PIANO MUSIC TRANSCRIPTION MODELING NOTE TEMPORAL EVOLUTION

Andrea Cogliati and Zhiyao Duan

University of Rochester
AIR Lab, Department of Electrical and Computer Engineering
207 Hopeman Building, P.O. Box 270126, Rochester, NY 14627, USA.

ABSTRACT

Automatic music transcription (AMT) is the process of converting an acoustic musical signal into a symbolic musical representation such as a MIDI piano roll, which contains the pitches, the onsets and offsets of the notes and, possibly, their dynamic and source (i.e., instrument). Existing algorithms for AMT commonly identify pitches and their saliences in each frame and then form notes in a post-processing stage, which applies a combination of thresholding, pruning and smoothing operations. Very few existing methods consider the note temporal evolution over multiple frames during the pitch identification stage. In this work we propose a note-based spectrogram factorization method that uses the entire temporal evolution of piano notes as a template dictionary. The method uses an artificial neural network to detect note onsets from the audio spectral flux. Next, it estimates the notes present in each audio segment between two successive onsets with a greedy search algorithm. Finally, the spectrogram of each segment is factorized using a discrete combination of note templates comprised of full note spectrograms of individual piano notes sampled at different dynamic levels. We also propose a new psychoacoustically informed measure for spectrogram similarity.

Index Terms— Automatic music transcription, multi-pitch estimation, spectrogram factorization, onset detection

1. INTRODUCTION

Automatic Music Transcription (AMT) is the process of extracting a symbolic representation from a music audio file. The output of AMT can be a full musical score or an intermediate representation, such as a MIDI piano roll, which includes note pitches, onsets, offsets and, possibly, dynamics and instruments playing the notes. The complete AMT problem can be divided into several subtasks, not necessarily in this order: multi-pitch estimation, onset/offset detection, loudness estimation, source recognition, note tracking, beat and meter detection, rhythm detection. The biggest efforts so far have been spent on the multi-pitch estimation and onset detection stages [1].

Many music transcription methods attempt to identify pitches in each time frame, and then form notes in a post-processing stage [1]. This approach, however, does not model the temporal evolution of notes. We can call this approach *frame-based* transcription. Spectrogram decomposition-based approaches fall into this category. Non-negative matrix factorization (NMF) is a method for factorizing a large non-negative matrix into the product of two, low-rank non-negative matrices [2][3]. NMF has been applied to source separation and AMT [4]. For AMT, NMF is usually applied in a supervised way by pre-learning a dictionary of templates in which each template corresponds to the long-time average spectrum of a note. NMF is computationally inexpensive, jointly estimates multiple pitches at the same time and provides a salience of each estimated pitch with its activation weight. One of the drawbacks of NMF is that it does not model the temporal evolution of notes.

Piano notes are characterized by significant temporal evolutions, in both the waveform and the spectral content. In particular, different partials decay at different rates: higher frequency partials decay faster than lower frequency ones [5]. There are very few methods that consider temporal evolution of notes. A tensor can be used to represent multiple vectors evolving in time, e.g., a dictionary of spectrograms. Non-negative tensor factorization (NTF), an extension of NMF, has been applied to source separation and AMT [6][7][8]. An alternate formulation of NMF named Probabilistic Latent Component Analysis (PLCA) was proposed by Smaragdis et al. in 2006 [9]. PLCA is numerically equivalent to NMF but its formulation provides a framework that is easier to generalize and extend [9]. Grindlay and Ellis proposed a generalization to PLCA to account for the temporal evolution of each note [10].

All the spectrogram factorization methods described in the previous paragraphs form notes in the post-processing stage. To our knowledge, only one existing method exploits the percussive nature of piano notes to detect the onsets from the audio signal as the initial step [11]. In this method, the algorithm first detects the note onsets, then analyzes the audio signal between two successive onsets, assuming that the pitches do not change. We call this approach *note-based* transcription, as opposed to frame-based transcription.

