School of Electronic Engineering and Computer Science

Final Report

Programme of study:

Electrical Engineering and Electronics

Project Title:

Concatenative Sound Synthesis Using Non-Negative Matrix Factorization Techniques

Supervisor:

Dr. Bob Sturm

Student Name:

Michael Buch

Queen Mary University of London

Date: 25th of April 2016

ABSTRACT

Recent work explored applying non-negative matrix factorization (NMF) to concatenative sound synthesis (CSS). In this case, a target sonogram is factored into a product of the corpus sonogram and a matrix of activations. Various constraints can also be used to elicit specific properties of the activations, e.g., sparsity. An inverse short-term Fourier transform (STFT) then synthesises the result from the product of the corpus STFT and activation matrix. In this paper, we make four contributions CSS with NMF: 1) we develop a simpler method for synthesis than the inverse STFT; 2) we test other non-negative features (Constant-Q spectrograms and chroma); 3) we propose an interactive editing of the activation matrix ("activation sketching"); and 4) we provide a free MATLAB software implementation (`https://github.com/Michael137/MFAMC`).

1. INTRODUCTION

[10] adapts and applies NMF methods to audio mosaicing also known as concatenative sound synthesis (CSS). In the field of musical sound synthesis, CSS is a popular method of combining sound units to model a target sound.

Non-Negative Matrix Factorization (NNMF) is the process of factorizing a non-negative matrix into non-negative factors. Although several approaches to NNMF exist, [10] and this project focus on the traditional NNMF algorithms described in [9]. Although NMF finds applications mostly in areas like source separation and pattern recognition, its non-negativity constraints translate well into the field of audio as well since one of the main signal analysis tools used in audio is a non-negative matrix: the spectrogram.

To enforce certain behavior onto the activation matrix, one of the factors resulting from NMF, Driedger et al. [10] extend the basic NMF algorithms with a set of modifications to contribute to the development of patterns within the activation matrix. More specifically, these update rule modifications create diagonal structures in the learned matrix and thus allow features from the source sound to be preserved during the synthesis process, which is usually not the case with NMF synthesis. The features discussed in [10] are timbre and temporal characteristics and analyzed by inspection of the synthesis spectrograms.

In an attempt to further investigate the the application of mathematical algorithms and concepts such as NMF when applied to a completely unintended field like audio synthesis and enhance the understanding of the relation between mathematical and more subjective musical interpretation, our contributions with this project include: 1) A simple synthesis method 2) Feature based NNMF synthesis 3) "Activation Sketching" and "Template Manipulation" tool which allows interactive synthesis using NNMF 4) NiMFKS: an open source Matlab implementation encompassing all synthesis techqniues and topcis discussed in this paper.

With its interactive functionality, our application allows users to customize every step of the synthesis process without limitation and due to the plug 'n' play model (i.e. new modules such as new features for analysis or new algorithms for various steps of the synthesis process can be implemented into the application for experimental purposes) adopted in its implementation we encourage further research into this method of sound synthesis not only in exploring a variety of sound combinations but different features and algorithms involving matrix representation and factorization applied to sounds.

2. BACKGROUND

2.1 Audio Analysis Terminology

This section will give a brief overview of the necessary terminology and topics from the field of Audio Signal Processing that are mentioned throughout the paper.

Spectrogram: Three-dimensional representations of a signal often used to analyze its spectral content. The horizontal axis denotes time, the vertical frequency and the color intensity. Using this information, it follows that each column of the spectrogram corresponds to the spectrum (range of frequencies and their weights) within the signal at the corresponding time frame. In this project these spectra are also referred to as "templates".

It is computed from the magnitude of the Short-Time Fourier Transform (STFT). More specifically, a spectrogram can be created by taking the Fast Fourier Transform (FFT), a computationally efficient implementation of the Discrete Fourier Transform (DFT) used to find the set of sinusoids or frequency components that make up a signal, of windowed segments of a signal and stacking the outputs next to each other forming a matrix where each column is a FFT sequence at specific time frames of the signal [1]. The window and associated variables such as its length, hop size and overlap effect the structure and length of the individual segments used in the creation of the spectrogram. The window size defines the length of the segments taken from the original signal by the STFT to perform the FFT onto. The hop size defines how many samples away the next segment in the signal is from the previous one. Overlap is the hop size as a percentage of window size.

An example of a spectrogram can be seen in figure 10.

Constant-Q: When applying the DFT to a signal e.g. during the construction of a spectrogram, the resulting frequency components are equally spaced when plotted as linear frequency. However, on a log-frequency (dB) scale, certain frequencies are squeezed while others are expanded. The "Constant-Q Transform" (CQT) [2] is a modified version of the DFT that provides a more suitable mapping of the frequency components of a signal to musical scales and when plotted against log-frequency, also called "Constant-Q Spectrograms" creates a constant pattern in the frequency domain.

Pitch: A subjective feature based on one's perception of a sound. More specifically, it is the property of a sound that allows a listener to match or order it according to another sound in terms of frequency [3]. In addition to using it for the construction of audio analysis structures such as "Chromagrams", in this paper it is also referred to when discussing how "high" or "low" a sound seems compared to another [4].

Timbre: This can be understood as the feature that allows a listener to distinguish two sounds of the same pitch and loudness [4]. Being a multidimensional attribute, it is not easily quantifiable. However, in an abstract sense it can be viewed as simply the detection of one sound within another. This definition will be used when mentioning timbre throughout this paper and observed by visual inspection and comparison of a sound's spectrogram to others.

Chroma: Based on the idea that a musical note can be represented by a combination of other musical notes, "Chroma Representation" allows the spectrum of a signal to be represented by a "Pitch Class" i.e. discrete set of bins where each bin represents a pitch [3]. This representaion is also referred to as a "Chromagram" an example of which can be seen in figure 12. For computation of chromas and creation of chromagrams we use the implementation from [5].

2.2 Concatenative Sound Synthesis

2.2.1 Technical Details

Given a database of source sounds called a "corpus" which consists of a wide range of sound segments called "units", the goal of CSS is to find and concatenate these source sounds to produce a synthesis that best fits another given sound called the "target". Provided with best match criteria, selection algorithms scan through the database, compare the units to the target sound and pick out the closest matching sources to finally combine these into an approximation of the target. The analysis that the algorithm performs is usually based on different types of features within the source and target and varies between applications. At a high level, CSS can be described as a three step process: 1) Analyse corpus and target sound 2) Select best matching segments 3) Combine segments into synthesis [6]. The analysis step segments the corpus into units and assigns each with "descriptors", which are a way of classifying sound according to a feature or set of features within it. These can vary from low-level descriptors, i.e. based on features derived directly from the signal's physical properties such as signal energy, to high-level descriptors that have very little to do with the actual signal but can describe music in a more abstract sense and provide semantic meaning, such as musical notes or timbre. Based on the approach to the steps involved in CSS, [7] provides a taxonomy and description of existing applications within this field. Whereas tools like "MATConcat" [8] focus on low-level features, other applications such as "Caterpillar" [6] make use of high-level descriptors to select the best matching corpus units. Our application attempts to differentiate itself from other state of the art synthesis systems by providing a new approach to synthesis using NMF while also making it interactive.

2.3 Non-Negative Matrix Factorization

2.3.1 Technical Overview

The NMF procedure described in [9] uses iterative update rules to find an approximation of a matrix in terms of two other matrices based on a cost metric defining the approximation error from multiplying the factors.

The basic idea of NMF for sound synthesis is that the two factors that result from applying the operation to a spectrogram, are a source and an activation matrix, which when multiplied together approximate the original spectrogram. If the source matrix is now fixed to a spectrogram of a source sound, the goal of this synthesis method is to find an activation matrix H, that when multiplied by the source sound spectrogram will approximate a target. When converting back the resulting reconstruction, we obtain a sound that is an approximation of a target in terms of a source (also corpus), similar to the output a CSS application provides.

2.3.2 NMF For CSS

We now review the method proposed by Driedger et al. [10]. Denote the short-term Fourier transform (STFT) of the target as $\mathbf{S}_T$, of the corpus as $\mathbf{S}_C$. NMF algorithms allow us to find a non-negative matrix $\mathbf{H}$ such that

$$|\mathbf{S}_T| \approx |\mathbf{S}_C|\mathbf{H} \tag{1}$$

where $|\cdot|$ applies the modulus element-wise to the argument. Given an "activation matrix" $\mathbf{H}$ then, an inverse STFT of the product $\mathbf{S}_C\mathbf{H}$ synthesizes the time-domain result, e.g., using the Griffin-Lim Algorithm [11]. Our application uses the Matlab implementation of the ISTFT from [12].

Finding the factor $\mathbf{H}$ is typically done iteratively, with update rules coming from how the cost function (approximation error) is defined [13]. Driedger et al. [10] use the Kullback-Leibler divergence, which specifies the update rule for the (l, m) element of $\mathbf{H}$ (denoted $[\mathbf{H}]_{lm}$):

$$[\mathbf{H}]_{lm} \leftarrow [\mathbf{H}]_{lm} \frac{\sum_i \frac{[|\mathbf{S}_C|]_{il}[|\mathbf{S}_T|]_{im}}{[|\mathbf{S}_C|\mathbf{H}]_{im}}}{\sum_k [|\mathbf{S}_C|]_{kl}} \tag{2}$$

with $\mathbf{H}$ initialised by sampling all its elements from a uniform random distribution over $[0, 1)$.

For comparison and an additional option for NiMFKS, update rule (2) is replaced by an expression which aims to minimize Euclidean distance between the target and the reconstruction:

$$[\mathbf{H}]_{lm} \leftarrow [\mathbf{H}]_{lm} \frac{([|\mathbf{S}_C|]^T[|\mathbf{S}_T|])_{lm}}{([|\mathbf{S}_C|]^T[|\mathbf{S}_C|]\mathbf{H})_{lm}} \tag{3}$$

2.3.3 Extensions with Constraints

Driedger et al. [10] experiment with how to avoid particular behaviours in the activation matrix, i.e., encouraging diagonal structures over horizontal and vertical ones. To suppress horizontal structures like corpus unit repetitions (called "Repitition Restriction" in [10]), they process the activation matrix after its full update in the kth iteration of K by defining

$$[\mathbf{H}]_{lm} \leftarrow \begin{cases} [\mathbf{H}]_{lm}, \text{ if } [\mathbf{H}]_{lm} = \max\{[\mathbf{H}^T\mathbf{e}_l]_{m-r:m+r}\} \\ [\mathbf{H}]_{lm}(1 - \frac{k+1}{K}), \text{ else} \end{cases} \tag{4}$$

where $\mathbf{e}_l$ is the lth standard vector, and r describes the number of columns to the right and left of the mth activation column in which one wishes to reduce repeated non-zero values. To suppress many non-zero values in each column of $\mathbf{H}$ (called "Polyphony Restriction" in [10]), they process the activation matrix after its full update in the kth iteration of K by defining

$$[\mathbf{H}]_{lm} \leftarrow \begin{cases} [\mathbf{H}]_{lm}, \text{ if } [\mathbf{H}]_{lm} \in \max_p\{\mathbf{H}\mathbf{e}_m\} \\ [\mathbf{H}]_{lm}(1 - \frac{k+1}{K}), \text{ else} \end{cases} \tag{5}$$

where $\max_p\{\cdot\}$ returns the subset of p largest values of its argument. To promote diagonal structures, they process the activation matrix after its full update in the kth iteration of K by filtering it with a size c identity matrix (called "Continuity Enhancement" in [10]):

$$\mathbf{H} \leftarrow \mathbf{H} * \mathbf{I}_c. \tag{6}$$

After all of these steps, the elements of the activation matrix are once again updated using (2), and the matrix is processed again. In summary, this CSS process proposed by Driedger et al. [10] thus involves the definition of the total number of NMF iterations K, the horizontal neighbourhood r, the vertical magnitude neighbourhood p, the diagonal kernel size c, and finally the inverse STFT.

3. FROM NMF TO NIMFKS

3.1 Extending Extensions

With the problem set outlined as is in the beginning of sections 2.3, room for investigation and further questions arises. The following sections describe the four main points that drove the need for a standalone NNMF-based sound synthesis application.

3.1.1 What are These Diagonal Patterns All About?

The main intent of the modifications to the NNMF algorithm applied by equations (4), (5) and (6) is the creation of diagonals throughout $\mathbf{H}$. The former two control the number of consecutive activations in a row and column respectively, while the latter enforces diagonals onto each point in the activation matrix by convolving them with a matrix of that shape. During reconstruction, through the process of matrix multiplication, $\mathbf{S}_C\mathbf{H}$, these diagonals produce a scaled repitition of the corresponding part of the source matrix in the product, $\mathbf{S}_T$. This effect is visible in the following example:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 3 \end{bmatrix} = \begin{bmatrix} a & b & 0.5a & 3b \\ c & d & 0.5c & 3d \end{bmatrix} \tag{7}$$

This implies that diagonal structures in the activation matrix cause small scale repition of the source within the target sound thus enhance its perception in the synthesis. Because timbre is closely related to recognition of a sound when played together with other sounds [14], these structures provide the observer with seemingly preserved timbre. A reverse diagonal will result in the same effect with the individual source segments within the output to be in reverse.

By modifying the identity matrix $\mathbf{I}_c$ in (6) we are able to choose which type of structure to enforce within the activation matrix. This provides an algorithmic approach of controlling activation patterns using this synthesis method of which additional ones include: 1) Consecutive Horizontal/ Vertical Activations 2) Rectangular Activations. While horizontal and vertical activations are referred to as "unit repitition" and "polyphony" and supressed in Driedger et al.'s [10] trials, we identify cases where these patterns are of advantage 6.2.

Rectangular patches of activations are caused by filtering the activation matrix with a set of weights causing a blurring effect.

3.1.2 More on Spectrograms and Features

In the description of NNMF synthesis so far we have considered the matrices in (1) to be spectrograms. These are not the only but in fact the less common way of representing signals, particularly in music analysis. Representation such as chromagrams or constant-q spectrograms are much more natural and useful methods of analyzing audio as they provide deeper insight into the melodic and harmonic features of a sound. Since the NNMF synthesis techqniue described previously puts no constraint on the type of matrix used in its update rules apart from non-negativity, different matrix representations of audio can be substituted into the algorithm to serve as alternative feature analysis, similar to how CSS applications differ in the type of features their algorithms use for unit selection.

