MODELING MUSICOLOGICAL INFORMATION AS TRIGRAMS IN A SYSTEM FOR SIMULTANEOUS CHORD AND LOCAL KEY EXTRACTION

Johan Pauwels†, Jean-Pierre Martens†, Marc Leman‡

†Digital Speech and Signal Processing group (ELIS-DSSP)
‡Institute for Psychoacoustics and Electronic Music (IPEM)
Ghent University, Belgium
johan.pauwels@elis.ugent.be, jean-pierre.martens@elis.ugent.be, marc.leman@ugent.be

ABSTRACT

In this paper, we discuss the introduction of a trigram musicological model in a simultaneous chord and local key extraction system. By enlarging the context of the musicological model, we hoped to achieve a higher accuracy that could justify the associated higher complexity and computational load of the search for the optimal solution. Experiments on multiple data sets have demonstrated that the trigram model has indeed a larger predictive power (a lower perplexity). This raised predictive power resulted in an improvement in the key extraction capabilities, but no improvement in chord extraction when compared to a system with a bigram musicological model.

Index Terms— Chord extraction, key extraction, music signal processing, music information retrieval

1. INTRODUCTION

In Western polyphonic music, the backbone of harmony is formed by a sequence of chords. They are the building blocks of the accompaniment over which a melody is played. We define a chord as a collection of simultaneously sounding accompaniment notes, although the distinction between a chord note and a melody note can be questionable in some cases. The associated chord name refers to a reference note, the root, and a set of tonal distances to this reference, the chord type. All notes together, whether played concurrently (forming a chord) or sequentially (forming a melody), establish a broader musical context, which we call a key. Its associated name refers to a tonal center, the tonic, and a set of tonal distances in relation to this tonic, called the mode.

Chord extraction is the process of converting an audio recording into a stream of chord symbols with associated times. Because a sequence of chords provides a compact description of a song, automated extraction has multiple applications. For instance, the resulting chord symbols can be directly used to learn how to play a basic accompaniment of a music piece. Furthermore, they can serve as an intermediate representation for a variety of indirect applications, such as automatic playlist generation, partly based on harmonic similarities.

Key extraction is a similar process for extracting key symbols from the recording. A distinction can be made between global and local key extraction. The former assigns a single label that applies to the whole song whereas the latter assigns a label to each time segment. While studying harmony, one is mostly interested in the movement of chords within a key. This requires the extraction of both chords and keys.

Since successive chords and keys in a sequence are not mutually independent, most chord and key extraction systems model these dependencies. Usually, a finite state automaton is used, with each state either representing a chord [1, 2, 3, 4], a key [5] or a key-chord combination [6, 7, 8], depending on the desired output. Most of the time, a first order Markov assumption is made such that transitions between states only depend on the previous state. However, it is musically intuitive to consider a bigger context while it has also been shown that higher order modeling decreases the perplexity [9]. Nonetheless, very little research has been done that goes beyond bigram modeling. Only Khadkevich & Omologo [10] have used trigram and 4-gram modeling in a chord extraction system before. They report that adding a trigram model to a baseline system comprising no contextual modeling at all yields an improvement: the frame accuracy could be raised from 67.86 % to 69.52 %. However, substituting the trigram by a 4-gram model did not offer any additional improvement. Surprisingly, the authors do not report any figures for a bigram model, while our personal experience is that this already increases the chord extraction performance by 2 % [11]. Note too that the system of Khadkevich & Omologo first extracts one global key before it starts to extract the chords. This means that the influence of higher order contextual modeling on key extraction was not investigated. In the present paper this particular influence is assessed in detail because we conduct all our experiments with a system that performs a simultaneous extraction of chords and local keys.

In the remainder of this paper, we will first give a detailed description of our system in Section 2. Then the used data sets and evaluation measure are discussed in Section 3 and the experimental results in Section 4. We end with some concluding remarks in Section 5.

2. ARCHITECTURE OF THE SYSTEM

2.1. Overview

The input audio file is supplied to a three stage front-end. First, it is resampled to 8 kHz and converted to mono. The resulting waveform is then split into 150 ms long frames with a step size of 20 ms, and for each frame, a chroma profile is calculated. This is a logarithmic spectral representation where the octave information has been discarded such that it represents the intensity of each of the 12 pitch classes in the frame. Since its introduction by Fujishima [12], the chroma profile has been the preferred feature of almost all key and chord extraction algorithms. Finally, the subsequent chroma profiles are integrated over 11 frames (220 ms), and the integrated profiles are supplied to the back-end. Due to the smoothing, they can be supplied at a rate of one per 220 ms which improves the processing speed.

