Real-time tempo detection with harmonic-percussive source separation via median filtering

Martín Rodriguez
Maseeh College of Engineering and Computer Science
Portland State University
Portland, OR
mtr@pdx.edu

***Abstract*—Harmonic-percussive source separation with the goal of real-time tempo detection is explored via the use of median filtering across the spectrogram of monaural audio. A filter is designed and applied to the instantaneous power of the percussive signal output and peak detection is used to determine its tempo. A real-time result is obtained for the tempo which agrees with other currently available methods.**

***Index Terms*—audio signal processing, beat tracking, real-time audio, non-linear filter, median filter, FIR filter**

I. Introduction

In the field of audio processing, tempo detection has several names: beat detection, beat tracking, and beats-per-minute (BPM) detection. These terms all generally refer to the isolation of percussive or periodic elements in music and the description of their temporal or statistical properties. An algorithm may output the time of occurrence of percussive elements or estimate the tempo as a single number, often in units of BPM. Applications of tempo detection include DJing software, remixing, automatic music recognition and cataloging, and others. Many methods are presented in the literature, both for offline and online processing of audio. Some interesting examples use the statistical properties of the onset envelopes [1], comb filter networks [2], discrete wavelet transforms [3], or a combination of different techniques. This paper focuses on the online handling of tempo detection. Because tempo can fluctuate within a piece of music, it is expected than a successful algorithm is able to update its estimate when new data is received.

A. Harmonic-percussive source separation (HPSS)

Harmonic-percussive source separation (HPSS) is a preprocessing step often taken in many audio processing applications. It entails the separation of percussive elements, or drum-like sounds, from elements of the audio which carry pitch information [4]. Although some types of music do not have a clear delineation between the harmonic and percussive elements, this binary categorization of music is often helpful for applications like remixing, beat detection, pitch tracking, and others. There are several possible approaches discussed in the literature; these include tensor factorization [5], non-negative matrix factorization with support vector machines (SVMs) [6], and complementary diffusion of the energy across the spectrogram [7]. Perhaps the simplest and most computationally efficient method was put forth by Fitzgerald [4], who applied anisotropic median filtering, a technique often used in image processing, to the audio spectrogram for the purpose of HPSS.

B. Median filtering

The median filter is a non-linear filter that is often used as a spatial filter in image processing for removal of so-called salt-and-pepper noise. This type of noise arises from large-valued, impulse-like points in the input signal. Because the median is statistically insensitive to outliers, it works well for this purpose.

For a given kernel size n, the median filter output $y[k]$ can be calculated at each time step k as follows:

$$y[k] = \text{median}\left(x\left[\frac{k-(n-1)}{2} : \frac{k+(n-1)}{2}\right]\right)$$

if n is odd, and

$$y[k] = \text{median}\left(x\left[k-\frac{n}{2} : k+\frac{n}{2}-1\right]\right)$$

if n is even. Unfortunately, one of the weaknesses of median filters is that they cannot be characterized in the same way that linear filters can, since the filter itself necessarily depends on the surrounding values of the input signal.

Looking at a spectrogram, percussive elements more closely resemble impulses in that their frequency content is spread out in a more broadband sense than harmonic content. Put another way, they often appear as vertically-oriented lines in the spectrogram rather than the horizontal lines that correspond to the harmonic elements. When a median filter is applied across frames of the spectrogram, the percussive content is suppressed, and when it is applied across frequency bins of the spectrogram, harmonic content is suppressed. In HPSS, the median filter is applied to the spectrogram of the input signal twice: once to create a soft mask for the percussive elements and a second time to create a soft mask for the harmonic elements [4]. The resulting soft masks can be applied to the original magnitude spectrogram to recover both the complex harmonic spectrogram and the complex percussive spectrogram. These masks are derived from Wiener filtering methods [4].

Given an input magnitude spectrogram $\boldsymbol{S}$, where S_i is the ith time frame and S_h is the hth frequency bin, the median filter $\mathcal{M}$ is applied in each direction resulting in the corresponding enhanced frame.

