ESTIMATING NOTE INTENSITIES IN MUSIC RECORDINGS

Sebastian Ewert

University of Bonn
Bonn, Germany

Meinard Müller

Saarland University and MPI Informatik
Saarbrücken, Germany

ABSTRACT

In this paper, we present automated methods for estimating note intensities in music recordings. Given a MIDI file (representing the score) and an audio recording (representing an interpretation) of a piece of music, our idea is to parametrize the spectrogram of the audio recording by exploiting the MIDI information and then to estimate the note intensities from the resulting model. The model is based on the idea of note-event spectrograms describing the part of a spectrogram that can be attributed to a given note event. After initializing our model with note events provided by the MIDI, we adapt all model parameters such that our model spectrogram approximates the audio spectrogram as accurately as possible. While note-wise intensity estimation is a very challenging task for general music, our experiments indicate promising results on polyphonic piano music.

Index Terms— note intensities, audio parametrization, music synchronization, performance analysis.

1. INTRODUCTION

The score of a piece of music basically specifies note parameters such as the pitch, the onset position and the duration. Musical nuances beyond the score are subject to the interpretation by a musician. For example, timings and dynamics (intensities) are not taken as fixed constants and offer a musician the artistic freedom to form a piece of music in his or her own way. Also parameters referring to the timbre are often strongly influenced by the musician. Capturing these musical nuances is important for many different fields in music signal processing. For example, it allows for an automated analysis of differences between several interpretations of a piece of music, as done in the field of performance analysis [1, 2]. A compact description of the nuances might also lead to more efficient compression approaches or higher quality in applications of source separation.

In this contribution, we focus on the estimation of note intensities in recordings of polyphonic piano music. For this task, techniques related to source separation are necessary to distinguish individual note events in an audio recording. In our scenario, we employ a score-informed strategy, where note-event information is given by an additional score-like MIDI file. Similar approaches have been previously applied mainly for source separation. For example, in [3] the authors describe a score-informed source separation system that can be used to separate audio recordings into instrument tracks, where the score is only used to replace a multiple pitch estimation step. In [4], MIDI files are employed to synthesize audio files that are subsequently used to initialize a probabilistic source separation framework. Finally, the system presented in [5] allows for removing solo instruments from a polyphonic recording. However, none of these approaches estimates properties related to individual note events.

Given a MIDI file and an audio recording of a piece of music, our idea is to employ a parametric model that describes a spectrogram as a sum of note-event spectrograms. Here, each note-event spectrogram describes the part of a spectrogram that can be attributed to a specific note event. Our approach starts by initializing the pitch, onset and duration parameters in our model using the note events provided by the MIDI file. In the second step, we adapt the onset and duration parameters by aligning the note events with their corresponding occurrences in the audio using a high-resolution music synchronization approach [6]. In the third step, we iteratively modify parameters in our model related to the acoustic representation of a note event such that our model spectrogram approximates the audio spectrogram as accurately as possible. In a final step, the individual note intensities are estimated using the adapted note-event spectrograms described by the model. An example of the final estimation result is given in Fig. 1.

Because of a lack of annotated ground truth data, evaluating the quality of estimated note intensities is a challenging task itself. In our experiments, we use audio recordings obtained by a Yamaha Disklavier. Equipped with optical sensors and electromechanical devices, such pianos allow for recording the key movements along with the acoustic audio data. On the one hand, the key movement data can be used to derive expected note intensities as described below. On the other hand, we can use our procedure to estimate the note intensities using the audio data. Comparing the expected with the estimated note intensities allows for a first assessment of the estimation quality.

The remainder of the paper is organized as follows. In Sect. 2, we introduce our novel procedure for estimating note intensities in audio recordings. Our experiments on piano music are described in Sect. 3, and conclusions and prospects on future work are given in Sect. 4. Further related work is discussed in the respective sections.

2. PARAMETRIC SPECTROGRAM MODEL AND NOTE INTENSITY ESTIMATION

To describe an audio recording of a piece of music parametrically, many different musical and acoustical aspects have to be considered [7, 8]. Here, one requires parameters to encode the pitch as well as the onset position and duration of note events. Other parameters reflect tuning and timbre of specific instruments or encode amplitude and activity progressions. In this section, we start with a description of our model and its parameters (Sect. 2.1). Then, in Sect. 2.2 and 2.3, we describe how we exploit the information provided by a MIDI file to find parameters such that our model accurately approximates a given audio recording. Finally, in Sect. 2.4, we describe how note intensities are derived from the model parameters.