The back-end traces the most likely state sequence through a huge finite state machine. If a trigram contextual model is used, each state of that machine represents a distinct combination of two key-chord pairs representing the actual situation and the context respectively. The machine comprises (24 * 48) * (24 * 47) ≈ 1.3 million states. The 24 refers to the 24 keys (12 tonics and two modes: major/minor) under consideration, the 48 to the 48 chords (12 roots, 4 chord types: major, minor, diminished and augmented) and the 47 reflects the assumption that a key change can only occur in combination with a chord change. Obviously, the search for the best solution will gradually extend only those partial solutions whose probability is within a range of that of the best one. This so-called beam search strategy is borrowed from continuous speech recognition systems (e.g. [13]). By storing the history in each state, no back-tracking is needed and the required storage capacity remains manageable.

2.2. Chroma extraction

Simply folding a logarithmic frequency spectrum into one octave produces a chroma profile that contains contributions of the harmonics because a note usually contains harmonics of its fundamental frequency. For instance, a third harmonic would add evidence to the chroma a fifth above the fundamental, even though that note has not necessarily been played.

Instead of accounting for harmonics in the templates, like in [3], we chose to deal with this phenomenon in the chroma extraction step. We therefore adopt multiple pitch tracking techniques to resolve partials. A comb filter is applied on a peak-picked spectrum to discover harmonic relations between the peaks such that their energy can be assigned to one of the candidate fundamental frequencies once there is enough harmonic evidence to support this hypothesized F0. Because the sample frequency was reduced to 8000 Hz and because we require each fundamental to be supported by at least one harmonic, the highest detectable fundamental frequency is 2000 Hz. We argue that any note higher than this upper limit is most likely a melody note and hence, not contributing to a chord. Similarly, we impose a lower limit of 100 Hz on the grounds that any note below this limit is most likely a bass note, not contributing a unique chroma to the chord. A more thorough description of the algorithm can be found in [14]. The advantage of the chosen approach is that it enables the use of binary chord templates in the back-end, and these templates can be directly derived from music theory.

2.3. Probabilistic framework

The back-end implements a unified probabilistic framework for the simultaneous recognition of chords and keys. Its objective is to retrieve the most likely sequence of states $\hat{Q}$ for the acoustic observation sequence $\mathbf{X}$. Each state $q_n = (k_n, c_n, k_{prev}, c_{prev})$ represents the combination of the key-chord pair (k_n, c_n) assigned to a vector $\mathbf{x}_n$ and the key-chord pair (k_{prev}, c_{prev}) from which the transition to the present pair was made. Using Bayes's rule, the desired state sequence $\hat{Q}$ follows from

$$\hat{Q} = \arg\max_{Q} P(Q)\, P(\mathbf{X}|Q)$$

2.3.1. Acoustic model

The acoustic likelihood $P(\mathbf{X}|Q)$ is calculated by means of an acoustic model. We consider the key and chord labels k_n and c_n as two independent means of testing whether an observation vector complies with a certain state, i.e. $P(\mathbf{x}_n|q_n) = P(\mathbf{x}_n|c_n)P(\mathbf{x}_n|k_n)$. By making the standard assumption that acoustic observations emitted in the same state are independent, the acoustic likelihood can be factorized as follows

$$P(\mathbf{X}|Q) = \prod_{n=1}^{N} P(x_n|c_n)P(x_n|k_n)$$

Because of the scarcity of training data (in comparison with other fields such as speech processing), we did not attempt to train any acoustic models (e.g. Gaussian mixture models that incorporate a lot of free parameters). Instead, we opted for a model that just penalizes the dissimilarities between $\mathbf{x}_n$ and a template vector representing either k_n or c_n. Note that we also use the same observations for computing both key and chord acoustic likelihoods, as opposed to [7] where the inputs of the key acoustic model are the result of an integration over a much longer time than the one used for generating the inputs to the chord acoustic model. Such a long integration time possibly allows for a better key estimation on acoustic features alone, but we fear that this also smears the key boundaries.

The chord acoustic model $P(\mathbf{x}_n|c_n)$ utilizes a template for c_n. This template is composed of binary components: 1 for a chroma that is theoretically present in the chord and 0 for one that is not. The components are treated independently of each other so that the acoustic likelihood is obtained as a product of 12 likelihoods. The latter are computed by means of two Gaussian models: one for each possible value (0 or 1) of the template component [11].