Percussion-enhanced frame:

$$P_i = \mathcal{M}\{S_i, l_{perc}\}$$

Harmonic-enhanced frame:

$$H_i = \mathcal{M}\{S_h, l_{harm}\}$$

The soft masks are formed (based on Wiener filtering).

Percussive soft mask:

$$\boldsymbol{M}_{\boldsymbol{P}_{h,i}} = \frac{\boldsymbol{P}_{h,i}^p}{(\boldsymbol{H}_{h,i}^p + \boldsymbol{P}_{h,i}^p)}$$

Harmonic soft mask:

$$\boldsymbol{M}_{\boldsymbol{H}_{h,i}} = \frac{\boldsymbol{H}_{h,i}^p}{(\boldsymbol{H}_{h,i}^p + \boldsymbol{P}_{h,i}^p)}$$

where p is the power to which each element is raised (typically 1 or 2).

The complex spectrograms can then be reconstructed by applying the soft masks to the original magnitude spectrogram $\hat{\boldsymbol{S}}$.

Percussive complex spectrogram:

$$\hat{\boldsymbol{H}} = \hat{\boldsymbol{S}} \otimes \boldsymbol{M}_{\boldsymbol{H}}$$

Harmonic complex spectrogram:

$$\hat{\boldsymbol{P}} = \hat{\boldsymbol{S}} \otimes \boldsymbol{M}_{\boldsymbol{P}}$$

where $\otimes$ is elementwise multiplication.

II. Methodology

The piece of music chosen for analysis is a personal edit of a song by Lost Souls Of Saturn called Ring Transmission (Mathew Jonson Space Opera Remix). It was exported via Ableton Live with a sample rate of 48 kHz and a bit depth of 24. It was processed via Librosa's data stream functionality with the following characteristics:

- Frame length (number of samples per frame): 2048
- Block length (number of frames per block): 16

The main algorithmic steps toward BPM detection based on median-filtered HPSS follow.

HPSS tempo detection

1) Short-time Fourier transform (STFT) across a block to obtain magnitude spectrogram
2) HPSS
   - Apply median filter across frames
   - Apply soft mask to mask percussive content
   - Apply median filter across frequency bins
   - Apply soft mask to mask harmonic content
3) Inverse short-time Fourier transform (ISTFT) of complex percussive spectrogram to recover percussive signal
4) Calculate Instantaneous power of block
5) Apply smoothing filter to instantaneous power
6) Thresholding and peak detection
7) Tempo estimation using peak periods

A median filter length of 31 was selected and produced good results for this particular example. [4] suggests a filter length of 17, or a range between 15 and 30 for both the harmonic and percussive filters. A filter length of 17 reduced average computation time by a factor of about 0.6, but it resulted in fewer successful tempo detections. It is possible that the filter length may need to be adjusted for other genres of music. A power factor of 2 was selected based on the fact that this led to increased separation between harmonic and percussive elements, as noted in [4]. A threshold of 0.17 was applied to detect peaks in the instantaneous power. This also may need to be adjusted to adapt to other pieces of music. A filter length of 17 required a slightly higher threshold (0.18) in order to eliminate incorrect detections.

The tempo for each block is calculated by taking the sample rate and dividing it by the mean of the detected maxima's first differences, which gives beats per second. This is multiplied by 60 to make units of BPM.

A. Low-pass filter design

A finite impulse response (FIR) filter based on the Hann window was selected due to it being computationally efficient, easy to implement, and a common window used in digital audio filtering due to its linear phase characteristic. One drawback is that since FIR filters are sample rate dependent, the filter would need to be recalculated if working at a different sample rate. A rough estimate of the necessary filtering was calculated based on the temporal characteristics of most popular music:

180 beats per minute is taken to be an upper limit. Then

$$180 \frac{\text{beats}}{\text{min}} \times \frac{1}{60} \frac{\text{min}}{\text{secs}} = 3 \frac{\text{beats}}{\text{sec}} = 3 \text{ Hz}$$

It was determined that this amount of filtering would require about 8000 samples, incurring roughly 167 ms in delay. This amount of delay is unacceptable for most real-time audio applications, so the filter's specifications were relaxed. It was discovered that a filter length of 801 provided enough filtering of the instantaneous power signal such that the peaks could be reliably detected in this particular example.

