EVALUATION OF A SCORE-INFORMED SOURCE SEPARATION SYSTEM

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

In this work, we investigate a method for score-informed source separation using Probabilistic Latent Component Analysis (PLCA). We present extensive test results that give an indication of the performance of the method, its strengths and weaknesses. For this purpose, we created a test database that has been made available to the public, in order to encourage comparisons with alternative methods.

1. INTRODUCTION

Source separation is a difficult problem that has been a topic of research for several decades. It is desirable to make use of any available information about the problem to constrain it in a meaningful way. Musical scores provide a great deal of information about a piece of music. We therefore use this information to guide a source separation algorithm based on PLCA.

PLCA [1] is a technique that is used to decompose magnitude spectrograms into a sum of outer products of spectral and temporal components. It is a statistical interpretation of Non-Negative Matrix Factorization (NMF) [2]. The statistical framework allows for a structured approach to incorporate prior distributions.

Extraction of a single source out of a sound mixture by modeling a user guidance as a prior distribution was presented in [3]. In our previous work [4], we based ourselves on that approach and extended it to a complete source separation system informed by musical scores, finally demonstrating it by separating sources in a single real-world recording.

We perform source separation by decomposing the spectrogram of a given sound mixture using PLCA, and then performing reconstructions of groups of components that correspond to a single source. Before using PLCA on the sound mixture, we first decompose synthesized versions of those parts of musical scores that correspond to the sources that we wish to separate (also using PLCA). The temporal and spectral components obtained by these decompositions of synthesized sounds are then used as prior distributions while decomposing the real sound mixture.

In this work we make a detailed evaluation of such source separation system and its overall performance. To the best of our knowledge, a comprehensive and extensive dataset to use as ground truth for such problem does not exist, mainly because we also require the corresponding scores as additional information to the source separation system. We therefore construct a test set of our own, mimicking realistic conditions as well as possible even though it is synthetic. This also allows us to make detailed evaluations of how the results are affected by common performance practices, like changes in tempo or synchronization. To get objective quality measurements of this method, we use the metrics defined in the BSS_EVAL framework [5], which are widely adopted in related literature.

2. SCORE-INFORMED SOURCE SEPARATION WITH PLCA

We're not the first to propose source separation based on score information. A method based on sinusoidal modeling has been proposed by Li [6], and Woodruff [7] used scores as information source for the separation of stereo recordings. Our PLCA-based system for score-informed source separation is set up as shown in fig. 1 :

- The complete score gets synthesized;
- Dynamic Time Warping (DTW) matches the spectrogram of the sound mixture to that of the score;
- The resulting path is used to match single parts or sections from the score to the mix;
- Components for each of the parts to extract are learned using PLCA on separately synthesized parts;
- These components are used as prior distributions in the subsequent PLCA decomposition of the mix;
- With the learned components 'fitted' to the mix, we can now resynthesize only those components from the mix that we want.

