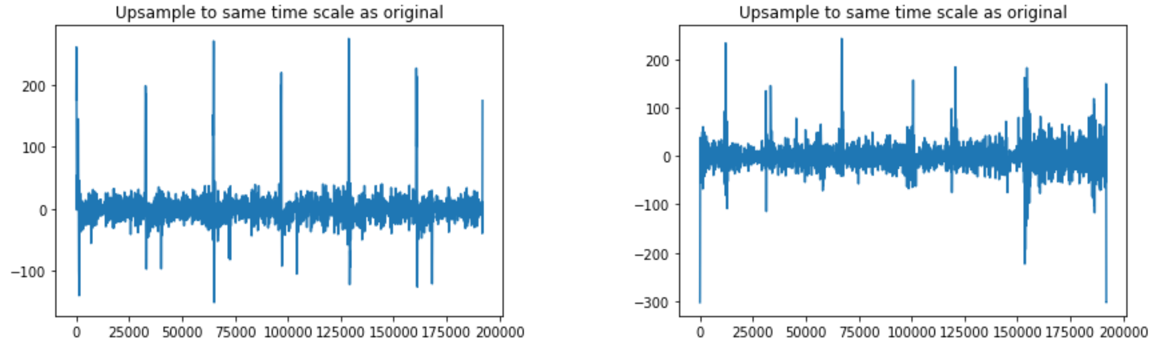
STFT differentiated



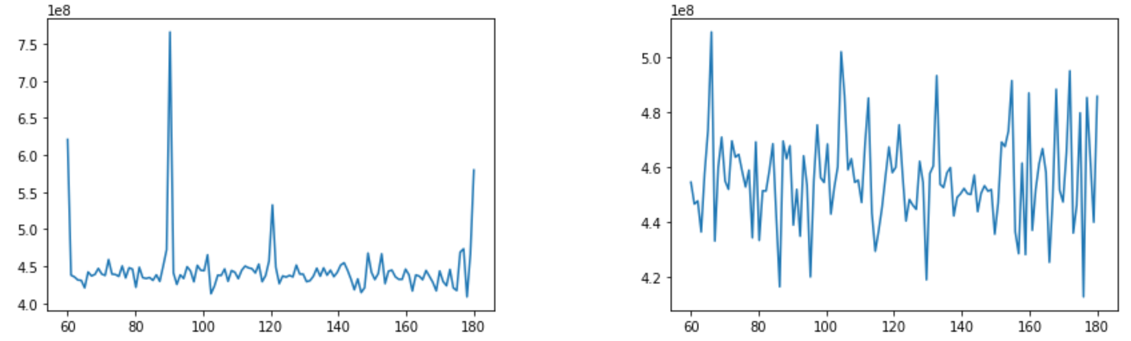
Difference accumulated for all frequencies



Normalize waveform to zero



Upsample to original time scale



BPM Histogram

2. Tempo Analysis

2.1 Comb Filter [2]

After the novelty curve is determined, we need to find a regular beat and a bpm value. In this function, we "try" different BPMs from a given minimum and maximum value constraints to find the single most likely BPM value.

We try different BPM values by convolving regular beats (in the form of a Kronecker delta train) with the audio signal. The most likely BPM value will come from convolution of the beat. Interval between deltas is defined as

$$t_{beat} = 60/bpm$$

$$\text{Index } n_{beat} = t_{beat} * f_{sampling}$$

This is the most expensive step computationally, so we tried multiple implementation to find the fastest approach.

First, we tried a recursive filter using a difference equation in feedback form.

$$y[n] = x[n] + \alpha y[n - t_{beat}]$$

However, this difference equation mimics an infinite-length comb filter, and thus yielded incorrect results.