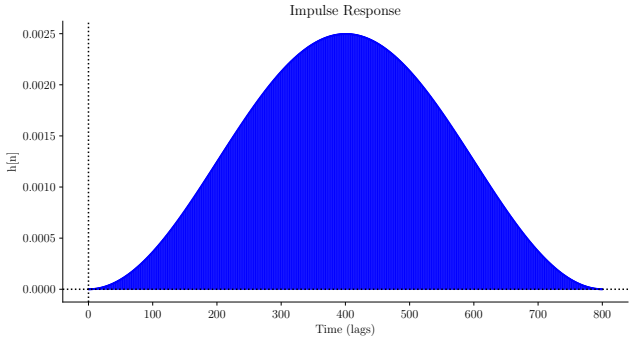


Fig. 1: Impulse response for Hann window (FIR lowpass)

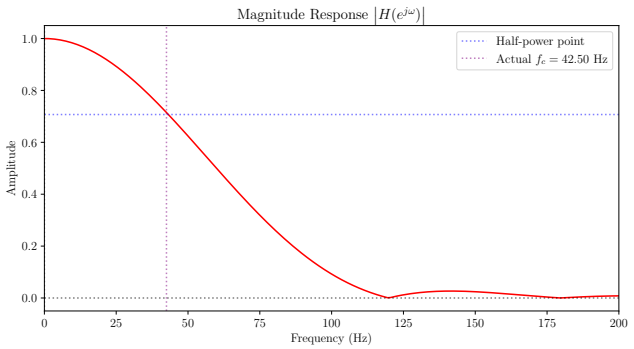
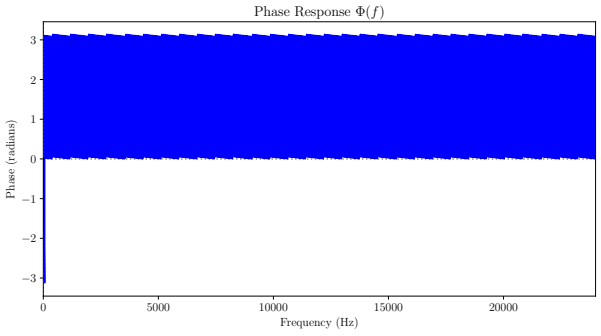
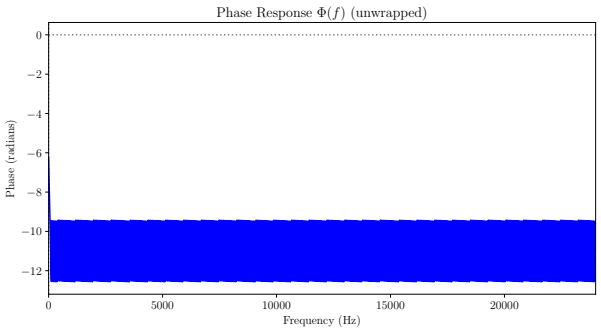


Fig. 2: Magnitude response for Hann window (FIR lowpass)

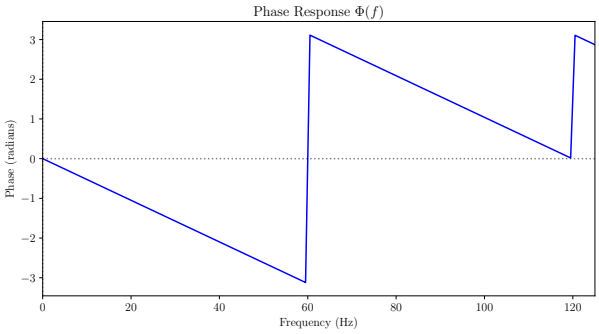


(a) Phase response

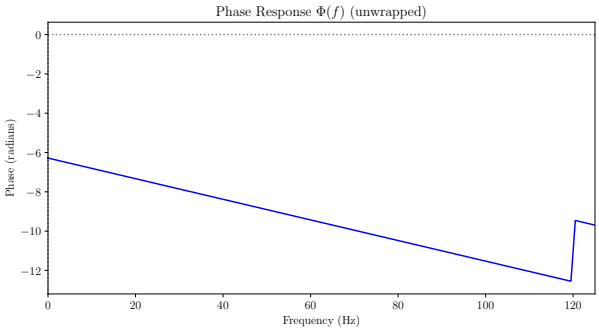


(b) Unwrapped phase response

Fig. 3: Phase response for Hann window (FIR lowpass)