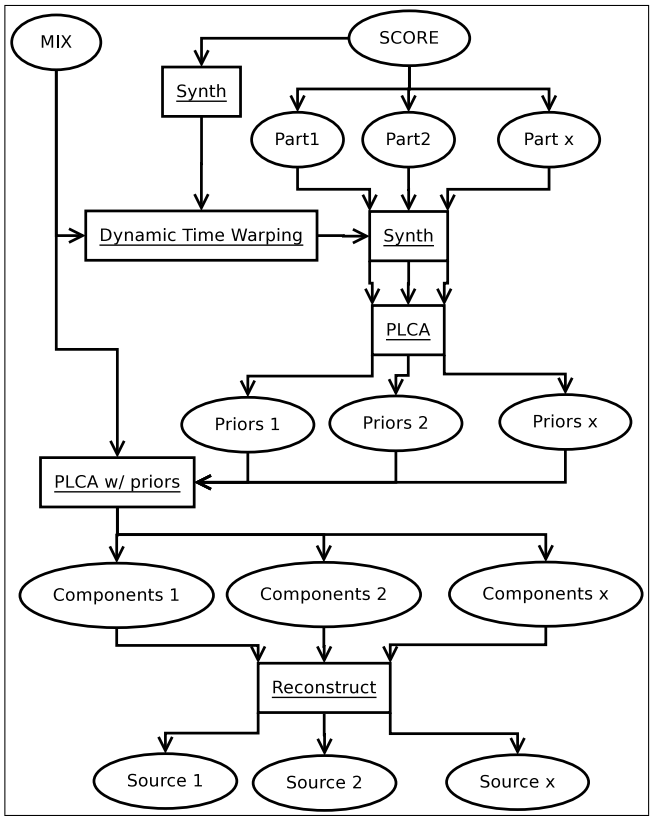

Figure 1. Architecture of the score-informed PLCA-based source separation system

The PLCA method that we adopt does not presuppose any structure, instead it learns the best representation for a spectrogram through an EM (expectation-maximization) algorithm. Both temporal and spectral components can assume any shape. The dictionary of spectral and temporal components resulting from decomposition of the synthesized score parts is only used to initialize the subsequent PLCA decomposition of the sound mixture. The EM-iterations decomposing this mixture optimize those spectral and temporal components further in order to make them explain the sources in the sound mixture. A drawback of PLCA is that it operates on magnitude spectrograms and does not take into account the phase, which easily leads to some audible distortion in the resynthesized sounds.

We implemented our system largely in Matlab, with the DTW routine provided from [8]. We always work on mono audio. The method does not work in real-time - on a modern dual-core 3.0GHz computer with 4GB RAM memory, processing a 1 minute sound file (44100Hz samplerate) takes about 3 to 4 minutes of calculation time with high quality settings. The DTW subroutine has a memory complexity that is quadratic in spectrogram size, due to the calculation of a complete similarity matrix between the spectrograms of the sound mixture and the synthesized score. Alternatives to DTW exist and could be used, there is e.g. prior work on aligning MIDI with audio without computing a complete rendering of the MIDI file (also available through [8]).

It needs to be noted that the spectral and temporal components of the synthesized score parts are initialized with random data. Starting from these random probability distributions, the EM-algorithm then iteratively estimates better candidates that fit the data. The resulting estimates of the components from the score data will be slightly different on each run. This will in turn affect the subsequent PLCA analysis of the real data and its path towards convergence. We will quantify this in more detail later, but it is important to keep in mind that all measurements presented in this paper are subject to a certain error margin that is a direct result of this random initialization.

3. TEST SETUP

In order to do large scale comprehensive testing of this method, we need a database of real sources and their scores which we can mix together and then try to separate. To the best of our knowledge, a carefully crafted database for research purposes containing separate sources and their scores for a wide range of instruments and/or styles does not yet exist [9]. For source separation, evaluation databases with multitrack recordings are available (e.g. [10]), but usually they don't come with scores, MIDI files, or any other symbolic information.

We decided to create our own database, generating short random MIDI tunes, and then running them through different synthesizers. In testing, one of the synthesized sounds then can take on on the role of 'real performance', while the other is used as 'synthesized score'. To better simulate real performance, we generated several versions of each file with tempos regularly changing, up to half or double the speed of the original. This also allows the database to be used to test alignment algorithms. The resulting dataset is available online [1].

We generated a set of 10 second sound files using PortSMF [11]. The files were synthesized once using Timidity++ [12] with the FluidR3 GM soundfont on Linux, and once with the standard built-in MIDI player capabilities in Windows XP (DirectSound, saved to file using WinAmp [13]). Each file contains on average 20 note onsets, spread randomly over 10 seconds. We reduced our test set to 20 commonly used instruments, both acoustic and electric. This was done in part because of a lot of the sounds standardized in General MIDI are rarely found in scores (helicopter sounds, gunshots), and to keep the size of the resulting data manageable. With 380 possible duos with different instruments out of 20 instruments, it allowed us to run repeated experiments on all of these combinations.

