REPET Algorithm: Background Music Separation Using Auto-correlation

Tongyu Lu, 31-March-2021

This is a note for Zafar Rafii's REpeating Pattern Extraction Technique (REPET) algorithm in its basic form. REPET is a super-simple algorithm which could separate human voice from accompanied music, although it is effective only in typical cases when background music is highly repetitive.

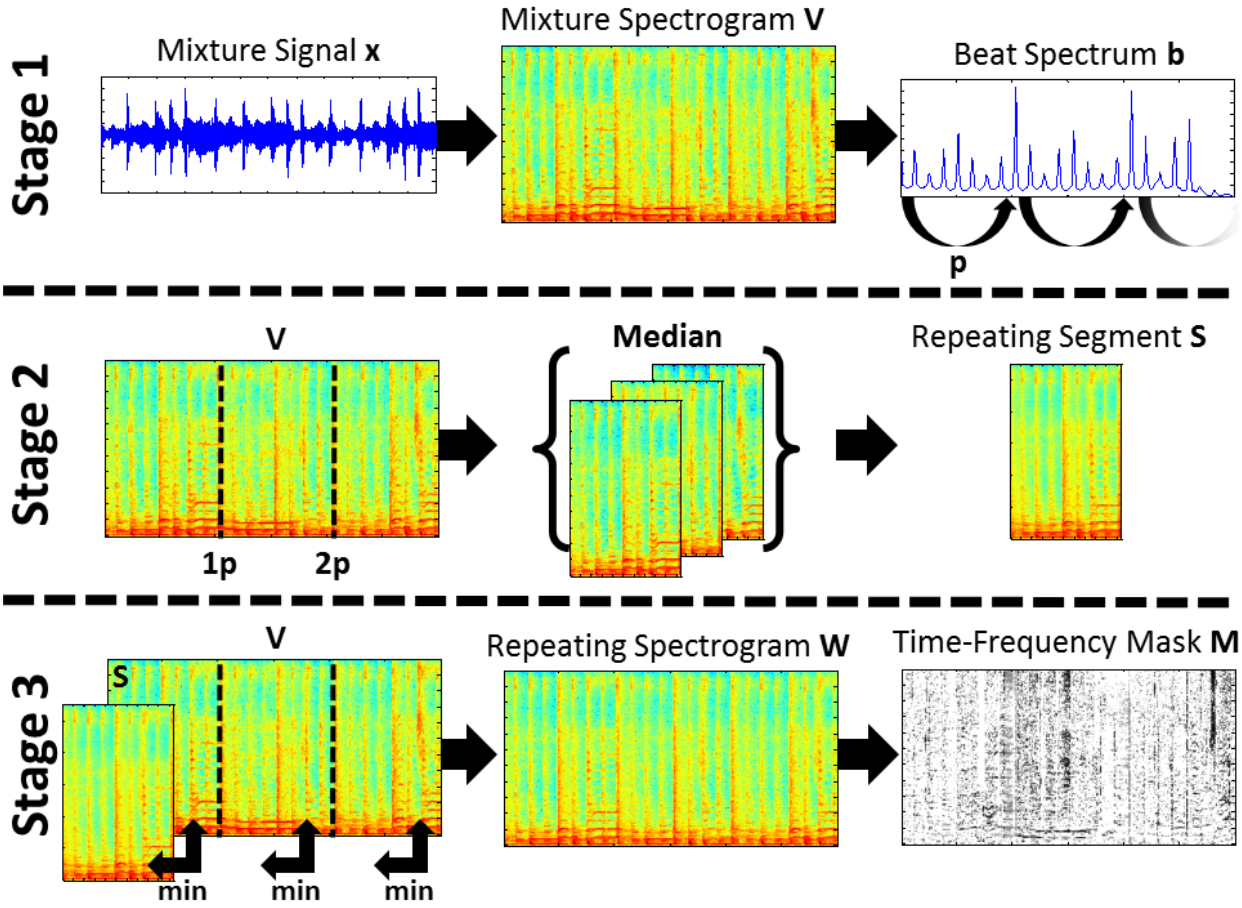
REPET code:

> https://github.com/zafarrafii/REPET-Python

TEPET paper:

> Z. Rafii and B. Pardo, "Repeating pattern extraction technique (REPET): A simple method for music/voice separation," IEEE transactions on audio, speech, and language processing, vol. 21, no. 1, pp. 73–84, 2012.

The REPETidea is shown in the following figure (which is cited from the original paper):



The calculation procedure is illustrated as follows:

- Define $\mathbf{STFT}$ and $\mathbf{iSTFT}$ calculation function for music audio
  Suppose the input audio is $x_{1:T}$. We calculate $X_{1:F,1:N} = \mathbf{STFT}(x)$ by first segmenting x into N segments by windowing the audio, and then calculate FFT for each segment; finally, concatenate them together. (The traditional STFT calculation procedure.) Similarly, we could define inverse STFT.
- Define auto-correlation function $\mathbf{acorr}(\cdot)$ using Wiener–Khinchin theorem:
  1. input a vector $x_{1:N}$
  2. compute its power spectrogram $S_{1:F} = |\mathbf{STFT}(x)|^2$
  3. compute the iSTFT for power spectrogram $r_{1:\tau} = \mathbf{Re}[\mathbf{iSTFT}(S_{1:F})]$
- Stage 1: calculate auto-correlation for each freq column and summarize them into beat spectrum
  1. Given $X_{1:F,1:N}$, we have F sequences of frequency over time. Treat those frequency-fluctuation series as signals and calculate auto-correlation for each of them, getting $R_{f,1:\tau} = \mathbf{acorr}(X_{f,1:N}), f = 1, \dots, F$, which becomes $R_{1:F,1:\tau}$
  2. Summarize $R_{1:F,1:\tau}$ over frequency domain, getting $b_{1:\tau} = \sum_{f=1}^{F} R_{f,1:\tau}$
  3. find the salient repeating period
     Give a start lag $\tau_0 > 1$, because $b_{\tau=1}$ is always the largest; give a ending lag $\tau_e \leq [\tau/3]$; select period $P = \arg\max_t b_{t \in [\tau_0:\tau_e]}$.
- Stage 2: calculate the median of STFT spectrogram periodical segments
  1. cut $X_{1:F,1:N}$ into $L = [N/P]$ segments: $X^s_{1:F,1:P,l} = X_{1:F,(l-1)P+1:lP}, l = 1, \dots, L$
  2. calculate median on the L axis: $X^m_{1:F,1:P} = \mathbf{median}_{l=1:L}(X^s_{1:F,1:P,l})$
- Stage 3: calculate the background music and foreground music

1. for $l = 1, \ldots, L$, calculate $X^{bg}_{f,p,l} = \min\{X^{m}_{f,p}, X^{s}_{f,p,l}\}$, where $\min$ operation compares the absolute value. Reshape $X^{bg}_{f,p,l}$ into $X^{bg}_{1:F,1:N}$, and we get the background music spectrogram.
2. foreground music could be calculated using $X^{fg} = X - X^{bg}$.
3. compute $iSTFT$ for X^{bg} and X^{fg}, getting the final output.

REPET is only effective when the background music has repeating pattern. However, this algorithm is super easy to implement, and it needs no training data. REPET might be a hint for designing further MSS algorithms by considering repeating patterns. And this algorithm also need to be adapted into multi-instrument case.
I think that the basic REPET algorithm could serve as de-noising front-end processing block for voice separation applications.