SOURCE SEPARATION BY SCORE SYNTHESIS

Joachim Ganseman, Paul Scheunders
IBBT-Visielab
Department of Physics, University of Antwerp
2000 Antwerp, Belgium

Gautham J. Mysore, Jonathan S. Abel
CCRMA
Department of Music, Stanford University
Stanford, California 94305, USA

ABSTRACT

A musical score provides a great deal of information about a piece of music. In this paper we consider the incorporation of a music score to guide source separation on a single channel recording. We propose a method based on synthesizing lines of music in the score. Dynamic time warping (DTW) is used to to fit the synthesized data to the recording. These are then used as prior distributions in a statistical model of the recorded sound mixture. Probabilistic Latent Component Analysis (PLCA) is then used for the source separation. Preliminary results on a Bach work for string orchestra saw good separation with few artifacts.

1. INTRODUCTION

Music scores in many file formats are becoming abundantly available, as are software synthesizers. A music score contains a lot of very detailed information about a piece of music: it tells us which note is played when, how loud, etc., for each instrument present. This makes the task of source separation much easier: from the score we know what to look for in an audio file. When trying to decompose a mixed sound into its components, we can use this knowledge to our advantage.

Source separation using data used from symbolic representations of audio (mostly MIDI) is not a new idea. A system for the separation of voice and piano using sinusoidal modeling was presented in [5]. A more general approach using spectral filtering was proposed in [2]. In this paper, we extend the separation-by-humming method described in [8], which is based on PLCA. The scores that we used in our experiments were gathered from the Mutopia database [6], which stores scores in the lilypond format. For score synthesis, we used the Timidity sound synthesizer [4] with the Fluid R3 GM soundfont, after conversion of the score to MIDI.

Using synthesized music as a guide to separate sources of a real recording has several advantages over using the symbolic data or even trying to do blind separation. The synthesized sound contains many of the characteristics of the sound that we are actually trying to extract: it is timbrally very similar, and features like onset and time-frequency envelope correspond pretty well to those of a recording. PLCA allows us to insert this knowledge as prior distributions into the decomposition algorithm, therefore immediately starting with a good guess of the result from which we can optimize further. One difficulty that arises is that a recorded piece of music will differ from the score in subtle ways. Dynamic time warping provides a straightforward and easy-to-use method to align a synthesized recording with a real recording. The result is a practically usable system for source separation of everyday music recordings, needing only a digital score of the same piece as additional input.

In [8], voice input was used as guidance for PLCA. This approach works well on small fragments of audio where only the extraction of a few seconds of a single component is necessary. On the downside, the extracted audio takes on some of the timbral characteristics of the voice input. Also, it is not very scalable: it would be really tedious to hum all parts of all instruments of a recording separately, in order to extract them. Our system is exactly meant to be used in those large-scale cases. We need the score to synthesize the sound that will guide the separation process. This process is scalable to complete databases of audio recordings and their corresponding scores.

2. PROBABILISTIC LATENT COMPONENT ANALYSIS

The main component of the complete system, which performs the source separation, is the PLCA algorithm [8]. It is an iterative method which factors a magnitude or power spectrogram into a sum of outer products of spectral and temporal components - they can be interpreted as spectral bases and their corresponding weights. PLCA interprets the spectrogram as a histogram and the spectral and temporal components as distributions along time and frequency (see fig. 1), and uses the EM-algorithm to perform the decomposition.

Groups of components which together capture the characteristics of a single instrument, can be formed according

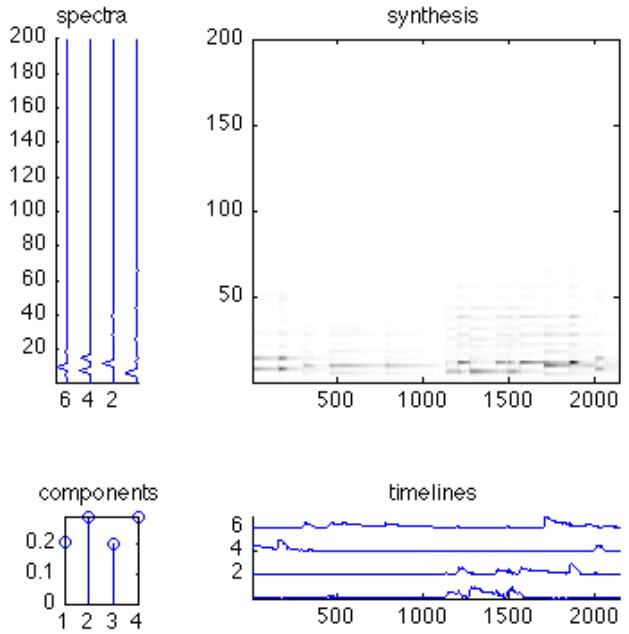

Figure 1. *PLCA*

to prior distributions, and can be used to separate that particular instrument from the mix. As a side note, it has been proven that PLCA in this 2-dimensional case is numerically equivalent to Nonnegative Matrix Factorization [7]. The advantage of PLCA lies in the fact that it assumes a probabilistic model, having parameters that can be adjusted intuitively and easily.

