NICHT-NEGATIVEMATRIXFAKTORISIERUNGNUTZENDESKLANGSYNTHESENSYSTEM (NIMFKS): EXTENSIONS OF NMF-BASED CONCATENATIVE SOUND SYNTHESIS

Michael Buch
Centre for Digital Music
Queen Mary University of London
London, UK
m.buch@se13.qmul.ac.uk

Elio Quinton
Centre for Digital Music
Queen Mary University of London
London, UK
e.quinton@qmul.ac.uk

Bob L. Sturm
Centre for Digital Music
Queen Mary University of London
London, UK
b.sturm@qmul.ac.uk

ABSTRACT

Concatenative sound synthesis (CSS) entails synthesising a "target" sound with other sounds collected in a "corpus." Recent work explores CSS using non-negative matrix factorisation (NMF) to approximate a target sonogram by the product of a corpus sonogram and an activation matrix. In this paper, we propose a number of extensions of NMF-based CSS and present an open MATLAB implementation in a GUI-based application we name NiMFKS. In particular we consider the following extensions: 1) we extend the NMF framework by implementing update rules based on the generalised β-divergence; 2) We add an optional monotonic algorithm for sparse-NMF; 3) we tackle the computational challenges of scaling to big corpora by implementing a corpus pruning preprocessing step; 4) we generalise constraints that may be applied to the activation matrix shape; and 5) we implement new modes of interacting with the procedure by enabling sketching and modifying of the activation matrix. Our application, NiMFKS and source code can be downloaded from here: `https://code.soundsoftware.ac.uk/projects/nimfks`.

1. INTRODUCTION

Given a dataset of digitally recorded sound material called a "corpus", the goal of concatenative sound synthesis (CSS) is to concatenate, or mix together, this sound material to approximate a "target" sound according to some criteria [1–7]. This criteria is typically in terms of distance within a descriptor space, e.g., features like spectral centroid and MFCCs, but can also involve considerations of context, e.g., concatenation cost and/or transformation cost [4, 7]. The motivations of CSS can be creative [2, 5, 8], and also to achieve high-quality sound synthesis [9, 10].

Recent work of Driedger et al. [3] explores the use of non-negative matrix factorization (NMF) for CSS. NMF [11] is an iterative procedure that attempts to express a non-negative matrix $\mathbf{V} \in \mathbb{R}_+^{N\times M}$ as a product of two other non-negative matrices, $\mathbf{V} \approx \mathbf{WH}$. The idea is to represent each column of $\mathbf{V}$ as a linear combination of the columns of $\mathbf{W} \in \mathbb{R}_+^{N\times K}$, by the weights in $\mathbf{H} \in \mathbb{R}_+^{K\times M}$. As such, the columns of $\mathbf{W}$ are referred to as *templates* and the rows of $\mathbf{H}$ as *activations*. In audio applications, it is common for $\mathbf{V}$ to be the magnitude or energy spectrogram (sonogram) [12], which is non-negative. NMF has found application in several areas of audio processing, e.g., polyphonic music transcription [12] and source separation [13–15].

In their CSS research, Driedger et al. [3] proposes to adapt NMF with additional constraints to enforce particular characteristics in the activation matrix $\mathbf{H}$. In particular, these modifications aim to reduce repetition, suppress simultaneity, and preserve context of sound material in the corpus. As a follow-up, Su et al. [16] extends on this work to create a certain "8-bit music" and contrast different methods of NMF when used for CSS. They develop a simple time-domain synthesis method to avoid sonic artifacts in the inversion of complex STFT, a technique we also develop here as an option in our application (see Sec. 3.1).

The work we present here makes several contributions to NMF for CSS. First, we propose a generalisation of existing procedures and introduce some extensions. Namely, we generalise the NMF framework and the post-processing constraints, by using the generalised β-divergence as an optimisation objective and a general convolution kernel respectively. As an additional constraint on the activation matrix, we implement a monotonic algorithm to enforce sparsity constraints in the NMF optimisation [17]. Second, we address computation issues with NMF that makes it unsuitable for very large corpora (i.e. dictionary martix $\mathbf{W}$). In order to overcome this limitation, we introduce a pruning strategy as a pre-processing step in which only a subset of the corpus frames are selected to be used in the NMF procedure (i.e. reducing the dimensionality of $\mathbf{W}$). Third, we present an open MATLAB Graphical User Interface (GUI), thereby making these techniques accessible. In addition, we propose and implemented other modes of interacting with the procedure, e.g., editing or "sketching" the activation matrix $\mathbf{H}$. To encourage further research and applications, we design our application NiMFKS to facilitate the integration of new modules through separation of concerns between the interface and implementation details and modular design with individual components responsible for different parts of the synthesis. Essentially one can simply plug-in desired NMF algorithms and modify existing ones without breaking the application.

2. PRIOR WORK

NMF aims to find non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{V} \approx \mathbf{WH}$. This takes the form of an optimisation problem where the factorisation attempts to minimise a reconstruction error function, which we denote $D(\mathbf{V}||\mathbf{WH})$. Common choices when applying NMF to audio data are Euclidian distance and Kullback-Leibler (KL) divergence [12–15].