2.1. Parametric Spectrogram Model

Let $Y \in \mathbb{R}_{\geq 0}^{K \times N}$ denote the magnitude spectrogram of a given audio recording. Our strategy is to approximate Y by means of a model spectrogram Y_λ, where λ denotes the free model parameters. The

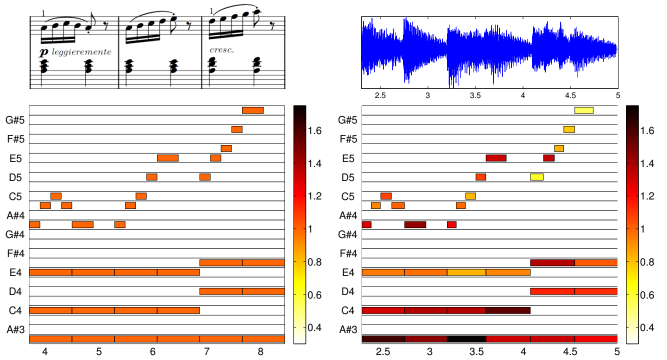

Fig. 1. Illustration of our note intensity estimation procedure using three measures of Burgmüller, Op. 100, Etude No. 2 as an example. The color encodes the note intensity. **Left:** Note events taken from a MIDI generated from score data. The note intensities are constant for all notes. **Right:** Note events derived from our model approximating an audio recording.

first element of λ is a set $\mathcal{M} := \{\mu_m \mid m \in [1 : M]\}$ with note events $\mu_m = (p_m, t_m, d_m)$. Here, p_m describes the MIDI pitch, t_m the onset position and d_m the duration of the note event. Using the index set $[1 : M]$ we define Y_λ as a sum of note-event spectrograms $Y_{m,\lambda}$. More precisely, we define Y_λ at frequency bin $k \in [1 : K]$ and time frame $n \in [1 : N]$ as

$$Y_\lambda(k, n) := \sum_{m\in[1:M]} Y_{m,\lambda}(k, n),$$

where each $Y_{m,\lambda}$ denotes the part of Y_λ that is attributed to μ_m. Each $Y_{m,\lambda}$ consists of a component describing the amplitude or activity over time and a component describing the spectral envelope of a note event. We define

$$Y_{m,\lambda}(k, n) := \alpha_m(n) \cdot \varphi_{p_m,\tau,\gamma}(\omega_k),$$

where ω_k denotes the frequency in Hertz associated with the k-th frequency bin. Here, $\alpha_m \in \mathbb{R}^N_{\geq 0}$ denotes the activity of the m-th note event in the N time frames. We set $\alpha_m(n) := 0$, if the time position associated with frame n lies in $\mathbb{R}\setminus[t_m, t_m + d_m]$. The spectral envelope associated with a note event is described using a function $\varphi : \mathbb{R} \to \mathbb{R}_{\geq 0}$. More precisely, to describe the frequency and energy distribution of the first L partials of a specific note event, φ depends on a pitch $p \in [1 : 127]$, a parameter $\tau \in [-1, 1]^{127}$ related to the tuning and a parameter $\gamma \in [0, 1]^L$ related to the energy distribution over the L partials. We define for a frequency ω in Hertz

$$\varphi_{p,\tau,\gamma}(\omega) := \sum_{\ell\in[1:L]} \gamma_\ell \cdot \kappa(\omega - \ell \cdot f(p + \tau_p)).$$

The function $\kappa : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is used to describe the shape of a partial in frequency direction. Here, we use a Gaussian centered around zero with a suitably chosen, fixed variance. Furthermore, $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$ defined by $f(p) := 2^{(p-69)/12} \cdot 440$ denotes the mapping of the pitch to the frequency scale. To account for non-standard tunings, we express the fundamental frequency associated with pitch p by the term $f(p + \tau_p)$. This completes our model and $\lambda := (\mathcal{M}, \alpha, \tau, \gamma)$ denotes the set of all free parameters, where $\alpha := \{\alpha_m \mid m \in [1 : M]\}$.

Note, that the number of free parameters is kept low by sharing the parameters τ and γ between all note events. Here, a low number allows for an efficient parameter estimation process as described below. Furthermore, sharing the parameters prevents model overfitting, which we observed when using a higher number of parameters.