This original test set of 20 sounds was expanded by introducing timing variations in the MIDI files. Several sets of related files were generated, in which the tempo in each

[1] https://ccrma.stanford.edu/~jga/ismir2010/ismir2010.html

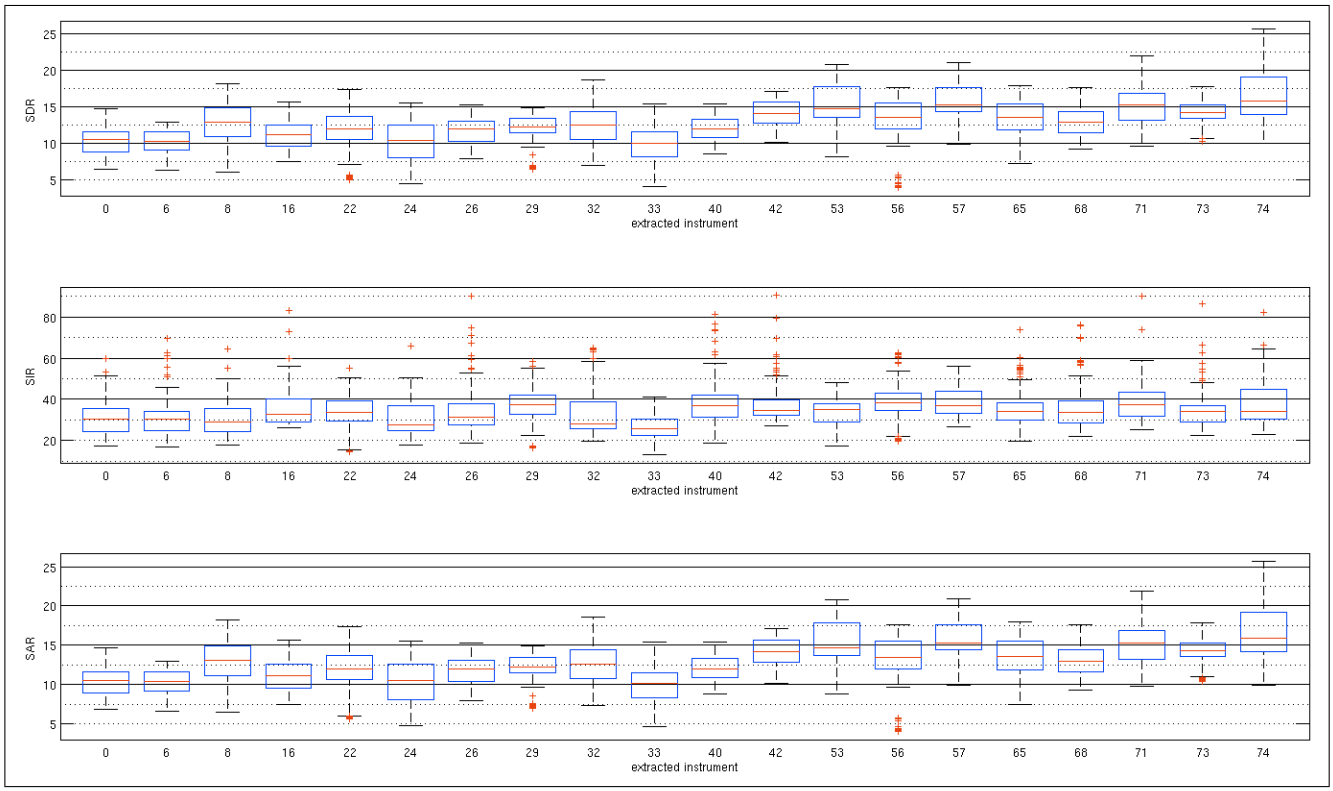

Figure 2. Overall source extraction scores per instrument, mixed with any other instrument. On the x-axis, the MIDI Program Change number. In this and all other figures, standard Matlab style boxplots are used.

file was changed 5 times - this to be able to test the effects of the method used to align symbolic data and recordings, which is part of the system. 2 distinctly different tempo curves were defined, and for each of these 2 curves, 5 new renditions were made for every original source. The first of these 5 would have the tempo changed by up to 10%, either slower and faster, while the last would allow deviations from the original tempo up to 50% . Thus we have a dataset of 20 original sources, and for each original file also 10 files with all different variations in tempo.