Second, we tried multiplication in the Fourier domain with three delta functions. This took too much time, as it was performing an FFT and an IFFT on the 6 bands of the entire audio signal

Third, we tried a linear filter function (scipy.signal.lfilter) with the same principle with three delta functions. However, because the filter length was tens of thousands of coefficients long, it took an even longer time than the FFT method.

$$y[n] = x[n] + x[n - t_{beat}] + x[n - 2t_{beat}]$$

Finally, the most time-efficient approach was to convolve the delta train in the time domain. This method, however requires a large amount of memory.
Because the comb filter is very short in length, convolution can be computed quickly in the time domain. Using the delta function property

$$f(t) * \delta(t - t_0) = f(t - t_0)$$

and the distributive property of convolution, we can obtain the following:

$$f(t) * (\delta(t) + \delta(t - t_{\text{beat}}) + \delta(t - 2t_{\text{beat}}) = f(t) + f(t - t_{\text{beat}}) + f(t - 2t_{\text{beat}})$$

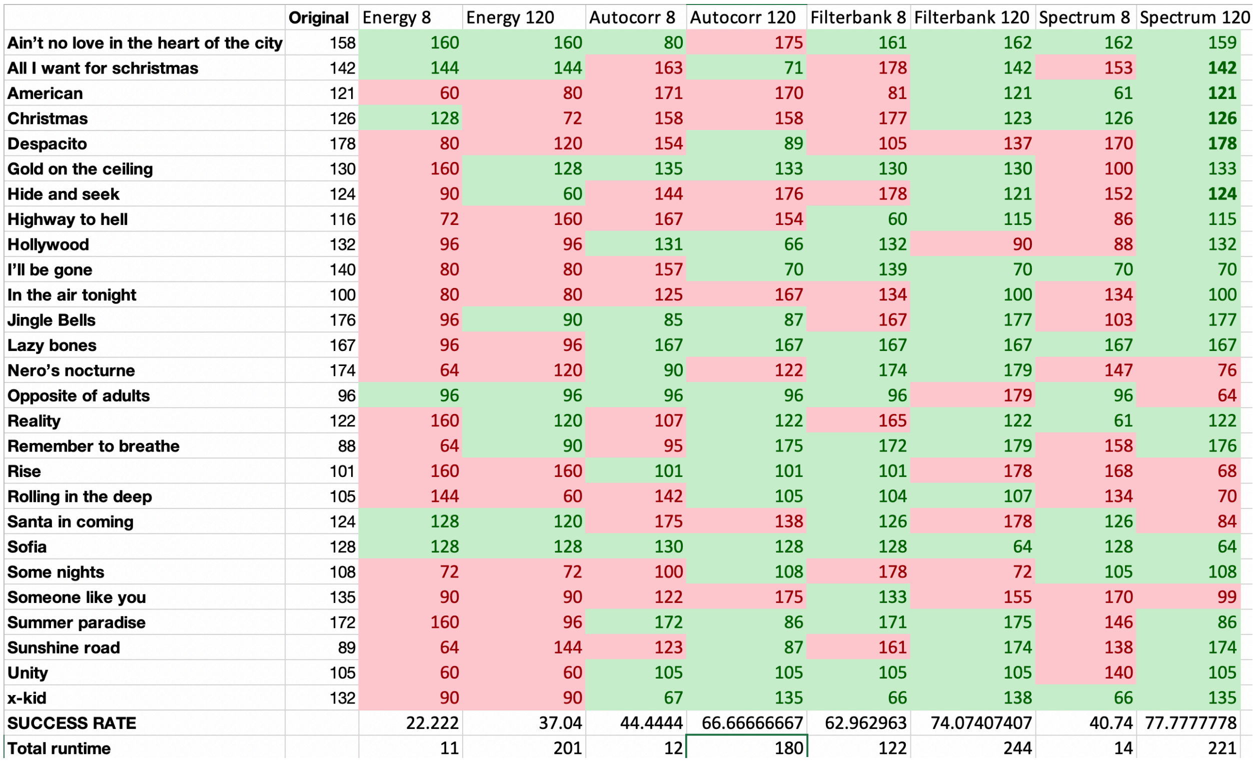
Energy is then computed in the time domain

$$E = \sum |f(t)|^2 = \sum |F(\omega)|^2$$

This method allows us to get a probability distribution of the most likely BPMs, since the higher the energy, the more likely it is for the beats to be correct. It also allows us to filter “bad” candidates, since if the beats don’t appear in regular pattern, and if the spikes are much larger than the noise on the novelty curve, it won’t affect the probability.

The limitation on this method is that increasing the resolution of BPM to be tried means longer runtime and larger memory required. It also means that any changes in the BPM during the music won’t be detected, whereas alternate methods such as Fourier tempogram can indicate when and what the tempo changes were.

