Score-Informed Audio Decomposition and Applications

Jonathan Driedger
International Audio Laboratories Erlangen*
jonathan.driedger@audiolabs-erlangen.de

Harald Grohganz
Bonn University
grohganz@cs.uni-bonn.de

Thomas Prätzlich
International Audio Laboratories Erlangen
thomas.praetzlich@audiolabs-erlangen.de

Sebastian Ewert
Queen Mary University of London
sebastian.ewert@eecs.qmul.ac.uk

Meinard Müller
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

ABSTRACT

The separation of different sound sources from polyphonic music recordings constitutes a complex task since one has to account for different musical and acoustical aspects. In the last years, various score-informed procedures have been suggested where musical cues such as pitch, timing, and track information are used to support the source separation process. In this paper, we discuss a framework for decomposing a given music recording into note-wise audio events which serve as elementary building blocks. In particular, we introduce an interface that employs the additional score information to provide a natural way for a user to interact with these audio events. By simply selecting arbitrary note groups within the score a user can access, modify, or analyze corresponding events in a given audio recording. In this way, our framework not only opens up new ways for audio editing applications, but also serves as a valuable tool for evaluating and better understanding the results of source separation algorithms.

Categories and Subject Descriptors

H.4 [**Information Interfaces and Presentation**]: Sound and Music Computing

Keywords

Score-informed processing, source separation, audio editing, alignment, music synchronization

1. INTRODUCTION

In recent years, the task of separating a mixture of superimposed sound sources into its constituent components has been a central research topic in the field of digital signal processing. In speech, for example, these components could be the individual conversations of simultaneously speaking persons ("Cocktail party scenario", see [1]). In the context of music, the sources might correspond to the main melody, a bassline, a drum track or another instrument track [5, 6, 13, 14, 15]. To guide the source separation process in such a scenario, it has become a common strategy to provide the algorithm with additional information. Such information can, for example, be given in the form of user-specified annotations [2], or by a musical score. In *score-informed* procedures the explicit timing, pitch and instrument information encoded by the score is utilized to guide and support the source separation processes.

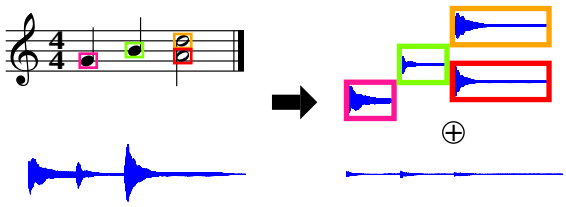

Figure 1: Score-informed decomposition of a given audio recording into note-wise audio events and a residual signal.

Most current score-informed approaches are designed for extracting individual instruments as specified by the score, see [8, 10, 18]. In this paper, we go beyond this scenario by introducing a framework for decomposing a music recording into elementary building blocks or sound events. More precisely, a musical score can be considered as a composition of elementary events given by the individual musical notes. These notes have some explicit musical meaning (in terms of pitch, onset time, and duration) and are directly intelligible by a human. The core idea of this contribution is to decompose a given music recording into a set of note-wise audio events, where each audio event is directly associated with a note in the musical score, see Figure 1. Based on this decomposition, we introduce an intuitive interface that allows a user to directly access the audio recording in a note-wise fashion, which opens up explicit ways of editing and manipulating audio material. Such an interface also provides novel possibilities to better understand the quality achieved by the underlying source separation algorithm. For example, subtracting all note-wise audio events from the original recording yields a *residual signal* which can be interpreted as the part of the recording that was not captured by the source separation process (for example because it was not reflected by the given musical score). Analyzing this residual can then reveal parts in the original recording where the source separation algorithm typically fails or where data inconsistencies occur.

The remainder of this contribution is structured as follows. In Section 2 we summarize a recent score-informed source separation algorithm used in our experiments. Furthermore, we show how to derive the note-wise decomposition of the audio recording and discuss some manipulation strategies. In Section 3, we present a

*The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer - Institut für Integrierte Schaltungen (IIS).

MM'13 Barcelona, Catalunya, Spain

http://dx.doi.org/10.1145/2502081.2502143.

prototype of a user interface for intuitive score-based audio editing and analysis. Finally, in Section 4, we close this paper with conclusions and future work.