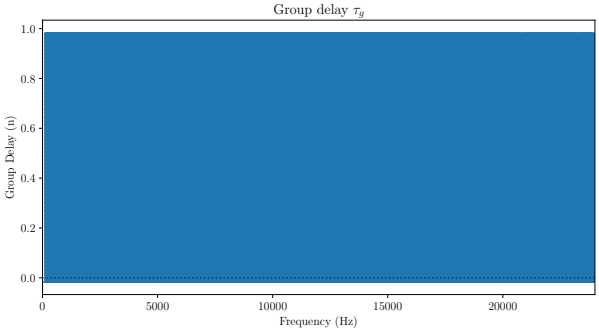


(a) Phase response in passband

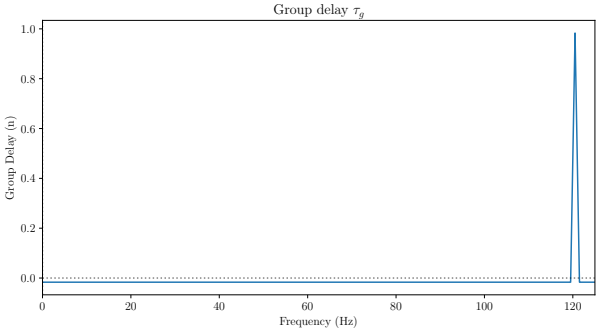


(b) Unwrapped phase response in passband

Fig. 4: Phase response in passband for Hann window (FIR lowpass)



(a) Group delay



(b) Group delay in passband

Fig. 5: Group delay for Hann window (FIR lowpass)

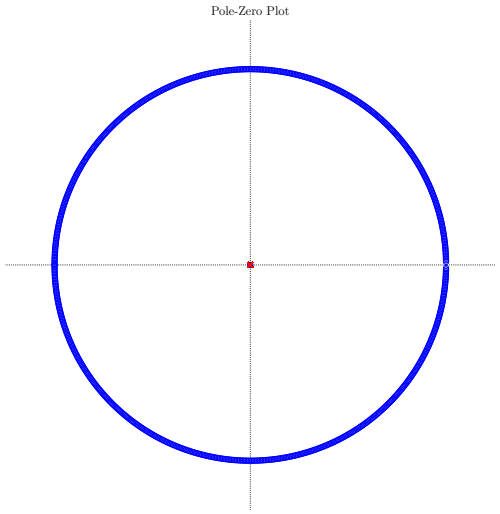


Fig. 6: Pole-zero plot for Hann window (FIR lowpass)

III. Results

The spectrograms produced for the offline analysis of the HPSS step were formed by an FFT length 256, were padded to a length of 2^{16} and had an overlap of 128. Parameter optimization was focused on other areas of the algorithm, so it is possible that other values (such as a longer FFT length) could yield better results in offline processing, as noted in [4].

