CROSS-SYNTHESIS BASED ON SPECTROGRAM FACTORIZATION

Juan José Burred

Paris, France

jjburred@jjburred.com

ABSTRACT

Spectrogram factorization techniques decompose a sound into a set of characteristic spectral shapes and a set of corresponding temporal evolutions. This can be exploited for a cross-synthesis-like processing by combining the spectral shapes of one sound with the temporal evolutions of the other. A system is proposed that implements such a task in an unsupervised way by means of a comparison of the involved spectral shapes in terms of timbral similarity, and a phase reconstruction algorithm for the resynthesis. The system enables cross-synthesis at the level of intra-note resonances, transients or temporalities. Some illustrative sound generation examples will be presented and discussed.

1. INTRODUCTION

Matrix factorization methods, when applied to spectrograms, have the ability to reveal underlying, dynamic patterns that constitute the sound being analyzed, and as such, they have been extensively used in analytic applications such as audio content analysis, music information retrieval and source separation. Their usage for musical composition or sound synthesis has only recently started to be explored, though [8]. In very broad terms, it can be said that they decompose a sound as a set of time-frequency layers. Such layers can then be processed separately, or be used as building blocks for resynthesis. This suggests that spectrogram factorization could offer an attractive alternative to other analysis/resynthesis techniques such as sinusoidal or source/filter modeling.

In its most general definition, matrix factorization aims at approximating an observed matrix $\mathbf{X}$ as a product of two factor matrices: $\mathbf{X} \approx \mathbf{WH}$. A particular case of interest is when the combined size of the factor matrices is smaller than the size of the observed matrix. If matrix $\mathbf{W}$ is of size $F \times K$ and $\mathbf{H}$ is of size $K \times T$, then this condition will write: $F \times K + K \times T \ll F \times T$. The internal product dimension K is usually a fixed parameter given to the factorization algorithm. When this condition holds, matrix factorization is also called *factor analysis* or *low-rank approximation*. Apart from its benefits for data compression purposes, the usefulness of low-rank approximation is explained by the fact that setting a small K forces the algorithm to "condense" as much information as possible from the observation matrix into the two smaller factor matrices. If the approximation is successful, the factor matrices will end up containing the essential information needed to reconstruct $\mathbf{X}$ as closely as possible from much less data points, and such information corresponds to "hidden factors" (latent variables) present in the observed data.

One possible way to interpret the decomposition produced by matrix factorization is to consider that the approximation equals the sum of a set of K matrices $\mathbf{C}_k$, all of the same size than $\mathbf{X}$, and each one of which has been generated by the outer product of the k-th column of $\mathbf{W}$ with the k-th row of $\mathbf{H}$:

$$\mathbf{X} \approx \mathbf{WH} = \sum_{k=1}^{K} \mathbf{C}_k = \sum_{k=1}^{K} \mathbf{w}_k \otimes \mathbf{h}_k, \qquad (1)$$

where $\mathbf{w}_k$ is the k-th column of $\mathbf{W}$, $\mathbf{h}_k$ is the k-th row of $\mathbf{H}$, and $\otimes$ denotes the outer product (note that the outer product of two vectors, given by $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T$ produces a matrix, whereas the inner or scalar product $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b}$ produces a scalar). The component matrices $\mathbf{C}_k$ are sometimes called *rank-1 terms*. A graphical representation of Eq. 1 is given in Fig. 1.

In the specific case of matrix $\mathbf{X}$ being a magnitude spectrogram of F frequency bins and T temporal frames (i.e., its colums are the spectral frames, and its rows the evolution of each frequency bin over time), then the columns of $\mathbf{W}$ can be interpreted as spectral shapes, and the rows of $\mathbf{H}$ as coefficients over time. Thus, each component $\mathbf{C}_k$ can be thought of as a magnitude spectrogram generated by multiplying a static spectral shape $\mathbf{w}_k$ by a time-varying weight $\mathbf{h}_k$. In this context, the $\mathbf{w}_k$s are sometimes called *spectral bases* (or just *bases*) and the $\mathbf{h}_k$s are called *activations*. The activations indicate the importance of each spectral shape at each time frame, and thus contain powerful information that can be exploited for analysis or separation. For instance, if the algorithm finds a spectral basis that strongly resembles the spectrum of a particular note of an instrument present in the observed signal, then a high value at a given time frame of the corresponding activation will indicate an onset of this note, and this can be exploited for transcription applications [6]. Also, by resynthesizing the component spectrograms $\mathbf{C}_k$ (or appropriate subsets thereof), one might be able to perform source separation with simple mixtures [9]. Even if current state-of-the-art systems for content analysis or source separation employ more sophisticated signal models than a single bases/activations decomposition, spectrogram factorization remains the core element of most of the best-performing approaches. The most popular technique used for spectrogram factorization is Non-