2. AUDIO DECOMPOSITION

In the last years, techniques based on non-negative matrix factorization (NMF) have been applied to decompose a magnitude spectrogram into a set of template (column) vectors and activation (row) vectors [16]. To better control this factorization, additional score information has been used to constrain NMF and to yield a musically more meaningful decomposition [7]. In this section, we summarize the score-informed procedure as introduced in [3] (Section 2.1) and then describe how to decompose a given audio recording x into note-wise audio events x_m, $m \in [1:M]$, where M is the number of note events specified in the score, and a residual signal r such that $x = \sum_m x_m + r$ (Section 2.2).

2.1 Constrained NMF-based Source Separation

Given a matrix $V \in \mathbb{R}_{\geq 0}^{S \times T}$, the goal of classical NMF is to derive two matrices $W \in \mathbb{R}_{\geq 0}^{S \times K}$ and $H \in \mathbb{R}_{\geq 0}^{K \times T}$, such that the distance, typically a modified Kullback-Leibler divergence, between V and WH is minimized [12]. In the context of source separation, given an audio recording x with spectrogram X, the goal is to factor the magnitude spectrogram $V := |X|$ into a matrix W of *template vectors* (every column corresponding to the prototype spectrum of a certain tone) and a matrix H of *activations* (every row encoding when and how loud a corresponding tone is played). In standard NMF the matrices W and H are derived by iteratively updating two randomly initialized matrices using multiplicative update rules. However, the result of this process is often not musically meaningful as discussed for example in [3].

To overcome this issue, [3] proposed a score-informed approach, where the score information is used to initialize both matrices W and H to guide the NMF update process in a musically meaningful direction. More precisely, having the score of the audio recording at hand in form of a MIDI file, high-resolution synchronization techniques are used to temporally align the MIDI events with the audio recording [4, 11]. Each of the M note events of the synchronized MIDI file yields information about the pitch, the onset time and the duration of a corresponding audio event that should occur in the recording according to the score. For each occurring pitch in the MIDI file, a harmonic template (column of the matrix W), which encodes the rough harmonic structure of the pitch, is initialized. This template is defined to have non-zero entries at frequency bins that are related to the fundamental frequency and the overtones of the given pitch and zero entries otherwise. Similarly, the activation matrix H is initialized by the specified onset times and note durations obtained from the synchronization procedure. For later use, we link the initialization of H with the corresponding synchronized MIDI events as follows. A binary constraint matrix $C_m \in \mathbb{R}^{K \times T}$ is constructed for each $m \in [1:M]$, where C_m is 1 at entries that correspond to the pitch and temporal position of the m^{th} MIDI event and 0 otherwise. Each C_m therefore constitutes a link between specific entries in H and a MIDI note event. The union (OR-sum) of all C_m is then used as initialization of H. At this point, the crucial observation is that the multiplicative NMF update rules can only change the non-zero entries. Therefore, applying NMF to the initialized matrices W and H yields a decomposition, where the relations expressed by the C_m between MIDI note events and entries in the activation matrix H are preserved, see Figure 2a-d. The result after the NMF-learning procedure can be seen as a refinement of the initially constrained harmonic template and activation matrices.