Driedger et al. [3] use the KL divergence. At each iteration, after the NMF update, Driedger et al. [3] introduces subsequent updates to the activations to avoid particular behaviours, e.g., encouraging diagonal structures over horizontal and vertical ones. To suppress horizontal structures (e.g., corpus unit repetitions), they

modify $\mathbf{H}$ after the NMF update at iteration l of L according to

$$[\mathbf{H}]_{km} \leftarrow \begin{cases} [\mathbf{H}]_{km}, \ [\mathbf{H}]_{km} = \max\{[\mathbf{H}^T\mathbf{e}_k]_{m-r:m+r}\} \\ [\mathbf{H}]_{km}(1 - \frac{l+1}{L}), \ \text{else} \end{cases} \quad (1)$$

where $\mathbf{e}_k$ is the k^{th} standard vector, and $r \in \{0, 1, ...m\}$ describes the number of columns to the right and left of the m^{th} activation column over which one wishes to reduce repeated non-zero values.

To suppress non-zero values in each column of $\mathbf{H}$, i.e., discourage too many templates from being active simultaneously, they modify $\mathbf{H}$ according to

$$[\mathbf{H}]_{km} \leftarrow \begin{cases} [\mathbf{H}]_{km}, \ [\mathbf{H}]_{km} \in \max_p\{\mathbf{H}\mathbf{e}_m\} \\ [\mathbf{H}]_{km}(1 - \frac{l+1}{L}), \ \text{else} \end{cases} \quad (2)$$

where $\max_p\{\cdot\}$ returns the subset of $p \in \{0, 1, ...m\}$ largest values of its vector argument.

To promote diagonal structures, i.e., continuity of the corpus units, Driedger et. al. [3] process the activation matrix after its full update in iteration l by filtering. This can be expressed as a two-dimensional convolution between $\mathbf{H}$ and a diagonal kernel

$$\mathbf{H} \leftarrow \mathbf{H} * \mathbf{G} \quad (3)$$

where $*$ is convolution, and $\mathbf{G} = \mathbf{I}_c$ (identity matrix of size c) in Driedger et al. [3].

The application of these constraints removes the NMF algorithm's convergence guarantee. Thus the sequence of updates (1) to (3) is repeated until a user-specified stopping criteria, such as number of iterations. In summary, this NMF approach to CSS requires the specification of the following parameters: the total number of iterations L, the horizontal neighbourhood r, the vertical magnitude neighbourhood p, the filter kernel size c, and finally the parameters involved in computing the STFT and its inverse.

3. EXTENSIONS

We now present our three extensions to the application of NMF to CSS: 1) time-domain synthesis; 2) sparse NMF; and 3) pruning large corpora.

3.1. Time-domain synthesis

Given a magnitude spectrogram, or the product $\mathbf{WH}$ in the present case, the Griffin-Lim algorithm [18] can be used to project back to the time-domain. If we have the corpus as a time-domain waveform, then we can simply window, scale, and add the corresponding waveforms to create the synthesis. Since the m^{th} row of $\mathbf{H}$ specifies the activation of the m^{th} column of $\mathbf{W}$, the time domain synthesis is achieved by concatenating the corresponding portions of the corpus waveform scaled by their activation (i.e. elements of $\mathbf{H}$). This approach to synthesis circumvents the restrictions on window shape and overlap for invertibility inherent to the Griffin-Lim algorithm [18], and also makes possible the use of other kinds of features, e.g., chroma. It can be parallelised to make it faster.

3.2. Sparse NMF

The generalised β-divergence offers a family of divergences that are parametrised by β [19, 20]. It is defined as follows:

$$D_\beta(x||y) = \begin{cases} \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta-1} & \beta \in \mathbb{R}\backslash\{0,1\} \\ x\log\left(\frac{x}{y}\right) - x + y & \beta = 1 \\ \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 & \beta = 0 \end{cases} \quad (4)$$

When $\beta = 2$, this becomes the Euclidean distance; the Kullback-Leibler (KL) divergence for $\beta = 1$, and the Itakuro-Saito (IS) divergence for $\beta = 0$. In our case, we denote the β-divergence of $\mathbf{V}$ and $\mathbf{WH}$ as $D_\beta(\mathbf{V}||\mathbf{WH})$. In this case, we apply (4) to each element of the two matrices, and then sum over all elements. Fevotte and Idier [21] show that one of the possible multiplicative update rules using β-divergence is given by

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T\left[\mathbf{V} \odot (\mathbf{WH})^{(\beta-2)}\right]}{\mathbf{W}^T(\mathbf{WH})^{(\beta-1)}} \quad (5)$$

where $\odot$ denotes the element-wise multiplication, and exponentiation as well as division are element-wise.

