SMSA Mini Project

Emmanouil Dimogerontakis

August 13, 2022

1 Introduction

In this project, a musical transcription algorithm will be presented as part of the SMSA course. An alternative way to re-synthesize the individual sources, or to transport the extracted musical information into meaningful controls for a modular synthesizer, will be introduced. The Non-Negative Matrix Factorization will be used as the algorithm to analyze and detect the individual sources of a monophonic drum recording, after this information will be converted into information that will control a modular synthesizer.

First, the theoretical part of the Non-Negative Matrix Factorization and basic principles of Control Voltages signals will be introduced. In the second part, the implementation will be explained. And in the last part, the basic conclusions and future implementations will be presented.

2 Theoretical Background

As it is mentioned above, the Non-Negative Matrix Factorization (NMF) will be used as the technique to analyze and extract musical information from an audio signal of a monophonic drum recording. This information will be transformed to Control-Voltages to control a Eurorack synthesizer.

2.1 NMF for Drum Transcription

Mathematically, NMF is based on iteratively computing the approximated spectrogram $\tilde{V} \in R$ of an initial spectrogram V[1]. Rephrasing the above sentence, the approximated $\tilde{V}$ can be defined as the combination of a matrix with the frequency templates W and a matrix with the activations over time H. Such as:

$$V \approx \tilde{V} = WH \tag{1}$$

Algorithmically, to compute the approximation of the V, W, and H are initialized by non-negative random numbers (in the unsupervised method), and then by iterative updates, we try to minimize the error. This technique of minimization is called the multiplicative gradient descent algorithm. And can be described as:

$$minD(V \parallel WH) \tag{2}$$

where W and $H \geq 0$, and D as the measure of the 'divergence'.

For computing D there are some different choices. In the NMF algorithm with the so-called β-divergence or cost functions, there are 3 choices for D. With $\beta = 0$ to be the Itakura-Saito (IS) divergence, which is scale-invariant. That means, low and high-energy components have the same weight (same chances to be detected). With $\beta = 1$ as the Kullback-Leibler (KL) cost function, this one is scale-variant. The last one is called Euclidean and like the KL is scale-variant [1] [2].

Practically, depending on the case (the usage of the algorithm, the type of the musical signal) different types of the cost function can be used. For example, to detect and analyze a drum signal to each source/component, it is possible to use KL divergence. That is because drums usually are transient signals with lots of energy in a small segment of time [3]. For signals with more dynamics, the optimal choice is the IS divergence.

The output of the NMF is the matrices W(bases) and H(activations), inside these matrices the information of the components, is stored. Inside the W matrix, the frequency information for each component is stored in columns, and the number of the columns depends on the number of the wanted

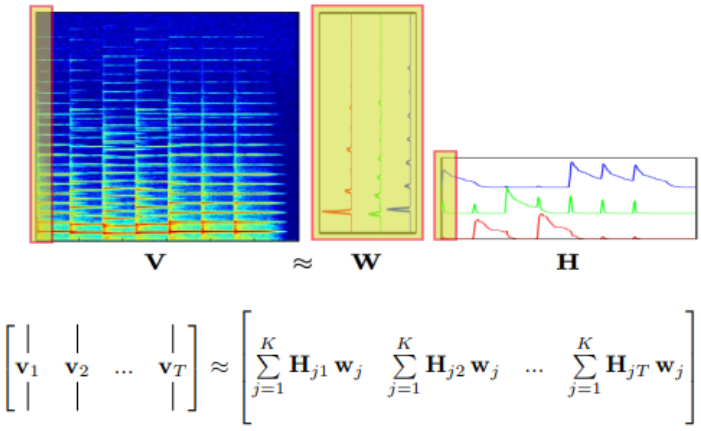

Figure 1: How a decomposition of 3 sources is made by NMF[2]

.

components. Into the H matrix, the time envelopes/activations are stored in rows, and the same principle, as in the W matrix, yields for the number of rows, as is observed in the figure 1. To simplify what this information means, we can think of H as gain envelopes (ADSR) and the W as the filter or IR of each component.

2.2 CV signals in Analog Synthesizers

Control voltages are DC electrical signals for controlling automatically various parameters in a modular synthesizer. The most known values are from 0V-5V, although this can differ from manufacturers and the audio units. With that being said, these envelopes inside of the matrices can be mapped to voltages and control different parameters in a modular system. The interpretation of the H matrix (activations) can be mapped directly to the CV of a VCA and re-create these envelopes for each individual source. In the case of the W matrix (bases), the most simple idea is to control the frequency and the amplitude of bandpass filters with spectral information. In our case, the spectral envelopes are converted into FFT filters which are filtering white noise.

3 Implementation

For the implementation, Max/MSP was used and the the externals from FluCoMa[1]. This patch is taking a monophonic audio signal and decomposes it into a number of sources that the user can select. Then, with the external object `fluid.bufnmf` the signal is decomposed into the W and H matrices. These matrices are stored in buffers. Initially, there are 2 buffers with the information of the activations and the bases. For better manipulation, these buffers are separated from each other.