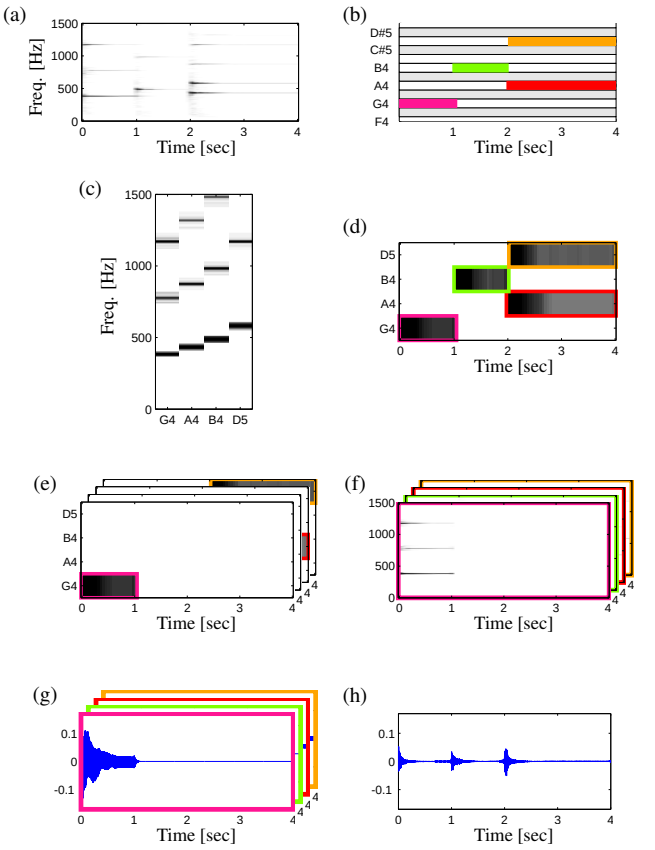

Figure 2: (a) Magnitude spectrogram V. (b) Synchronized MIDI note events. (c) Template matrix W learned by NMF. (d) Activation matrix H learned by NMF. (e) Note-wise activation matrices H_m. (f) Note-wise spectrograms X_m. (g) Note-wise audio events x_m. (h) Residual signal r.

2.2 Note-Based Audio Decomposition

Let W and H denote the template and activation matrices after applying the NMF learning procedure. We now use the note-wise constraints given by the matrices C_m, $m \in [1:M]$, to derive the note-wise audio events x_m. To this end, we first compute a note-wise activation matrix $H_m := H \odot C_m$, where the operator $\odot$ denotes the point-wise multiplication. Afterwards, we derive a spectral mask $M_m := (WH_m) \oslash (\sum_m WH_m + \epsilon)$, where $\oslash$ is understood as point-wise division and ϵ is a small positive constant to avoid a potential division by zero. The mask M_m can be interpreted as a weighting matrix that reflects the contribution of the m^{th} note event to the original spectrogram X. Finally, we compute the note-wise spectrogram $X_m := X \odot M_m$ and apply the inverse short-time Fourier transform to obtain the audio event x_m, see Figure 2e-h. The audio events x_m represent a decomposition of the original signal x (the music recording) according to the note events specified by the given musical score. Obviously, this decomposition becomes problematic in the case that the recorded performance deviates from the musical score. More generally, synchronization inaccuracies, i. e. deviations in the alignment of the MIDI events and their expected realization in the music recording, may lead to local errors in the decomposition. Furthermore, simplifying model

assumptions (such as the assumption that the partials' relative energy distribution is independent of the loudness), deviations in the expected tuning, or additional sound components caused by resonance or reverberation may cause artifacts in the decomposition. Therefore, we also compute a residual signal $r = x - \sum_m x_m$. The signal r holds a lot of valuable information since it does not only give a deeper insight into the source separation process, but it may also reveal inconsistencies between the musical score and the audio recording. Therefore, it is a natural idea to analyze r in more detail. In the next section we present an interface that supports such an analysis by enabling the user to study the decomposition and the residual signal more thoroughly.

3. APPLICATIONS

In this section, we show how the decomposition of an audio recording can be utilized as a basis for various applications. To this end, we developed a user interface (Section 3.1) which offers a user-friendly access to such a decomposition[1]. Furthermore, we show how this interface can be used for intuitive audio editing (Section 3.2) and the analysis of the underlying source separation procedure by investigating the residual signal r (Section 3.3).

3.1 User Interface

Besides standard audio player functionalities, our interface comes with a set of additional tools related to the score-informed audio decomposition introduced in the previous section. In its current state, our interface provides a piano roll representation of the musical score. While playing back the audio recording, the synchronized MIDI events are displayed and allow a user to directly access every single corresponding note-wise audio event, see Figure 3. Additionally, the interface also provides a set of plugins that can be used to manipulate, analyze and visualize the note-wise audio events. Some plugins included in our interface are discussed next. In general our Matlab-based framework is not restricted to these use cases and is flexible enough to support a wide range of further applications.

3.2 Audio Editing

Our interface provides easy-to-use possibilities for manipulating the audio recording in a musically informed manner. For example dragging a MIDI event in the piano roll representation and dropping it at a different position is an intuitive way of changing the onset time (horizontal displacement) and the pitch (vertical displacement) of a note. Having the note-wise audio decomposition at hand, we are able to transfer the same manipulations to the audio recording. If the user wishes to displace a note with respect to time and pitch, the corresponding audio event is first subtracted from the original recording and a suitably time- and pitch-shifted version (using a standard pitch shifting procedure such as [17]) of the event is added afterwards. We keep track of all the applied manipulations such that it is possible to manipulate a previously edited note again. By using similar strategies it is also possible to change the duration or the volume level of notes, to remove notes completely from the audio recording, or to add additional notes by copying and manipulating existing ones.