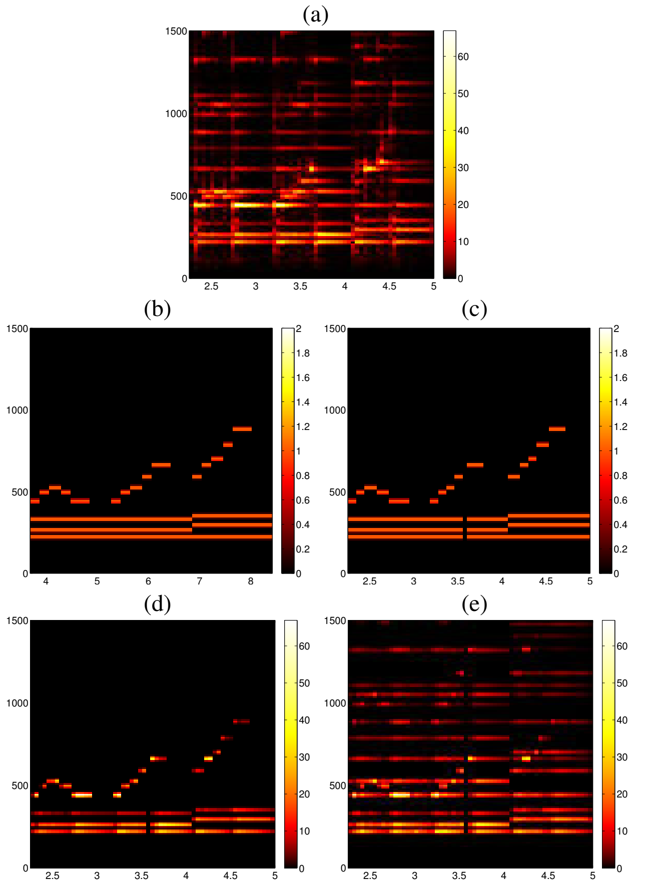

Fig. 2. Illustration of the first iteration of our parameter estimation procedure continuing the example shown in Fig. 1. **(a):** Audio spectrogram Y to be approximated. **(b)-(e)** show the model spectrogram Y_λ after certain parameters are estimated. **(b):** Parameter $\mathcal{M}$ is initialized with MIDI note events generated from score data. **(c):** Note events in $\mathcal{M}$ are synchronized with the audio recording. **(d):** Activity α and tuning parameter τ are estimated. **(e):** Overtone energy distribution parameter γ is estimated.

Next, we describe how a set of parameters λ can be found, such that a given audio spectrogram Y is approximated by Y_λ as accurately as allowed by the model. More precisely, we look for a λ^* with

$$\lambda^* = \underset{\lambda}{\operatorname{argmin}} \, \|Y - Y_\lambda\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. In the following, we illustrate the individual steps in our parameter estimation procedure in Fig. 2, where a given audio spectrogram, shown in Fig. 2(a), is approximated by our model (Fig. 2(b)-(e)).

2.2. Initialization and Estimation of Note Timing Parameters

To initialize our model, we exploit the available MIDI information and fill $\mathcal{M}$ with the MIDI note events. For the m-th note event, we then set $\alpha_m(n) := 1$, if the time position associated with frame n lies in $[t_m, t_m + d_m]$. Furthermore, we set $\tau = (0, \ldots, 0)$ and $\gamma =$

$(1, 0, \ldots, 0)$. An example model spectrogram after the initialization is given in Fig. 2(b).

Our parameter estimation procedure starts by modifying the onset positions and durations of the model note events in $\mathcal{M}$. To this end, we employ a high-resolution music synchronization approach described in [6] to align the note events with their corresponding occurrences in the audio. The procedure is based on Dynamic Time Warping (DTW) and chroma features but extends previous synchronization methods by introducing novel onset-based features to yield a higher alignment accuracy. Using the resulting alignment we determine for each note event the corresponding position in the audio and update the onset position and duration of the note event in $\mathcal{M}$ accordingly. After the synchronization, the parameter $\mathcal{M}$ remains constant during the estimation of the remaining parameters. Fig. 2(c) shows an example of a model spectrogram after the estimation of note timings.

