

# Separation of drums from music signals

Project Contributors:

Manasi Dalal (150166535)

Gaurang Sharma (150165798)

## Overview of the problem and solution

The separation of drums (percussive components) from music signals is the challenge to be solved. To tackle this problem, we find anisotropy in the percussive and harmonic components. On the spectrogram, the harmonic component has a steady pitch and creates parallel ridges with smooth temporal envelopes, but the energy of a percussive tone is concentrated in a short time frame, forming a vertical ridge with a broad spectral envelope [1]. The spectrogram below explains this anisotropy. The horizontal lines represent the harmonic components, while the vertical lines represent the percussive components.

In other words, harmonic components have temporal continuity along time (thus the horizontal line on the spectrogram), whereas percussive components have temporal continuity along frequency. In order to make use of this discovery, we try to find spectrograms that maximize anisotropy. To maximum efficiency, this can be done iteratively. The algorithm is based on the referenced research article.

## Algorithm implementation

The algorithm is based on the referenced research article [1].

The functions implemented in code are explained below.

**HnP\_Cal(hh,ii,W,H,P):**

This function calculates H and P iteratively for the number of epochs specified and then the final H and P are normalised after the last iteration is completed. The H and P values are updated based on the calculated delta.

hh and ii are the max values of frequency and time bins.

W is the normalised STFT of the input signal and H and P are the harmonic and percussion components.

This function returns H and P.

This is the main function of our algorithm.

**spectrograms\_plots(file):**

This function plots the spectrograms of the original audio and the harmonic and percussion components.

**ISTFT\_cal():**

This function calculates the inverse STFT of the obtained harmonic and percussion components:

**evaluation\_params(e,SNRh,SNRp,SNR):**

This function calculates the evaluation parameters like SNR for the reconstructed signal, harmonic and percussion components also the accuracy of reconstructed signal.

The algorithm is run iteratively for different values of  $\alpha$  and  $\gamma$

## Evaluation

On two audio samples provided by the course: police03short.wav and project test1.wav, three different assessment metrics were used: signal-to-noise ratio, spectrogram, and auditory evaluation.

We have calculated accuracy by taking the ratio of the sum of error array and the original audio signal array.

Following are the results when  $\alpha = 0.3$  and  $\gamma = 0.3$ .