3.3 Source Separation Analysis

Analyzing the decomposition of a music recording, and especially the residual signal r offers novel possibilities to investigate

[1]A demo of the proposed interface can be found at `http://www.audiolabs-erlangen.de/resources/2013-ACMMM-AudioDecomp/`.

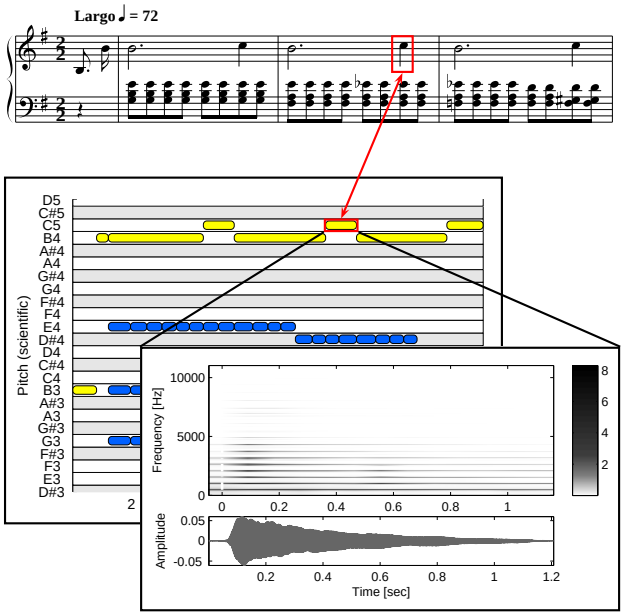

Figure 3: Top: Score of the first three measures of Op. 28 No. 4 by Frédéric Chopin. Bottom: Our user interface showing the corresponding part in an audio recording of the piece. Each note-wise audio event can be accessed separately.

the behavior of the underlying source separation algorithm. Positions in the original audio recording where r shows high energy indicate passages where the source separation procedure could not assign all of the recording's energy to the note-wise audio events. To analyze such positions, our interface has been equipped with a plugin that plots the color-coded short-time energy of r in the background of the standard visualization, see Figure 4. This way, one can directly observe temporal relations between bursts of energy in r and the synchronized MIDI note events.

As an illustrative example how this tool can be used, we consider a short excerpt of Chopin's Prelude Op. 28 No. 4 as shown in Figure 4. Often a musical score does not completely describe what is actually played by the performing musician. An example for this are ornamental notes which are not reflected directly by the score (see, e.g., the pink boxes in Figure 4). Such deviations typically lead to local misalignments between the MIDI events and the audio recording. Even worse, additionally played notes that are not contained in the MIDI file may neither have an appropriate template vector in W, nor entries in the activation matrix H. It is therefore impossible for the score-informed source separation procedure to properly capture these notes. As Figure 4 shows, the residual r can reveal such inconsistencies between the notated score and the performance.

Local energy peaks in the residual are commonly aligned with note onsets (see the green and orange boxes in Figure 4). While smaller peaks, like shown in the lower green box, commonly emerge from oversimplifications in the musical model of the source separation algorithm (the derived template vectors in W can often not describe the sound of an onset accurately), more massive bursts of energy often arise from slightly misaligned MIDI events (e.g., the two bass notes marked in orange are played slightly earlier than they are encoded in our synchronized MIDI file).

Another aspect that can not be captured appropriately by the used source-separation procedure are acoustical phenomena like resonance or reverberation (see the red boxes in Figure 4). At the be-