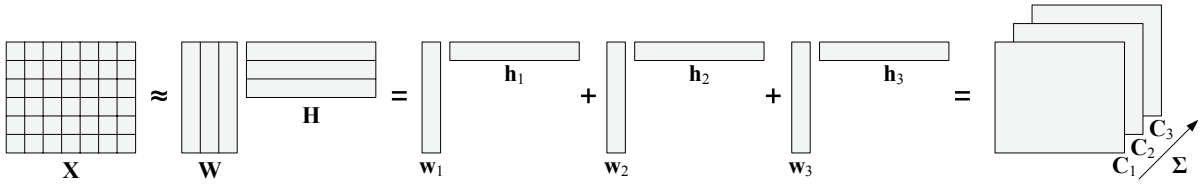


Figure 1. Interpretation of matrix factorization as a sum of rank-1 terms.

negative Matrix Factorization (NMF), which is also used here and will be introduced in Sect. 3.

There have been a few previous uses of spectrogram factorization in musical composition or sound synthesis, as will be introduced in the next section. Most of them involve the separate processing of the extracted components from a single input sound. In the present article, a new technique is proposed, in which two sounds are subjected to spectrogram factorization (a source sound and a target sound). The obtained sets of source and target bases and activations can be then combined with each other, and an automatic method is proposed to that aim. For instance, all (or some) of the source sound's activations can be imposed on the bases of the target sound to generate an hybrid sound that combines the timbre of the target with the temporality of the source. The idea is similar to that of cross-synthesis via source/filter models or vocoders (in which the global spectral envelope of one sound is imposed onto the other), but allows more flexibility by enabling cross-synthesis-like processing at the level of the components. As an example, it can be used to separately control the individual resonances or formants of a target sound by the separate temporal evolutions of the resonances of the source sound. Automatizing such procedure raises some questions such as the appropriate mapping between source and target spectral bases and the problem of phase reconstruction for resynthesis. Solutions to these issues will be proposed throughout the article.

It should be noted that another key difference between traditional source/filter-based cross-synthesis and the current approach is that, in the former, the temporalities of both source and target signals have a direct influence on the hybrid sound, whereas here, temporality and timbre are fully decoupled and contributed separately by source and target. In this sense, a related idea is *spectral reanimation* [3], in which timbre and time are also decoupled, albeit in a different way: each spectral frame from the source is replaced, keeping the sequence, by the most similar one from the target.

2. USES OF SPECTROGRAM FACTORIZATION IN MUSICAL CREATION

As mentioned above, the use of spectrogram factorization as a tool for musical composition or performance has been seldom explored. A review of five recent musical works using factorization techniques is included in [8]. Most of them are based on Probabilistic Latent Component Analysis (PLCA), which can be understood as a probabilistic reformulation of NMF. The cited works use an implementation of PLCA decomposition called SoundSplitter to decompose a sound into individual time-frequency components, which are then processed or manipulated separately. As an example, *Decomposing Autumn* (2010) by David Plans Casal uses PLCA to generate a database of components extracted from a recording of a piano etude by Ligeti. The stored components are then matched live with the sounds produced by the performer using music information retrieval techniques.

Another example cited therein and more relevant to the work presented here is the piece *Elementary Sources* (2011) by Spencer Topel. Again, an input sound is decomposed by PLCA, and some of the components are then selected and resynthesized. It contains some elements of cross-synthesis, since spectral bases and activations are manually recombined in different ways through trial and error. The recreated components were then transcribed into musical notation for interpretation by a string quartet. During performance, the quartet is mixed with live electronics derived from the PLCA components.

In [4], a complex natural soundscape is decomposed by PLCA. The extracted temporal activations are then used to drive several parameters of a group of synthesizers. In this example, the decomposition is used to impose temporal structure, rather than to hybridize timbre.