As an extension of Driedger et al. [3], we implement sparsity constraints in the NMF update, which effectively restrict both polyphony and the corpus diversity within the NMF framework. There exist a number of algorithms for applying sparsity constraints on the activation matrix because it has proven useful for a variety of audio and music tasks, see for example [22, 23]. In our implementation, we use the penalty introduced in [17], notated Υ here, so that the cost function we minimise with respect to $\mathbf{H}$, $\mathbf{W}$ is:

$$D_\beta(\mathbf{V}||\mathbf{WH}) + \lambda\Upsilon \quad (6)$$

where λ is a parameter to control the weight of the penalty. In order for the regulariser Υ to enforce a sparsity constraint, it is common to set $0 < \beta \leq 1$. Typically, the closer β is to 0, the stronger the sparsity constraint will be. We set $\beta = 0.5$ in our application by default, as it enforces strong sparsity constraints, so that the templates that only account for little variance in the target matrix and their corresponding activation tend to zero during the NMF updates. By virtue of the multiplicative updates, components of zero magnitude remain zero for the remaining updates and are therefore effectively pruned out of the model [17]. The corresponding update rule for $\mathbf{H}$ is [17]:

$$\mathbf{H} \leftarrow \mathbf{H} \odot \left[\frac{\mathbf{W}^T\left(\mathbf{V} \odot (\mathbf{WH})^{\left[-\frac{3}{2}\right]}\right)}{\mathbf{W}^T(\mathbf{WH})^{\left[-\frac{1}{2}\right]} + \lambda\Psi_{\mathbf{H}}}\right]^\varphi \quad (7)$$

where $\Psi_{\mathbf{H}} = \mathbf{H}^{-1/2}$ is a term resulting from the application of the penalty Υ with $\beta = 0.5$ and $\varphi = \frac{2}{3}$ is a term ensuring the descent of the cost function at each iteration.

3.3. Working with Large Corpora

Optimisation approaches applied to CSS [3, 4, 7], carry far higher computational costs than "greedy" approaches, e.g., Catepillar [2] and MATConcat [5]. With the templates $\mathbf{W}$ fixed, NMF need only iteratively update the activations $\mathbf{H}$. Using the Euclidean distance, the computational complexity of each iteration is then $\mathcal{O}(K^2M)$. The KL divergence carries a higher complexity. These increase

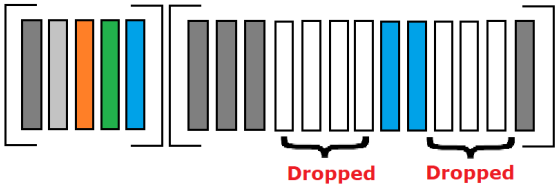


Figure 1: The corpus pruning algorithm removes those corpus frames that are dissimilar from any of the frames in the target. Algorithm 1 describes the procedure.

even more when we include the constraints detailed in section 2. Hence, the complexity is linear in the duration of the target, but quadratic in the duration of the corpus. This imposes limits on the size of the corpus one can work with in NMF for CSS, which can affect the quality of the results.

Algorithm 1: Corpus template pruning: Given two sequences of vectors in the same space $\mathbb{R}^N$ — $\mathcal{W}$ from the corpus and $\mathcal{V}$ from the target — and three user-specified parameters — $\gamma, \rho_{\min}, \theta$ — produce a subset of the index into $\mathcal{W}$ according to the dissimilarity function $d : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}_+$.

function Prune $(\mathcal{W}, \mathcal{V}, \gamma, \rho_{\min}, \theta)$;
Input :
1. $\mathcal{W} = (\mathbf{w}_k \in \mathbb{R}^N)_{k\in\mathcal{K}}, \mathcal{K} = \{1, \ldots, K\}$ (sequence of corpus frames)
2. $\mathcal{V} = (\mathbf{v}_m \in \mathbb{R}^N)_{m\in\mathcal{M}}, \mathcal{M} = \{1, \ldots, M\}$ (sequence of target frames)
3. $\gamma \geq 0$ (similarity pruning parameter)
4. $\rho_{\min} \in [0, 1)$ (relative energy pruning parameter)
5. $\theta > 0$ (comparisons parameter)

Output: $\mathcal{K}_{\text{pruned}} \subseteq \mathcal{K}$ (pruned index into $\mathcal{W}$)

$\mathcal{K}_{\text{remain}} \leftarrow \{k \in \mathcal{K} : \|\mathbf{w}_k\|_2 > \max_{l\in\mathcal{K}} \|\mathbf{w}_l\|_2 \rho_{\min})\}$;
$\mathcal{M}_{\text{remain}} \leftarrow \{m \in \mathcal{M} : \|\mathbf{v}_m\|_2 > \max_{l\in\mathcal{M}} \|\mathbf{v}_l\|_2 \rho_{\min})\}$;
$\mathcal{K}_{\text{pruned}} \leftarrow \emptyset$;
while $\mathcal{M}_{remain}$ *is not empty* **do**
  $m \leftarrow \mathcal{M}_{\text{remain}}(1)$ (first element of set);
  $\mathcal{D} \leftarrow (d(\mathbf{v}_m, \mathbf{w}_k) : k \in \mathcal{K}_{\text{remain}})$ (dissimilarities of the target frame to remaining corpus frames);
  $\mathcal{J} \leftarrow \{k \in \mathcal{K}_{\text{remain}} : d(\mathbf{v}_m, \mathbf{w}_k) < (1 + \gamma) \min \mathcal{D}\}$ (indices of corpus frames acceptably similar to the target frame);
  $\mathcal{K}_{\text{pruned}} \leftarrow \mathcal{K}_{\text{pruned}} \cup \mathcal{J}$ (store indices of corpus frames);
  $\mathcal{K}_{\text{remain}} \leftarrow \mathcal{K}_{\text{remain}} \backslash \mathcal{J}$ (remove indices of corpus frames from further consideration);
  $\mathcal{M}_{\text{remain}} \leftarrow \{m' \in \mathcal{M}_{\text{remain}} : d(\mathbf{v}_m, \mathbf{w}_{m'}) > \theta\}$ (indices of target frames that acceptably dissimilar from current target frame);
end
return $\mathcal{K}_{\text{pruned}}$;

To address this complexity issue, we propose preprocessing a corpus by pruning. Figure 1 depicts the basic concept. Frames in the target (left) and corpus (right) are colour coded by their content. Pruning keeps only the corpus frames that are measured similar enough to the target frames. We then execute NMF with an $\mathbf{W}$ built by concatenating the remaining templates. This assumes that the dissimilar corpus frames would not have been used in approximating $\mathbf{V}$ anyway.