2.3. Estimation of Remaining Model Parameters

To estimate the remaining parameters of λ, we look for (α, τ, γ) that minimize the function $d(\alpha, \tau, \gamma) := \|Y - Y_{(\mathcal{M},\alpha,\tau,\gamma)}\|_F$, where d describes the distance between the audio and the model spectrogram. Here, we need to consider range constraints for the parameters. For example, τ is required to be an element of $[-1, 1]^{127}$. While there are several choices for a minimization approach, we employ a variant of the interior points method described in [9][1]. To this end, we fix two parameters and minimize d regarding the third. For example, to get a better estimate for α, we fix τ and γ and minimize $g(\cdot, \tau, \gamma)$. This process is repeated until all three parameters converge. Figs. 2(d) and (e) illustrate the first iteration of our parameter estimation. Here, Fig. 2(d) shows the spectrogram described by our model after the estimation of the tuning parameter τ and the activity parameter α. Fig. 2(e) shows the spectrogram model after the estimation of the energy distribution parameter γ. Here, one can observe that our model focuses on the harmonic parts of a spectrogram while mainly ignoring the noisy percussive elements.

The main ideas behind [9] can be summarized as follows. Let $h : \mathbb{R}^d \to \mathbb{R}$ be a function to be minimized and $x \in \mathbb{R}^d$. Then the method computes the first and second derivative of h numerically at x and derives a quadratic approximation of h using the Taylor series. The Taylor approximation of h is assumed to be meaningful only in a neighborhood of x, the so called trust region. Instead of minimizing h, the method then minimizes the Taylor approximation within the trust region yielding a new value x^*. If $h(x^*) > h(x)$, then the trust region was too large and the approximation of h was insufficient. In this case, the trust region is decreased and the process is repeated for x. If $h(x^*) \leq h(x)$, then x is set to x^* and the process is repeated until x converges. Possible parameter range constraints are considered in [9] by a barrier approach. Here, the basic idea is to reformulate the function h by including penalty terms that get increasingly large when the value for a parameter is invalid.

2.4. Note Intensity Estimation

After the parameter estimation, the audio spectrogram Y is approximated by Y_λ as precisely as allowed by the model. In particular, the model describes the note-event spectrograms $Y_{m,\lambda}$, that can be used to derive an intensity value for each note event. The energy related to the m-th note event in frame n is given by

$$E_m(n) := \sum_{k\in[1:K]} Y_{m,\lambda}(k, n)^2.$$

To describe the intensity of a note event by a single value, we define

$$\mathcal{I}(m) := \max_{n\in[1:N]} E_m(n)^{0.3}.$$

Here, the exponent 0.3 is used so that the note intensity roughly approximates the perceived loudness of the human auditory system [10, chapter 8]. The result of our method is illustrated in Fig. 1. Here, the left half of Fig. 1 shows note events from a MIDI file generated from score data. The note intensities are constant for all note events. The right half shows the result of our procedure, where note timings and intensities are estimated from an audio recording.

3. EXPERIMENTS

For our evaluation, we employ a database consisting of Disklavier recordings of various pieces from the Western classical music repertoire[2]. For each audio recording, the Disklavier automatically generates a MIDI file, which can be regarded as a kind of ground truth annotation. For example, the MIDI onset-positions and durations correspond closely to those in the audio recordings. Particularly interesting for our evaluation are the velocity values, which symbolically encode the dynamics in a MIDI file. Here, a translation of the symbolic velocity values to physical note intensities would allow for a simple way to evaluate our method. However, this translation is not trivial for polyphonic music because of the complex interaction of multiple sound sources and other acoustical effects. To approximate such a translation, we built up a dictionary that maps a pitch and a velocity to a physical note intensity. Here, in a first step, we employed training data consisting of single note events played at several velocities. In a second step, we refined the dictionary using five pieces from our database. We marked these pieces in our evaluation results with a star. However, initial experiments using monophonic recordings revealed that the dictionary is only capable of estimating the note intensity with an accuracy of about five to ten percent.

Given a piece from our database, we used our dictionary in combination with the prealigned annotation MIDI file to estimate the intensity for each note event. We denote the resulting values by $(\mathcal{I}_{\text{dic}}(m))_{m\in[1:M]}$, where $[1 : M]$ denotes the index set for the note events. Then, ignoring the given velocity values, we used our proposed method to estimate the individual note intensities using the MIDI and the audio data. The resulting values are denoted by $(\mathcal{I}(m))_{m\in[1:M]}$. To compensate for different overall recording levels, we only consider relative note intensities in our evaluation. To this end, we normalize the M-dimensional vectors $\mathcal{I}_{\text{dic}}$ and $\mathcal{I}$ with regard to the Euclidean norm and compare them in terms of a percentage error defined by

$$\text{PE} := \left(100 \cdot \left|\frac{\mathcal{I}(m) - \mathcal{I}_{\text{dic}}(m)}{\mathcal{I}_{\text{dic}}(m)}\right|\right)_{m\in[1:M]}$$

The mean and standard deviation of PE are given in the third and fourth column of Table 1. For example, for Bach's BWV849-02 one gets a mean percentage error of 9.3 with a standard deviation of 5.5. The average over all pieces is 16.9 with a standard deviation of 9.3. With an intrinsic error of five to ten percent induced by our estimated dictionary, this indicates a reasonable estimation quality.