Another domain of new musical application of matrix factorization is sound effects processing. In [5], new sound effects are proposed by analyzing the sound with NMF and reordering the extracted components according to features calculated on the bases or the activations. The components are then assigned different weights according to their new position, and the final signal is reconstructed by adding back the weighted components.

3. PROPOSED SYSTEM

Figure 2 shows a block diagram of the proposed cross-synthesis system. By convention, the current system is designed so that the temporality of the output sound is provided by the source sound, and its timbre is mainly provided by the target sound. The following subsections will go through the different processes taking place.

3.1. Source and target analysis

The first step of the process is to analyze both source signal $s(n)$ and target signal $t(n)$ by means of matrix factor-

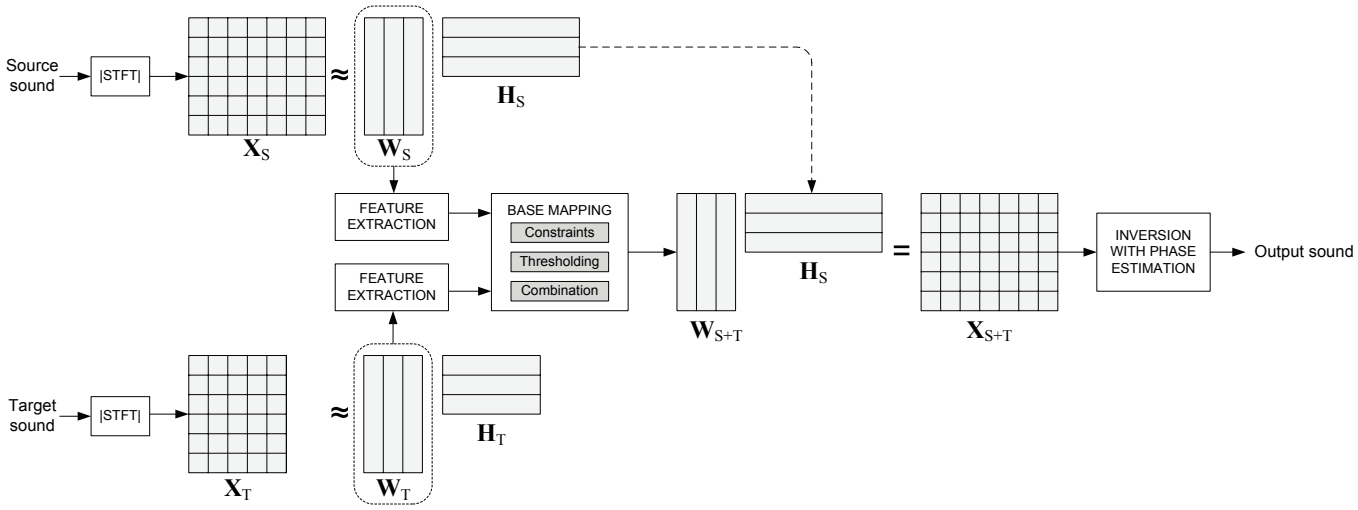


Figure 2. Overview of the factorization-based cross-synthesis system.

ization applied to their respective magnitude spectrograms $\mathbf{X}_S = |\mathrm{STFT}\{s(n)\}|$ and $\mathbf{X}_T = |\mathrm{STFT}\{t(n)\}|$, where STFT denotes the Short Time Fourier Transform. In particular, each magnitude spectrogram is subjected to NMF, which is a particular type of matrix factorization that enforces all elements of the output factor matrices to be positive or zero (and assumes the same about the input matrix). When working with magnitude audio spectrograms, NMF is by far the most popular method for low-rank approximation, since the resulting bases and activations have a direct physical interpretation as respectively, static magnitude spectra and temporal gain functions (other factorization methods such as Principal Component Analysis or Independent Subspace Analysis can produce negative values in both bases or activations).