In this paper we propose a novel note-based spectrogram factorization algorithm that exploits the temporal evolution of piano notes. The method uses an artificial neural network to detect note onsets from the audio spectral flux. Then, the notes present in each audio segment between two successive onsets are estimated with a greedy search algorithm. The log-frequency, linear magnitude spectrogram of each segment is factorized by using a discrete combination of note templates comprised of full note spectrograms of individual piano notes sampled at different dynamic levels. The use of full length spectrograms as dictionary templates allows the algorithm to take into account the temporal evolution of the notes. In the paper we also investigate how the recording level, i.e., the global loudness, of the audio signal and the different dynamic levels used to play each note, i.e., the local loudness, affect the pitch estimation, and we propose a method to reduce the dimensionality of the search space by separating the estimation of these two parameters.

2. RELATION TO PRIOR WORK

The proposed method builds on and expands existing spectrogram factorization techniques. The key idea of the proposed method is to combine a note-level transcription with a factorization algorithm exploiting the temporal evolution of piano notes.

Existing spectrogram factorization techniques originated and stemmed from NMF [1]. An alternate formulation of NMF, named Probabilistic Latent Component Analysis (PLCA), was proposed by Smaragdis et al. in 2006 [9]. PLCA is numerically equivalent to NMF but its formulation provides a framework that is easier to generalize and extend [9]. A notable extension to PLCA was proposed by Benetos and Dixon in 2012 [12]; in their formulation, the authors extend asymmetric PLCA to accommodate for different instrument sources and for pitch shifting. Grindlay and Ellis proposed a different generalization to PLCA to account for the temporal evolution of each note [10], using dictionary templates consisting of long spectrograms. In this paper, we adopt a similar idea, but we use a discrete combination of templates to reduce the dimensionality of the search space. Another AMT method exploiting the temporal evolution of notes, called harmonic temporal structured clustering, was proposed by Kameoka et al. [13]. This method jointly estimates multiple notes and their onsets, offsets and dynamics using a maximum likelihood estimator. The method proposed in this paper uses two separate stages to detect onsets and to estimate the pitches.

A separate onset detection stage has been proposed in previous AMT systems. SONIC, a relatively old but still competitive piano transcription system, uses an onset detection stage to improve the performance of the algorithm [14]. However, the results of the onset detection are only used in the post-processing stage and not in the pitch estimation process, which is frame-based. The method proposed by Constantini et al. [11] uses an initial onset detection stage that feeds its output to the pitch estimation stage, but only considers a single window of 64 ms after the onset for the pitch estimation. The proposed method utilizes the entire portion of the audio between two successive onsets and also analyzes the temporal evolution of the notes in identifying pitches.

3. PROPOSED METHOD

The proposed method operates in two stages: onset detection and pitch estimation. The first stage analyzes the audio file looking for note onsets. The percussive nature of the piano helps to detect the onsets, even at softer dynamics. Onsets temporally close enough can be considered as a single onset without excessive loss of precision in the following stage. The second stage analyzes the audio between two successive onsets and identifies the pitches in the segment by decomposing the spectrogram into a summation of note templates. Each template is a spectrogram of a note with a certain dynamic, and is learned from a database of isolated piano note samples. A greedy algorithm and a psycoacoustically motivated similarity measure were proposed for the decomposition.

3.1. Onset detection

Onset detection is another challenging problem in computer audition, and several different methods have been proposed to address it [15]. SONIC uses a feedforward neural network for the onset detection stage [16]. Eyben et al. proposed a universal onset detection method based on bidirectional long short-term memory neural networks [17]. The usage of recurrent neural networks instead of feedforward networks improves the performance of the detection stage because the network can take into account the temporal evolution of the notes and adapt itself to the context in the musical piece. The drawback of this approach is the complexity of the model. In this paper we use a nonlinear autoregressive network with exogenous inputs (NARX), which produces better results than a simple feedforward network, but with a simpler model than Eyben's.

