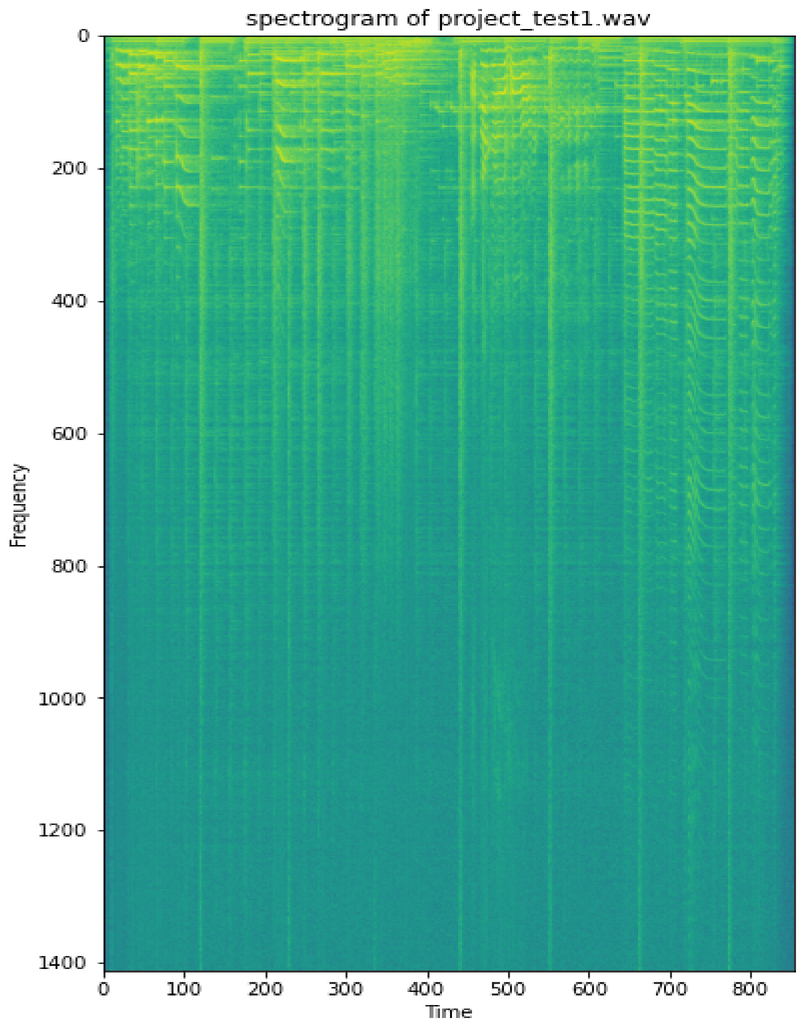
Documentation of Separation of drums from music signals

Introduction:

Here In this project we will separate a monaural audio signal into harmonic and percussive components, which is much useful for multi-pitch analysis, automatic music transcription, drum detection, modification of music,etc.Here we are going to employee a simple concept to separate the harmonic and percussion components of the music signal. We make use of the anisotropy of them on spectrograms. (i.e), We transform our music signal into spectrograms. We can clearly see that the percussion components are local to the frequency and harmonic components are local throughout the time.

Below image we can see the spectrogram of one of the given music signals "project_test1.wav", where we can clearly see both harmonic and percussion components.



Name: Prabu Mohan student number: 50505270

We use this property to separate our given music signal. We use optimization methods to solve it, as the reference paper suggests.

$$J(\boldsymbol{H},\boldsymbol{P}) = \frac{1}{2\sigma_H^2}\sum_{h,i}(H_{h,i-1}-H_{h,i})^2 + \frac{1}{2\sigma_P^2}\sum_{h,i}(P_{h-1,i}-P_{h,i})^2 \qquad (1)$$