There are several existing NMF algorithms, with different cost functions and optimization methods. For the purpose of this article, a multiplicative update algorithm based on the minimization of the Frobenius norm (sum of element-wise squared errors) of the approximation error was used [2]. Such error measure is given by

$$D_F = \|\mathbf{X} - \mathbf{WH}\|_F = \sum_{f=1}^{F}\sum_{t=1}^{T}(\mathbf{X}_{(t,f)} - (\mathbf{WH})_{(t,f)})^2. \quad (2)$$

The choice of the number of components for source and target (respectively, K_S and K_T), is critical. They are the two main parameters of the system, and they must be carefully set depending on the desired results. As a general guideline, the values of K_S and K_T (which do not need to be the same) should depend on the number of pitches present in either analyzed sound. Two cases are considered here:

- **Note-level analysis.** If K equals the number of notes of different pitches present in the analyzed sound, the resulting $\mathbf{w}_k$s will very likely contain one spectrum per pitch. This is useful if a cross-synthesis at the note level is desired. However, the resynthesis of the individual note-level components will lack dynamics, since each note will correspond to a rank-1 term (static spectrum multiplied by a time-varying gain).

- **Resonance-level analysis.** More interesting results can be obtained if cross-synthesis is performed at the resonance-level, i.e., if K is set higher than the number of pitches (e.g., as a multiple thereof). In this case, each pitch on the sound will be decomposed into a subset of bases. If prominent resonances or formants are present in each note, they will be revealed and separated by the factorization into different $\mathbf{w}_k$s. The same is true for other non-resonant salient events such as attacks or transients. This enables transformations on the internal temporality of the sounds. On the other hand, one must be careful not to set K to high to avoid noisy, non-informative components.

Figure 3(a) shows an example of a three-note, note-level analysis with $K = 3$ (piano sound). The functions plotted on the top of the spectrogram are the temporal activations $\mathbf{h}_k$. The functions to its left are the spectral bases $\mathbf{w}_k$. It can be seen that, since the number of components equals the number of pitches, the factorization algorithm has assigned each pitch to a component, thus, the spectral bases contain spectra showing each note's characteristic harmonics, and the activations clearly indicate each note's onset and global energy evolution.

On the other hand, Fig. 3(b) shows an example of resonance-level analysis where the analyzed sound is a single note played by a bell with clearly audible resonances. Since the component number $K = 3$ is in this case higher than the number of pitches, the algorithm has split the resonances among different components. Note that the activations clearly show the energy oscillations of the individual resonances. These example sounds, and separated components, can be listened to on the compan-

ion website to this article that will be presented in Sect. 4.

3.2. Mapping of spectral bases

After the analysis stage, a set of bases and activations for source and target sounds are obtained and stored. The main question that motivated the design of the present system was how to automatically combine the obtained bases and activations in order to obtain satisfactory hybrid sounds. Directly multiplying the target base matrix with the source activation matrix will almost invariably produce poor results, because of the following two reasons.

Firstly, NMF suffers from what is called the *permutation problem*: the ordering of its output components is random. This is due to the random initializations used by the multiplicative update algorithm. Thus, by directly multiplying activations and bases from two unrelated factorizations, high-energy bases of the source (i.e., important spectral shapes associated with a high average activation) might get multiplied by low-energy or noisy activations of the target. Vice-versa, noise-like spectral shapes can get unnaturally amplified by high-energy activations. In the end, the output will likely be noisy.

Secondly, more natural hybrid sounds will be obtained if the source and target components are matched following some notion of similarity. In the present version of the system, temporal information is fully provided by the source sound, and timbral information by either the target sound, or by a combination of target and source [1]. Thus, a comparison of the spectral bases in terms of timbral similarity is of interest here.

The purpose of the *base mapping* module is to circumvent these two issues. It computes a similarity matrix $\mathbf{S}$ of size $K_S \times K_T$ between all possible couples of source/target spectral bases. A simple and widely-used approximation of timbre similarity is used here: the bases are first subjected to feature extraction and converted to Mel Frequency Cepstral Coefficients (essentially, a compact description of the spectral envelope), and subsequently compared to one another by means of the Mahalanobis distance:

$$\mathbf{S}_{(i,j)} = \sqrt{(\mathbf{m}_{S,i} - \mathbf{m}_{T,j})^T \mathbf{\Sigma}^{-1} (\mathbf{m}_{S,i} - \mathbf{m}_{T,j})}, \quad (3)$$

where $\mathbf{m}_{S,i} = \mathrm{MFCC}\{\mathbf{w}_{S,i}\}$, $\mathbf{m}_{T,i} = \mathrm{MFCC}\{\mathbf{w}_{T,i}\}$, and $\mathbf{\Sigma}$ is the covariance matrix computed over the whole dataset (i.e., of all feature vectors $\mathbf{m}_{S,i}$ and $\mathbf{m}_{T,i}$ put together). The Mahalanobis distance is preferred to the Euclidean distance due to its scale-invariance.