The key acoustic model $P(\mathbf{x}_n|k_n)$ also uses templates, namely the non-binary templates defined by Temperley. They represent the stability of the 12 pitch classes relative to a given key and are based on the Krumhansl–Schmuckler profiles, but specifically adjusted for computational key-finding [15]. The measure used in our system is the cosine similarity between the key template and the observation vector.

2.3.2. Prior transition model

The prior probability $P(Q)$ is computed by making the second order Markov assumption. This means that

$$P(K, C) = \prod_{n=1}^{N} P(k_n, c_n|k_{n-1}, c_{n-1}, k_{prev}, c_{prev})$$

We distinguish two types of state transitions: self-transitions and transitions to another state. The self-transition probabilities are supposed not to depend on the context (k_{prev}, c_{prev}). Furthermore, since a key change is always presumed to entail a chord change as well, the self-transitions actually constitute a chord duration model. We opted for a very simple geometric model, represented by a single $P_c = P(q_n = q_{n-1})$ for all states. The parameter P_c is chosen in such a way that the mean chord duration becomes equal to some predetermined value $\overline{d_c}$ (e.g. retrieved from an annotated corpus). This is achieved by satisfying the relation

$$\frac{P_c}{1 - P_c} = \frac{\overline{d_c}}{h} \iff P_c = 1 - \frac{h}{\overline{d_c}}$$

where h is the time shift between successive frames being processed.

The transitions between different states constitute a musicological model, in our case a stochastic trigram model, which is further decomposed into a key transition model and a chord transition model:

$$P(K, C) = \prod_{\substack{n=1 \\ c_n \neq c_{n-1}}}^{N} P(k_n|k_{n-1}, k_{prev}, c_n, c_{n-1}, c_{prev}) \\ P(c_n|k_{n-1}, k_{prev}, c_{n-1}, c_{prev})$$

The key transition model $P(k_n|...)$ is further simplified by arguing that the influence of the chords c_n, c_{n-1}, c_{prev} on the identity of the current key label is negligible compared to that of the keys k_{n-1}, k_{prev}. Consequently, the key transition model gets reduced to $P(k_n|k_{n-1}, k_{prev})$. This trigram key model now allows us to explicitly take into account the fact that the musical concept of key does not allow for key changes at every chord. We realize this by permitting k_n to differ from k_{n-1} only when the latter is equal to k_{prev}. This means that

$$\begin{aligned} P(k_n|k_{n-1}, k_{prev}) &= P(k_n|k_{n-1}) \qquad (k_{n-1} = k_{prev}) \\ &= 1 \qquad (k_{n-1} \neq k_{prev} \;\&\; k_n = k_{n-1}) \\ &= 0 \qquad (k_{n-1} \neq k_{prev} \;\&\; k_n \neq k_{n-1}) \end{aligned}$$

and consequently, that only a bigram key transition model is required. We can thus reuse our previously developed bigram model [11]. That model is based on Lerdahl's regional distance [16, p.68], which expresses numerically the perceptual distance between two keys. We assume that keys which are perceptually close are also likely to appear in sequence and thus receive a high transition probability. A weaknesses in this assumption however, is that this gives inadequate probabilities to some key changes such as the "gear change" or the "one up" which are common in pop music, but not in music of the Common Practice period on which Lerdahl's theory is based.

Since a chord change will usually not be accompanied by a key change, the key transition model must definitely accommodate a strong self-transition. We therefore assign a probability P_k to the chance of staying in the same key and we fix this to 0.99. The remaining probability mass is divided over the key changes according to Lerdahl's regional distance. A distance d is converted to a probability by applying a normalized exponential of the form $e^{-\nu d}$, where ν is inversely proportional to the mean perceptual distance between keys.

The chord transition model $P(c_n|k_{n-1}, k_{prev}, c_{n-1}, c_{prev})$ can be simplified enormously by rewriting it in terms of relative chords and key modes. A relative chord c' in a key k is obtained by expressing the root of the chord c as a distance to the tonic of a key k and combining it with its chord type. This representation is more in line with the way scholars study harmony. We also interpret all chords in the same key, namely k_{n-1}. This is not just a simplification to reduce the number of parameters, but we argue that this also is in accordance with common practice in harmonic analysis. Chords at a key change often give an indication of both the preceding and the following key (by also belonging to the other key, being a so called pivot chord, or by being perceptually close to a chord that does belong to the other key). They thus are also meaningful when interpreted in the key at the other side of the change, whereas the movement of relative chords interpreted in different keys does not really make sense. Then we construct a model where the chord transition probability only depends on the mode m_{n-1} of key k_{n-1} and on the relative chords. This way we exploit the parallelism between keys that differ in tonic, but not in mode. We end up with distinct transition models for major and minor keys, for which distinct idiomatic chord sequences exist. Systems such as the ones proposed in [1, 2, 3] which do not jointly extract the key and chord labels would not be able to do so.