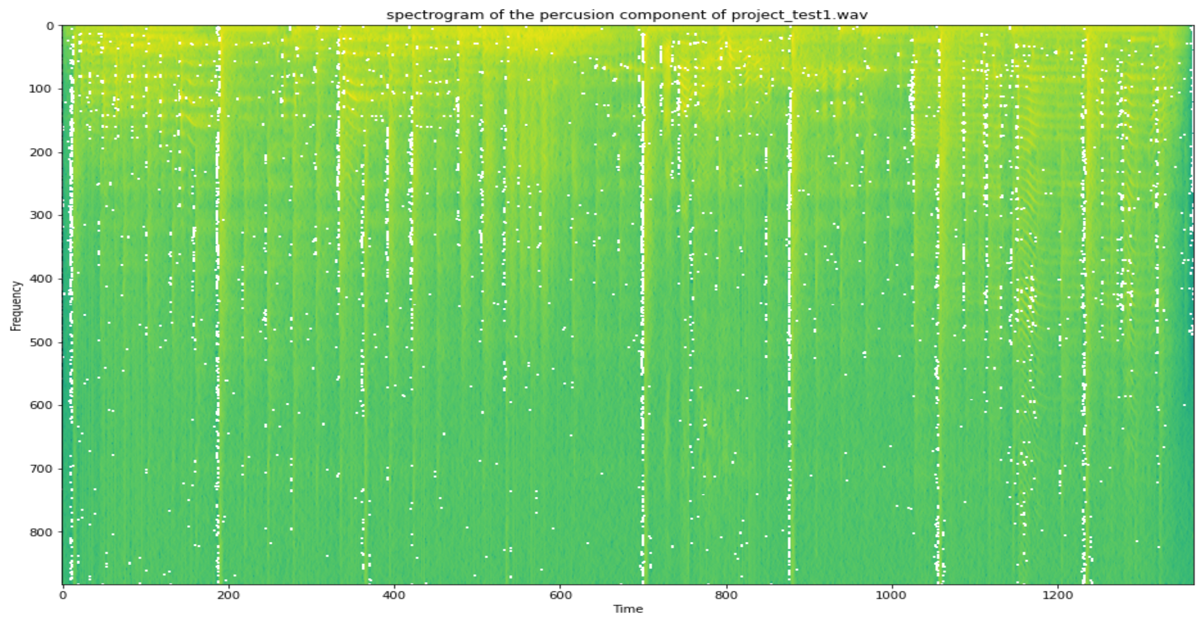
under the constraint as

$$H_{h,i}+P_{h,i} = W_{h,i} \qquad (2)$$

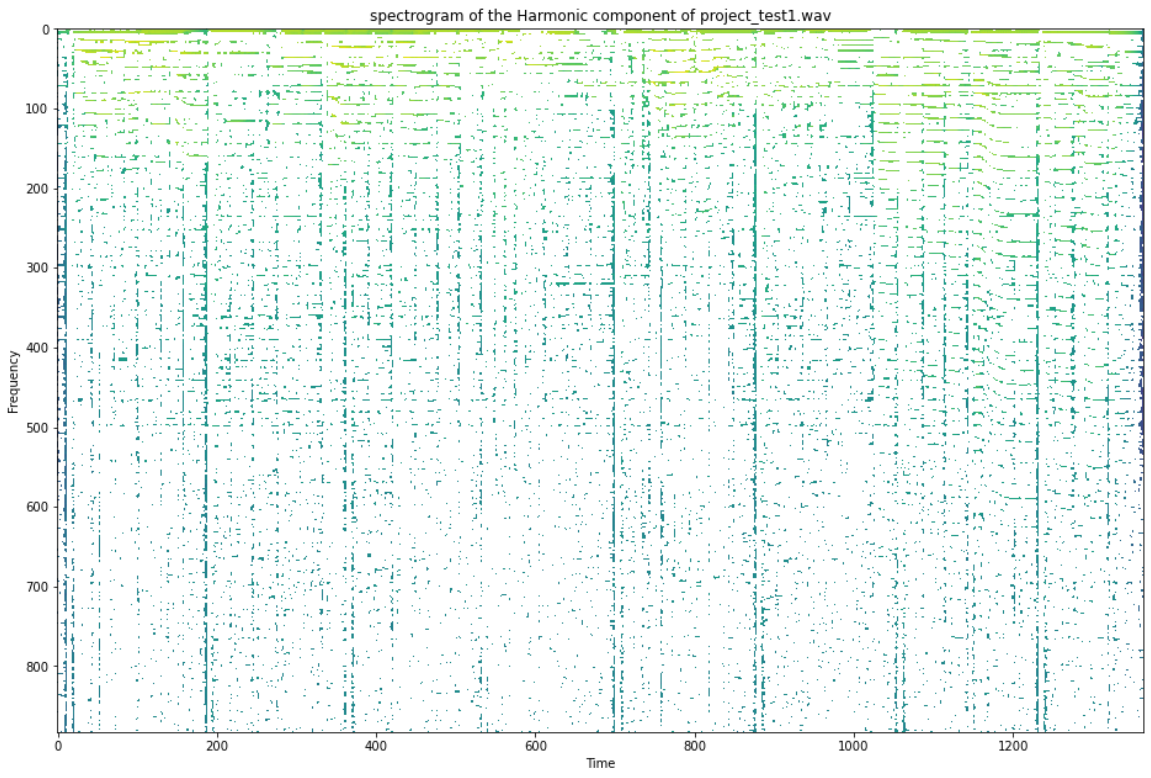
$$H_{h,i} \geq 0, \quad P_{h,i} \geq 0, \qquad (3)$$

where H is a Harmonic component and P is a percussion component with the same dimension as W that is (i,j). The delta(H) and delta(P) are used to handle the smoothness of the element. As mentioned in the reference paper we can do partial differentiation to solve it but it will take huge memory and computation resources. So to solve it we use an auxiliary function and arrive at an iterative method. **Once we separated our harmonic and percussion components we can approximately assess the performance of our algorithm by visually examining the STFT of original signal, percussion component, harmonic component.**

Below is the STFT of the percussion component of the project_test1.wav



Below is the STFT of the harmonic component of the project_test1.wav



Algorithm Implementation:

We have our main method named **“sep_percusion_harmony”** which calculates H(harmonic component) and P(Percussion component) iteratively for predefined number of times “k_max” it also gets the hyper parameter alpha and gamma as parameters, which are finalized by trial and error method. It returns H and P in both time domain and in time-frequency domain.

Later we plot our H,P in both time domain and in time-frequency domain, to visually assess the performance of our algorithm.