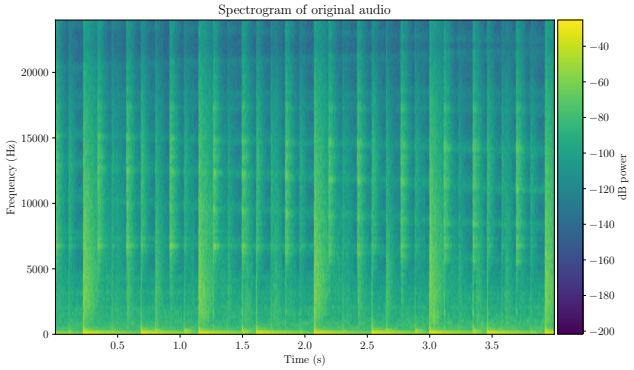


Fig. 7: Spectrogram of original signal

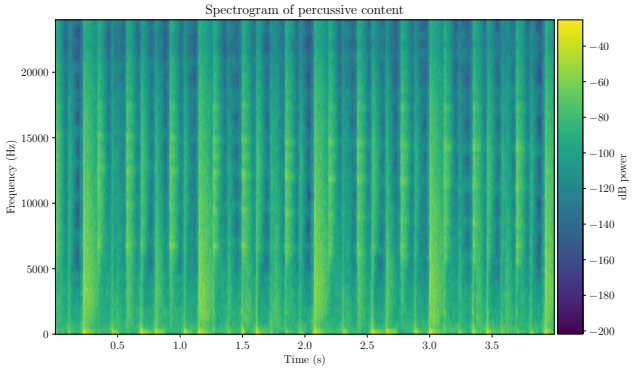


Fig. 8: Spectrogram of percussive content

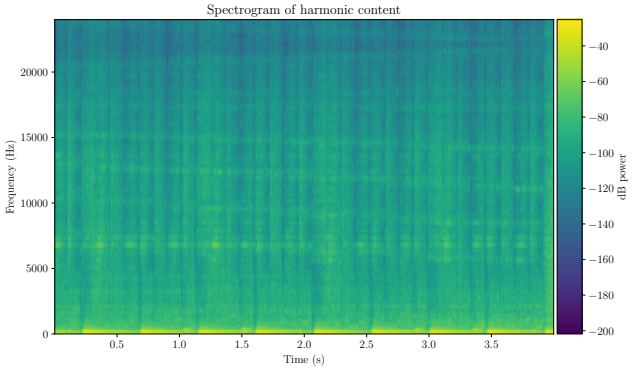
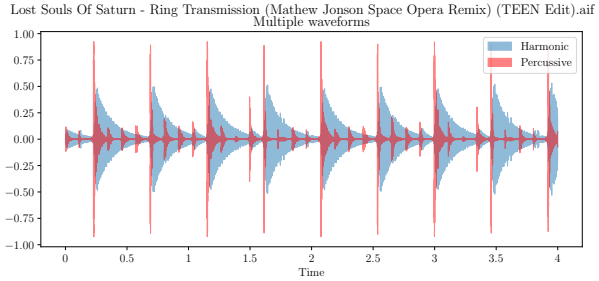
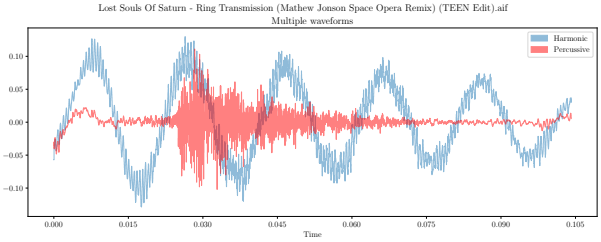


Fig. 9: Spectrogram of harmonic content



(a) Waveforms after HPSS



(b) Zoomed-in waveforms after HPSS

Fig. 10: Multiple waveform view of separated harmonic and percussive content

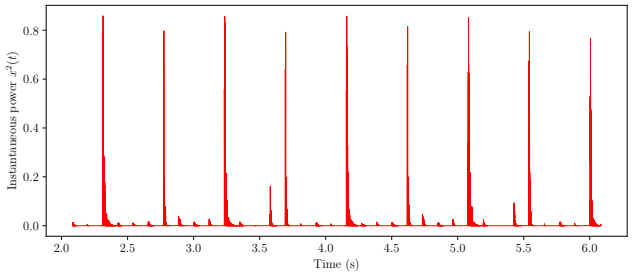


Fig. 11: Example instantaneous power of percussive content

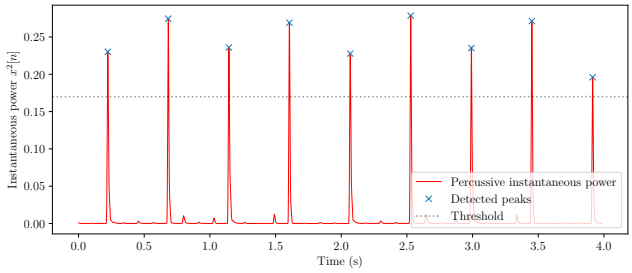


Fig. 12: Example peak detection on filtered instantaneous power of percussive content