The neural network is used to analyze the normalized spectral flux of the audio input. The linear magnitude spectrogram from the original audio is generated using STFT of 46.4 ms, Hamming window and a hop size of 2.9 ms. The spectral flux is calculated by summing all the positive bin-to-bin differences between two successive frames. The very short hop size is necessary to obtain a high time resolution, which is critical for the pitch estimation stage.

The normalized spectral flux is then processed by a nonlinear autoregressive network with exogenous inputs (NARX) with two hidden layers, with 18 and 15 neurons respectively, and 4 delays (see Fig. 1). The neural network has been trained on 10 pieces from the Disklavier Ambient corpus of the MAPS database [18] using the Levenberg-Marquardt algorithm. The 10 pieces had a combined total length of 2,840 s and a combined total of 11,630 onsets.

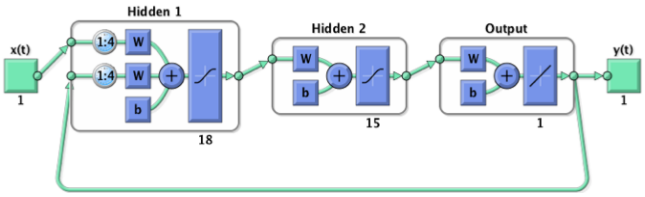


Fig. 1. Recurrent neural network for onset detection.

3.2. Pitch estimation

3.2.1. Dictionary creation

The dictionary of templates is generated from the University of Iowa Musical Instrument Samples[1], in which the individual notes of a piano have been recorded at three different dynamic levels, i.e., *pp*, *mf* and *ff*. Each note template is a log-frequency, linear magnitude spectrogram, and is calculated using a constant Q transform (CQT) with 36 bins per octave and a hop size of 23.2 ms. All templates are limited to 128 frames, corresponding to about 3 s. The raw spectrograms contain significant energy due to resonance and reverb of the keystroke so we filter the spectrogram to only keep the active partials, i.e., partials of currently active notes. We found that the amplitude evolution of active partials can be approximated by a sum of two decaying exponentials

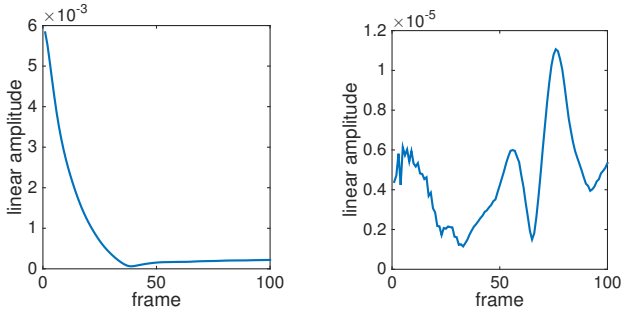
$$A(t) = ae^{bt} + be^{ct}, \tag{1}$$

while non-active bins show a more noisy, random or oscillatory envelope, as shown in Fig. 2. The spectrogram is initially filtered with a median filter operating over the frequency axis with a window of 3 bins; then, using a curve fitting algorithm, only the bins that can be properly approximated by (1) are retained. All the other bins are set to 0. The results of the filtering are illustrated in Fig. 3. The filtered spectrograms are also normalized and quantized to 16-bit integers to optimize the memory footprint and speed up computation.

3.2.2. Pitch estimation algorithm

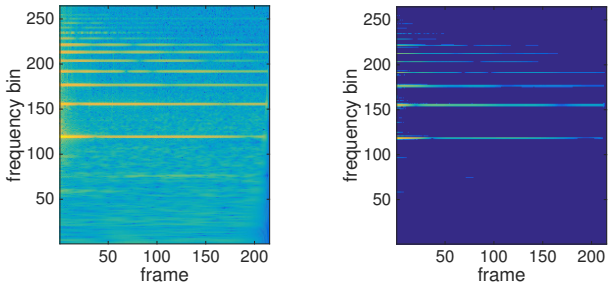
Each individual inter-onset audio segment is modeled as a discrete, additive combination of templates with the same dynamic from the dictionary, scaled by a global gain factor. The model only takes into account newly activated notes, i.e., partials belonging to notes played in a previous segment are filtered out. We assume that all the notes

[1] http://theremin.music.uiowa.edu/MIS.html



(a) Active partial (b) Non-active bin

Fig. 2. Comparison of an active partial versus a non-active frequency bin in a piano note.