For the activations, these matrices are being read as a signal with the object `groove` , this object reads from a buffer and playback it. After that, the signals are going to the `matrix` object to be separated into individual channels. Making this route the user is able to merge or separate the playbacks. Then the outputs are going to `VCA` objects from the BEAP library [2]. For the bases, the spectral information of the components is stored in individual buffers. Then, these buffers are read with the `index` object inside of the STFT process. To create an FFT filter in Max/MSP is straightforward, first the `pfft` object is used as the STFT process, this gives 3 outlets, the real and the imaginary part of the signal and the bin index of the spectrum. A conversation is made to magnitude and phase from real and imaginary parts with the object `cartopol` and by synchronizing the buffer reading with the FFT bins of the signal and then multiplying each buffer bin (FFT bin of the buffer) with the magnitude of each FFT bin of the signals, the FFT filter is created. To recreate the audio signal the inverse STFT is applied with the `ipfft` object. The input of the filters is noise,

[1] https://www.flucoma.org/
[2] https://cycling74.com/tutorials/beap-analog-model-curriculum-outline/

Figure 2: The patch. In the upper left, we can observe the content of the activations and the bases buffers.

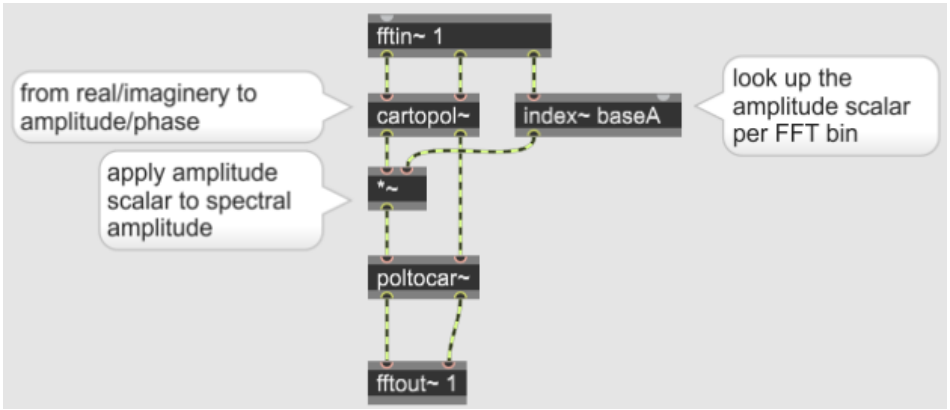

Figure 3: Inside the `pfft` object.

the filtered noise from the spectral envelopes of the bases is patched to the VCA and is controlled by the time envelopes of the activations.

4 Conclusions and Further work

The project was made inside the Max/MSP software, although by using an audio interface with DC coupled outputs, the control of a hardware modular synthesizer is a straightforward procedure. In a future implementation, modal synthesis needs to be tested. Using the information from the extracted bases, to recreate the source by tuning its modes.

References

[1] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription."

[2] N. Bryan and D. Sun, "Source separation tutorial mini-series ii: Introduction to non-negative matrix factorization," 2013.

[3] D. T. Fakultät and C. D. aus Jena, "Source separation and restoration of drum sounds in music recordings quellentrennung und restauration von schlagzeugklängen in musikaufnahmen."

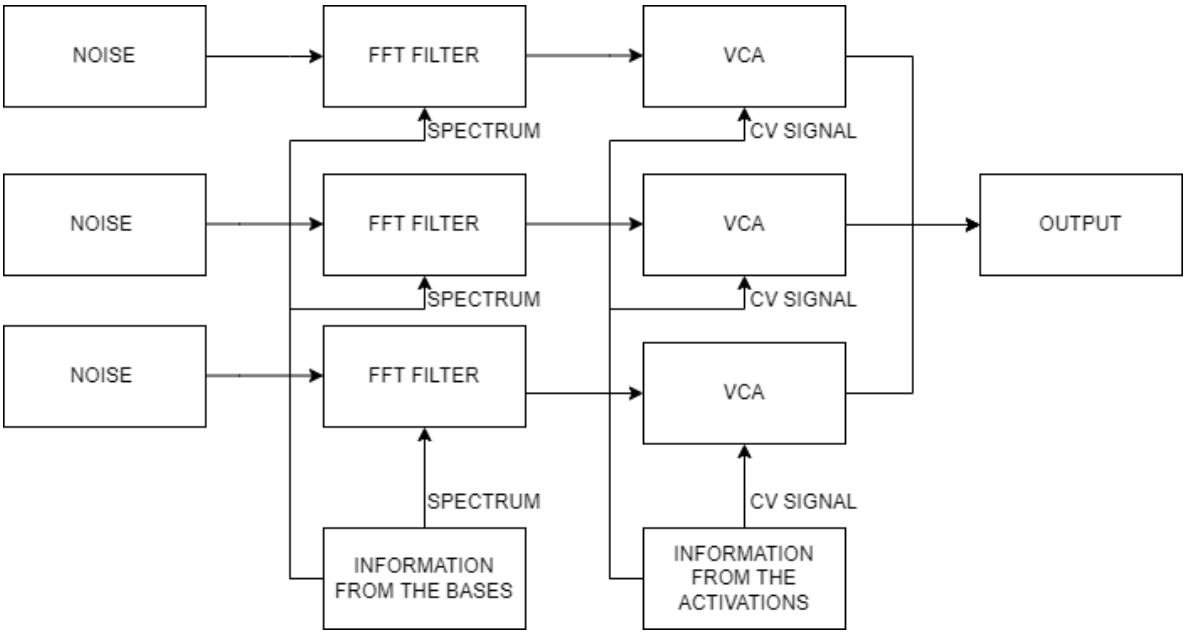

Figure 4: Signal flow of the implementation.