All together, the musicological model can be simplified to a product of two partial models comprising but a limited number of transitions:

$$P(K,C) = \prod_{\substack{n=1 \\ c_n \neq c_{n-1}}}^{N} P(k_n|k_{n-1}, k_{prev}) \\ P(c'_n|m_{n-1}, c'_{n-1}, c'_{prev})$$

Due to this limited number of transitions, one can harvest much more training examples per transition from the training corpus and get more reliable probabilities.

2.3.3. Summary

For computational reasons, the search algorithm works with log-probabilities and in order to have control over the relative importances of the different sub-models, multiplicative balance parameters α, δ, μ and κ are being introduced. Thus, the function to maximize is given by

$$\hat{Q} = \arg\max_{Q} \sum_{n=1}^{N} \Big[\log P(\mathbf{x_n}|c_n) + \alpha \log P(\mathbf{x_n}|k_n) \\ + \delta \mathcal{L}_D + \mu \mathcal{L}_M \Big]$$

$$\begin{aligned} \mathcal{L}_D &= \log P_c & (q_n = q_{n-1}) \\ &= \log(1 - P_c) & (q_n \neq q_{n-1}) \\ \mathcal{L}_M &= \big[\kappa \log P(k_n|k_{n-1}, k_{prev}) + & (c_n \neq c_{n-1}) \\ &\quad (1-\kappa) \log P(c'_n|c'_{n-1}, c'_{prev}, m_{n-1})\big] & \\ &= 0 & (c_n = c_{n-1}) \end{aligned}$$

3. DATA SETS AND EVALUATION MEASURE

3.1. Data sets

In order to assess the performance of our system, we need data with accompanying ground truth labels, as well as an evaluation measure. We have two audio collections at our disposal. The first one is a private collection of 142 manually annotated 30 s excerpts of music pieces in a variety of genres and tempi, hereafter called the SEMA set. We will use it here as the development set for optimizing the balance parameters. The second set consists of the 210 songs that were used in the MIREX 2009[1] chord estimation contest. It is composed of full albums by the Beatles (174 songs), Queen (18 songs) and Zweieck (18 songs).

For investigating how well a relative chord model learned on one data set scales to another data set, we have built three relative chord transition models: one on SEMA, one on MIREX and one on 9GDB [17], a symbolic data set of 855 songs with chord and key labels (without duration information) distributed over 9 genres. The 9GDB set was originally used to classify songs into genres solely based on their constituent chords and keys.

	SEMA		MIREX		9GDB	
N	major	minor	major	minor	major	minor
1	984	1268	12862	2191	34011	6026
2	866	1125	12379	2074	32580	5600
3	766	1012	11867	1897	31286	5252

Table 1. Total number of valid N-grams per key mode in the three data sets introduced in the text.

3.2. Chord transition models

To construct an N-gram relative chord model, all annotated chords are mapped to triads and subsequent identical chords (after their mapping to triads) are merged. Then, a sliding window of length N is moved over the chord annotations where every chord $c_i, \forall i \in \{1, \ldots, N\}$ is converted to a relative chord c'_i by interpreting it in the key of c_{N-1}. Based on these new annotations, we determine the N-gram counters from which we finally derive a Kneser-Ney back-off model [18] for each key mode. All trigrams whose count is larger than $K = 4$ are converted to a genuine trigram probability using a discount of $D = 2$. The remaining trigram probabilities are then computed by backing off to a bigram model. The same procedure, this time applied on bigram counts, is adopted to compute the bigram probabilities (with the same K and D). Finally, if not all relative chords are seen in the training data, 5 % of the unigram probability mass is reserved for these unobserved chords.

Table 1 comprises the number of valid N-grams (entirely composed of maj–min-dim–aug chords) per key mode (major–minor) for each of the three data sets mentioned above.