3.1.3 Sound Recovery

After obtaining the activation matrix, the next major step in the synthesis process is to create the actual time-domain signal. This is done by inverting the spectrogram obtained when multiplying $\mathbf{S}_C$ and $\mathbf{H}$. This seemingly simple task of spectrogram inversion, is a topic of extensive research with numerous methods for various scenarios available. [11] shows that a signal can be reconstructed into its time-domain representation simply from the magnitude of its spectrogram. This process is called the Griffin-Lim Algorithm (GLA) and has since been extended to fit specific needs and is the algorithm used in [10] to synthesize the target sound.

By providing an alternative approach that exploits the fact that we have an activation matrix that specifies the expected source and target template positions within the synthesis, we can provide universality to the synthesis algorithm to allow for other matrices apart from spectrograms to be used in place of $\mathbf{S}_C$. This universal NMF synthesis method is also substantially simpler than the GLA in its implementation. Since this technique is based solely on multiplying and positioning source templates into a new output signal, we provide more control over the algorithm and potentially more room for experimentation such as editing which templates to use in the resynthesis. For this reason NiMFKS also features a Template Editing tool for additional customizability of the synthesis procedure.

3.1.4 Iterations

[1] mentions the possibility of enforcing the restrictions once a certain amount of iterations of the regular update rules has been completed. This results in slightly more accurate reconstruction of the activation matrix before adapting it to a sparse diagonalized matrix. However, the amount of iterations does not only affect the likeliness of the algorithm to converge and yield different reconstruction errors but affects the computation time substantially. Based on the idea of adapting the restrictions at later stages of the update rules, an alternative version of the procedure described in 2.3.3 is only applying restrictions to the very last iteration of the algorithm to garantuee convergence of NMF while preserving characteristics of the source with a decrease in computational time. With this slight modification additional effects are introduced depending on the parameter choices going into the synthesis (see section 4.1).

4. OUR APPLICATION

We now present our contributions to the work of Driedger et al. [10].

4.1 Modified Restrictions

Since the main focus of this synthesis method and driving force behind the synthesis outcome is the activation matrix itself and the patterns it contains, further possible modifications to the restrictions applied previously are investigated.

4.1.1 Cost? Why bother?

As mentioned in [10], the aim of imposing restrictions on the NMF algorithm is not enhance cost minimization but retain characteristics, most notably timbre, of the source sound during synthesis. We investigated this point further and asked whether scenarios existed in which cost reduction actually yielded a perceptually more desirable result and whether a balance of system parameters existed in which the algorithm still converges to a minimum but retains most of the effects the restrictions produce.

Cost minimization for any of the two NMF algorithms discussed in this paper ((2), (3)), means finding $\mathbf{H}$ for which the corresponding cost measure equations yield the lowest value or the value that the cost function converges to. Generally, a lower cost is associated with higher accuracy of an algorithm output. In our application a lower cost implies a more accurate construction of $\mathbf{H}$ and thus in theory a more accurate reconstruction of the target when multiplied by a source spectrogram.

Although applying the NMF procedure on a spectrogram leads to higher factorization accuracy, passing the local minimum of the cost minimizing function can lead to not only unnecessary computational time but also negative effects on the synthesis if used together with the modifictions discussed in 2.3.2 (see section 5.6) including "smearing" of the sound. This can be attributed to the mathematical formulation of these modifications. By increasing K in equations (4) and (5) we decrease the rate with which horizontal and vertical activaitons are suppressed respectively, and expose the activation matrix and these potentially "noisy" activations (i.e. activaitons that contribute to the approximation error) for longer. Additionally, each time we perform the convolution in (6) combined with the previous two modifications, we change the lengths of the diagonals without any control and can activate templates for longer durations than intended.

To find the right number of iterations for different needs e.g. preserving source characteristics or accurate reconstruction of a sound, our NMF implementation monitors the cost at each iteration of the update rules. When minimizing the Euclidean distance we calculate the cost as the Frobenius Norm of the difference between the reconstruction and the target spectrogram:

$$\textit{cost} = ||\mathbf{S}_T - \mathbf{S}_C\mathbf{H}|| \tag{8}$$

We calculate the cost of the Kullback-Leibler Divergence [9] based update rule as follows:

$$\textit{cost} = \sum_i \sum_j ([\mathbf{S}_T]_{ij} \log \frac{[\mathbf{S}_T]_{ij}}{[\mathbf{S}_C\mathbf{H}]_{ij}} - [\mathbf{S}_T]_{ij} + [\mathbf{S}_C\mathbf{H}]_{ij}) \tag{9}$$

This allows for additional control over the NMF modifications and provide the ability to either apply the modifications or stop the NMF synthesis upon reaching a local minimum.

4.1.2 Alternative NMF Implementation

Although well suited for preservation of timbre, these constraints have been found to be computationally intensive (see section 5), especially when applied to the Divergence minimizing update rule (2). One reason is that at each update iteration all values in the activation matrix are modified by (4) and (5) after which an additional convolution is applied to $\mathbf{H}$. This scales poorly for long source sounds and high number of iterations. From Driedger et al. [10], minimizing cost is not one of the goals of the constraints they proposed and is possibly never the case when the activation restrictions are applied since the activations are forcefully changed at each iteration.

As part of the NiMFKS we propose a slight change to the algorithm where learning the activations is of stronger focus when synthesizing and the feature preserving restrictions are only applied once the $\mathbf{H}$ is computed as accuartely as possible such that the synthesis reflects the target more accurately. Thus substantial speedup in synthesis speed is achieved and is particularly useful when synthesizing with a large corpus. Another reason to impose the restrictions is to ultimately give the source more weight in the synthesis. However, using our proposed "Template Synthesis" (4.2) the synthesis will already heavily incorporate the source signal. This allows the "Fast Restricted NNMF" algorithms to provide similar feature preservation properties as inteneded while performing faster for large sets of data and contain less noisy activations (5).

4.2 Simpler synthesis

4.2.1 Technical Overview

If we notice that the mth row of the "activation matrix" $\mathbf{H}$ specifies the presence of the mth column of the "template matrix" $\mathbf{S}_C$ in the time-domain synthesis of $\mathbf{S}_T$, then we can forgo the need to perform the inverse STFT of the $\mathbf{S}_C\mathbf{H}$ (and avoid the restrictions on window shape and overlap for invertibility) by just simply concatenating the relevant portions of the corpus waveform scaled by the elements of $\mathbf{H}$. Symbolically, this is equivalent to $\mathbf{CH}$ where the mth column of $\mathbf{C}$ is the windowed time-domain corpus waveform used to compute the discrete Fourier transform in the mth column of $\mathbf{S}_C$.

4.2.2 Implementation

Algorithm 1 Template Addition Synthesis

```
1: parameters: H, source signal, window size, overlap
2:
3: init: $N \leftarrow$ number of rows of $[\mathbf{H}]$,
4: $M \leftarrow$ number of columns of $[\mathbf{H}]$,
5: $X \leftarrow$ source signal vector, $j \leftarrow 1$, $k \leftarrow 1$,
6: $C, Y \leftarrow$ zero vector of output length
7:
8: for $j = 1, M$ do
9:     for $i \leftarrow 1, N$ do
10:        $C(j,:) \leftarrow C(j,:) + X(i : i+winsize) \times [\mathbf{H}]_{ij}$
11:    end for
12: end for
13:
14: while $j \leq M - overlap$ do
15:     $Y \leftarrow Y(j : j + winsize) + C(k,:)$
16:     $j \leftarrow j + overlap$
17:     $k \leftarrow k + 1$
18: end while
```

The pseudocode of our basic synthesis algorithm 1 shows that the only pieces needed to perform "Template Addition Synthesis" are the time-domain vector of the source sound, the activation matrix $\mathbf{H}$ and optionally the window size and overlap. Extensions to the algorithm include the windowing of each new addition of a scaled template to $\mathbf{C}$ and optimizations such as sparse matrix multiplication. Since this synthesis technique eliminates the reliance on the target and source spectrograms and thus any complex valued signal representations and utilizes the activation matrix within its synthesis computation, we are able to explore several advantages over the alternative ISTFT spectrogram inversion techqniue.

The first major advantage and reason for introducing this synthesis technique is the capability of applying it even with other methods of deriving the source and target matrices in the NNMF algorithm. Since the $\mathbf{S}_T$ and $\mathbf{S}_C$ are spectrograms derived using the STFT, inverse STFT algorithms can be used as the natural spectrogram inversion techqniue to obtain the time-domain signal of the synthesis $\mathbf{S}_C\mathbf{H}$. However, when using a broader class of features such as chromagrams as part of the NNMF procedure, this method of synthesizing will not apply since a separate spectrogram inversion algorithm accompanies each new matrix representation of a signal (4.3). Our proposal skips the step of multiplying the source spectrogram and the activations and immediately uses $\mathbf{H}$ to create a time-domain signal without creating a spectrogram that needs inversion in the first place. This universality of the synthesis method in combination with our ability to plug in different types of NMF algorithms into NiMFKS creates a diverse sandbox environment for experimenting with different sound and parameter combinations.

From experiments (see 5), we have been able to identify additional scenarios in which "Template Addition" outperforms ISTFT in terms of synthesis quality and cases in which it occasionally fails to produce a sensible output.

4.3 Broader class of features

Armed with our simpler synthesis method, we can now employ other kinds of non-negative features, e.g., constant-Q and chroma.

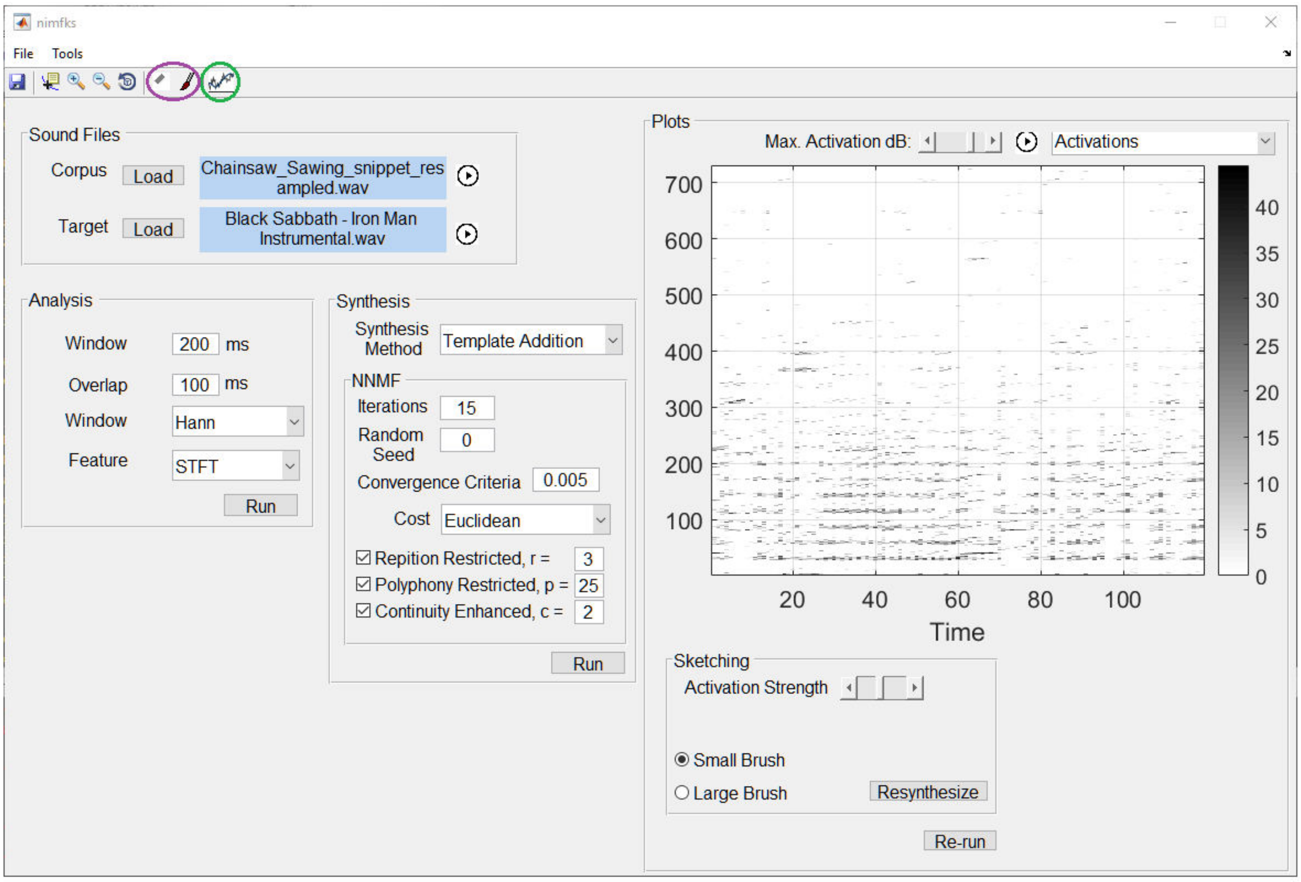
By default NiMFKS computes the spectrograms for the NNMF synthesis using the STFT. Although these are useful in analyzing the spectral content of a signal, it does not provide useful insight into melodic or harmonic characteristics of sound which is particularly important in music analysis. This resulted in our investigation of alternative audio representations that could potentially provide more meaningful information about the sounds to be synthesized, not only for the user but at a computational level that results in different quality syntheses. Currently, NiMFKS supports the use of Constant-Q Spectrograms and Chromagrams as features for anaylsis. However, due to the semantic independent nature of NMF, any type of matrix representation of signals can be chosen.

In the current implementation of NiMFKS, chromagrams are the only alternative option to choose from. In this project chromagrams are used as a visual evaluation criteria for synthesis quality but demonstrations in section 5 show that these can also be used to compute the activation matrix which can either be applied to the spectrogram of the source for synthesis or directly to the source's chromagram when using Template Addition Synthesis.

4.4 Activation sketching

A simple extension of these ideas is to let the user interactively edit the resulting activation matrix, e.g., erasing or adding activations. Based on the same idea of focusing the synthesis on the structure and patterns within the activation matrix as explained in 4.1 NiMFKS enables the user to draw and erase activations to suit the resulting synthesis.

Enabling the tool, the user will have the option to draw activations of different weight onto the visualization of the current activation matrix. This will overwrite the values in the matrix and create new set of activations that can be used to resynthesize the song using the patterns the user drew.