A very detailed description of PLCA, its applications, and some extensions and generalizations that we don't use here, can be found in [7]. We assume the magnitude spectrogram F of a recording to consist of M components. Each of these has in this particular case a frequency distribution and a time distribution, which we'll represent as $P(f|z)$ and $P(t|z)$ respectively. This makes for a 2-dimensional model. The distribution of the components $P(z)$ can be interpreted as their weight in the total mix. The model represents the spectrogram as

$$F = \sum_{z=1}^{M} P(z)P(f|z)P(t|z) \tag{1}$$

The vectors $P(f|z)$ and $P(t|z)$ are multinomial distributions, their conjugate prior distribution is a Dirichlet distribution. The prior distribution can be interpreted as an "example" that we use to bias the results towards. It is defined by a set of so-called hyperparameters, denoted here by $\alpha(f|z)$ and $\alpha(t|z)$. An EM-algorithm can be used to learn optimal $P(z)$, $P(f|z)$ and $P(t|z)$ in such a way that the priors are used as bias [8]. In the expectation step we update $P(z)$ as follows:

$$P(z|f,t) = \frac{P(z)P(f|z)P(t|z)}{\sum_{z'} P(z')P(f|z')P(t|z')} \tag{2}$$

In the maximization step, the priors $\alpha(f|z)$ and $\alpha(t|z)$ are blended in using weight factors μ_z and κ_z. We calculate updated values for $P(f|z)$, $P(t|z)$, and $P(z)$:

$$P(f|z) = \frac{\sum_t F_{f,t}P(z|f,t) + \kappa_z\alpha(f|z)}{\sum_{f'}\sum_t F_{f',t}P(z|f',t) + \kappa_z\alpha(f'|z)} \tag{3}$$

$$P(t|z) = \frac{\sum_f F_{f,t}P(z|f,t) + \mu_z\alpha(t|z)}{\sum_f\sum_{t'} F_{f,t'}P(z|f,t') + \mu_z\alpha(t'|z)} \tag{4}$$

$$P(z) = \frac{\sum_f\sum_t F_{f,t}P(z|f,t)}{\sum_{z'}\sum_f\sum_t F_{f,t}P(z'|f,t)} \tag{5}$$

3. SOURCE SEPARATION

We modify and extend the approach used in [8] and use it to extract all instruments from a polyphonic recording, by synthesizing what we wish to extract. This has several benefits compared to vocalization or otherwise mimicking the sound:

- Using a good synth, a better timbral correspondence with the target sound can be expected, thus allowing for fewer iterations of the PLCA algorithm or being more likely to converge to the optimal maximum.
- Transients in attack, release or temporal envelope are better modeled.
- Starting from the score, we force the algorithm to do a meaningful decomposition. From the score, we know how many tones occur at approximately which time and what frequency. By forcing the decomposition process to explain these tones and nothing else, we implicitly annotate the mix and the extracted sounds. It can be used for all kinds of analysis of recordings (error detection etc.)

For a large dataset to work with, we looked at the Mutopia database [6]. This database contains a few thousand music scores in the lilypond format. It is straightforward to generate MIDI data from a score, which in turn can be easily synthesized. It provides us with a ground truth for source separation tasks.

The separation process itself is conducted as follows:

- Compute the spectrogram F of a recorded mix, and the spectrograms F_1 through F_M of the synthesized instruments

- Learn from every F_m a number of components Z_m and their corresponding $P(f|z_m)$ and $P(t|z_m)$, using standard PLCA.

- Initialize the prior-based PLCA algorithm with the learned components for all active instruments, use it to estimate a mixture model of F with $\sum Z_m$ components.

- Each instrument m from the mix can now be resynthesized by only using the optimized $P(z_m)$, $P(f|z_m)$ and $P(t|z_m)$.