We acknowledge that there are a couple of drawbacks with this dataset. The first one that the files are randomly generated, while in most popular and classical music, harmonic structure makes separation more difficult due to overlapping harmonic components. The second that even using two different soundbanks to synthesize can result in the two synthesized versions of a single file to be more similar to eachother than they might be similar to a real recording. We found however that the timbres of the two soundbanks used differ quite significantly. As for the random generation: not using real data frees us from dealing with copyright issues, and generating it randomly allowed us to quickly obtain a large and comprehensive body of test files, not presupposing any structure or style.

In the following sections, we use the files generated on Windows as sources for the 'performance' sound mixture, and the files rendered on Linux as 'scores' from which we obtain priors. The BSS_EVAL toolbox [5] calculates 3 metrics on the separated sources given the original data. The Signal-to-Interference Ratio (SIR) measures the inclusion of unwanted other sources in an extracted source; the Signal-to-Artefacts Ratio (SAR) measures artefacts like musical noise; the Signal-to-Distortion Ratio (SDR) measures both the interference and artefacts.

4. MEASUREMENTS ON IDEAL DATA

4.1 Error margin on the results

As mentioned previously, due to randomness in the initialization, separation results might differ with every run, and so might the SDR, SIR and SAR scores. To properly quantify what we are dealing with, we ran the system with standard parameters that give decent results (sampling rate of 44100Hz, 2048-point FFT with 75% overlap, 50 components per source, 50 iterations) 10 times on each of the possible 380 instrument duos in the test set. This was done

	min std	max std	mean std	median std
SDR	0.062	1.55	0.48	0.41
SIR	0.056	14.21	1.89	1.20
SAR	0.061	1.55	0.47	0.41

Table 1. Reliability of the results: statistics on the standard deviation of SDR, SIR and SAR scores of 10 runs of the algorithm with the same parameters on 380 data pairs.