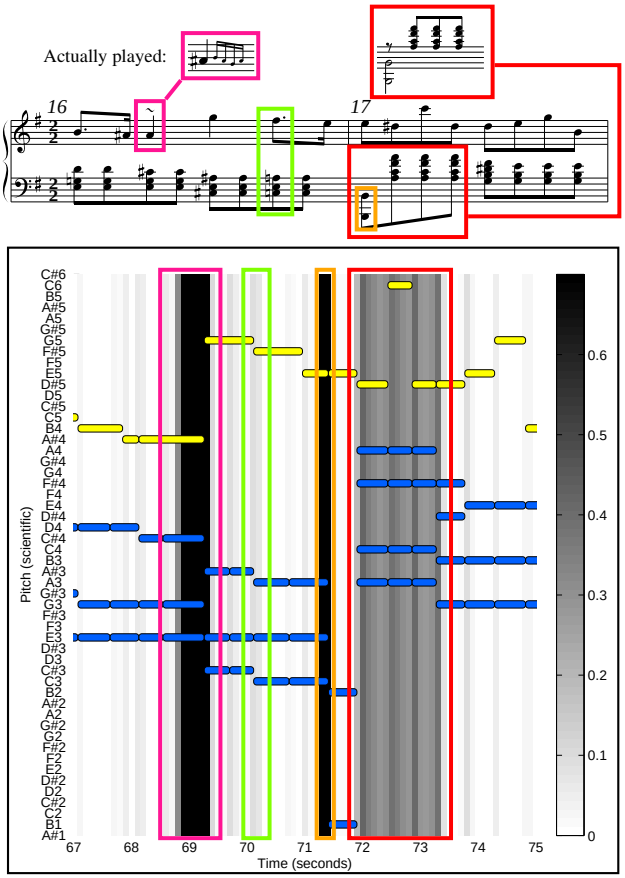

Figure 4: Top: Score of measures 16 and 17 of Op. 28 No. 4 by Frédéric Chopin. Bottom: Our user interface showing the corresponding part in an audio recording of the piece. The short-time energy of the residual signal r is visualized in a color-coded format in the background.

ginning of measure 17, the performing musician holds the pedal of the piano and all played notes are therefore sustained until the pedal is released again in the middle of the same measure. Furthermore, the pressed pedal allows all strings of the piano to resonate with the actually played notes, thus creating an even more complex sound mixture. This information is not reflected in the audio decomposition and a large amount of energy migrates to the residual in the source separation process.

4. CONCLUSION AND FUTURE WORK

In this paper we presented a framework that allows for decomposing a given audio recording into score-based, and therefore musically meaningful audio events. Furthermore, we showed how this decomposition can be used for audio editing and analysis purposes. As discussed in Section 3.3, the residual signal r provides valuable information about deviations between a performance and a corresponding score as well as about misalignments of the MIDI events with the audio recording. As for future work, building a classifier that can automatically distinguish between different kinds of error sources may not only be beneficial for the source separation procedure itself, but could also aid in performance analysis applications or in improving alignment techniques. A further challenge is to investigate to which extend our decomposition may serve as an instance of object-oriented sound representation with applications to parametric audio coding and interactive remixing, see [9].

5. REFERENCES

[1] C. E. Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustic Society of America (JASA)*, 24:975–979, 1953.

[2] J.-L. Durrieu and J.-P. Thiran. Musical audio source separation based on user-selected f0 track. In LVA/ICA, pages 438–445, 2012.

[3] S. Ewert and M. Müller. Using score-informed constraints for NMF-based source separation. In *Proc. of the IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 129–132, Kyoto, Japan, 2012.

[4] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *Proc. of the IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1869–1872, Taipei, Taiwan, 2009.

[5] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 16(3):529–540, 2008.

[6] M. Goto. A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings. In *IEEE Intern. Conference on Acoustics, Speech, and Signal Processing ICASSP*, volume 2, pages 757–760, 2000.

[7] R. Hennequin, R. Badeau, and B. David. Time-dependent parametric and harmonic templates in non-negative matrix factorization. In *Proc. of the Intern. Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, 2010.

[8] R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. In *Proc. of the IEEE Intern. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 45–48, Prague, Czech Republic, 2011.

[9] J. Herre and L. Terentiv. Parametric coding of audio objects: Technology, performance and opportunities. In *Proc. of the Audio Engineering Society Conference (AES)*, Ilmenau, Germany, 2011.

[10] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models. In *Proc. of the Intern. Conference for Music Information Retrieval (ISMIR)*, pages 133–138, Philadelphia, USA, 2008.

[11] C. Joder, S. Essid, and G. Richard. A conditional random field framework for robust and scalable audio-to-score matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(8):2385–2397, 2011.

[12] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of the Neural Information Processing Systems (NIPS)*, pages 556–562, Denver, CO, USA, 2000.

[13] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal on Selected Topics in Signal Processing*, 5(6):1088–1110, 2011.

[14] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1247–1256, 2007.

[15] M. Ryynänen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[16] P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Proc. of the Intern. Conference on Independent Component Analysis and Blind Signal Separation (Lecture Notes in Computer Science 31959*, pages 494–499, Grenada, Spain, 2004.

[17] W. Verhelst and M. Roelands. An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech. In *Proc. of the IEEE Intern. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Minneapolis, USA, 1993.

[18] J. Woodruff, B. Pardo, and R. B. Dannenberg. Remixing stereo music with score-informed source separation. In *Proc. of the Intern. Conference on Music Information Retrieval (ISMIR)*, pages 314–319, 2006.