If necessary, extra components can be added to model parts of the sound that are not present in the score, resulting in a residual component. However, if we have all instruments accounted for, not having residual components forces the algorithm to assign as much spectral energy as possible to the instrument components - a very useful property if the recording is known to be relatively clear and correct. Parts of instruments can be extracted by synthesizing only parts of the score.

The complete process has been implemented in Matlab. In the next section we'll describe the first successful experiment which appears to validate our approach. Audio results can be reviewed online [1].

4. REAL-WORLD DATA

We work on a recording of J.S. Bach's Air (BWV1068), part of a suite for baroque orchestra, and in our recording performed by 2 violins, viola and continuo, here a cello 2. This is an inherently difficult piece to analyze, because of the violins playing notes close together and very harmonically related, and all instruments being string instruments and thus timbrally related too. Nevertheless, the separation succeeds and only really encounters troubles on moments when instruments play unisono.

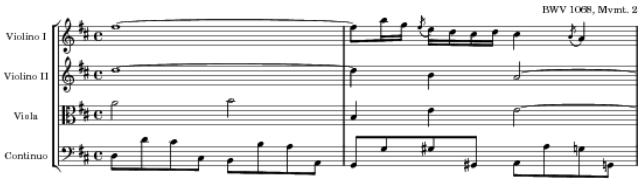

Figure 2. *J.S. Bach, "Air", BWV1068*

After synthesizing the complete score, it can be matched and time stretched to the real recording using DTW and a phase vocoder [1]. This way we obtain a one to one correspondence between spectrogram frames of the score and of the recording. If this approach is used, care must be taken to use settings for the vocoder that yield good quality results. The same alignment data can be used to match and time stretch the separately synthesized instruments that we wish to extract. On this data, we can let the PLCA algorithm do its work, first learning the parameters from the synthesized recording, then using these parameters as priors for a PLCA analysis of the real recording.

For long recordings, the spectrogram decomposition requires a lot of time and memory. It is beneficial to chop the recordings into smaller frames (a few seconds or even less long), and perform the PLCA analysis on each of these frames separately. This way, as the method tries to find optimal $P(f|z)$ and $P(t|z)$ over the portion of the spectrogram under consideration, we get more accurate and locally valid results when keeping that spectrogram's size small in time. On the downside, it is possible that this introduces hearable inconsistencies on frame borders.

Each of these frames can be handled independently, and we used frames of approximately 1 second long for the results that we present here. Other parameters used were: 25 components per instrument, 50 iterations of the PLCA algorithm, an FFT size of 2048 and 75% overlap. The latter overlap was necessary for the DTW and subsequent resizing using a phase vocoder [1] to work correctly, and we kept it for the PLCA algorithm. A binary mask was used on the output of the algorithm as this improved the separation performance. For resynthesis, we gather all components of the source we want to resynthesize, make a spectrogram out of it using to equation 1, and resynthesize that spectrogram. The method works on magnitude spectrograms; for the phase we copy the phase of the mix to each of the extracted sources. The results can be seen in figure 3.

5. FUTURE WORK

Future work consists of, amongst other things, studying the different parameters and how they influence the analysis, and objectively evaluating the results using a framework like BSS_EVAL [3]. The latter was not possible yet during these preliminary experiments since it requires access to the separate tracks of the real recording to compare the separation results with the original, which we did not have available. We may further investigate incorporating phase information (here we only work on the magnitude spectrum), applying sparsity constraints and further developments of the PLCA model as described in [7], or using one separated recording to separate another. Also, instead of stretching the spectrum of the score to match the spectrum of the mix, we are likely better off using an alignment method that directly matches the score data to the mix, only then to synthesize the score.