Using the comb filter method, sweeping BPM values from 60 to 179 in increments of 1, here’s the BPM value we get for the library of songs tested.



	Original	Energy 8	Energy 120	Autocorr 8	Autocorr 120	Filterbank 8	Filterbank 120	Spectrum 8	Spectrum 120
Ain’t no love in the heart of the city	158	160	160	80	175	161	162	162	159
All I want for schristmas	142	144	144	163	71	178	142	153	**142**
American	121	60	80	171	170	81	121	61	**121**
Christmas	126	128	72	158	158	177	123	126	**126**
Despacito	178	80	120	154	89	105	137	170	**178**
Gold on the ceiling	130	160	128	135	133	130	130	100	133
Hide and seek	124	90	60	144	176	178	121	152	**124**
Highway to hell	116	72	160	167	154	60	115	86	115
Hollywood	132	96	96	131	66	132	90	88	132
I’ll be gone	140	80	80	157	70	139	70	70	70
In the air tonight	100	80	80	125	167	134	100	134	100
Jingle Bells	176	96	90	85	87	167	177	103	177
Lazy bones	167	96	96	167	167	167	167	167	167
Nero’s nocturne	174	64	120	90	122	174	179	147	76
Opposite of adults	96	96	96	96	96	96	179	96	64
Reality	122	160	120	107	122	165	122	61	122
Remember to breathe	88	64	90	95	175	172	179	158	176
Rise	101	160	160	101	101	101	178	168	68
Rolling in the deep	105	144	60	142	105	104	107	134	70
Santa in coming	124	128	120	175	138	126	178	126	84
Sofia	128	128	128	130	128	128	64	128	64
Some nights	108	72	72	100	108	178	72	105	108
Someone like you	135	90	90	122	175	133	155	170	99
Summer paradise	172	160	96	172	86	171	175	146	86
Sunshine road	89	64	144	123	87	161	174	138	174
Unity	105	60	60	105	105	105	105	140	105
x-kid	132	90	90	67	135	66	138	66	135
SUCCESS RATE		22.222	37.04	44.4444	66.66666667	62.962963	74.07407407	40.74	77.7777778
Total runtime		11	201	12	180	122	244	14	221

Figure 2.1. Results of different method

Songs like x-kid, Sofia, and Lazy bones performed well in all of the methods, because of a very regular percussive instrument playing throughout the song. However,

One surprising result was that Christmas songs with regular sleigh bells failed to produce consistent results in most of the methods. All I Want for Christmas and Jinge Bells failed to produce accurate results in 8-second sample. Santa is Coming was a particularly difficult song, as the beats are very unstable.

Overall, Filterbank and STFT seemed to yield the most reliable result, while instantaneous energy test didn’t fair so well. It was surprising that passing the original audio through the comb filter produced results that were far better than energy method, and almost as good as the filterbank. This suggests that our selection of songs that were tested had too regular and strong percussion to test music with different styles.

3. Applications

There are many exciting applications for automated beat detection. One of them will be presented during our demo, where we control an array of LEDs based on the beats of a music. This can be used in Disco balls, and Christmas trees!

It will also play a role in creating water fountain shows, light shows, movie editing, and game development as well, by aligning visual changes with the beats.

It can also be used in a factory setting for industrial automation, where we can check nominal operation of rotary machines with contactless monitoring by detecting regular pulses of sound.

References

1. Bello, Juan Pablo, et al. “A tutorial on onset detection in music signals.” *IEEE Transactions on speech and audio processing* 13.5 (2005): 1035-1047
2. Scheirer, Eric D. "Tempo and beat analysis of acoustic musical signals." *The Journal of the Acoustical Society of America*103.1 (1998): 588-601.
3. Tzanetakis, George ,Georg Essl, and Perrk Cook, “Audio analysis using the discrete wavelet transform.” *Proc. Conf. in Accoustics and Music Theory Applications.* Vol. 66 2001
4. http://archive.gamedev.net/archive/reference/programming/features/beatdetection/index.html

Special thanks to Eric Bezzam for providing us with the LED array and its firmware that will be used in the presentation on Tuesday.