The template pruning algorithm is shown in Algorithm 1. The user must specify three parameters, and a dissimilarity function. The dissimilarity function we use is the cosine distance

$$d(\mathbf{v}, \mathbf{w}) := 1 - \frac{\mathbf{v}^T\mathbf{w}}{\|\mathbf{v}\|_2\|\mathbf{w}\|_2}. \tag{8}$$

The parameter $\gamma \geq 0$ controls the severity of the pruning procedure, with smaller values proucing a higher degree of pruning. For $\gamma = 0$, only one corpus frame will be kept for each target frame considered. The parameter $\theta > 0$ controls the number of comparisons between target frames and corpus frames. As $\theta \to 0$ every target frame is considered, but this can be unnecessary when there is high similarity between target frames. As θ grows, we potentially consider fewer and fewer target frames in the pruning. Finally, the parameter $\rho_{\min} \in [0, 1)$ controls a preprocessing pruning procedure, removing the indices of the corpus and target frames that have norms that are too small. The effects of this are bypassed if $\rho_{\min} = 0$; and if $\rho_{\min} \to 1$, only the frame with the largest norm is kept.

After pruning, we use the columns of $\mathbf{W}$ indexed by $\mathcal{K}_{\text{pruned}}$ to create a $\mathbf{W}'$, and then use NMF as described above to find a $\mathbf{H}'$ such that $\mathbf{V} \approx \mathbf{W}'\mathbf{H}'$. We then upsample the activation matrix $\mathbf{H}'$ to form $\mathbf{H}$ having a size commensurate with the original problem. This involves distributing the rows of $\mathbf{H}'$ according to the indices $\mathcal{K}_{\text{pruned}}$, and leaving everything else zero. We can then apply modifications to $\mathbf{H}$, such as continuity enhancement and polyphony restriction, but at the conclusion of the NMF procedure.

4. THE MATLAB APPLICATION NiMFKS

Driedger et al. [3] do not supply a reproducible work package, but their procedure is described such that it can be reproduced. We have done so by implementing a GUI application in MATLAB, named NiMFKS. We organise the interface into four panes. Figure 2 shows an example screenshot of the interface.

The "Sound Files" pane, top left, lets the user load the audio files to be used as corpus and target. In this example figure, the user has selected multiple audio files, and a target instrumental recording of "Mad World" by "Gary Jules". If several audio files are specified by the user, they are concatenated to form the corpus. If the sampling rates of the target and corpus are different, NiMFKS resamples the target signal to have the same sampling rate as the corpus.

The pane labeled "Analysis" lets the user set the parameters for computing features of both the corpus and target audio. In the example of Fig. 2, the user has specified the STFT feature to be computed using a Hann window of duration 400ms with 50% overlap.

In the "Synthesis" pane, the user may select the method used for synthesising the output audio waveform and set the related parameters. Building on this observation and work reported in [16] where Su et al. demonstrate the benefits of synthesis approaches alternative to the ISTFT, we make both Time-Domain (cf. section 3.1) and ISTFT synthesis available in NiMFKS. The "Synthesis Method" drop-down menu lets the user chose between these.

The NMF sub-pane lets the user set all the parameters that influence the factorisation. The "Prune" parameter, γ (cf. section