It can be quantified by using the “signal to noise ratio” where we try to find the ratio between the input signal and the error of the reconstructed signal. A large value means we have less noise in our reconstructed signal as the error e(t) is the denominator. We also used this property to tune our hyperparameters alpha and gamma.

Report:

Report 1:
Just from playing two recreated sound signals we can clearly see that this algorithm is not suited for audio signals with voice information like singing due to the time varying pitch. Additionally while observing keenly as mentioned in the given research paper, The duration of some percussion, typically bass drum, was almost separated into Harmonic components since it has a smooth temporal envelope, too. Inversely, the attack of pitched tone had a tendency to be classified as a percussion component.

Report 2:
The separation quality can be assessed by a couple of methods, like visually examining the STFTs of the original signal, harmonic component and percussion components. We can also listen to the individual H and P in time domain to access the audio separation quality. It can also be measured by signal to noise ratio. I.e, we can use the ratio of between original and difference between original and recreated signal to find the SNR, In the ideal case the noise will be zero so the SNR may tend to infinity but it is nearly impossible to not have noise. So a good separation will have higher value for SNR.

the signal to noise ratio(SNR) of police3short.wav is 14.410584116926943

the signal to noise ratio(SNR) of project_test1.wav is 14.985863805119486

Name: Prabu Mohan   student number: 50505270