Sound Files
Corpus Load Chainsaw_Sawing_snippet_resampled.wav
Target Load Black Sabbath - Iron Man Instrumental.wav

Analysis
Window 200 ms
Overlap 100 ms
Window Hann
Feature STFT

Synthesis
Synthesis Method Template Addition
NNMF
Iterations 15
Random Seed 0
Convergence Criteria 0.005
Cost Euclidean
☑ Repition Restricted, r = 3
☑ Polyphony Restricted, p = 25
☑ Continuity Enhanced, c = 2

Sketching
Activation Strength
● Small Brush
○ Large Brush

Figure 1. Shows a preview of the NiMFKS GUI. The "Analysis" panel bundles the parameters necessary to perform the sound analysis step in which features are analyzed and visualizations for e.g. spectrograms or chromagrams are constructed. The "Synthesis" panel allows users to define the different variables in the update rules and any other part of the synthesis procedure such as the actual time-domain construction of the target (i.e. "Synthesis Method").

4.5 NiMFKS

While Driedger et al. [10] do not supply a reproducible work package, their work is described in such a manner that it is simple to reproduce their results. We make freely available our application (NiMFKS) exploring all of the above. In addition to the NMF algorithm using the Kullback-Leibler divergence, we implement the one using the Euclidean distance.

The design choice was driven by consideration of the target user. Those who are familiar with the topic of NMF or CSS in general, are able to customize most steps involved in the synthesis process discussed throughout this paper whereas users who wish to experiment with sound synthesis or NMF for the first time can simply load a corpus and target sound and run the synthesis with default parameters already set. Figure 1 shows the main interface of NiMFKS.

First choose the "Corpus" and "Target" sound. Then select a "Cost Metric" (i.e. NMF algorithm) and decide on the "Synthesis Method" (i.e. ISTFT or Template Addition). In the "Analysis" panel you have the option to change the type of matrix to use in the NMF process e.g. chromagrams, STFT or constant-q spectrograms and specify parameters of the STFT used to compute the spectrograms. The "NNMF" parameters that can be customized include the number of iterations (variable L in equation (2)), "Convergence Critera" (percentage change of the cost measure at which we assume the NMF algorithm converged) and which constraints to apply to the update rules. After all parameters are set, hit "Run" and enjoy the output by hitting the play button above the axes. If you want to use the "Activation Sketching" feature from the tools menu, the "Skteching" panel provides the option of choosing brush size, until what dB level activations are displayed and with what strength they are drawn. Upon enabling the Activation Sketching feature, the toggle buttons circled in blue appear. Use the left button to erase activations and the right one to draw. Click the "Resynthesize" button to run the synthesis again after modifying activations without the need to re-run the NMF procedure. Finally, the dropdown menu above the axes allows the visualization of synthesis characteristics like spectrograms, cost at each iteration or the activation matrix. When viewing "Source Templates" enable the "Template Editing" tool from the "Tools" menu and toggle the button circled in green to scroll through (use your mouse wheel) and remove (hit "del" on the keyboard) templates from the source. The "Re-run" button will run the NMF procedure again using the modified corpus matrix (figure 17).

The actual implementation of our application is based on Matlab's GUIDE GUI design tool. Each component visible in the GUI is linked to a "Callback" that has access to the current state of the figure i.e. state of each component including the "handles" that store user data. The Callbacks can then be used to call functions like the NMF procedures. Figure 11 shows the program flow after hitting the "Run" button in the Synthesis panel.

Each component of the GUI that the user can interact with is associated with an "action". The "Synthesis Controller" determines what the application needs to perform based on the action by the user. In the example flow chart the user hits button press indicates to the application and thus the controller that a synthesis needs to be performed. Once an action is routed, the controller reads in the data within each component of the GUI through the "handles" data structure and uses them as parameters for the "Synthesis" method which is part of the "Synthesis Object" that holds the actual methods and information about the current synthesis session. The parts indicated in blue occur in the controller while the red tile is part of the synthesis object. Once the task is run and the object updated, the new synthesis object is stored in the GUI handle such that it can be saved or modified at a later stage. This again shows the modular design of the application and highlights that by simply altering the colored within the flow chart and adding new actions to the GUI and controller, new functionality can easily be implemented into NiMFKS.

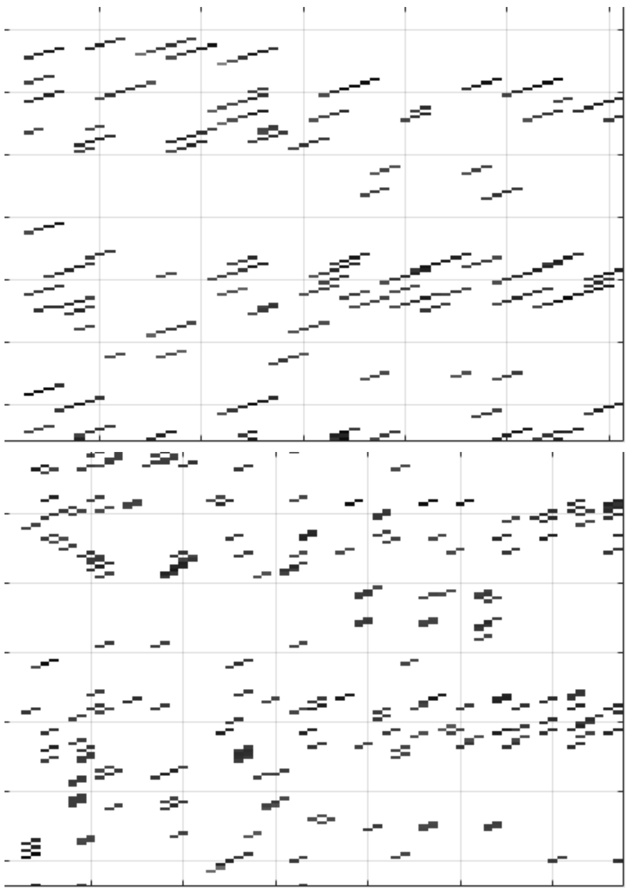


Figure 3. Activation matrices obtained by using a) "Euclidean" b) "Fast Euclidean" algorithms

.

5. DEMONSTRATION

We now demo our application.

5.1 Model Synthesis

Before previewing the features NiMFKS has to offer, this section lays out a model run of the application and will be used as the basis for some of the following feature demonstrations in this section. The corpus selected is a set of consecituve sounds of "Wind" blowing (taken from [10] investigations) concatenated together whose frequency range increases with the duration of the sound file. The target is a female opera singer. To prevent any issues during replay of the synthesis we make sure the sampling frequencies match. Figure 2a shows the spectrogram of the target (top) and synthesis (bottom). The activation matrix is shown in figure 2b. The synthesis parameters are the "Euclidean" NMF update rule with modification variables r, p, c (see figure 1) set to 3, 25, 3 respectively. We let the synthesis process run for 12 iterations without a convergence criteria after analyzing its STFT with a window size of 200ms and 50% overlap. Finally, the output is constructed using "Template Addition Synthesis".

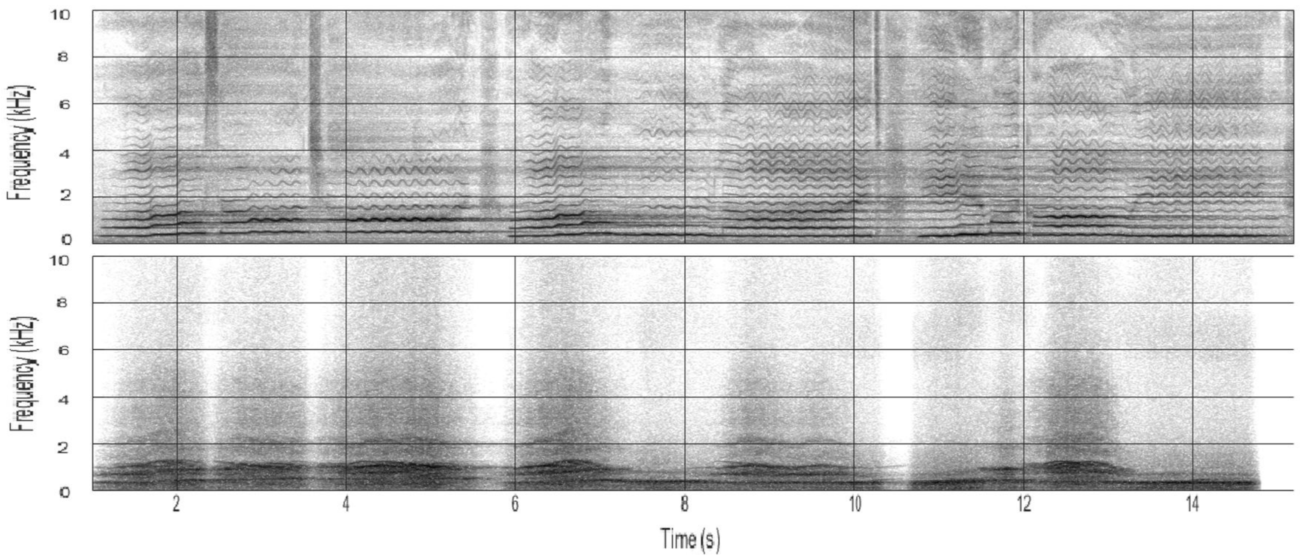
5.2 Playing with Restrictions

A set of four different NMF algorithms are available in NiMFKS; one algorithm that minimizes Kullback-Leibler divergence and another that minimizes Euclidean distance (see **??**) and an alternative implementation of each called "Fast Euclidean" and "Fast Divergence", where "fast" refers to the point at which restrictions are applied to the update rules. In the "fast" algorithms, the restrictions described in **??** only take effect in the last iteration. We now demonstrate the difference between the constraint algorithms. The corpus used is a "Chainsaw Sawing" [10] and the target to approximate is "Iron Man" by Black Sabbath. Figure 3 constrasts the activation matrices learned from applying the "Euclidean" NMF algorithm with variables of constraints (4), (5) and (6) set to $r = 3$, $p = 15$ and $c = 3$ respectively in the "fast" and original implementations.

Seeing that other features of the constraints can be varied, we additionally modify constraint (6) such that the activation matrix develops structures other than diagonals during the NMF process, changing the identity matrix $\mathbf{I}_c$ to a matrix of different shape while observing the effect in figure 4. It shows outtakes of the activation matrices produced with the synthesis discussed above.

5.3 Template Addition Synthesis

Running the application using the same set of parameters as in the model synthesis but different synthesis methods we obtained the two plots seen in figure 8a and 8b. The figures show the spectrogram and corresponding time-domain synthesis using the ISTFT and Template Addition respec-



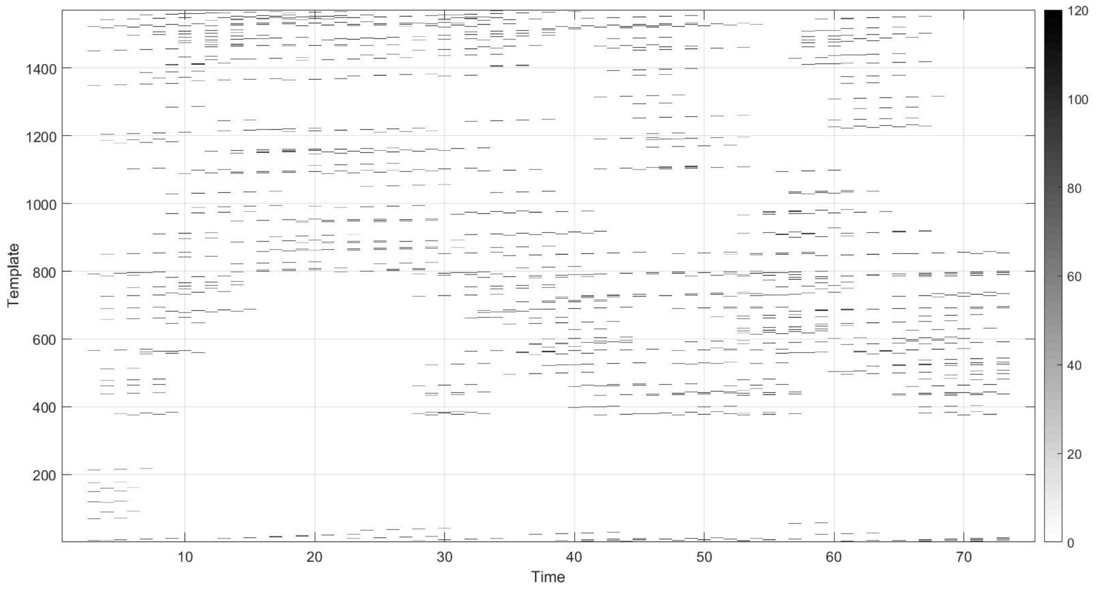


Figure 2. The top figure shows the target and synthesis spectrogram from the model synthesis demonstration respectively, while the bottom figure shows the associated activation matrix

tively.

A second synthesis (Algorithm: Fast Euclidean, Restrictions: 3, 0, 3, Source: "Chainsaw Sawing", Target: "Iron Man by Black Sabbath", Iterations: 15) demonstration (see figure 9) shows the difference between the synthesis method with a different set of sound combination "types". Difference in "type" refers to not only different musical genres but also differences in features such as harmonic range and musical structure (see section 6 for a discussion on the significance of sound and target choice for the synthesis proces).

5.4 Other Features

By simply replacing $\mathbf{S}_C$ with a chromagram of the same sound we are able to apply both the NMF algorithm and Template Addition Synthesis. Our demonstration takes a different approach and instead computes the activation matrix from the chromagrams, i.e. $\mathbf{S}_C$ and $\mathbf{S}_T$ are substituted with the respective chromagrams. We then perform the time-domain synthesis using the STFT spectrograms with the activations obtained previously. The result is shown in figures 12, 13, 14.