### police03short.wav:

SNR of harmonic component: 7.868925333023071

SNR of percussive component: 1.0576969385147095

SNR: 140.74824333190918

Accuracy: 99.99999287128034

### project test1.wav:

SNR of harmonic component: 12.470238208770752

SNR of percussive component: 0.34626904875040054

SNR: 140.34231185913086

Accuracy: 99.99975545488269

## Auditory evaluation

### police03short.wav



police03short.wav



police03short\_reconstructed.wav



police03short\_harmonic.wav



police03short\_percussion.wav

### project\_test1.wav



project\_test1.wav



project\_test1\_reconstructed.wav



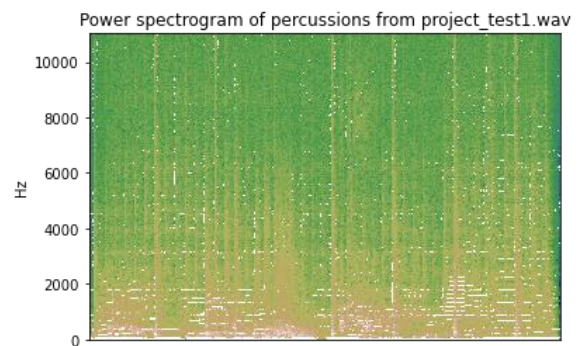
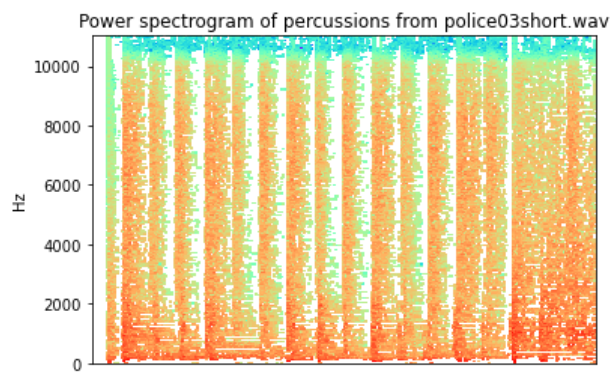
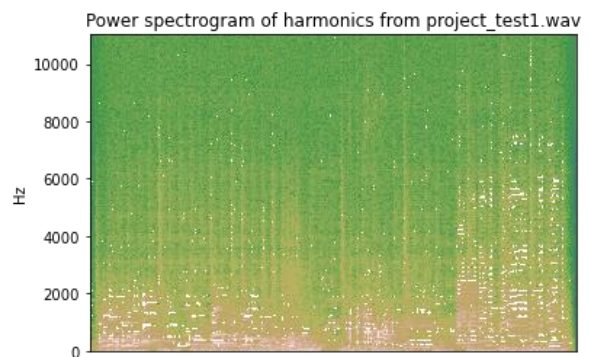
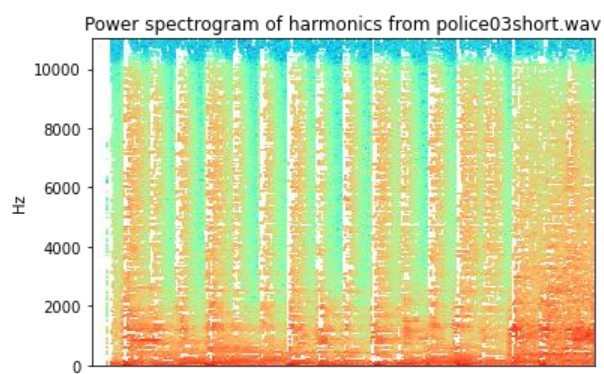
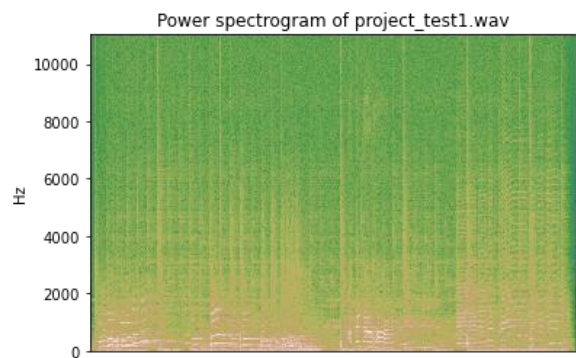
project\_test1\_harmonic.wav



project\_test1\_percussion.wav

# Spectrogram

Usage for different types of audios:



## Signal-to-noise ratio

The signal-to-noise ratio is defined as

$$SNR = 10 \log_{10} \left( \frac{\sum_t s(t)^2}{\sum_t e(t)^2} \right)$$

where  $s(t)$  represents the original signal and  $e(t)$  represents the original signal less the separated signal. A ratio larger than one (greater than 0 dB) shows that there is more signal than noise. To investigate the impact of the range compression coefficient and the number of search iterations on the quality, signal-to-noise ratios were determined. The observations on police03short.wav are depicted in the two charts below.

From these plots we can see that setting the range compression value to 1 (the highest possible value) will yield the highest SNR, minimizing the effect of noise. Changing the number of iterations, however, does not significantly affect the SNR. Similar observations are also observed in project\_test1.wav. Therefore, we recommend setting  $\gamma$  to 1 and the number of iterations to at least 20 but increasing it won't yield noticeable improvement.

## What kind of audio material is the algorithm suitable for and why?

The method is appropriate for any audio that has percussion sounds, however it performs best with instrumental samples.

This method works the best on separation of pitched instruments and drums together.

## How should the separation quality be measured and assessed?

When we compare these spectrograms, we can clearly see the harmonics (horizontal lines) and the percussions (vertical lines) (straight, vertical lines). We listened to the sounds and compared them to the original sample for auditory evaluation. In addition, when we tried to discriminate between the original and rebuilt samples, we discovered that the reconstructed one is exactly matched. This means that the algorithm did an excellent job on this audio sample.

## References

- [1] Nobutaka Dno, Kenichi Miyamoto, Jonathan Le Roux, Hirokazu Kameoka, and Shigeki Sagayama "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram". Proc. EUSIPCO, Aug 2008
- [2] Wikipedia: Signal-to-noise ratio. Available at: [https://en.wikipedia.org/wiki/Signal-to-noise\\_ratio](https://en.wikipedia.org/wiki/Signal-to-noise_ratio)