By default, the base mapping module assigns each source basis to its closest target basis according to the similarity matrix $\mathbf{S}$, and rearranges the target bases so that their new ordering (as columns of the new matrix $\mathbf{W}_{S+T}$) is the same as the ordering of their corresponding similar bases from the source. When the final hybrid spectrogram is obtained with the multiplication $\mathbf{W}_{S+T}\mathbf{H}_S$, the source activations are now acting on components of similar spectral envelope from the target[2].

The rearrangement of the bases and activations is often enough to get satisfactory hybrid sounds (sound examples will be discussed in Sect. 4). Nevertheless, several optional operations have been implemented into the base mapping module, to gain further control on the cross-synthesis process: injectivity constraint, similarity thresholding and combination of bases.

3.2.1. Injectivity constraint

There is no guarantee that the base mapping will be a one-to-one (injective) relation. For instance, several bases from the source might be close together in terms of feature similarity, so that they will be assigned to the same single basis of the target. This might be a desirable behavior if the main goal is to preserve the timbral fidelity in the process. However, if many source bases point to very few target bases, this will result in a target sound with little internal dynamics.

To control such a trade-off between timbre fidelity and temporal variability, the user might enable an injectivity constraint, that ensures that the base mapping results in a one-to-one assignment. If that option is activated, a list of the target bases that have already been assigned to a source based is maintained. If a target base is marked as already selected for that source base, the second closest target base to that source base is selected, and so on.

3.2.2. Similarity thresholding

Another way to control the timbral fidelity is to define a similarity threshold. All the bases that are less similar to each other than the threshold are discarded (and so are their corresponding activations). In the current implementation, the optional threshold is defined as the median value of all the similarity values contained in the similarity matrix $\mathbf{S}$.

3.2.3. Combination of spectral bases

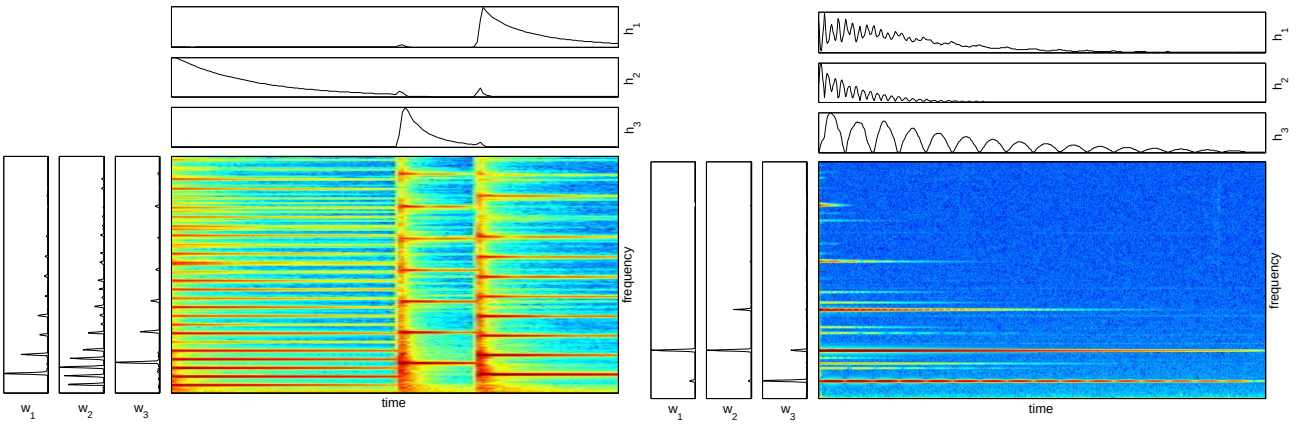
Finally, instead of discarding the bases that are too different, the user might choose to have them replaced by the corresponding source bases. In this case, the hybrid base matrix $\mathbf{W}_{S+T}$ will end up containing bases from both source and target signals: those target bases that are similar to some source bases, and those source bases that do not have a close match. The motivation of this option is to allow keeping the dynamics of the source sound even if the timbre of source and target sounds are very different.