5.5 Sketching Activations

We have seen previously that we can modify choose to enforce different types of patterns in the activation matrix by modifying the identity matrix in the continuity enhancement step in [10]. These patterns are constant throught $\mathbf{H}$ and using the algorithms provided there is no simple way of enforcing different types of patterns within the same ac-

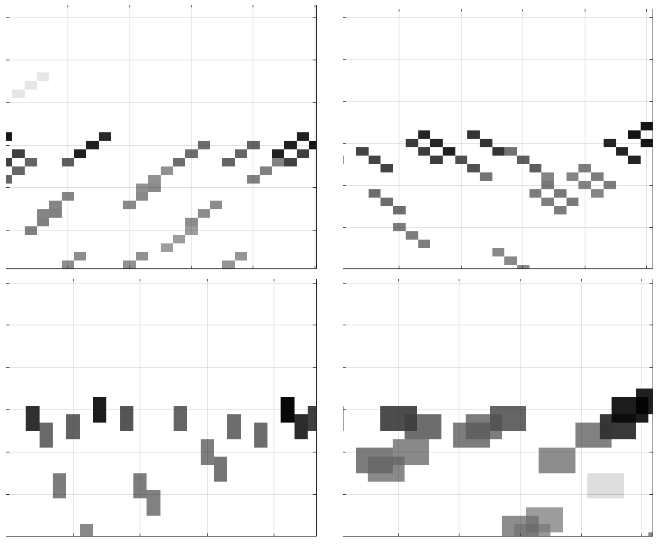


Figure 4. Activation matrices containing patterns such as a) Diagonals b) Reverse Diagonals c) Vertical Activations d) Blurred Activation Patches
.

tivation matrix.

Our activation sketching feature allows the user to overcome this limitation. An activation matrix like 5a can be turned into 5b, in which the single matrix contains all of the patterns described in section 3.1.1. Moreover, the user can decide on the strength and size of the activations and delete them.

5.6 Additional Experiments

To further our understanding of NMF for sound synthesis and the strengths and limitations of our application, we used NiMFKS in an additional set of trials.

The following sections feature three of the experiments conducted using our application: 1) Testing the effect of synthesis parameters 2) Comparison of the different NMF algorithms discussed in this paper 3) Attempt at finding novel applications for NiMFKS.

5.6.1 Exploring Parameter Choices

The aim of these experiments is to find a set of source and target sound combinations that yields a perceptually appropriate synthesis result when used in NiMFKS. This aids the analysis of the effect of "NNMF" and "Analysis" parameters on synthesis quality and allows possible identification of patterns or variables that contribute more towards the synthesis output than others.

Due to the large number of possible parameter combinations that can be used to synthesize a set of sound using NiMFKS, the focus of our experiments falls onto the NNMF algorithm used (including different types of restrictions), choice of source/ target sounds and the synthesis technique.

The choice of source sounds was driven from an observation made when investigating the experiments and results in [1]. All source seemed to have a pattern of in-

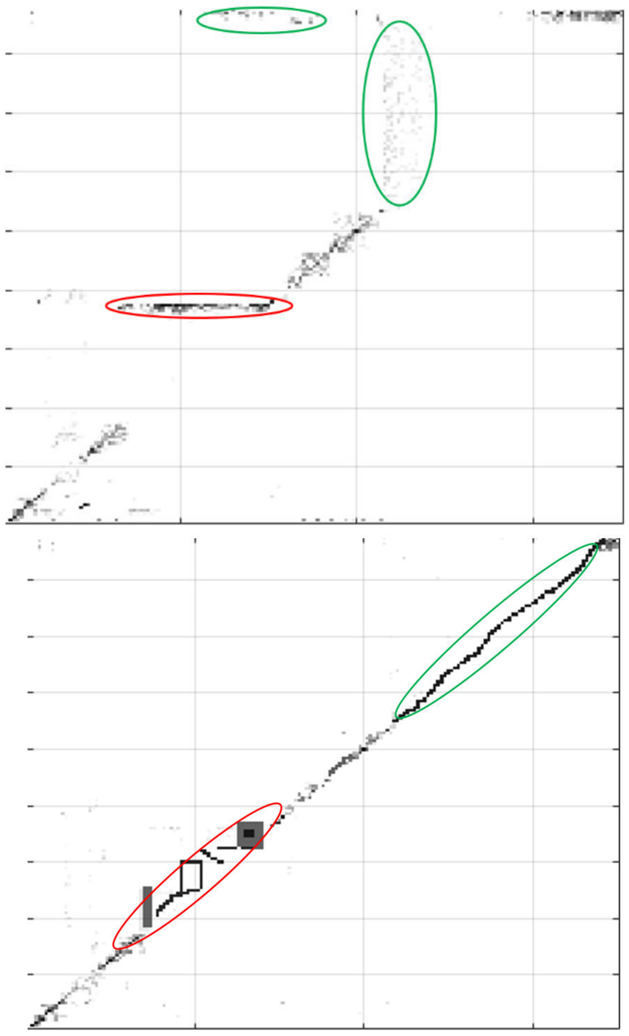


Figure 5. Activation matrix for a sample synthesis in which "Glockenspiel" is the target and source a) before b) after "Activation Sketching"
.

creasing frequency range. A sound segment was taken and repeated with a wider range of frequencies (see 10). This is called "Pitch Shifting" and can be achieved manually using a "Phase Vocoder", a tool used to scale a signal while leaving its length undisturbed [3]. In our investigations, the already provided source sounds from [10] were chosen as sound samples for comparison. This shows similar corpus requirements to applications like "MATConcat", where a preferably wide range of possible templates is made available for the selection algorithm to choose from.

To contrast these sound samples, source sounds without modifications were taken which are naturally rich with large frequency ranges like "Glockenspiel". Remaining source sounds include single segment or a sequence of sounds like "Dogs Barking", "Dog Bark", monkey noises. Since speech synthesis is a major application of sound synthesis but was not described by [10], speech and singing was included in the set of source sounds.

Target sounds mainly included songs from different genres since this was the easiest choice of targets for our perception based evaluation criteria and was the only investigation point described in [1]. Additionally, speech was also used as targets with potential desired effects to be achieved like the creation of a voice beatboxing or singing acapella.

Each of the source sounds was used to synthesize the target sounds and the resulting synthesis was evaluated by listening to it. The synthesis was first run using the default NiMFKS settings (Window Size: 100, Overlap: 50, NNMF Algorithm: Euclidean, Restrictions: None, Synthesis Method: ISTFT, Iterations: 20). If the result was sensible, i.e. a synthesis of the two sounds provided is observable, the parameters are changed in an attempt to improve sound quality.

The optimal synthesis result had to contain the melody, rhythm and temporal characteristics of the target with features like pitch, timbre and small snippets of the source clearly noticeable. The second factor that contributed to the selection of appropriate sound combinations was amount of noise and ringing. This was found to be mainly affected by the synthesis method and choice of window size.

Table 1 shows the final set of sound combinations that not only yielded a "successful" result but also provided more insight of how parameter choice can effect synthesis quality.

5.6.2 Algorithm Comparison

To specifically test the impact NMF algorithm choice has on the synthesis, we run two of the trials from table 1 that we know yield a reasonable result, and simply change the update rule used, to each of the ones available in NiMFKS (i.e. two "Divergence" and "Euclidean"). We then compare the synthesis spectrograms for each to the target spectrogram looking for preservation of timbre, noise, reconstruction accuracy and potentially additional patterns. The structures present within the activation matrix and actual replay of the output are also taken into consideration. The two combinations selected are in rows 1 and 3 from table 1. See figure **??** for the experiment results.

5.6.3 Taking NiMFKS Further

As a final investigation we explore the possible applications of NiMFKS and NMF in CSS further and find any limitations it sets on the synthesis, particularly compared to other CSS applications. We look into the following applications: 1) Steganography using Activation Sketching 2) Sound detection using NMF 3) Creation of special effects applied to voice signals 4) Resynthesizing and contrasting the combinations in rows 1 and 3 from table 1 in the application "MatConcat".

Hiding information within other media like text or images is referred to as "Steganography" [15]. We demonstrate that the activation sketching feature enables a user to encode (i.e. sketch) a decodable message into an activation matrix. Letting $\mathbf{S}_C$ be the spectrogram of the sound synthesized using the modified activation matrix the message can be decoded by applying the NMF algorithm again and view the resulting activation matrix. Figure 6a shows an arbitrary activation matrix in which the phrase "Hello World" was encoded. Decoding the message results in the

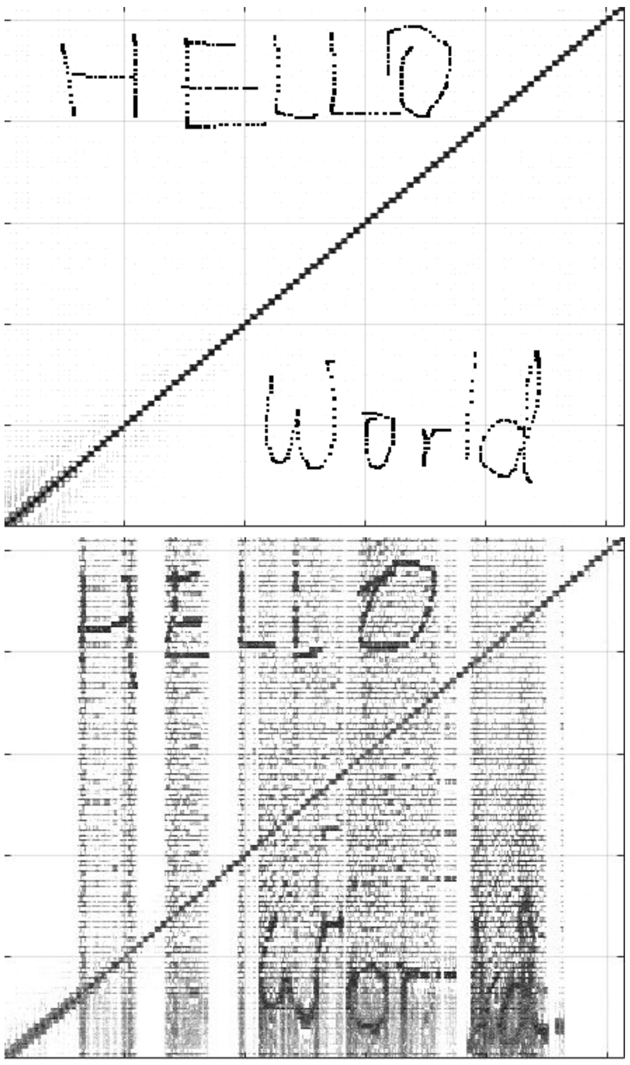


Figure 6. Activations a) after encoding the message b) after decoding the message

.

activation matrix in 6b we can see the two hidden words with high accuracy.

The second test run shows that by observing "strong diagonal structures" within an activation matrix, we can detect i.e. verify the existence of a sound within another sound. We took five seconds of female speech and combined it with a five second segment of the song "Let It Be" by "The Beatles". Taking this combination as the $\mathbf{S}_T$ and the speech as the $\mathbf{S}_C$ in a run of the NMF synthesis, we obtain the activation matrix shown in figure 18. The diagonal circled in green indicates that the corpus sound has been "detected" within the target at that position.

6. DISCUSSION & CONCLUSION

Before commenting on the results from NiMFKS' trials and demonstrations in section 5 we have to describe the evaluation criteria with which we assess output characteritics such as "synthesis quality" or "perceptual superiority". Quality is judged based on how well the actual synthesis compares to the expected. This includes visual measures such as how the spectrogram and chromagram structure of the target and to what extent source timbre is presrved but also oral evaluation in which NiMFK's users. By listening to the output we compare the amount of source sound perceived and preservation of pitch, melody and rhythm of the target.

This section discusses the findings of the demonsntrations and experiments from the previous section before commenting on the applications' current state and proposing additional extensions to NiMFKS.

6.1 Demonstration Discussion

6.1.1 Model Demo

The "Model Synthesis" from previous section provides an insight into several topics of discussion in the remainder of this section. First notice the two spectrograms in figure 2a and b follow the same overall structure. The target spectrogram has distinct visible frequencies between 0-4kHz and weaker frequency patterns going up to 8kHz. The synthesis spectrogram has a similar frequency pattern, with frequencies at the lower end of the spectrum going up until around 6kHz. However, these frequency patterns are less distinct, i.e. the specific frequencies contained in the sound are more difficult to pick out spearately, and seem to have a structure similar to that of noise. This demonstrates how our synthesis process operates and the potential disadvantages associated with it. The application correctly took segments from the corpus sound used them to mimc the target's overall spectrogram structure. However, since the "Wind" sound does not contain the frequency patterns i.e. rapid variations in frequency and pitch (appear as sinusoidal patterns in the spectrogram and chromagram respectively), the synthesis does not contain them either. Rather it carries patterns found in the source, in this case low variations in frequency and poor reconstruction at higher frequencies. The synthesis spectrogram shows accurate reconstruction for constant frequencies in the target but contains no signs of the other characteristics mentioned which are commonly found in voice. This could be viewed as a failure of the NMF algorithm for CSS but can potentially be corrected by choosing more granular corpus and target analysis i.e. smaller window size, hop and overlap. An additional implication is that this synthesis technique is not adequate for voice imitation applications or speech synthesis. Moving from visual to oral analysis, we realize that both the target and corpus are clearly notiecable. The blowing wind sound in particular is underlying the whole synthesis sound implying that timbre was preserved to a great extent, potentially due to the NMF modifications applied. Although not completly understandable, we even observe the short duration that the singer appears to speak providing the an effect such as "whispers/ voice in the air". From here we see the potential for additional experiments (section 5.6) to see and showcase the effect of changing NiMFKS's parameters on the synthesis and advantages customizability can actually provide with this synthesis techqniue.

6.1.2 Restriction Modifications Demo

The most striking difference between the two matrices in 3 is the diagonals of different lenghts. While in the faster variation of the algorithm they are of the length specified by the user ($c = 2$ in both figures), there is no specifc length the structures in the other activation matrix are formed with and no pattern to how their size is determined. The length of the diagonals affects how many templates (i.e. the frequency range) of the source sound the NMF algorithm selects. Since the length of the kernel in equation 6 determines the amount of the source that is found in the synthesis leading to more perceived timbre, more templates are placed into potentially unintended positions within the target causing effects such as polyphony, when a sound appears to consist of several other sounds, when the diagonals overlap within the synthesis and when listening to effects such as smearing or ringing. However, we also see that applying Driedger et al. [10]'s constraints only at the last iterations produces an activation matrix in which the activations learned are positioned in the same overall locations. Possible implications of this are that without losing much of the activation matrix structure that the original modifications provide, we are able to produce similar results with more control over the algorithm at lower computational costs such as time. Additionally, when using the "Template Addition Synthesis" method where activations have a direct effect on the output (since it shapes the corpus using the activation matrix directly), more control over the length of the diagonals can be of advantage. When listening to the synthesis, higher pitch is noticed in the output from the "fast" NMF algorithm and in the case of the combination of "Sine Sweep" and "Play That Funky Music" the target is reflected more in the synthesis. This is also seen in the syntheses spectro- and chromagrams. Applying the constraints throughout the whole NMF process causes weaker chroma patterns (dimmer in color) on figure 15 although the chromagram seems to be reconstructed more accurately in 16.