Each obtained model is characterized by a model perplexity, defined as the inverse of the exponent of the mean log probability of a relative chord c', given the previous $N-1$ relative chords and the mode:

$$PP(N) = e^{-\sum_{c'_1,\ldots,c'_N,m} P(c'_1,..,c'_N,m)\ \log P(c'_N|c'_1,..,c'_{N-1},m)}$$

The sum is taken over all valid relative chord combinations and $P(c'_1, .., c'_N, m)$ follows directly from the corresponding N-gram counter computed during the training phase.

Figure 1 shows the model perplexities as a function of N for the three models we derived: one per data set. Since $N = 0$ means that no context is taken into account and that all chords have the same probability, the figure shows that just taking the prior probabilities of the chords into account ($N =$

[1] http://www.music-ir.org/mirex/wiki/2009:Audio_Chord_Detection

1) is very effective since it reduces the perplexity by a factor 3.5 to 4. Taking one or two context symbols into account ($N = 2$ and $N = 3$) leads to a further decline in perplexity for all models, although with a progressively milder slope.

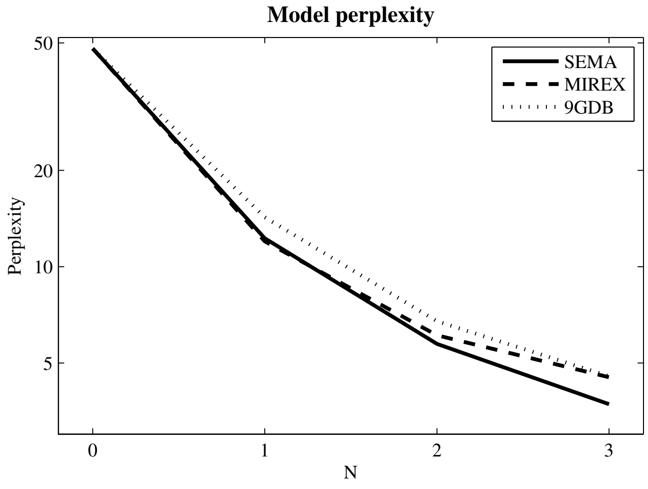

Fig. 1. The model perplexity (per key mode) as function of N for each data set.

3.3. Duration model

The duration model is kept constant and the controlling parameter P_c is set to attain a mean chord duration of 1.71 s. The latter is exactly the mean chord duration that was observed in the SEMA data set.

3.4. Evaluation measure

System performance is quantified by the percentage of the time the extracted key or chord labels match with the annotated key or chord labels. To avoid a disputable ranking of multiple possible mappings of complex chords to triads, we restrict the chord evaluation to frames where the annotated chord is one of the basic triads (maj–min–dim–aug, including inversions). This leaves us with 62.56 % of the data for the SEMA set and 77.44 % for the MIREX set. Key extraction performance is measured over the whole data set. For both chord and key evaluation, only perfect matches are considered correct.

4. EXPERIMENTAL RESULTS

Per data set (SEMA and MIREX) we have conducted two types of experiments: experiments with musicological models that were trained on the same data set and experiments with models that were trained on an independent data set, namely 9GDB. The optimal parameters $\alpha, \delta, \mu and \kappa$ balancing the different models in the auxiliary function being maximized are determined by means of a grid search on the SEMA set. The same factors are then reused in the experiments on the MIREX set. The results of our experiments are summarized in Table 2 and Table 3. They are compared with a bigram modeling system that was previously described in [8]. We use the bigram probabilities of the Kneser-Ney backoff model we constructed here to configure its relative chord model.

rel.chord	bigram		trigram		p-value
model	Chord	Key	Chord	Key	Key
SEMA	73.16	70.66	73.84	74.03	0.07
9GDB	72.69	63.06	73.12	66.80	0.05

Table 2. Chord and local key extraction score on the SEMA set for bigram and trigram relative chord models learned on the SEMA and the 9GDB set and significance of the key improvement

rel.chord	bigram		trigram		p-value
model	Chord	Key	Chord	Key	Key
MIREX	78.29	76.64	78.81	78.44	0.29
9GDB	76.97	65.10	77.15	69.36	10^{-4}

Table 3. Chord and local key extraction score on the MIREX set for bigram and trigram relative chord models learned on the MIREX and the 9GDB set and significance of the key improvement

Apparently, moving from a bigram to a trigram musicological model does not improve the chord recognition performance. For the key extraction on the other hand, trigram modeling does enhance the results. The p-values in the table were obtained using the sign test, and they show that in 3 out of the 4 cases, the improvements are significant. Most important is the significance of the results obtained the independent musicological model 9GDB. The results confirm our earlier finding [8] that key extraction is more sensitive to the musicological model than chord extraction. However, using a trigram musicological model strongly increases the computational requirements. The time to complete is multiplied by a factor of 34 for the trigram system, taking 431 % of the duration of the processed files instead of 13 % for the bigram system.