3.3. Phase reconstruction

After the hybrid magnitude spectrogram has been obtained as $\mathbf{X}_{S+T} = \mathbf{W}_{S+T}\mathbf{H}_S$, it must be inverted to gen-

[1] This is just a convention of the current system. In future extensions, merging source and target temporalities might be another option.

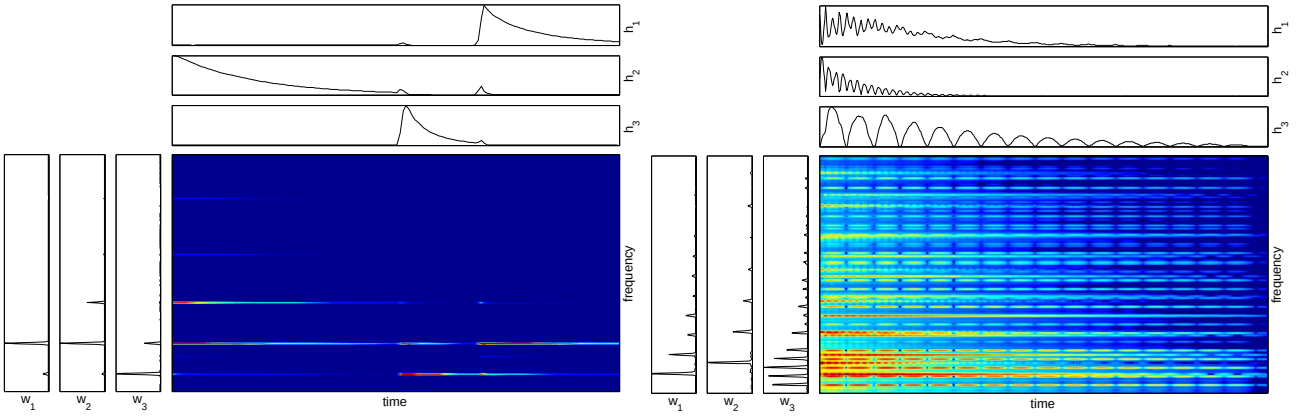
[2] Note that base mapping by a mere rearrangement of bases and activations requires $K_S = K_T$.



(a) Source analysis at the note level.

(b) Target analysis at the resonance level.

Figure 3. Examples of analysis by spectrogram factorization.



(a) Source-to-target cross-synthesis.

(b) Target-to-source cross-synthesis.

Figure 4. Examples of cross-synthesis by spectrogram factorization with the source and target from Fig. 3.

erate the output hybrid waveform. NMF (as many other methods used for spectral modifications), does not take into account phase information and only operates on magnitude and frequency. Thus, a technique for appropriate phase reconstruction is required.

In other factorization-based approaches requiring resynthesis, such as source separation or effects processing [5], phase is usually retrieved from the original input signal, either by directly attaching the phase spectrogram of the input to the processed magnitude spectrogram and inverting the STFT, or by time-frequency masking applied to the input (usually via Wiener filtering). Neither of those solutions is appropriate for the present application. Since originally unrelated pairs of base and activation matrices are being combined, there is no guarantee of phase coherence between input and output frequency bands. Furthermore, time-frequency masking is not applicable, since the output sounds are not generated from the input in a subtractive manner by filtering, but rather consist of completely new spectrogram components generated by the products $\mathbf{w}_{S+T,k} \otimes \mathbf{h}_{T,k}$.

Instead, the proposed system uses the Griffin and Lim (GL) algorithm [1] for constructing the output waveform from the hybrid magnitude spectrogram $\mathbf{X}_{S+T}$. The GL algorithm is an iterative gradient descent method that usually converges to a waveform that is perceptually satisfactory for a range of sound processing algorithms based on magnitude spectrogram modification or creation [7].

The user can optionally choose to invert the individual *cross-components* $\mathbf{w}_{S+T,k} \otimes \mathbf{h}_{T,k}$ in order to listen to them separately. In that case, the GL algorithm is performed on each cross-component spectrogram.