6.1.3 Synthesis Method Demo

To show the effect of the synthesis method on the output, we compare the signal plots and spectrograms for each synthesis. From figure 8 we observe several differences. Firstly, the ISTFT synthesis contains more distinct amplitude spikes and a peakier signal envelope (shape of the signal). This imposes a ringing effect when listening to the synthesis. The corresponding spectrogram also shows these rapid fluctuations in amplitude through a pattern of adjacent dark and bright patches along figure. These unsmooth transitions are not visible the plot and spectrogram of the Template Addition sytnehsis. One potential reason is that this techqinue is more robust to the window choice with which the source and targets are analyzed than synthesis by "ISTFT". In terms of the underlying structure i.e. shape of the envelope, positions of peaks within the signal plot, and patterns within the spectrogram we see that both synthesis results almost look identical.

Now from figure 9 we witness similar but also additional differences to the ones discussed previously. The main difference we see is not so much the peakiness of the signal envelope anymore, but also the overall shape and structure of bothe the plot and the spectrogram. Amplitude spikes such as around the seven second mark are suppressed to a greater extent in the ISTFT plot. The smoother transitions between segments is now less observable as the pattern of bright and dark bands in the spectrogram are now reduced. Additionally, the spectrograms show that in this case Template Addition produced an output with less noise. This is indicated by the white regions within the high frequency regions of the spectrogram. Orally, the Template Addition synthesis emphasizes the individual chords in the target to a greater extent i.e. are louder and more distinguishable. We can see this behaviour from the spectrograms as well since around the two, five, seven and eight second mark in which the constant frequency patterns picked out by the NMF algorithm are much more visible and extend up to 10kHz. These characteristics are not as clearly seen in the ISTFT spectrogram. Overall, the ISTFT synthesis sounds duller and overlayed with uniform noise. This implies that the Template Addition synthesis is more suitable than the ISTFT when the target or source contain noise and are not as clean and dissectable into harmonics and its individual notes.

6.1.4 Feature Analysis Demo

Figure 12 shows the chromagram of the target (a) and the synthesis (b). Both the target and corpus are set to the same sound and thus should yield the same chromagram once synthesized. The results show that the overall chromagram structure of the resynthesis is the same as the source with a few missing chromas especially around pitch-class four and six indicating that the synthesis causes the output to be less rich in harmonics and sound lower pitched than the target. The patterns circled in red are repeated in the synthesis chromagram due to the resampling used during the synthesis process to match the size of the activation matrix to the corpus spectrogram, causing mismatches in the size of the sound and the corresponding sampling frequency. This is one flaw currently part of NiMFKS when using chromas in the analysis step but can be improved in later versions of the software through careful redesign of the “Template Addition” synthesis such that it preserves sound length. This mismatching in sound length is also seen in the activation matrix (figure 13) obtained when using chromagrams during the NMF procedure. The expected result is a single diagonal going across the whole matrix. However, three diagonals are observed, indicating that the application correctly detects that the corpus and target are the same, but are is not producing a perfect diagonal through the middle due to resampling of the chromagrams. Finally, the spectrograms of the original (12a) and synthesis (12b) show that when using chromas instead of the STFT for synthesis, we add more noise to the result but obtain spectrograms with higher frequency resolution and thus seemingly higher pitch in the resynthesis. This demonstration implies that although not suitable as a replacement for STFT spectrograms yet, the NMF synthesis technique is extensible to use other matrix representations of audio, of which there are many more types to test out including other types of spectrograms e.g. narrow-band or wide-band spectrograms.

6.1.5 Activation Sketching Demo

In this demonstration we provide a case where the NMF algorithm has failed to produce the result we wanted, i.e. a diagonal going across the whole activation matrix since we are setting the corpus and the target to the same sound. One point of interest is the reason for false template selection by the application. In this case, the algorithm has trouble matching source templates to the target in positions where no music is playing (or playing with quietly) and instead selected templates that to the user would appear to just create noise. To correct this anomaly (circled in red in figure 5a) we erase and replace them with a set of different patterns (circled in red in figure 5b). Additionally we erase the anomalies circled in green in (a) and draw the diagonal we desired (circled green in (b)). These are two of the the use cases of Activation Sketching and shows that even if the algorithm does not pick out the templates appropriately this can be accounted for by NiMFKS.

6.2 Experiment Findings

When experimenting with different sets of parameters and sounds of different genres and musical structures, we found that the most signifcant influencers on the synthesis quality were the parameters defining window length and overlap parameters, synthesis method and when to apply the restrictions during synthesis. The remainder of the parameters are specific to the situation i.e. which target or source is used and goal of the synthesis.

The window parameters are cruical during the synthesis process. The overlap decides how many segments are selected for the analysis step and influence the overall success of finding a good approximation of a segment within the target by the corpus units. The window size then determines the granularity of the corpus units and what amount of the source sound will be heard in the output synthesis. Both also effect the synthesis method. While the Template Addition synthesis is not as susceptible to variations in window design, in the case of ISTFT synthesis these parameters are used to invert the reconstruction matrix and for example a large analysis window will lead to the ISTFT regarding the reconstruction as a low frequency resolution spectrogram and provide an output that has undesired characteristics such as noise, ringing effects and poor harmonic range.

Choosing the right balance between window size and overlap effects not only the quality of the synthesis but also computational time. To create a spectrogram, the program needs to move through the signal at intervals equal to the percentage overlap of the window while taking the FFT of each. Thus choosing a small window size or overlap yields higher resolution spectrograms, it drastically increases computational time.

The main difference between the algorithm choices was that the “Divergence” algorithms converged at a lower amount of iterations and offered better pitch preservation compared to the "Euclidean" update rules. Additionally, the "fast" implementations offered more control over the modifications applied and yielded less unexpected results. The modification parameters like (r, p and c in the formulas from section 2.3 can then be viewed as a determinant of what effect to achieve in the output. For example, longer diagonals lead to longer segments from the source to appear within the synthesis. Changing variable p allows to restrict polyphony. However, this phenomenon is somtimes preferred, e.g. in cases where a choir or echo sound is desirable such as the "Wind" and "Opera Singer" combination discussed earlier in this paper i.e. corpus' and targets with a wide harmonic range.

6.3 Final Comments on NiMFKS and Future Work

During its development NiMFKS was subject to constant change. It started as a simple set of scripts for experimenting with Driedger et. al's [10] existing NMF synthesis method. Noticing that alternative update rules such as the Euclidean distance based algorithms exist, we extended these implementations into a standalone application for investigation of NMF synthesis. After a set of possible extensions that arose from simply looking into alternative analysis methods and synthesis algorithms, the basic tool morphed into NiMFKS. During the early phases of its trials, we noticed the lack of customizability of the synthesis procedure and lack of control of the final output, especially compared to existing CSS applications. This drove us to create features such as Activation Sketching, Template Editing and control over when to apply modifications. The paper is meant to demonstrate the exhaustive set of possibilities that this synthesis method carries with it and show how easy unrelated DSP concepts can futher extend the application at hand.

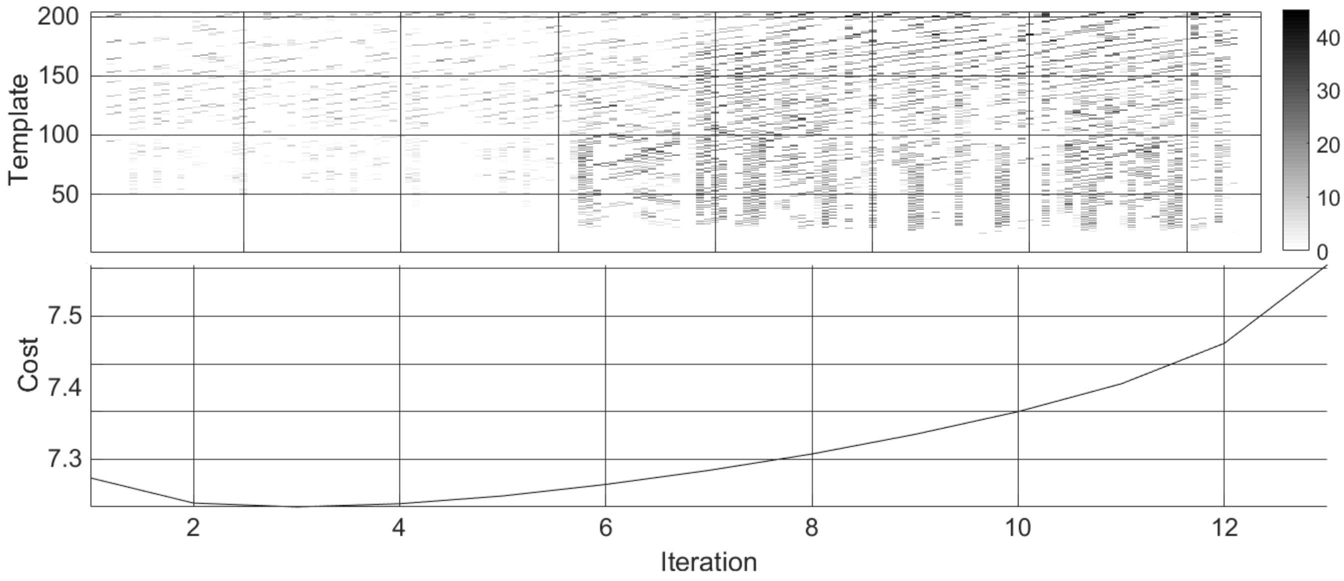
Although the presented investigations of some of the effects that the individual parameters in the synthesis impose, the amount of possible combinations, features to analyze and alternative implementations of the individual algorithms, provides far more room for experimentation. Particularly areas that concern performance of the application can provide even more insight into the effect of parameter choices and provide an edge over other synthesis applications limited by poor computational times with large corpus databases. Areas like complex-valued NMF, parallelization and optimization of the update rules and modifications to or scalable versions of "Template Addition Synthesis" will be investigated in future iterations of NiMFKS. Topics within optimization by cost reduction or certain concepts in machine learning can be eventually applied to the update rules e.g. to automatically detect the optimal break point out of the algorithms based on the cost and more complex measures of reconstruction quality.

NiMFKS successfully allows its users to perform fast and highly customizable synthesis with an easy implementation and room for extensions.

Acknowledgments

Special thanks to my supervisor of this project Dr. Bob Sturm, Queen Mary University of London, for the close support and supervision during the whole year of study.

A. APPENDIX



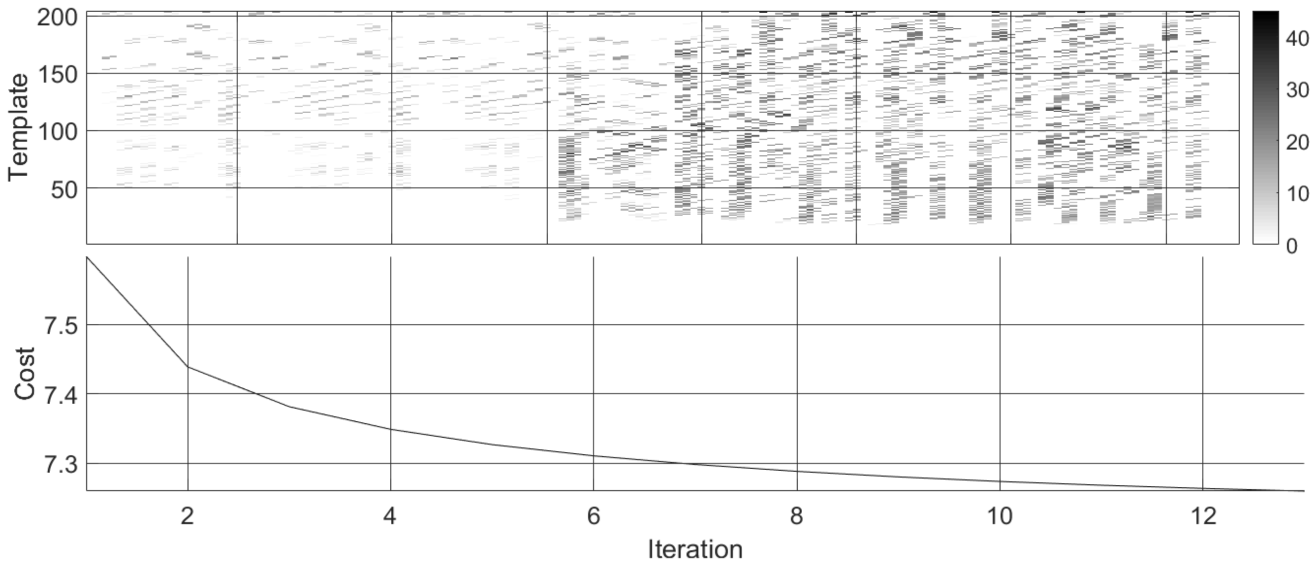
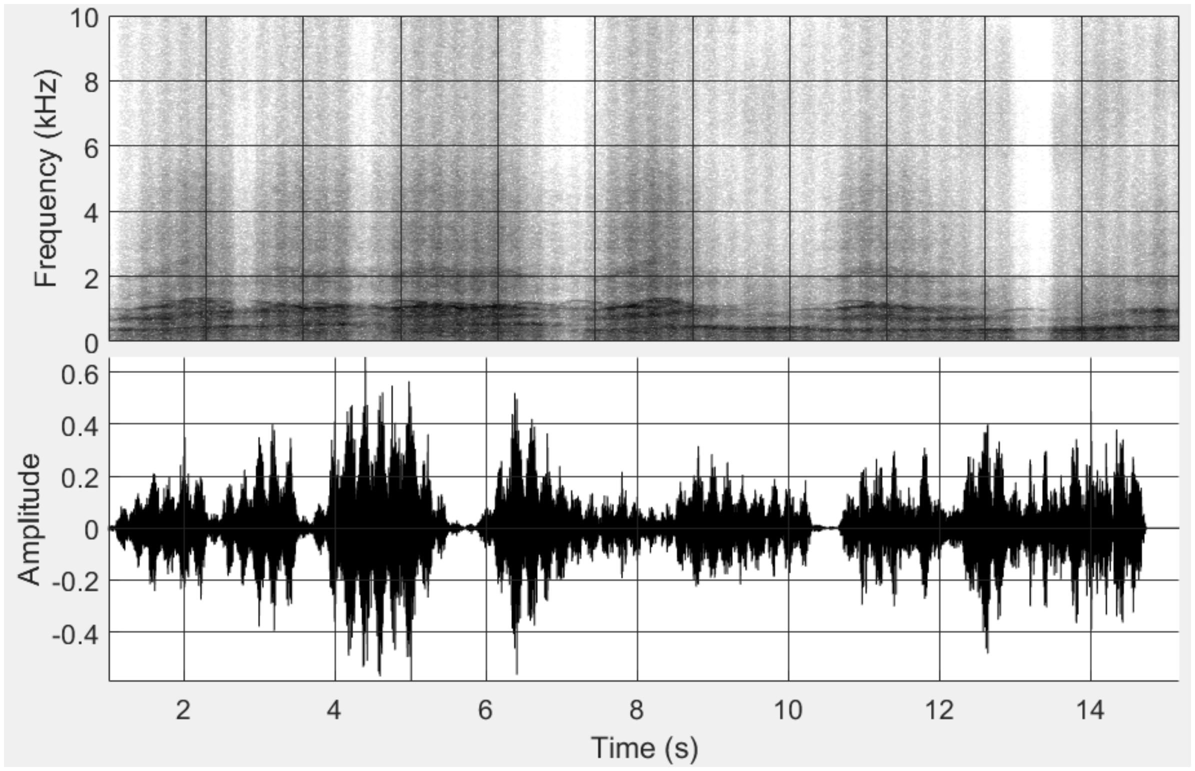


Figure 7. Activation matrices and cost obtained by using a) "Euclidean" b) "Fast Euclidean" algorithms for the synthesis in row one of table 1. The cost represents the update rule associated cost (see section 4.1.1) calculated after each iteration of the NMF algorithm.