[1] A popular implementation of this procedure can be found in the Matlab Optimization Toolbox software package.

[2] All files are part of the SMD database, which is available on request from the authors.

Composer	**Piece**	**Proposed**		**Proposed**		**Baseline**		**Baseline**	
		prealigned		distorted		prealigned		distorted	
		Mean	STD	Mean	STD	Mean	STD	Mean	STD
Bach	BWV849-01*	9.3	5.3	9.5	5.5	31.5	25.9	31.9	26.5
Bach	BWV849-02	9.3	5.5	9.5	5.7	28.7	23.9	29.3	24.5
Bach	BWV871-01	11.0	6.2	11.4	6.3	27.5	21.4	28.2	21.7
Bach	BWV871-02	7.7	5.1	7.8	5.2	24.6	20.0	25.0	20.3
Bach	BWV875-01	13.9	6.7	14.1	6.9	31.6	26.2	32.0	26.8
Bach	BWV875-02	8.3	4.9	8.5	5.0	28.2	25.4	28.8	26.1
Beethoven	Op027No1-01	12.5	7.1	13.0	7.2	39.4	28.0	40.5	28.6
Beethoven	Op027No1-02*	10.3	6.5	10.6	6.7	35.2	24.4	35.9	24.7
Beethoven	Op027No1-03	13.6	7.3	14.0	7.5	45.6	32.0	46.6	33.3
Beethoven	Op031No2-01	16.1	8.7	16.5	8.9	36.1	29.9	36.9	30.4
Beethoven	Op031No2-02	27.2	14.5	27.8	14.7	38.6	27.5	39.2	27.8
Beethoven	Op031No2-03	13.2	8.1	13.5	8.2	34.9	29.4	35.4	30.1
Brahms	Op010No1*	13.8	7.3	14.0	7.5	38.9	29.3	39.4	29.8
Brahms	Op010No2	13.6	7.9	14.2	8.0	41.3	32.6	42.5	33.8
Chopin	Op010-03	25.2	13.0	25.4	13.2	35.1	31.0	35.5	32.2
Chopin	Op010-04	25.0	13.2	25.8	13.6	36.0	34.2	36.9	35.1
Chopin	Op026No1	22.6	13.2	22.9	13.5	34.1	34.8	34.5	35.2
Chopin	Op026No2	23.6	14.2	23.8	14.6	34.7	33.2	35.0	33.8
Chopin	Op028-01	22.9	11.4	23.6	11.9	37.7	25.8	38.6	26.7
Chopin	Op028-03	19.0	12.2	19.3	12.6	33.4	36.8	33.9	38.0
Chopin	Op028-04	19.5	11.6	20.2	12.0	29.6	29.2	30.4	30.3
Chopin	Op028-11	18.8	9.1	19.0	9.3	25.6	23.3	25.9	23.5
Chopin	Op028-15	18.0	9.2	18.7	9.4	24.7	19.3	25.4	19.5
Chopin	Op028-17	22.1	10.7	22.9	11.0	31.1	24.9	32.0	25.4
Chopin	Op029*	20.1	11.6	20.8	11.9	32.7	34.9	33.6	35.9
Chopin	Op048No1	26.0	11.2	26.2	11.3	39.0	34.0	39.4	35.3
Chopin	Op066	22.4	13.5	22.7	13.8	31.0	37.3	31.4	38.6
Haydn	Hob017No4*	14.8	8.1	15.4	8.3	44.5	32.7	45.8	33.4
Rachman.	Op039No1	15.5	9.0	16.0	9.2	36.6	28.0	37.6	28.6
Skryabin	Op008No8	10.1	5.6	10.4	5.8	24.5	21.5	25.1	21.8
Average		**16.9**	**9.3**	**17.2**	**9.5**	**33.8**	**28.6**	**34.4**	**29.3**

Table 1. Estimation quality of our proposed method and a baseline. Shown are the mean and standard deviation of the percentage errors PE and $\mathrm{PE}_{\mathrm{base}}$. The results for using prealigned and temporally distorted MIDI files are listed separately. The stars indicate pieces, that have been used to refine our note intensity dictionary.