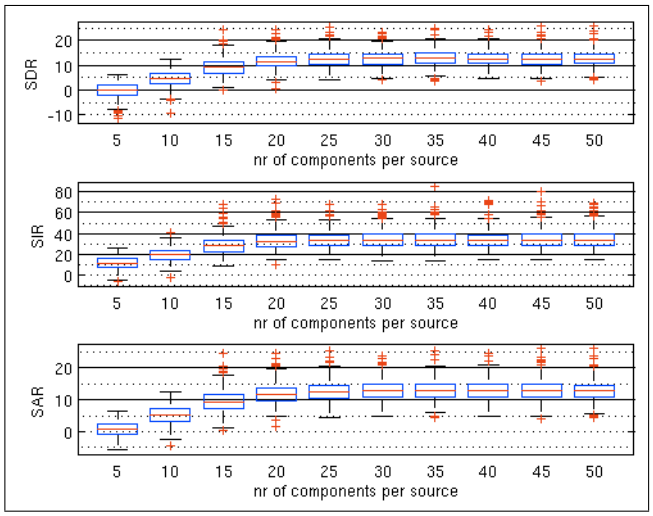

Figure 3. SDR, SIR and SAR vs. number of components used per source

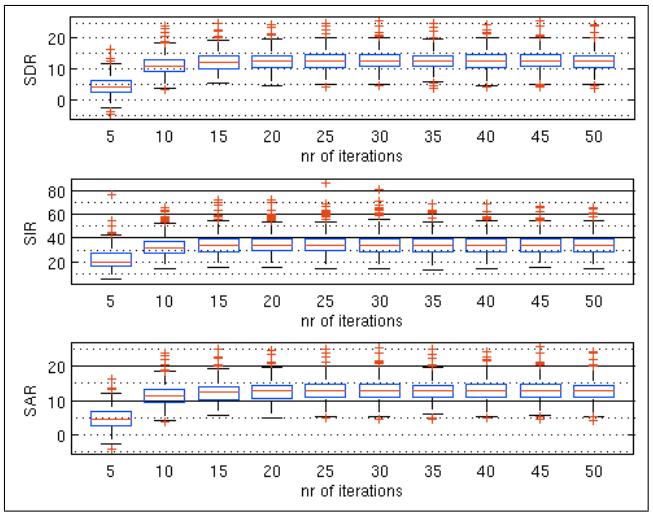

Figure 4. SDR, SIR and SAR vs. number of iterations in EM algorithm

on 'ideal' data, where the score would exactly line up with the sound mixture and no DTW was needed, so we compute the effect of the random initialization only. SDR, SIR and SAR values tend to be pretty consistent in between runs on the same pair, except in a few rare combinations where there is a lot of variance in the results. Even in these cases, the mean values of the scores are still within normal range.

The numbers in table 1 show that the mean standard deviation of calculated SDR and SAR scores stays below 0.5 dB, while there can be highly variable results in some SIR scores. Incidences of high variance in SIR score seem unrelated to each other, and almost every instrument had some combination with another instrument where SIR scores would be very variable in between runs of the algorithm. For evaluation purposes, SDR and SAR scores seem to be better suited to pay attention to.

Some instruments seem to be easier to extract from mixes with any other instrument, than others. Fig. 2 gives an idea of the 'overall easiness' with which an instrument can be extracted from a mix.

4.2 Components and iterations

The algorithm's running time will increase linearly with the amount of components. Generally, increasing the number of components that are available per source increases the ability to model the priors accurately, and thus also the overall separation results. We ran a small test of the effect of the number of components across all couples of sources. The effectiveness of the number of components is almost identical for every instrument, so we can generally plot the number of components versus the outcome of the metrics, which is shown in fig 3.

With on average 20 notes in each source, there is a huge climb in improvements up to 20 components after which the scores level off. There is some small improvement after this, but not drastic. We haven't run complete tests with significantly larger amounts of components, but from a couple of single tries we find that overfitting becomes an issue when the amount of components is chosen too large. Superfluous components of a single source risk to start modeling parts of other sources, which degrades separation performance again.

The number of iterations of the EM algorithm does not suffer from this - since the likelihood of subsequent EM iterations is monotonically increasing, more is always better. The only constraint here is how much time we're willing to spend on those iterations. We can see that the convergence towards a good solution is obtained rather fast: independent of instrument, above 25 iterations there is hardly any improvement of the scores (fig. 4).

4.3 Other parameters

The PLCA routine decomposes a magnitude spectrogram, and thus the properties of that spectrogram also play a role in the end result. Conducting a few small tests, we were able to conclude that the larger the FFT size, the better the results generally are. In subsequent tests, we used 2048-point FFTs. The overlap should be kept above 67.5% ; 75% is a safe value. Binary masking (assigning each spectrogram time-frequency bin to a single source instead of dividing it among different sources) significantly improves SIR scores, at the cost of a slight decrease in SDR and SAR scores.

It is possible to cut the spectrogram into 'timeslices' of variable length. Certainly when there are possibilities for parallelization, or when due to system limitations the spectrogram size needs to be kept to a minimum, it might be interesting to run the analysis on each slice separately.

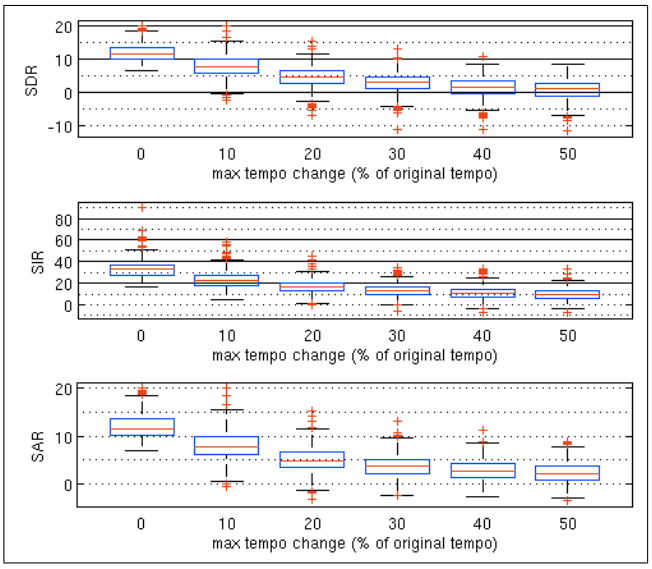

Figure 5. SDR, SIR and SAR vs. tempo deviation from reference, all sources with the same deviation