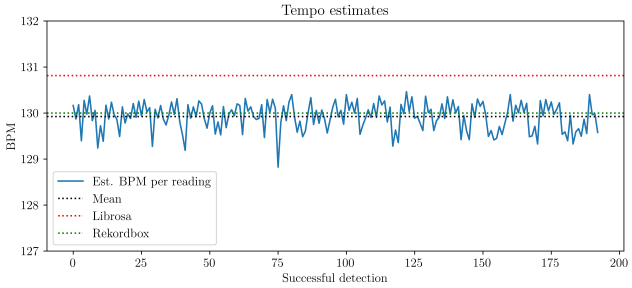


Fig. 13: Successfully detected tempos compared to other algorithms

IV. Discussion

HPSS is a compelling application of median filtering to the audio processing domain. It was able to successfully separate percussive elements from harmonic elements in a piece of electronic dance music, which naturally has some separable qualities compared to other genres. More investigation is needed to determine whether it can handle other genres with similar success.

Although the FIR smoothing filter design did not meet the initial specifications, these specifications were relaxed with considerations for incurred delay. It performed well at smoothing the instantaneous percussive power such that the beats could be accurately detected in the particular piece of music chosen for analysis. While not every processed block produced a successful tempo estimate, out of 603 blocks, 193 produced usable estimates. With a song length of 6:51, that means 2.129 seconds elapsed per successful detection in a simulated real-time scenario. This is sufficient for genres of music with nearly static tempo such as many electronic dance music genres, but may be underperforming on genres with rapid tempo changes.

Compared to Librosa's tempo detection algorithm, which uses envelope onset autocorrelation, and Pioneer rekordbox's proprietary algorithm, the results from using HPSS show a close match in accuracy to these existing methods. It was interesting to note that Librosa's implementation had a positive bias in its tempo estimation compared to the other methods, and rekordbox's estimate was very close to the tempo reported upon export from Ableton Live as well as the mean value estimated via HPSS peak detection.

The causal FIR filter design does come with some amount of lag, but at 32768 frames processed per iteration and and average rate of 15.51 blocks per second processing time, 507153.59 samples are processed per second on an M1 Mac with 16GB of RAM. Assuming a sample rate of 48 kHz, this would appear sufficient for most real-time processing scenarios.

V. Conclusion

An algorithm for near real-time tempo detection in monaural audio was presented. Good preliminary results were obtained from detecting beats in music via harmonic-percussive source separation (HPSS) using median filtering of the spectrogram followed by filtering the instantaneous power in an online scenario. More work remains in developing an algorithm that is both robust to large and instantaneous changes in tempo and that is able to correctly analyze other pieces of music within the electronic genre and music from other genres.

References

[1] G. Tzanetakis, "Tempo extraction using beat histograms," in *Proceedings of the 1st Music Information Retrieval Evaluation eXchange (MIREX 2005)*, 01 2005.

[2] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals." *The Journal of the Acoustical Society of America*, vol. 103, pp. 588–601, 1998.

[3] G. Tzanetakis, G. Essl, and P. Cook, "Audio analysis using the discrete wavelet transform," in *Proc. conf. in acoustics and music theory applications*, 2001.

[4] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *13th International Conference on Digital Audio Effects (DAFX10)*, 2010.

[5] D. Fitzgerald, E. Coyle, and M. Cranitch, "Using tensor factorisation models to separate drums from polyphonic music," in *Proceedings of the International Conference on Digital Audio Effects (DAFX09)*, 2009.

[6] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *2005 13th European Signal Processing Conference*, 2005, pp. 1–4.

[7] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *2008 16th European Signal Processing Conference*, 2008, pp. 1–4.