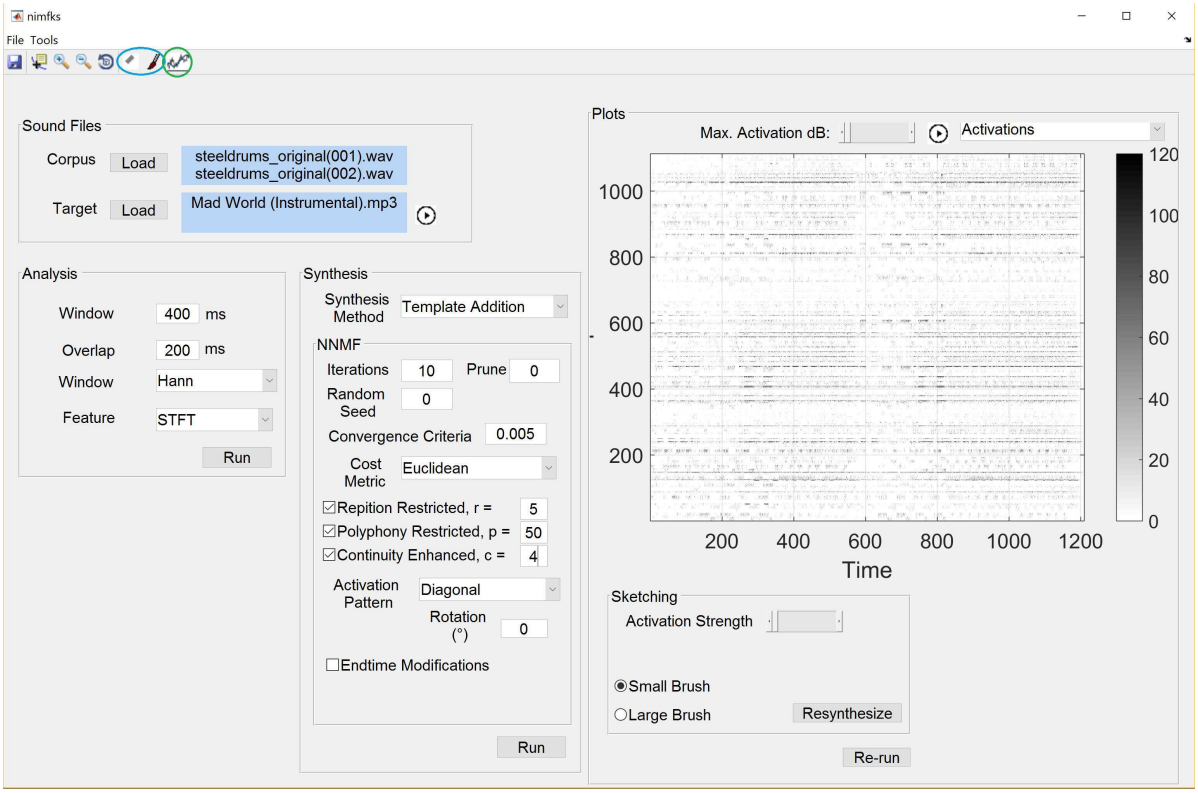


Figure 2: This screen shot shows the main window of NiMFKS. We see the target is "Mad World", and the corpus is multiple samples of steel drums. The user presses the run button in the "Analysis" pane to compute STFT features using a Hann window of duration 400ms with overlap of 50%. When the user presses the "Run" button in the "Synthesis" pane, 10 iterations of NMF with the Euclidean distance will be performed with the additional constraints (see Sec. 2). No pruning is performed in this case. For synthesis "Template Addition" is specified. On the right we see a visualisation of the activation matrix. Next to the slider is the play button to hear the resulting synthesis. Circled in blue at top are the "Activation Sketching" and "Activation Eraser" features (see Sec. 5). Clicking one of these buttons will reveal the "Sketching" pane in which you can choose the "Paint Brush" to perform the actual sketching. If new activations are added or removed by hand, the buttons "Resynthesise" and "Re-run" repeat the synthesis and NMF procedure respectively using the new activation matrix.

3.3), allows to tune how aggressively the corpus loaded by the user should be pruned before it is fed to the NMF optimisation. Currently, NiMFKS defines $\rho_{\min}$ to be -60 dB below the maximum observed in the corpus and target, and $\theta = 0.1$.

The NMF updates stop after a given number of iterations, or when the cost function reaches a convergence criteria. In the example of Fig. 2, NMF will perform no more than 10 iterations, but could exit before if the relative decrease in cost between subsequent iterations is less than 0.005.

NiMFKS relies on β-NMF update rules (cf. section 2) and provides three presets accessible from the "Cost Metric" dropdown menu: Euclidean ($\beta = 2$), Kullback-Leibler divergence ($\beta = 1$), and Sparse NMF with $\beta = 1/2$ as described in (4).

Finally, the remaining parameters control the modifications to be applied to the activation matrix $\mathbf{H}$ either at each iteration of the NMF optimisation or only after convergence if the "Endtime Modifications" parameter is toggled. The user can set the parameters of the filtering kernel, as described in section 2.

The pane labeled "Plots" displays various results from the procedure, accessible from the dropdown menu at top-right. These include the sound waveforms, their sonograms, and the activation matrix. In the example of Fig. 2, the user has selected to view the resulting activation matrix, and can adjust the contrast of the image by the slider labeled "Max. Activation dB".

Finally, in order to enable the "Activation Sketching" feature (cf. section 5.2), the user must select the "Activation Sketching" item from the "Tools" menu, which enabled the buttons located at the very top left of the application window and circled on Figure 2. The sub-pane labeled "Sketching" then lets the user specify the settings to be used when sketching activations.

5. ARTISTIC EXTENSIONS

We now describe a few artistic extensions that we have implemented in NiMFKS.

5.1. Activation Sketching

NiMFKS allows one to treat the activation matrix as an image on which the user can directly erase, draw, inflate or deflate activations. This feature is labeled as "Activation Sketching" in the GUI. These can be resynthesized on the fly to see any effects. This proved to be a useful experimentation tool to understand NMF and the synthesis methods in action. Figure 3 shows an example of an

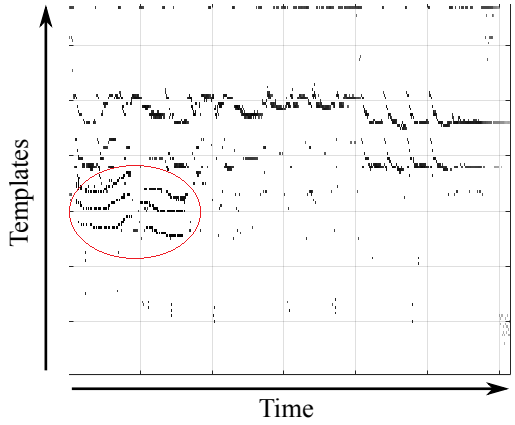


Figure 3: Image of an activation matrix produced by NMF using a specific corpus and target but on which we have sketched additional activations (circled).