4. EXAMPLES OF USAGE

Three examples of usage of the system will be briefly presented here. The corresponding sounds, and several more, can be listened to on the companion website[3]. Note that these are only a few possibilities among many others. All the sounds processed with the present, initial version of the system are mono (single-channel) and single-voiced (each source or target sound contains only one instrument).

[3] http://jjburred.com/research/icmc2013/

4.1. Note-level to resonance-level cross-synthesis

In this usage scenario, the source sound (which provides the temporality) is analyzed at the note level and the target sound (which mainly provides the timbre) is analyzed at the resonance level. The activations of the first are combined with the bases of the second. A simple example is shown in Fig. 4(a) (three-note piano source against single-note bell target). The onsets and energy evolution of the piano notes are applied to the resonances of the bell. Each piano event triggers a bell resonance, which then follows the piano's energy decay.

4.2. Note-level to resonance-level cross-synthesis

A substantially different effect is obtained by doing the inverse: analyzing the source at the resonance level and the target at the note level. The result is illustrated in Fig. 4(b), where the roles of the previously used source and target sounds have been reversed. The temporal (mostly oscillating) evolution of the resonances of the single bell strike are applied to entire piano notes. The result is an hybrid sound where each internal bell resonance has been replaced by piano-sounding oscillations, each one of which has also retained the original pitches played by the piano.

4.3. Speech to resonance-level cross-synthesis

Finally, some experiments involving speech have been performed, to further get an insight into the several important differences between this system and the classical source/filter or vocoder approaches (which are mostly used with voice). For instance, if the source is speech and the target is a musical sound rich in resonances, the individual vowels (formant structures) of the voice will match the most resembling target resonances (a related example sound is included on the webpage). This is due to the usage of MFCCs as features for the base mapping. Since the formants are matched individually, the resulting "speaking" hybrid sound will be less intelligible than hybrid sounds produced by source/filter approaches, where the global formant structure of the modulator is imposed on the carrier. The speech-like nature of the source is discernible, although in a more subtle way.

5. CONCLUSIONS

A system has been presented that uses spectrogram factorization to automatically analyze and cross-synthesize two sounds. The layer-like decomposition provided by the factorization allows to apply the internal, resonance-level temporality of one sound to the timbral structure of the other. A preliminary system has been presented, which deals with the issues of similarity-based mapping of the spectral bases and phase reconstruction from magnitude-only hybrid spectrograms. A first set of synthesis results have been presented and discussed.

The proposed framework can be subjected to evaluation under several other scenarios, depending on the nature of source and target sounds and the respective number of components chosen. A more systematic exploration of all these possibilities will be performed in the future. Also, a different choice of analysis features will obviously produce very different-sounding results.

There are also many ways in which the present system can be extended. The relatively simple factorization method used (NMF based on the Frobenius norm) could be replaced by more powerful and more recent methods from the field of source separation, such as the ones applying factorization separately on the source and filter parts. Another family of interesting methods is the one that uses temporal or spectral smoothness constraints, which could lead to more natural-sounding components.

For a practical use of the system as a composition or performance tool, an appropriate mapping of user actions to factorization parameters needs to be studied in detail. For performance, the system could be adapted to real time by the use of online matrix factorization algorithms.

6. REFERENCES

[1] D. Griffin and J. Lim, "Signal reconstruction from short-time Fourier transform magnitude." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 56 (5), pp. 236–243, 1984.

[2] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems*, Denver, USA, 2001.

[3] E. Lyon, "Spectral reanimation." in *Proc. Ninth Biennial Symposium on Arts and Technology*, New London, USA, 2003.

[4] R. Maguire, "Creating musical structure from the temporal dynamics of soundscapes," in *Proc. Int. Conf. on Information Sciences, Signal Processing and Applications (ISSPA)*, Montreal, Canada, 2012.

[5] R. Sarver and A. Klapuri, "Application of non-negative matrix factorization to signal-adaptive audio effects." in *Proc. DAFX*, Paris, France, 2011.

[6] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, 2003.

[7] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: a state of the art." in *Proc. DAFX*, Paris, France, 2011.

[8] S. Topel and M. Casey, "Elementary sources: Latent component analysis for music composition," in *Proc. ISMIR*, Miami, USA, 2011.

[9] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *Proc. International Computer Music Conference (ICMC)*, Singapore, 2003.