(a) Original CQT Spectrogram (b) Filtered template

Fig. 3. Dictionary templates contain full length spectrograms of piano notes.

in a single audio segment are played at the same dynamic level – this helps to rule out the interference of the reverb of previous notes. The global gain factor accounts for the loudness of the recorded audio, e.g., close microphones versus far microphones or gain adjustment in post-production. We assume that the general loudness does not change for the entire duration of a single piece. Finally, the model does not attempt to determine note offsets.

A spectrogram with the same parameters is generated from each segment detected in the previous step and it is filtered using the same approach used in Section 3.2.1. In most piano pieces, notes do not always start and end at the same time. So it is possible that new notes are played while old notes are still playing, e.g., a fast melody played on the right hand over longer chords played on the left hand. Active partials from old notes will then be present in the spectrogram of successive segments. To avoid the interference of these partials, we filter them out by comparing the first three frames of each segment with the last three frames of the previous one. For each active partial, we take the median of the magnitude in each segment. If the median value in the new segment is smaller than the median in the previous segment, the partial is assumed to be a continuation of a previously active partial and we set the entire frequency bin to 0.

Given a spectrogram difference measure (as described in Section 3.2.3), a greedy algorithm is used to find the combination of templates that minimizes the difference between the segment spectrogram and the reconstructed spectrogram with templates. A piano piece has a maximum polyphony of 10 notes, i.e., each single segment can only contain 10 active notes (this excludes four-hand piano performances and special playing techniques, such as large cluster chords played with the forearm). A concert piano has 88 different keys. The dictionary has 3 templates per note giving a total of 264 templates. The possible combinations to test are given by

$$3 * \left(\binom{88}{1} + \binom{88}{2} + \cdots + \binom{88}{10} \right) \simeq 15 \cdot 10^{12}. \quad (2)$$

The search space is too big for an exhaustive search, so a greedy algorithm is used instead. The greedy algorithm compares the spectrogram to each template in the dictionary and computes the corresponding cost function for reconstruction. The combination of note and dynamic with the lowest cost is selected. The dynamic level is fixed for the successive iterations. Then the algorithm tries to add a second note and, if the cost function decreases by at least 5%, the second note is selected. The algorithm stops when an additional note does not lower the cost function. All templates are truncated at the end to have the same length as the audio segment during the reconstruction.

We also considered a global optimization approach, i.e., non-negative tensor factorization, but we preferred a greedy approach instead, which has two main advantages: computational efficiency and discrete reconstruction. Each update iteration of NTF is computationally equivalent to an iteration of our greedy algorithm, but NTF generally requires tens of iterations to converge, while our greedy approach only needs a number of iterations up to the polyphony of the piece. The greedy approach also combines pitch and polyphony estimation in a single stage, while NTF requires a separate thresholding for the salience coefficients. The binary output of the greedy algorithm has the additional advantage of being categorical, i.e., each note is either active or not.

3.2.3. Spectrogram similarity

The spectrogram similarity is measured using a cost function derived from an L^2-norm scaled by a factor that depends on the amplitude of the spectrogram. Given V, the spectrogram of the original audio, and R, the reconstructed spectrogram, the difference of the two spectrograms is measured by

$$D(V||R) = \sum_{ij} \left(\frac{\alpha V_{ij} - R_{ij}}{\log_{10}(V_{ij} + 1) + 1} \right)^2, \quad (3)$$

where α is the global gain scale factor; i and j are frequency and time indices.