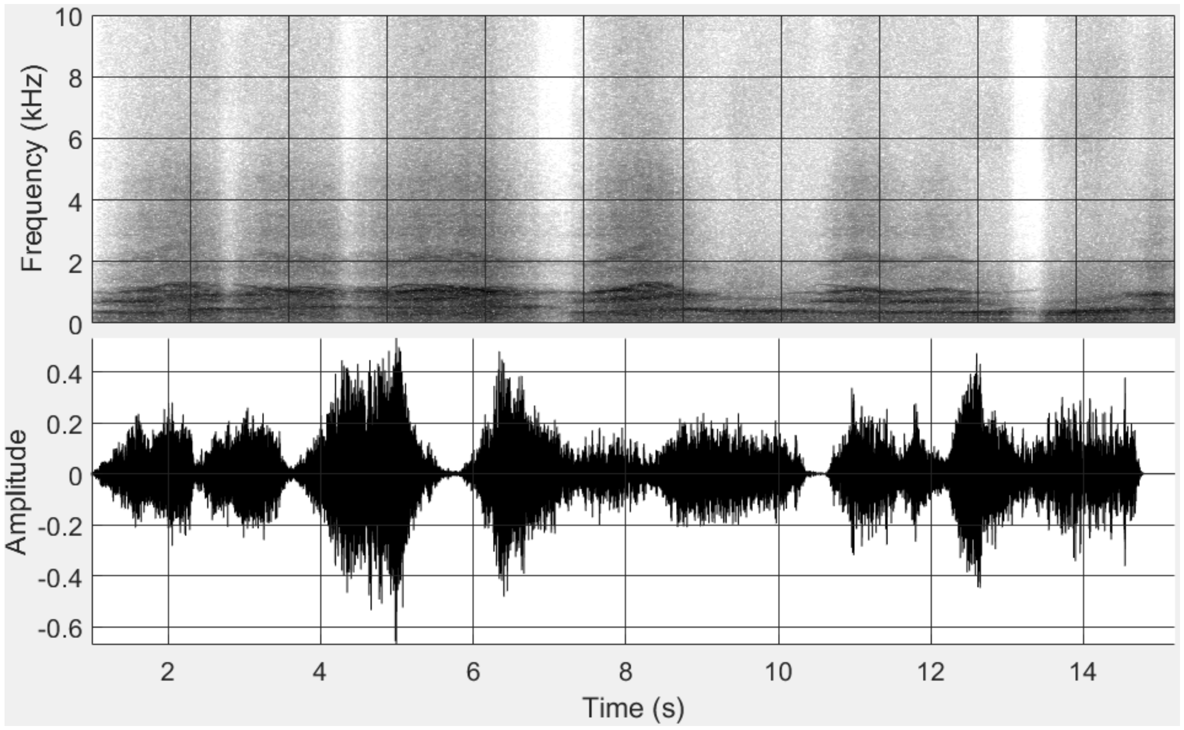


Figure 8. Synthesis plots and spectrograms of the combination one in table 1 using a) Template Addition b) ISTFT

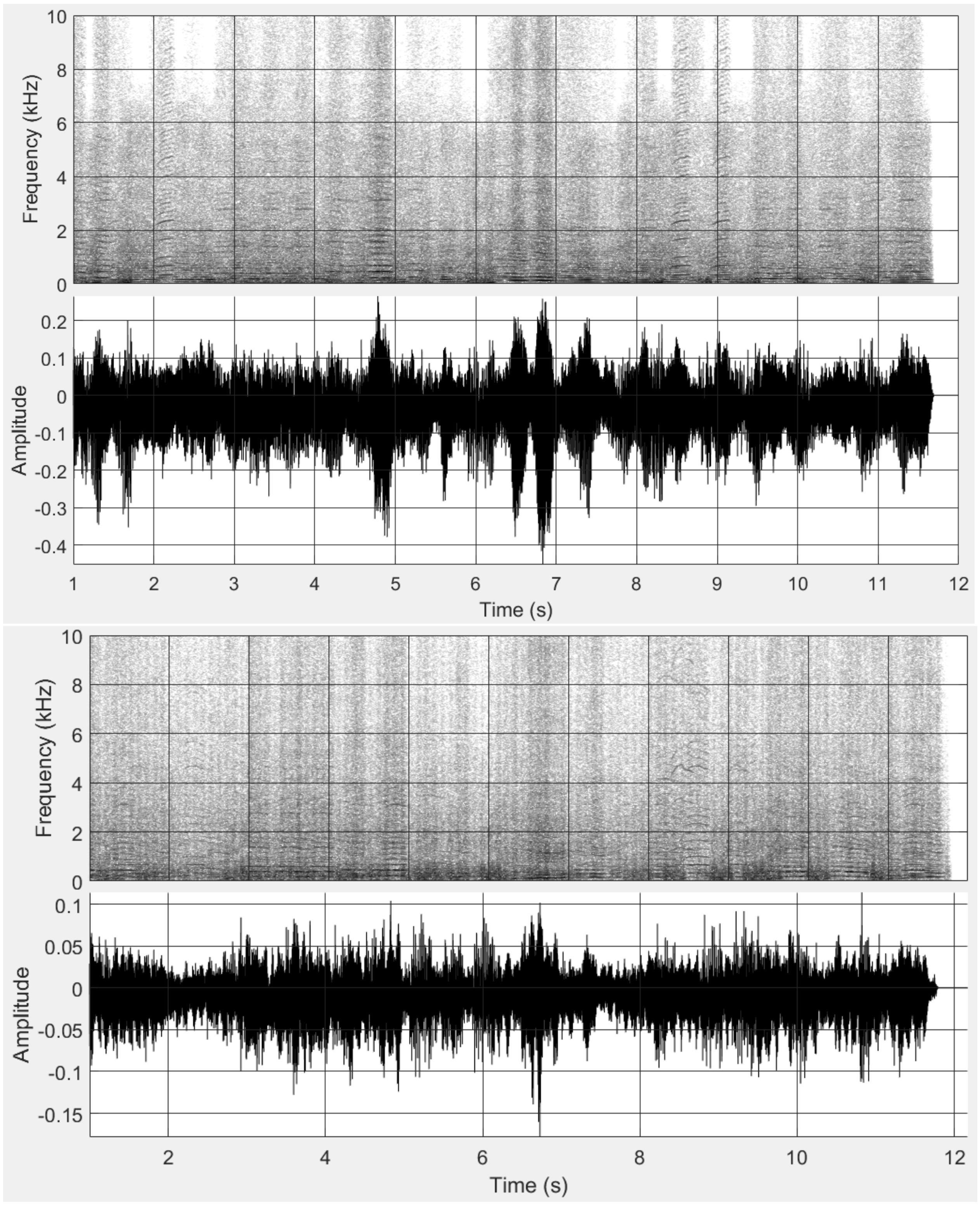


Figure 9. Synthesis plots and spectrograms of the combination of a "Chainsaw" and "Iron Man" by "Black Sabbath" using a) Template Addition b) ISTFT

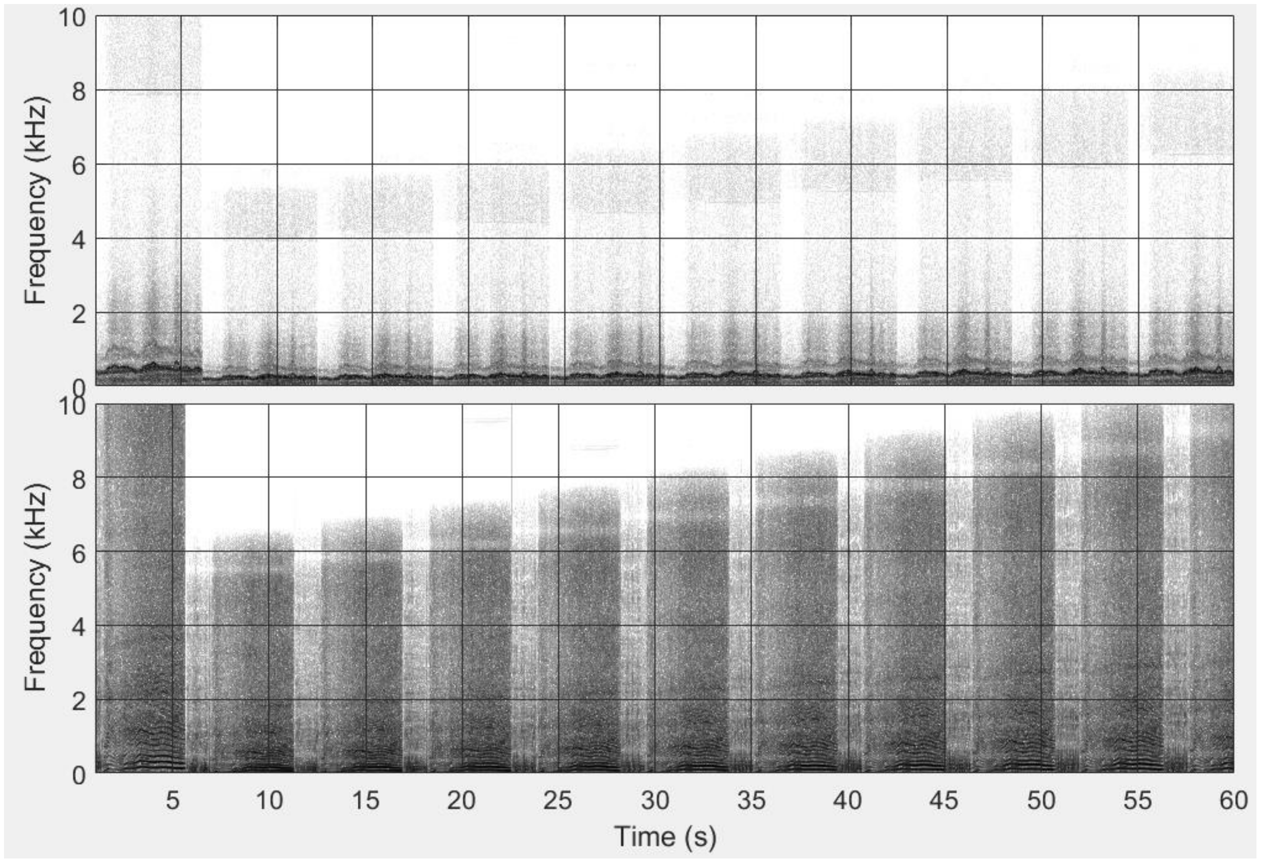


Figure 10. This figure shows the spectrograms of two different source sounds used in [10]'s investigations. "Bees Buzzing" on the top and "Race Car Engine" on the bottom