To further evaluate the influence of the music synchronization step, we randomly distorted the prealigned MIDI files by splitting them into 20 segments of equal length and by stretching or compressing each segment by a random factor within an allowed distortion range (in our experiments we used a range of $\pm 50\%$). The results are shown in the fifth and sixth column of Table 1. Here, the average error for Bach's BWV849-02 increases only moderately from 9.3 (prealigned MIDI) to 9.5 (distorted MIDI). Similarly, the average error also increases moderately from 16.9 to 17.2, which indicates that our synchronization works robustly in most cases.

To get a better understanding of these numbers, we conducted a simple baseline experiment. Our baseline method starts by computing the magnitude spectrogram for a given audio recording. Then, exploiting the onset, duration and pitch information given by a synchronized MIDI file, the method locates the spectrogram bins that are related to the first five partials of a given note event. To locate the partials correctly, we incorporate simple heuristics to estimate the fundamental frequency for each pitch. Using only the located bins, the baseline method then computes the energy in each time frame and derives a note intensity from the maximum of these energy values. In some sense, this method roughly represents what is possible without using a sophisticated overtone model. We denote the resulting intensity values by $(\mathcal{I}_{\mathrm{base}}(m))_{m\in[1:M]}$. After normalizing $\mathcal{I}_{\mathrm{base}}$, we compare $\mathcal{I}_{\mathrm{base}}$ and $\mathcal{I}_{\mathrm{dic}}$ in terms of a percentage error $\mathrm{PE}_{\mathrm{base}}$ as described above. The results of our baseline experiments using both prealigned and distorted Disklavier MIDI files are shown in columns seven and eight as well as nine and ten of Table 1, respectively. Using prealigned MIDI files, the error for Bach's BWV849-02 is 28.7, which is over three times higher compared to an error of 9.3 for our method. Furthermore, the average error when using the baseline method is with 33.8 twice as high as the error when using our proposed method. Overall, the baseline experiments indicate that our method indeed yields note intensities with a reasonable estimation quality.

4. CONCLUSIONS

In this paper, we have presented a first method for the estimation of note intensities in music recordings. While such an estimation is a very challenging task in general, our experiments on polyphonic piano music revealed measureable advantages of our method over a given baseline. However, for the future there are still several challenges to be solved. For example, the definition of a suitable ground truth for note intensities is an open problem. Here, one has to consider complex acoustical effects which strongly depend on the recording conditions. In particular, the room acoustics and the interaction of multiple sound sources with the resonance body of an instrument are important factors. However, this cannot be achieved with a simple dictionary based ground truth. A better ground truth is also a requirement for a more detailed analysis of the capabilities and limitations of our procedure, and our model in particular. A first manual inspection already revealed that our procedure tends to incorrectly model note events with very low pitches. Here, we need to consider other spectral representations or multi-resolution spectrograms that offer a higher frequency resolution for lower pitches. Furthermore, our procedure could be improved by incorporating perceptually oriented methods to assess the intensity or loudness of a note event [10].

5. REFERENCES

[1] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic, "In search of the Horowitz factor," *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.

[2] Craig Stuart Sapp, "Comparative analysis of multiple musical performances," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007, pp. 497–500.

[3] John Woodruff, Bryan Pardo, and Roger Dannenberg, "Remixing stereo music with score-informed source separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, Canada, 2006, pp. 314–319.

[4] Joachim Ganseman, Paul Scheunders, Gautham J. Mysore, and Jonathan S. Abel, "Source separation by score synthesis," in *Proceedings of the International Computer Music Conference (ICMC)*, New York, USA, 2010.

[5] Yushen Han and Christopher Raphael, "Desoloing monaural audio using mixture models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 145–148.

[6] Sebastian Ewert, Meinard Müller, and Peter Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[7] Pierre Leveau, Emmanuel Vincent, Gaël Richard, and Laurent Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 116–128, 2008.

[8] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009, pp. 327–332.

[9] Richard H. Byrd, M. E. Hribar, and Jorge Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[10] Hugo Fastl and Eberhard Zwicker, *Psychoacoustics, Facts and Models*, Springer, 2007 (3rd Edition).