Zooming in on the produced outputs, we see that changing the musicological model affects the key outputs of only 10 to 20 % of the processed files, but the accuracy usually changes from zero to perfect (or vice versa) for these files. Obviously, our strategy of strongly dissuading a key change inside a file is largely responsible for this behavior. Changes in the chord labels occur in nearly all files, but the changes are usually very localized and thus only slightly affecting the accuracies.

5. CONCLUSION

We investigated a further extension of our simultaneous chord and local key extraction system which consists of replacing a bigram model of prior musicological information by a more sophisticated trigram model. We showed that this extension required a fundamental modification of the search, increasing the system complexity. On the other hand, the extension does not require any new key transition model component (a bigram model), but the additional context key is used to constrain the key transitions. For the creation of the required relative chord transition models, we opted for a back-off model with Kneser-Ney smoothing.

An analysis of perplexities clearly showed that the predictive power of a trigram musicological model is significantly higher than that of a bigram model. The experimental validation of the new models on the other hand revealed that this gain is not translated into a better chord extraction. Nevertheless, there is a modest though statistically significant improvement of the key extraction accuracy. We contemplate that in order to improve the chord extraction, we need a better acoustical model, a possible direction for future work.

6. ACKNOWLEDGMENTS

This work originates from the "Semantic description of musical audio (GOASEMA)" project, funded by the "Bijzonder Onderzoeksfonds (BOF)", Ghent University under contract GOA-1250604. We also thank Carlos Pérez-Sancho for providing us with the 9GDB symbolic data set.

7. REFERENCES

[1] Alexander Sheh and Daniel P.W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proceedings of the 4th International Conference on Music Information Retrieval*, 2003.

[2] Juan Pablo Bello and Jeremy Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005.

[3] Hélène Papadopoulos and Geoffroy Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and HMM," in *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, 2007.

[4] Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[5] Katy Noland and Mark Sandler, "Influences of signal processing, tone profiles and chord progressions on a model for estimating the musical key from audio," *Computer Music Journal*, vol. 33, no. 1, 2009.

[6] Kyogu Lee and Malcolm Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.

[7] Thomas Rocher, Matthias Robine, Pierre Hanna, and Laurent Oudre, "Concurrent estimation of chords and keys from audio," in *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.

[8] Johan Pauwels, Jean-Pierre Martens, and Marc Leman, "Improving the key extraction accuracy of a simultaneous key and chord estimation system," in *International Workshop on Advances in Music Information Research*, 2011.

[9] Ricardo Scholz, Emmanuel Vincent, and Frédéric Bimbot, "Robust modeling of musical chord sequences using probabilistic n-grams," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[10] Maksim Khadkevich and Maurizio Omologo, "Use of hidden Markov models and factored language models for automatic chord recognition," in *Proceedings of the 10th International Conference on Music Information Retrieval*, 2009.

[11] Johan Pauwels and Jean-Pierre Martens, "Integrating musicological knowledge into a probabilistic system for chord and key extraction," in *Proceedings of the 128th Convention of the AES*, 2010.

[12] Takuya Fujishima, "Realtime chord recognition of musical sound: a system using Common Lisp Music," in *Proceedings of the International Computer Music Conference*, 1999.

[13] Jin'ichi Murakami and Shigeki Sagayama, "An efficient algorithm for using word trigram models for continuous speech recognition," in *Proceedings of the 4th Australasian International Conference on Speech Science and Technology*, 1992.

[14] Matthias Varewyck, Johan Pauwels, and Jean-Pierre Martens, "A novel chroma representation of polyphonic music based on multiple pitch tracking techniques," in *Proceedings of the 16th ACM International Conference on Multimedia*, 2008.

[15] David Temperley, *The cognition of basic musical structures*, MIT Press, 1999.

[16] Fred Lerdahl, *Tonal pitch space*, Oxford University Press, New York, 2001.

[17] Carlos Pérez-Sancho, David Rizo, and José M. Iñesta, "Genre classification using chords and stochastic language models," *Connection science*, vol. 21, no. 2–3, 2009.

[18] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1995, vol. 1.