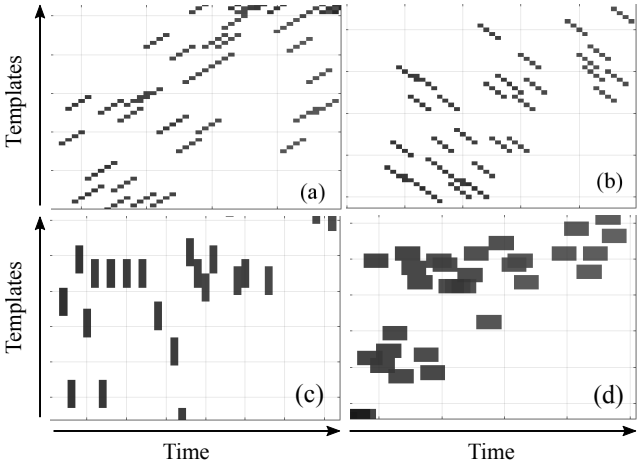


Figure 4: Clockwise from top-left, an activation matrix $\mathbf{H}$ has been filtered (3) with a kernel $\mathbf{G}$ that is diagonal, anti-diagonal, vertical bar and horizontal bar. More exotic options are possible.

activation matrix to which we have sketched additional activations. One can also remove activations with the eraser, and resynthesise the result.

5.2. Modifications of the activation matrix

Driedger et. al. [3] constrained the activation matrix to exhibit diagonal structures in order to preserve the original context of the corpus units. This was created by using a diagonal filter, i.e. setting $\mathbf{G} = \mathbf{I}_c$ in (3). We can define different kinds of filtering kernels, e.g., anti-diagonal, or block average in order to favour other types of structure. NiMFKS enables the user to choose from a selection of such kernels. For instance, the anti-diagonal filter promotes similar structures to the diagonal filter, but with a time-reversed context (the corpus units are not time-reversed, however). The vertical bar filter results in activating several templates simultaneously, while the horizontal bar filter produces repetitions of corpus units. Figure 4 shows several examples of activation matrices obtained using such structures. Additionally, the diagonal kernel may be rotated by an angle specified by the user.

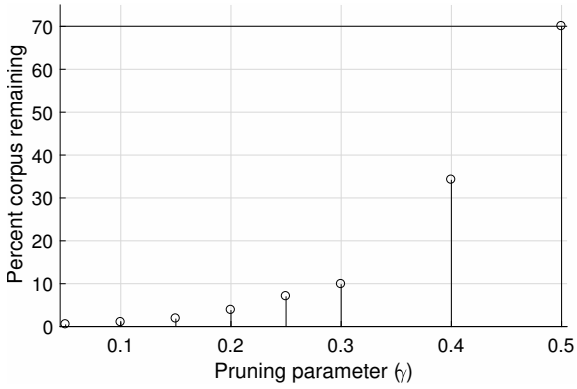


Figure 5: For several values of the pruning parameter, the percentage of the corpus remaining given a target. About 70% of the corpus remains after pruning by energy (keeping frames within 60 dB of the most energetic frame).

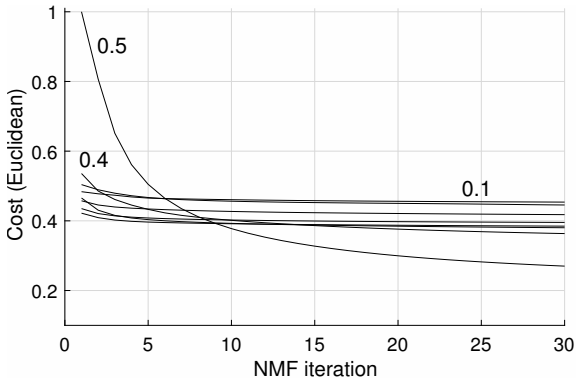
6. DEMONSTRATION

We now demonstrate the pruning strategy. We build a 40m58s corpus from a collection of 44.1 kHz sampled vocalisations from a variety of monkeys and apes. Our analysis of this corpus uses a Hann window of 100ms duration, and 50ms hop. We compute the discrete Fourier transform of each frame, sampled at 1024 points from DC to the Nyquist frequency. In total, there are 49,164 frames, which means $\mathbf{W}$ is a matrix of size 1024×49164, or 50,343,936 elements. Our target is a recording of "President" Trump saying, "Sadly. The American. Dream. Is *dead*!" Figure 5 shows the percentage of the corpus remaining after the pruning procedure for several pruning parameters. At the most strict value tested, $\gamma = 0.05$, only 247 frames remain. At the least strict value, $\gamma = 0.5$, the number of frames remaining is 34,433. The pruning by energy removes 14,731 frames.