[1] https://ccrma.stanford.edu/~jga/icmc2010/icmc2010.html

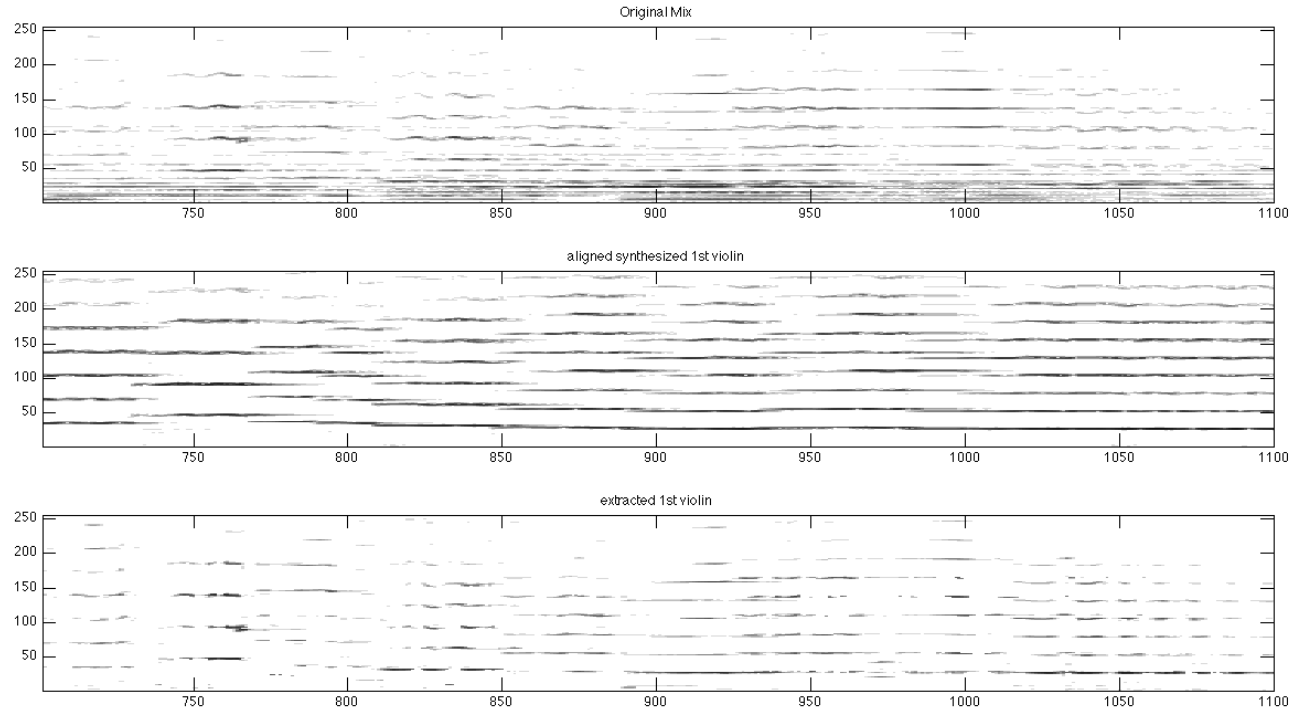

Figure 3. *Extraction of an instrument from a mix, corresponding to approximately the first 3 beats of the second measure of figure 2. On the x-axis are time frames, on the y-axis frequency bins, calculated using a 2048-point DFT with 75% overlap.*

6. CONCLUSION

In this paper we have presented a robust and scalable method for source separation using synthesized scores of the music under consideration as prior input to guide the analysis. The method can be used to extract single or multiple instruments from a recording and resynthesize them separately. The method does not require any human input, aside from setting a few parameters, and is thus readily usable on a large scale. The results are accurate and meaningful, and the first experiments on real recordings were successful and hold promise for the future .

7. ACKNOWLEDGEMENTS

This work has been partially funded by IWT-Flanders and FWO-Flanders.

8. REFERENCES

[1] D. Ellis, "Dynamic time warp (dtw) in matlab," accessed December 22, 2009, available at http://www.ee.columbia.edu/ dpwe/resources/matlab/dtw/.

[2] M. R. Every and J. E. Szymanski, "A spectral-filtering approach to music signal separation," in *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004, pp. 197–200.

[3] C. Févotte, R. Gribonval, and E. Vincent, "BSS_EVAL, a toolbox for performance measurement in (blind) source separation," accessed December 22, 2009, available at http://bass-db.gforge.inria.fr/bsseval/.

[4] M. Izumo and contributors, "Timidity++," accessed December 22, 2009, available at http://timidity.sourceforge.net.

[5] Y. Meron and K. Hirose, "Separation of singing and piano sounds," in *Proc. of the 5th International Conference on Spoken Language Processsing (ICSLP'98)*, Sydney, Australia, 1998, pp. 1059–1062.

[6] C. Sawer and D. Chan, "Mutopia," accessed December 22, 2009, available at http://www.mutopiaproject.org/.

[7] M. Shashanka, B. Raj, and P. Smaragdis, "Probabilistic latent variable models as non-negative factorizations." in *Computational Intelligence and Neuroscience Journal, special issue on Advances in Non-negative Matrix and Tensor Factorization*, May 2008.

[8] P. Smaragdis and G. Mysore, "Separation by humming: User-guided sound extraction from monophonic mixtures." in *Proc. of IEEE Workshop on Applications Signal Processing to Audio and Acoustics*, New Paltz, NY, October 2009.