This makes that the spectral components, and their temporal counterparts, can change from slice to slice. Due to the random initialization of the components, they are likely not related to the components in other slices at all, and each slice will have its components defined such that they represent the data in that slice optimally. During resynthesis, small artefacts can be introduced on the slice borders due to these changes in basis vectors that occur. Our tests indicated that, though a decline in scores remains small and only noticeable when slices were smaller than a second long, it is a good idea to have the length of a slice as large as possible. How this relates to the number of needed components or iterations remains to be studied in the future.

5. THE EFFECT OF DYNAMIC TIME WARPING

5.1 Quantifying the role of DTW

Whereas in the previous section we discussed metrics on ideally aligned data, this is not likely to occur in real life. Performers use their artistic freedom to make the notes on paper into a compelling concert. One of the main means of doing so are local changes in tempo. To cope with this, a DTW routine is attached at the beginning of the system. It serves to line up the score with the real performance.

In figure 5 the performance of the algorithm on ideally aligned sound mixtures (0% deviation from the score tempo) is compared to performance on mixtures with tempo deviations, where alignment is needed. The sources that were used were divided into 5 segments that each had a different tempo assigned to them, in such a way that the tempo was in every file partly below the reference tempo, and partly above. The amount of change has a pretty high influence on the effectiveness of the subsequent source separation.

Logically, the timing of an entire score applies to the individual instrument parts too. We provide the output from the DTW routine to a phase vocoder that dilates or compresses each of the synthesized parts in time, so that they match up with the the performance mixture. This is a quick and practical solution to make sure that in the following PLCA analysis steps, the temporal and spectral components of the performance mixture and their associated priors obtained from the synthesized score parts, have the same dimensions.

Both errors in alignment and the subsequent stretching of synthesized score parts introduce errors in the priors, which affect successful analysis. From the data in fig. 5, we conclude that heavy time warping and subsequent stretching of the spectrum puts the quality of the results at severe risk. The DTW routine and phase vocoder that we used [8] were chosen because they were readily available to plug into our code. It is however a bottleneck in our system. In future work, alternative methods to align scores with recordings are worth looking into [14]. If using DTW, in practical applications the possibility to manually correct or at least smoothen the time alignment should be available.

In tests where source files with different tempo curves were used in a single sound mixture (in order to simulate performers that are out of sync with each other), very similar results were observed. In such a case the time alignment is likely to contains errors for at least one of the sources, since notes that should be played together according to the score, are not necessarily played together in the mixture. We can conclude that the application of DTW and subsequent time dilating and compressing of the synthesized data with a phase vocoder can cause a considerable stir in the computed priors, to such an extent that in the subsequent decomposition of the mix, it becomes very difficult to get decent separation results.

5.2 Adaptations and alternatives

The DTW plus phase vocoder routine is the weak link in the complete process, and we ventured on to do a couple of experiments adapting that part of our system. Inspired by recent work by Dannenberg et al [15] we substituted the spectrograms used in the DTW routine by chromagrams, using code obtained from the same source [8]. The results are practically equal to those in fig. 5 . Just like in the case of DTW with spectrograms, some (manual) post-processing on the results of the DTW routine is likely to improve the test results.

We also undertook a small experiment skipping the use of a phase vocoder to stretch the spectrograms of the scores, instead only resampling the temporal vectors using piecewise cubic Hermite polynomials to maintain nonnegativity. It turns out that the mean SDR and SAR scores plummet, and the standard deviation increases drastically, resulting in a small but not negligible number of test results

that are actually better than what could be attained previously. Also, the SIR values stay remarkably high, and even at large tempo deviations. However, overall the system became highly unreliable and unfit for general use.