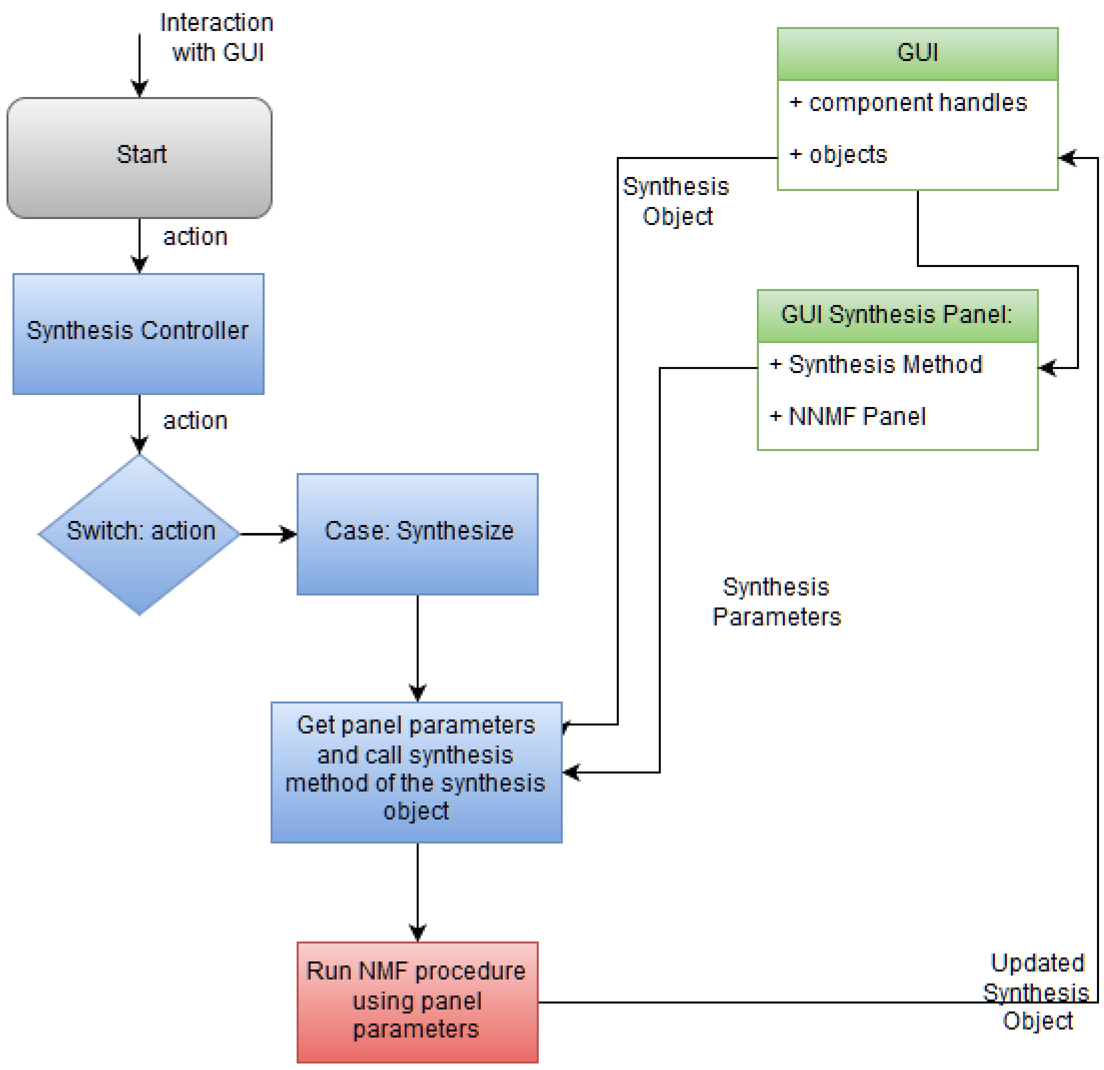


Figure 11. Represents the flow chart of NiMFKS after the “Run” button in the “Synthesis” panel is clicked. It goes through the steps and procedures the application takes in response to a user interaction. Blue tiles are part of the “Synthesis Controller” function, red tiles are part of the “Synthesis Object” and the GUI tiles represent the objects associated with the GUI and variables stored in them

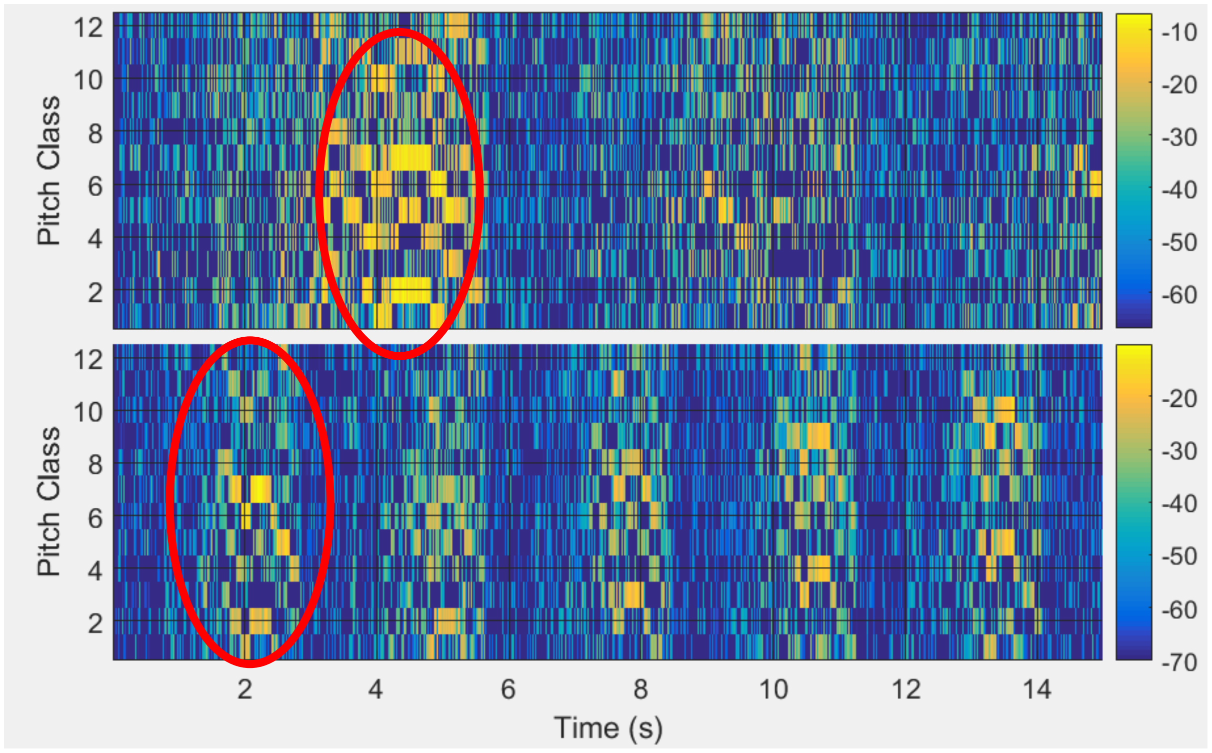


Figure 12. Shows the chromagram of the original "Chainsaw" target (and in this case corpus) on the top and the chromagram of the synthesis on the bottom. The synthesis to create these figures used chromagrams instead of STFT spectrograms during the "Analysis" step

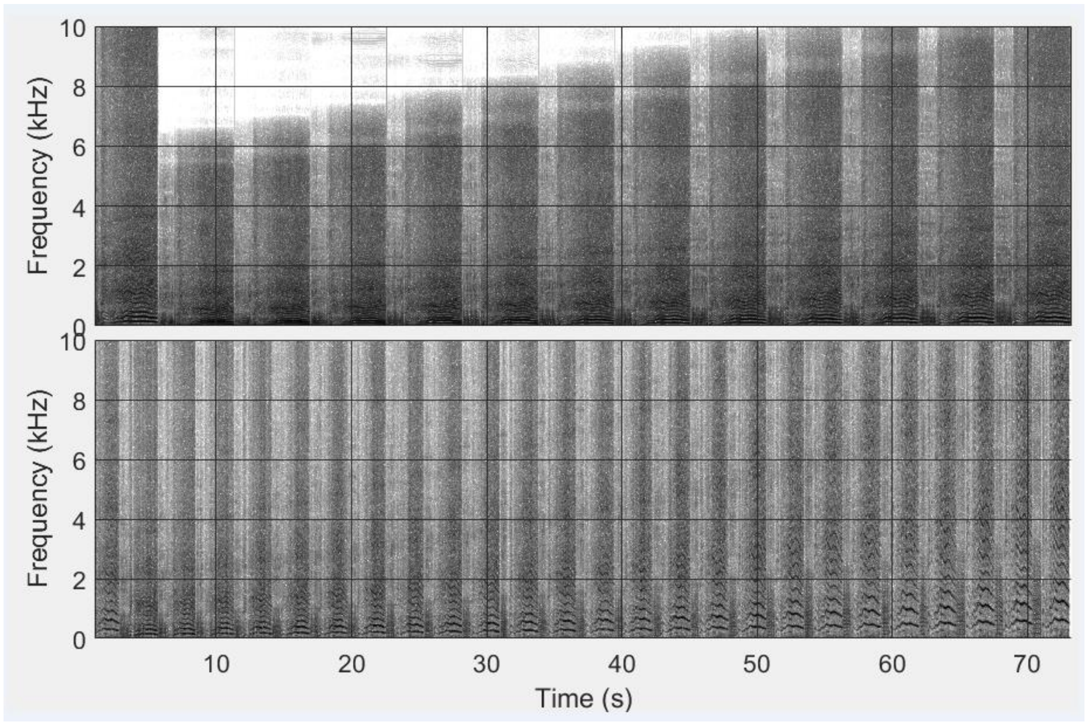


Figure 13. Shows the spectrograms obtained when synthesizing a "Chainsaw" target (a) with a Chainsaw" corpus (b) but using chromagrams as the analysis feature

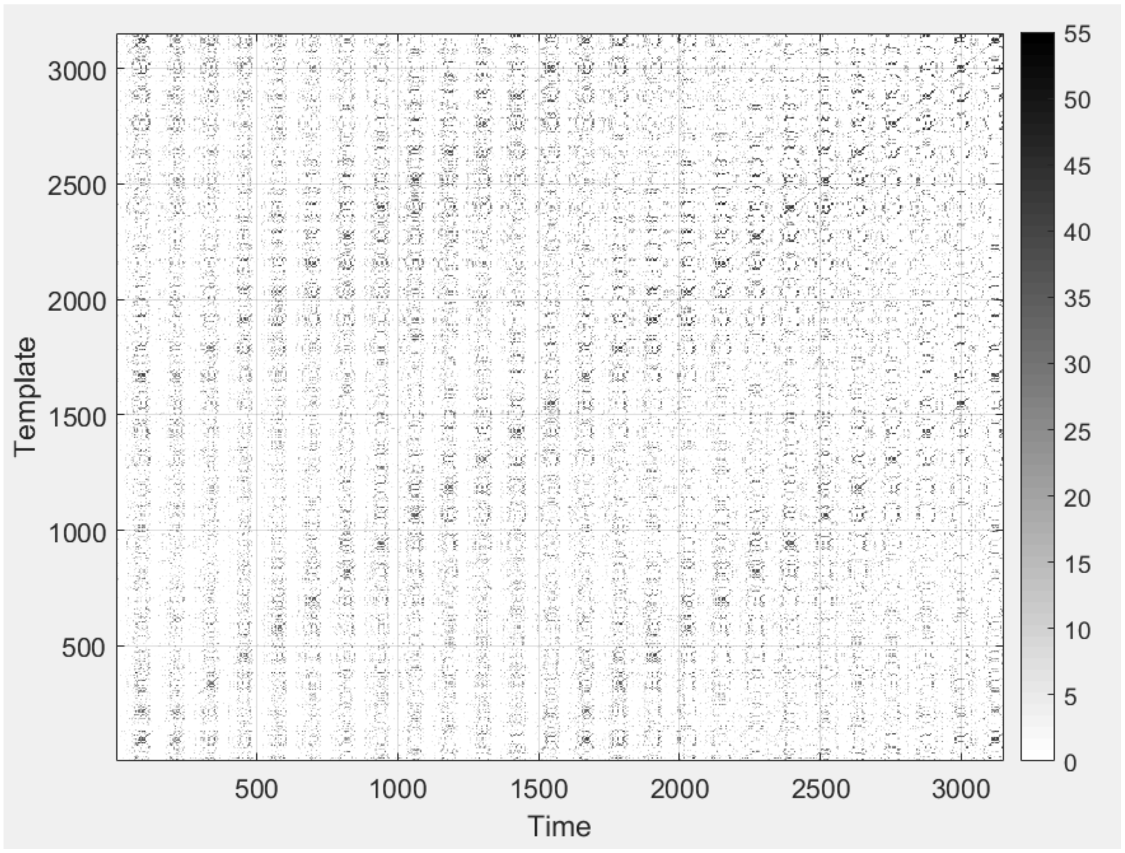


Figure 14. Shows the activation matrix obtained when synthesizing a “Chainsaw” target with a “Chainsaw” corpus but using chromagrams as the analysis feature

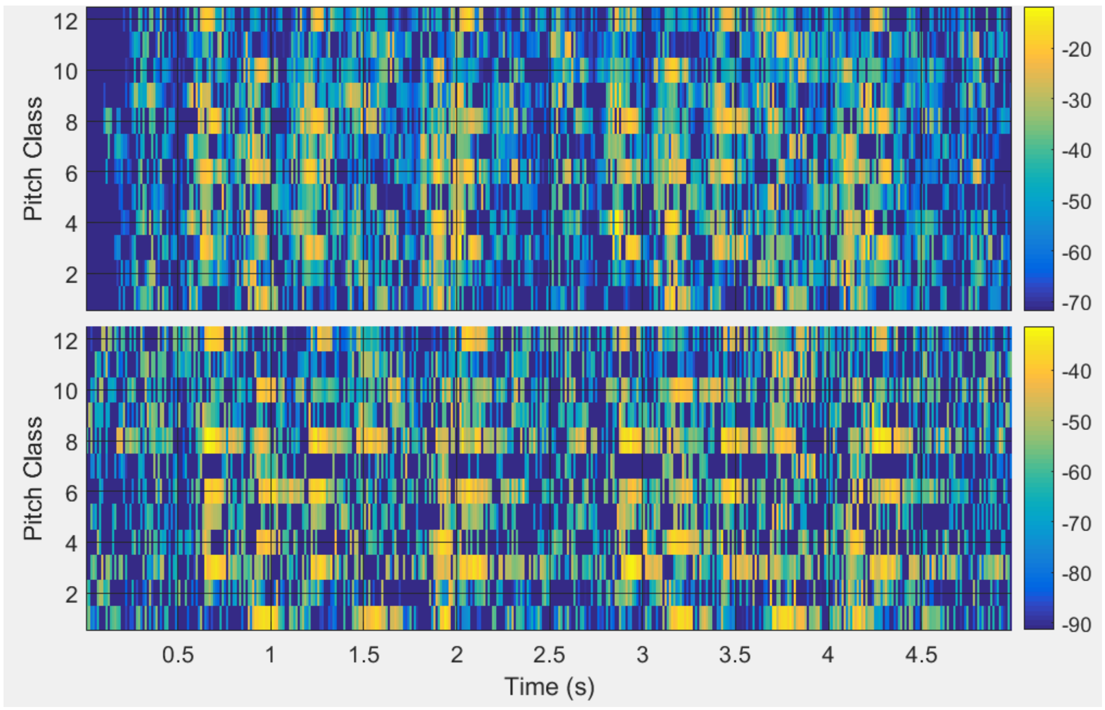


Figure 15. Synthesis chromagram obtained by using the "Euclidean" NMF algorithm during the synthesis (top). Chromagramof the original target ("Play That Funky Music") is shown on the bottom figure. The corpus during this synthesis was a "Sine Sweep"