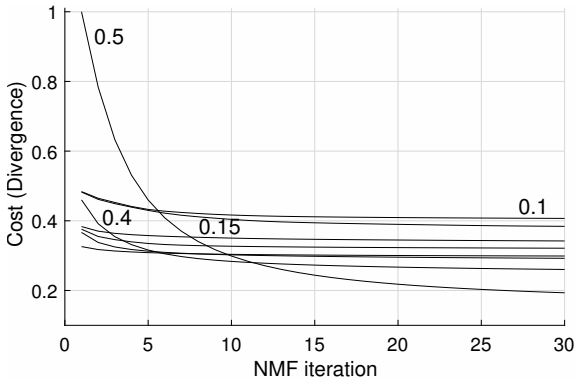
Figure 6 shows how the pruning parameter affects the cost of the factorisation over iteration. We see for this example that after 10 iterations the cost from using a larger corpus becomes smaller than when using the more pruned corpus. Figure 7 shows how the spectrogram of the original target matches with the results from NMF and three pruned corpora. The bottom spectrogram shows the results with a pruning $\gamma = 0.1$ and constraints with values $r = p = c = 3$. The convolution matrix used for continuity enhancement in (3) is diagonal with exponentially decreasing positive values along the diagonal.

7. DISCUSSION & CONCLUSION

The work of Driedger et al. [3] provides an excellent starting point for exploring NMF for CSS but their code is not available. We have implemented their approach and make the application freely available. Our application contributes a variety of extensions as well. We have implemented a time-domain synthesis method, which does not suffer from the limitations of STFT inversion. We make the modifications proposed in [3] applicable during the NMF update procedure, or at the end as a post-processing of the activation matrix. We also implement a pruning strategy for working with



(a) Euclidean distance



(b) Divergence distance

Figure 6: Change in cost over NMF iteration with pruned corpora (relative to highest cost for $\gamma = 0.5$).

large corpora. Although we find this hurts the cost in the factorising procedure, it can become useful when trying to get an idea of how a target and corpus could work together. In a more creative direction, we have implemented an interactive "sketching" procedure by which one may modify a resulting activation matrix, and more general activation filtering procedure.

Our future work will address the uniform segmentation of this NMF approach to CSS. We seek a way to make it compatible with a unit-based segmentation of a target and corpus. A simple way this could be solved is by performing parallel NMF with different time resolutions, and then a fusion of the results at the end with binary masking. We will also redesign the GUI to separate analysis, pruning, factorising, post-processing and synthesising.

8. ACKNOWLEDGMENTS

This work is supported by AHRC Grant No. AH/N504531/1.

9. REFERENCES

[1] A. Zils and F. Pachet, "Musical mosaicing," in *Proc. COST G-6 Conf. Digital Audio Effects*, Limerick, Ireland, 2001.

[2] D. Schwarz, "Concatenative sound synthesis: The early years," *J. New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[3] J. Driedger, T. Prätzlich, and M. Müller, "Let It Bee – towards NMF-inspired audio mosaicing," in *Proc. Int. Conf. Music Information Retrieval*, Malaga, Spain, 2015, pp. 350–356.

[4] J.-J. Aucouturier and F. Pachet, "Jamming with plunderphonics: Interactive concatenative synthesis of music," *J. New Music Research*, vol. 35, no. 1, 2006.

[5] B. L. Sturm, "Adaptive concatenative sound synthesis and its application to micromontage composition," *Computer Music J.*, vol. 30, no. 4, pp. 46–66, Dec. 2006.

[6] G Bernardes, *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*, Ph.D. thesis, Faculty of Engineering, University of Porto, 2014.

[7] G. Coleman, *Descriptor Control of Sound Transformations and Mosaicing Synthesis*, Ph.D. thesis, Universitat Pompeu Fabra, 2016.

[8] Juan José Burred, "Factorsynth: A max tool for sound analysis and resynthesis based on matrix factorization," in *Proceedings of the Sound and Music Computing Conference 2016, SMC 2016, Hamburg, Germany*, 2016.

[9] E. Lindemann, "Music synthesis with reconstructive phrase modeling," *IEEE Signal Processing Magazine*, vol. 24, no. 2, pp. 80–91, March 2007.

[10] M. Umbert, J. Bonada, M. Goto, T. Nakano, and J. Sundberg, "Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 55–73, Nov 2015.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, pp. 556–562. MIT Press, 2001.

[12] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on App. Signal Proc. Audio and Acoustics*, 2003, pp. 177–180.

[13] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio, Speech, and Lang. Proc.*, vol. 18, no. 3, pp. 550–563, 2010.

[14] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, and Lang. Proc.*, vol. 15, no. 3, pp. 1066–1074, 2007.

[15] S. Ewert and M. Müller, "Using score-informed constraints for NMF-based source separation," in *Proc IEEE Int. Conf Acoustics, Speech and Signal Proc., 2012*. 2012, pp. 129–132, IEEE.

[16] S. Su C. Chiu L. Su and Y. Yang, "Automatic conversion of pop music into chiptunes for 8-bit pixel art," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, March 2017, pp. 411–415.

[17] E. Quinton, K. O'Hanlon, S. Dixon, and M. B. Sandler, "Tracking Metrical Structure Changes with Sparse-NMF," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Proc.*, 2017.