Given that the parameters of the PLCA routine can be chosen optimally and that their effects are relatively well-known, most of the future effort in improving this score-informed separation system should clearly go into better and more accurate alignment and matching of the scores to the real performance data. Also more varied data and use cases need to be considered - here, we only worked on mixes of 2 instruments, and did not include common performance errors like wrongly struck notes. Several approaches to solve this problem, or parts of it, exist or are being worked on [14], and can contribute to a solution.

For alignment of scores with recordings, we have some future work set out, replacing the DTW and phase vocoder with methods more fit for our particular setup. In hindsight, with symbolic data and performance recordings available, we would very likely be better off applying a method that directly aligns symbolic information with a single spectrogram, to then modify the timing of the symbolic data, only then to synthesize it and compute priors from. For any future developments, we now have an extensive dataset to quickly evaluate the system.

6. CONCLUSIONS

In this paper we quantified the performance of a recently developed score-informed source separation framework based on PLCA. Several parameter options were explored and we paid special attention to the effect of DTW. The use of metrics that are prevalent in literature allows for future comparison with competing methods. We synthesized our own test dataset covering a wide range of instruments, using different synthesizers to mimick the difference between real-world data and scores, and mimicking some performance characteristics by introducing tempo changes. This dataset has been made freely available to the general public, and we exemplified its usability for extensive testing of alignment and source separation algorithms.

7. REFERENCES

[1] P. Smaragdis, B. Raj and M.V. Shashanka: "Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures," *Proc. of the 7th International Conference on Independent Component Analysis and Signal Separation*, London, UK, September 2007.

[2] D. Lee and H.S. Seung: "Algorithms for Non-negative Matrix Factorization," *Proc. of the 2000 Conference on Advances in Neural Information Processing Systems*, MIT Press. pp. 556562.

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

[4] J. Ganseman, G. Mysore, P. Scheunders and J. Abel: "Source separation by score synthesis," *Proc. of the International Computer Music Conference*, New York, NY, June 2010.

[5] C. Févotte, R. Gribonval and E. Vincent: "BSS_EVAL, A toolbox for performance measurement in (blind) source separation," *Available at http://bass-db.gforge.inria.fr/bss_eval/*, accessed May 27, 2010.

[6] Y. Li, J. Woodruff and D. L. Wang: "Monaural musical sound separation using pitch and common amplitude modulation," *IEEE Trans. Audio, Speech and Language Processing*, vol. 17, no. 7, pp. 1361-1371, 2009.

[7] J. Woodruff, B. Pardo and R. B. Dannenberg: "Remixing Stereo Music with Score-informed Source Separation," *Proc. of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, October 2006.

[8] D. Ellis: "Matlab audio processing examples," *Available at http://labrosa.ee.columbia.edu/matlab/* accessed May 27, 2010.

[9] A. Grecu: "Challenges in Evaluating Musical Instrument Sound Separation Algorithms," *Proc. 9th International Student Workshop on Data Analysis (WDA2009)*, Certovica, Slovakia, July 2009, pp. 3-9.

[10] E. Vincent, R. Gribonval, C. Févotte et al,. "BASS-dB: the Blind Audio Source Separation evaluation database." *Available at http://www.irisa.fr/metiss/BASS-dB/* , accessed May 27, 2010.

[11] R. B. Dannenberg and contributors: "PortSMF, part of PortMedia" *Available at http://portmedia .sourceforge.net/*, accessed May 27, 2010.

[12] Masanao Izumo and contributors: "Timidity++," *Available at http://timidity.sourceforge.net* , accessed May 27, 2010.

[13] NullSoft, Inc: "Winamp," *Available at http://www.winamp.com*, accessed May 27, 2010.

[14] R. B. Dannenberg and C. Raphael: "Music Score Alignment and Computer Accompaniment," *Communications of the ACM*, vol. 49, no. 8 (August 2006), pp. 38-43.

[15] R. B. Dannenberg and G. S. Williams: "Audio-to-Score Alignment In the Audacity Audio Editor," *Late Breaking Demo session, 9th International Conference on Music Information Retrieval*, Philadelphia, USA, September 2008.