This cost function was tested after poor results were obtained with L^1-norm, L^2-norm and KL-divergence. The metric was motivated by signal processing and psychoacoustical evidence. Given a spectrogram to model, a reconstructed spectrogram should have the same amount of energy as the model spectrogram at each time/frequency point. The error between the reconstructed spectrogram and the original spectrogram should be weighted according to the energy present in the original spectrogram, i.e., the error should be calculated as percentage variation. The logarithm in the scaling factor mimics the dB scale, while the addition of 1, before and after taking the logarithm, gives always a strictly positive value.

4. EXPERIMENT

We conducted three sets of experiments and compared the results against two state-of-the art transcription systems, Benetos's [12], the

	Precision	Recall	F
Benetos	0.580	0.498	0.534
Proposed ($\alpha = 1$, *mf* only)	0.618	0.504	0.555
Proposed ($\alpha = 1$, *pp, mf, ff*)	**0.736**	0.455	0.562
Proposed ($\alpha = 2.8$, *pp, mf, ff*)	0.674	**0.609**	**0.640**

Table 1. Results for the 100 random chords.

best performer in MIREX 2013[2], and SONIC [14], a less recent but still competitive piano transcription system. We used the original implementations for both methods. For the experiments, an estimated note is considered correct if the MIDI number corresponds to the ground truth MIDI number and its onset is within 50 ms from the ground truth; note offsets are not considered for the evaluation.

For the first experiment we took 100 random chords from [19], 10 chords for each polyphony from 1 to 10, with *normal* playing style. For this experiment we investigated the role of dynamic and loudness and how they affect the performance of the algorithm. We tested the proposed multi-pitch estimation algorithm with a dictionary containing only the *mf* dynamic and with all three dynamics, *pp*, *mf*, and *ff*. The results are shown in Table 1. The usage of multiple dynamics increases the level of precision by nearly 12% while reducing the level of recall by almost 5%. The F-measure is also slightly improved. We also tested different levels of gains for the original audio. We scaled the original audio by a series of factors, starting from 0.5 to 3.0 with an increment of 0.1. The best results were obtained with a gain of 2.8. This gain reduces the precision by about 6% but dramatically increases the recall by almost 15%. As a result, the F-measure is increased by almost 9%. This suggests that the correct recording level is very important for the proposed method, which is not surprising since the reconstruction of the spectrogram is done as a discrete combination of dictionary templates, i.e., a template is either added or not. A template with the right shape but different level will not be considered a good match by the similarity function. Also, in this case, the audio was most likely recorded at a lower level than the notes used for the dictionary; boosting the level of audio increases the recall because more templates from the dictionary can be properly matched, but it probably also introduces more noise and reverb that decrease the precision. We compared the proposed algorithm with Benetos's. The proposed method outperforms Benetos's in precision and F-measure. This suggests that considering the temporal evolution of the notes provides a better estimation.

For the second experiment we tested the 3 considered algorithms on the 30 pieces in the "Ambient" collection of MAPS [18]. This collection, being recorded in a reverberant ambient with microphones placed at 3-4 meters from the source is considered a more challenging testbed than audio recorded with microphones at close position, but it reflects a more realistic condition when dealing with recorded audio. The SONIC algorithm outperforms both Benetos's and the proposed algorithm in Precision, Recall and F-measure. The results are shown in Table 2.

The performance of the proposed method are most likely limited by three factors: the onset detection stage (see the results of the next experiment), short notes and recording level. Short notes, i.e., notes with a duration of less than 50ms, are poorly estimated by the algorithm, as the corresponding spectrogram is only 1 to 3 frames long; the proposed filtering method is not effective under these circumstances, and the long term evolution of the notes cannot be properly modeled. We tried to find the optimal gain, as we did in the previous experiment, but the results were inconclusive. This suggests that the pieces in MAPS were recorded with different setups or the gain was optimized in post-production.

[2] http://www.music-ir.org/mirex/wiki/2013:MIREX2013_Results

	Precision	Recall	F
Benetos	0.528	0.382	0.443
SONIC	**0.627**	**0.530**	**0.574**
Proposed ($\alpha = 1$, *pp, mf, ff*)	0.438	0.302	0.345

Table 2. Results on MAPS dataset.