[18] D. W. Griffin, J., and S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoustics, Speech and Signal Proc.*, pp. 236–243, 1984.

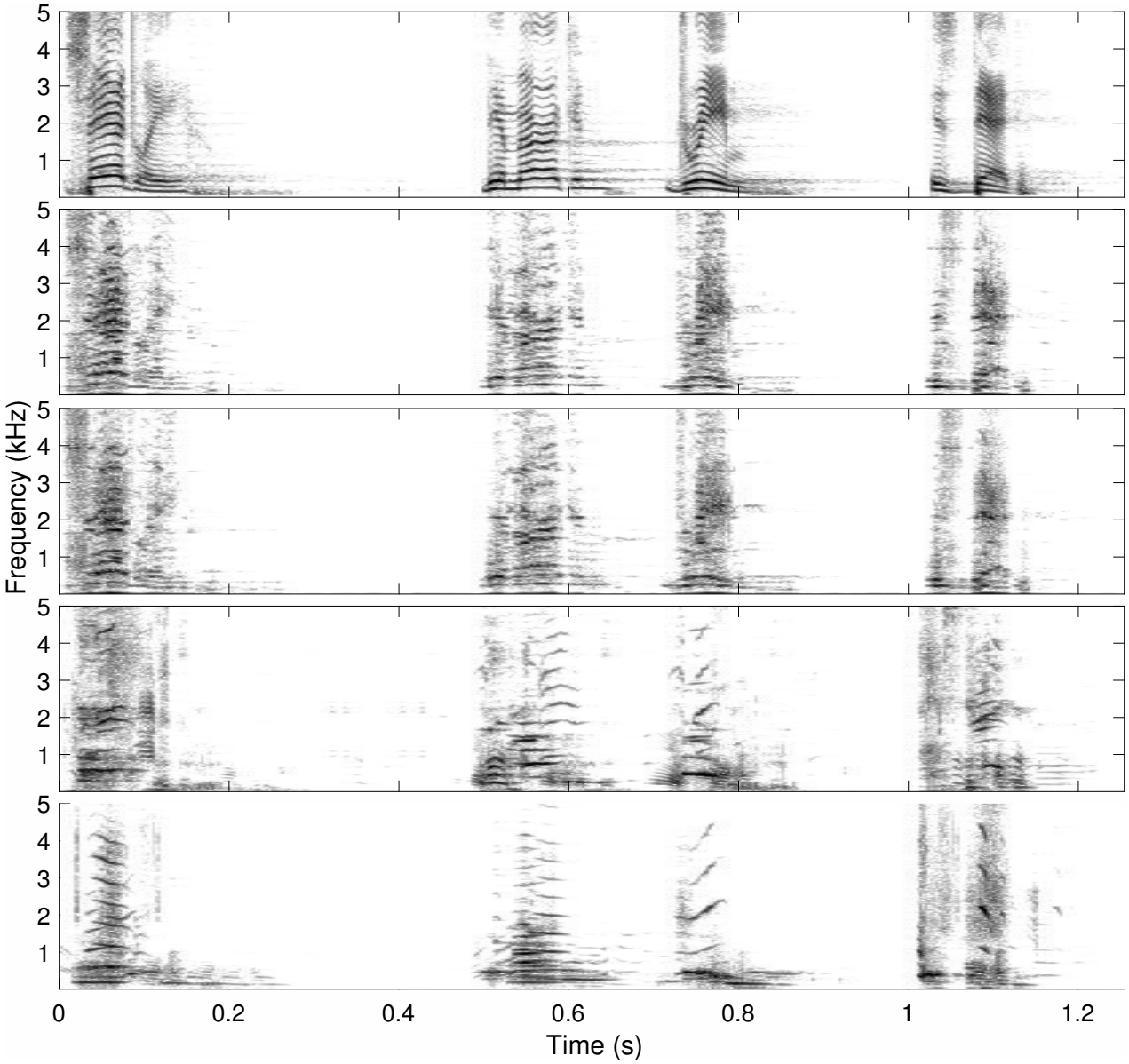


Figure 7: We use NMF-based CSS to resynthesise a recording of "President" Trump saying, "Sadly. The American. Dream. Is *dead*!" using a corpus of 40m58s of sampled vocalisations of monkeys, apes and other animals. The features are DFT magnitude frames. The spectrograms from top to bottom are: original signal; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.5$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.3$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.1$; synthesis by NMF with Euclidean distance and pruning with $\gamma = 0.1$, and constraints $r = p = c = 3$ with $\mathbf{G}$ in (3) diagonal that is exponentially decreasing.

[19] R. Kompass, "A generalized divergence measure for nonnegative matrix factorization," *Neural computation*, vol. 19, no. 3, pp. 780–791, 2007.

[20] A. Cichocki, R. Zdunek, and S. Amari, "Csiszar's divergences for non-negative matrix factorization: Family of new algorithms," in *Proc. Int. Conf. on Independent Component Analysis and Signal Separation*, 2006, pp. 32–39.

[21] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[22] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Machine Learning Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[23] J. Eggert and E. Korner, "Sparse coding and NMF," in *Proc. of the Int. Joint Conference on Neural Networks*, 2004, vol. 4, pp. 2529–2533.