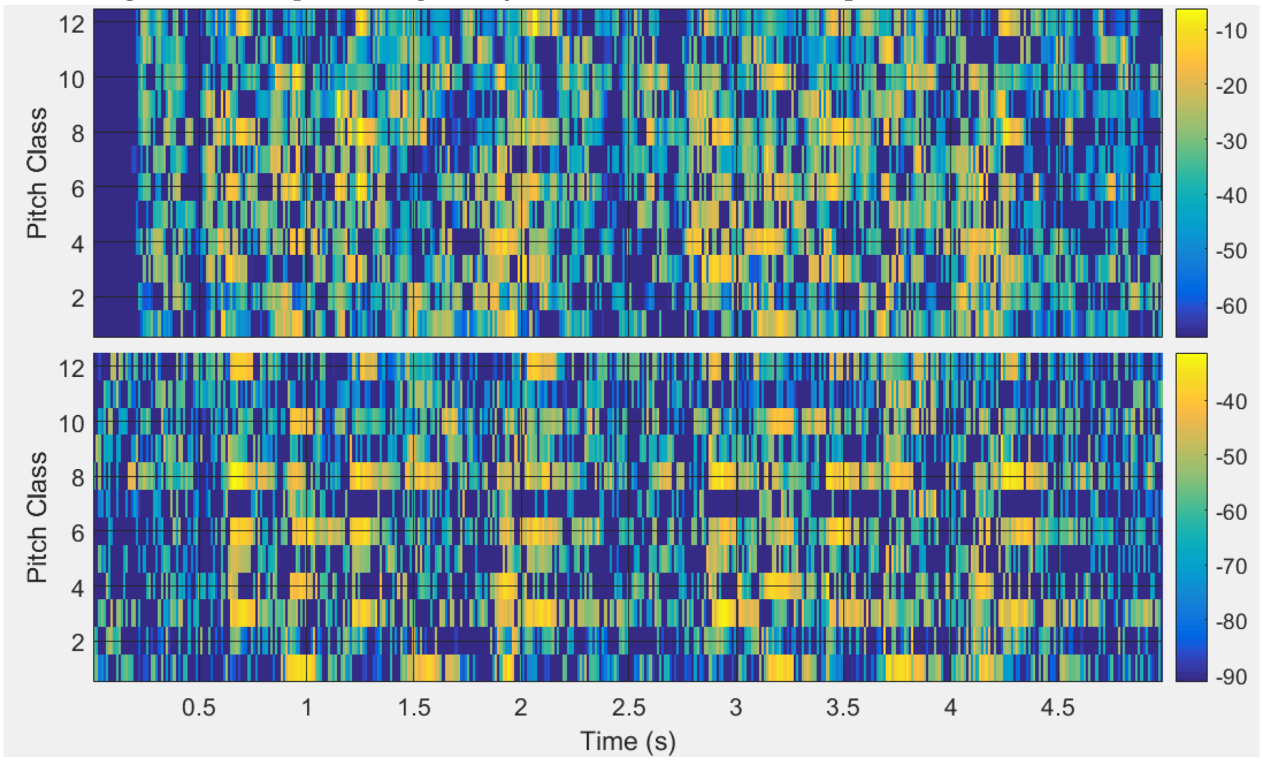
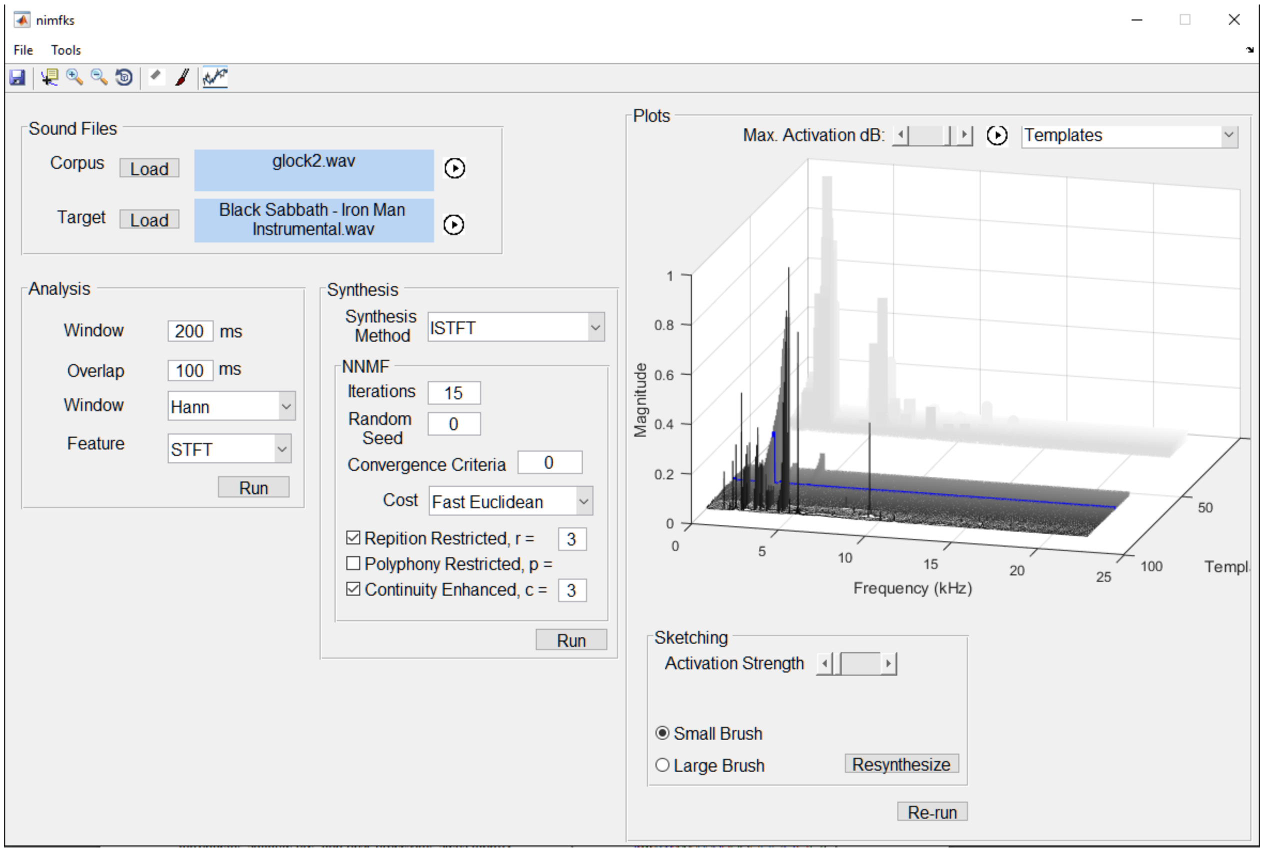


Figure 16. Synthesis chromagram obtained by using the "Fast Euclidean" NMF algorithm during the synthesis (top). Chromagram of the original target ("Play That Funky Music") is shown on the bottom figure. The corpus during this synthesis was a "Sine Sweep"



Sound Files
Corpus: glock2.wav
Target: Black Sabbath - Iron Man Instrumental.wav

Analysis
Window: 200 ms
Overlap: 100 ms
Window: Hann
Feature: STFT

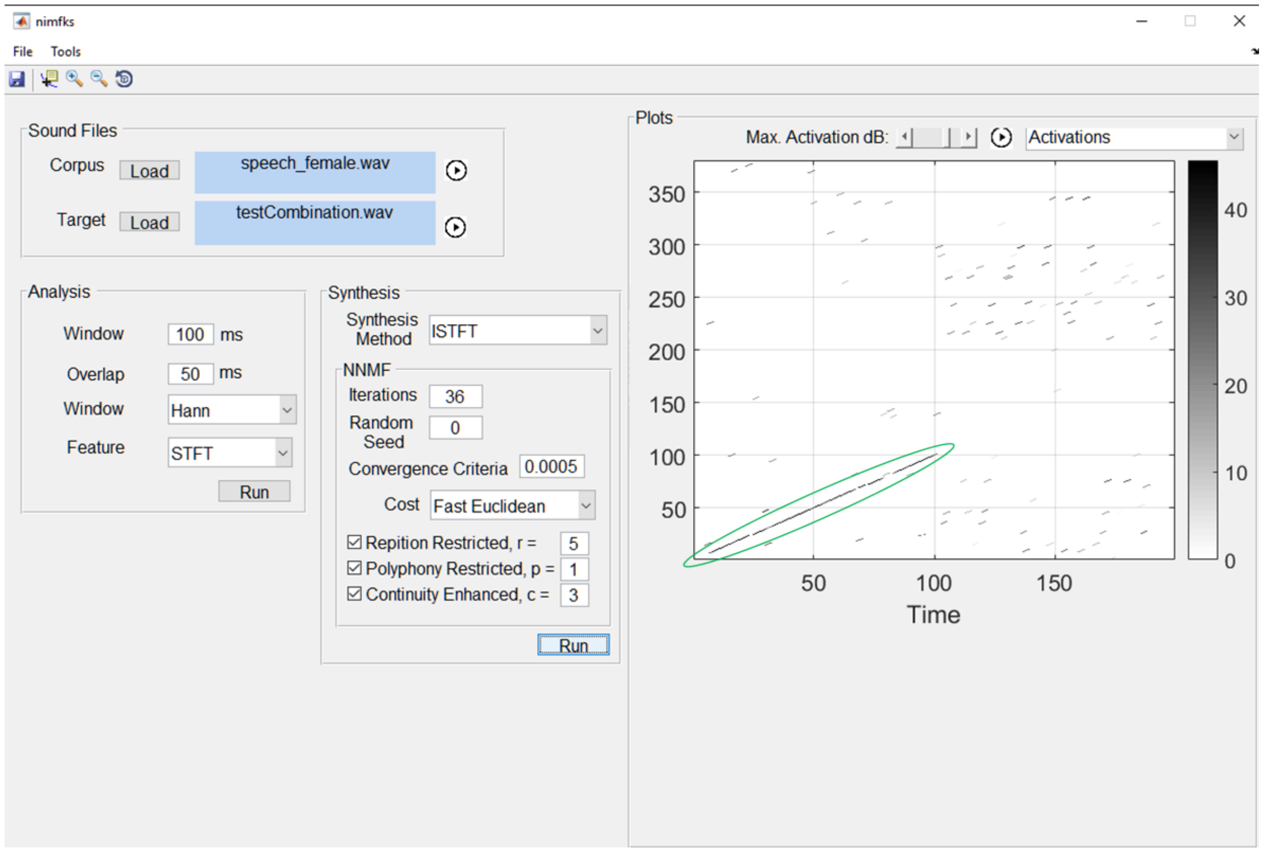
Synthesis
Synthesis Method: ISTFT
NNMF
Iterations: 15
Random Seed: 0
Convergence Criteria: 0
Cost: Fast Euclidean
☑ Repition Restricted, r = 3
☐ Polyphony Restricted, p =
☑ Continuity Enhanced, c = 3

Sketching
Activation Strength
[✓ Small Brush] Large Brush

Figure 17. Showcases the "Template Editing" feature in NiMFKS allowing to view and remove the currently selected (blue spectrum on axis) template in the corpus

Source Sound	Target Sound	Algorithm	Synthesis	Window (ms)
Sine Sweep	Play That Funky Music	Euclidean (3, N/A, 3)	Temp.	100
Race Car	Let It Be	Fast Eucl. (3, N/A, 4)	Temp.	200
Bees	Let It Be	Fast Div. (1, N/A, 2)	Temp.	100
Chainsaw Sawing	Sawtoothbirthday	Div. (3, 20, 3)	ISTFT	100
Sine Sweep	String Quartet	Div. (3, 50, 2)	ISTFT	100

Table 1. Shows the results of the experiments in section 5.6. These are the combinations chosen during experimentation with NiMFKS that perceptually yielded the precptually the best results for discussion and also provide a range of different parameter choices to showcase



Sound Files

Corpus Load speech_female.wav

Target Load testCombination.wav

Analysis

Window 100 ms

Overlap 50 ms

Window Hann

Feature STFT

Run

Synthesis

Synthesis Method ISTFT

NNMF

Iterations 36

Random Seed 0

Convergence Criteria 0.0005

Cost Fast Euclidean

☑ Repition Restricted, r = 5

☑ Polyphony Restricted, p = 1

☑ Continuity Enhanced, c = 3

Run

Figure 18. A demonstration of NiMFK's ability to detect a sound segment that is part of another sound. “testCombination” is the target that consists of a “Female Speech” and “Let It Be” combined. The corpus is set to the sound to detect i.e. “Female Speech”

B. REFERENCES

[1] R. Lyons, “dsp tips & tricks - the sliding DFT,” *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 74–80, mar 2003. [Online]. Available: http://dx.doi.org/10.1109/MSP.2003.1184347

[2] J. Brown, “Calculation of a constant q spectral transform,” *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, January 1991.

[3] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[4] B. C. J. Moore, *Introduction to the Psychology of Hearing*. Macmillan, 1977. [Online]. Available: http://www.amazon.com/Introduction-Psychology-Hearing-Brian-Moore/dp/0333197003%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0333197003

[5] D. Ellis. (2007) Chroma feature analysis and synthesis. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/

[6] D. Schwarz and I. C. Pompidou, “Data-driven concatenative sound synthesis,” Tech. Rep., 2004.

[7] D. Schwarz, “Concatenative sound synthesis: The early years,” *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[8] B. L. Sturm, “Matconcat: an application for exploring concatenative sound synthesis using matlab,” in *In Proceedings of the 7th International Conference on Digital Audio Effects (DAFx*, 2004.

[9] D. D. Lee and H. S. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[10] J. Driedger, T. Praetzlich, and M. Mueller, *Let It Bee - Towards NMF-Inspired Audio Mosaicing*. International Audio Laboratories Erlangen, 2015.

[11] D. W. Griffin, Jae, S. Lim, and S. Member, “Signal estimation from modified short-time fourier transform,” *IEEE Trans. Acoustics, Speech and Sig. Proc*, pp. 236–243, 1984.

[12] t. . Hristo Zhivomirov.

[13] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 556–562. [Online]. Available: http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf

[14] M. L. E. Stephen Handel, “Sound source identification: The possible role of timbre transformations,” *Music Perception: An Interdisciplinary Journal*, vol. 21, no. 4, pp. 587–610, 2004. [Online]. Available: http://www.jstor.org/stable/10.1525/mp.2004.21.4.587

[15] N. F. Johnson and S. Jajodia, “Exploring steganography: Seeing the unseen,” *Computer*, vol. 31, no. 2, pp. 26–34, Feb. 1998. [Online]. Available: http://dx.doi.org/10.1109/MC.1998.10029