	Precision	Recall	F
SONIC	0.727	**0.719**	**0.723**
Proposed ($\alpha = 1$)	**0.865**	0.619	0.722

Table 3. Results for the onset detection.

For the last experiment we tested the performance of the onset detection stage and compared its results with SONIC. Since in our approach an onset detection error (false alarm or miss) will always incur one or more transcription errors, we conducted this experiment to isolate onset detection errors from the overall music transcription errors. SONIC is only available in executable format, so we could not compare the results of our onset detection algorithm with their detection stage directly. Instead, we took the output of SONIC, a fully transcribed MIDI piano roll, and extracted the note onsets from there. The results on the 30 pieces in the "Ambient" collection of MAPS are shown in Table 3. The proposed method has a better precision than SONIC and a comparable F-measure. Still the overall performance is not satisfactory, and this might be another reason for the poor performance of the proposed method in the general transcription case compared to the performance in transcribing the chords.

5. CONCLUSIONS

We presented a new model for the challenging problem of automatic piano transcription. The proposed method performs a note-based transcription and matches the performance of state-of-the art algorithms in single-chord multi-pitch estimation and outperforms them in the best case scenario, in which the global gain of the recording matches the recording level of the templates.

Implementing a better onset detection stage seems a reasonable next step in order to improve the overall recall rate. A possible solution for the poor estimation of short notes would be to create a dictionary with increased resolution, i.e., shorter hop length, to be used when the time between two successive onsets is below a certain threshold. The biggest challenge is the estimation of the global gain, which plays an important role in the performance of the algorithms. A possible approach would be to iteratively try different gain levels and select the one that minimizes (3).

6. ACKNOWLEDGMENTS

The authors wish to thank Emmanouil Benetos, Research Fellow at City University London, for providing the code of his transcription system to compare the performance.

7. REFERENCES

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[3] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[4] Paris Smaragdis and Judith C. Brown, "Non-negative matrix factorization for polyphonic music transcription," *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.

[5] Murray Campbell and Clive Greated, *The Musician's Guide to Acoustics*, Oxford University Press, 1994.

[6] Tuomas Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[7] Joonas Nikunen, Tuomas Virtanen, and Miikka Vilermo, "Multichannel audio upmixing based on non-negative tensor factorization representation," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 33–36.

[8] Tom Barker and Tuomas Virtanen, "Non-negative tensor factorisation of modulation spectrograms for monaural sound source separation," in *Proc. Interspeech*, pp. 827–831.

[9] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "A probabilistic latent variable model for acoustic modeling," *In Workshop on Advances in Models for Acoustic Processing at NIPS*, 2006.

[10] G. Grindlay and D. P. W. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1159–1169, 2011.

[11] Giovanni Costantini, Renzo Perfetti, and Massimiliano Todisco, "Event based transcription system for polyphonic piano music," *Signal Processing*, vol. 89, no. 9, pp. 1798–1811, 2009.

[12] Emmanouil Benetos and Simon Dixon, "A shift-invariant latent variable model for automatic music transcription," *Computer Music Journal*, vol. 36, no. 4, pp. 81–94, 2012.

[13] Hirokazu Kameoka, Takuya Nishimoto, and Shigeki Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 982–994, 2007.

[14] Matija Marolt, "SONIC: Transcription of polyphonic piano music with neural networks," in *Audiovisual Institute, Pompeu Fabra University*, 2001, pp. 217–224.

[15] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[16] Matija Marolt, Alenka Kavcic, and Marko Privosnik, "Neural networks for note onset detection in piano music," in *Proc. 2002 International Computer Music Conference*.

[17] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves, "Universal onset detection with bidirectional long short-term memory neural networks," in *Proc. ISMIR*, pp. 589–594.

[18] Valentin Emiya, Roland Badeau, and Bertrand David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[19] Ana M. Barbancho, Isabel Barbancho, Lorenzo J. Tardn, and Emilio Molina, *Database of Piano Chords: An Engineering View of Harmony*, Springer Publishing Company, Incorporated, 2013.