APPROXIMATE NOTE TRANSCRIPTION FOR THE IMPROVED IDENTIFICATION OF DIFFICULT CHORDS

Matthias Mauch and Simon Dixon
Queen Mary University of London, Centre for Digital Music
{matthias.mauch, simon.dixon}@elec.qmul.ac.uk

ABSTRACT

The automatic detection and transcription of musical chords from audio is an established music computing task. The choice of chord profiles and higher-level time-series modelling have received a lot of attention, resulting in methods with an overall performance of more than 70% in the MIREX Chord Detection task 2009. Research on the front end of chord transcription algorithms has often concentrated on finding good chord templates to fit the chroma features. In this paper we reverse this approach and seek to find chroma features that are more suitable for usage in a musically-motivated model. We do so by performing a prior approximate transcription using an existing technique to solve non-negative least squares problems (NNLS). The resulting NNLS chroma features are tested by using them as an input to an existing state-of-the-art high-level model for chord transcription. We achieve very good results of 80% accuracy using the song collection and metric of the 2009 MIREX Chord Detection tasks. This is a significant increase over the top result (74%) in MIREX 2009. The nature of some chords makes their identification particularly susceptible to confusion between fundamental frequency and partials. We show that the recognition of these diffcult chords in particular is substantially improved by the prior approximate transcription using NNLS.

Keywords: chromagram, chord extraction, chord detection, transcription, non-negative least squares (NNLS).

1. INTRODUCTION

Chords are not only of theoretical interest for the understanding of Western music. Their practical relevance lies in the fact that they can be used for music classification, indexing and retrieval [2] and also directly as playing instructions for jazz and pop musicians. Automatic chord transcription from audio has been the subject of tens of research papers over the past few years. The methods usually rely on the low-level feature called chroma, which is a mapping of the spectrum to the twelve pitch classes C,...,B, in which the pitch height information is discarded. Nevertheless, this feature is often sufficient to recognise chords because chord labels themselves remain the same whatever octave the constituent notes are played in. An exception is the lowest note in a chord, the bass note, whose identity is indeed notated in chord labels. Some research papers have taken advantage of the additional information conveyed by the bass note by introducing special bass chromagrams [18, 12] or prior bass note detection [21].

There is much scope in developing musical models to infer the most likely chord sequence from the chroma features. Many approaches use models of metric position [16], the musical key [8, 21], or combinations thereof [12], as well as musical structure [13], to increase the accuracy of the chord transcription. Although in this work we will also use such a high-level model, our main concern will be the low-level front end.

Many previous approaches to chord transcription have focussed on finding a set of chord profiles, each chord profile being a certain chroma pattern that describes best the chroma vectors arising while the chord is played. It usually includes the imperfections introduced into the chromagram by the upper partials of played notes. The shape of each pattern is either theoretically motivated (e.g. [15]) or learned, usually using (semi-) supervised learning (e.g. [8, 9]). A few approaches to key and chord recognition also emphasise the fundamental frequency component before producing the chromagrams [5, 18] or use a greedy transcription step to improve the correlation of the chroma with true fundamental frequencies [19]. Emphasising fundamental frequencies before mapping the spectrum to chroma is preferable because here all spectral information can be used to determine the fundamental frequencies – *before* discarding the octave information.

However, in order to determine the note activation, the mentioned approaches use relatively simple one-step transforms, a basic form of approximate transcription. A different class of approaches to approximate transcription assumes a more realistic linear generative model in which the spectrum (or a log-frequency spectrum) Y is considered to be approximately represented by a linear combination of note profiles in a dictionary matrix E, weighted by the activation vector x, with $x \geq 0$:

$$Y \approx Ex \tag{1}$$

This model conforms with our physical understanding of how amplitudes of simultaneously played sounds add up [1] .

[1] Like the one-step transforms, the model assumes the absence of si-

Approaches to finding the activation vector x in (1) differ from the one-step transforms in that they involve iterative re-weighting of the note activation values [1]. To our knowledge, such a procedure has not been used to generate chromagrams or otherwise conduct further automatic harmony analysis. Unlike traditional transcription approaches, we are not directly interested in note events, and the sparsity constraints required in [1] need not be taken into account. This allows us to use a standard procedure called non-negative least squares (NNLS), as will be explained in Section 2.

The motivation for this is the observation that the partials of the notes played in chords compromise the correct recognition of chords. The bass note in particular usually has overtones at frequencies where other notes have their fundamental frequencies. Interestingly, for the most common chord type in Western music, the major chord (in root position), this does not pose a serious problem, because the frequencies of the first six partials of the bass note coincide with the chord notes: for example, a C major chord (consisting of C, E and G) in root position has the bass note C, whose fist six partials coincide with frequencies at pitches C, C, G, C, E, G. Hence, using a simple spectral mapping works well for major chords. But even just considering the first inversion of the C major chord (which means that now E is the the bass note), leads to a dramatically different situation: the bass note's first six partials coincide with E, E, B, E, G♯, B – of which B and G♯ are definitely not part of the C major triad. Of course, the problem does not only apply to the bass note, but to all chord notes [2].

This is a problem that can be eliminated by a perfect prior transcription because no partials would interfere with the signal. Section 2 focusses mainly on describing our approach to an approximate transcription using NNLS, and also gives an outline of the high-level model we use. In Section 3 we demonstrate that the problem does indeed exist and show that the transcription capabilities of the NNLS algorithm can improve the recognition of the affected chords. We give a brief discussion of more general implications and future work in Section 4, before presenting our conclusions in Section 5.

2. METHOD

This section is concerned with the technical details of our method. Most importantly, we propose the use of NNLS-based approximate note transcription, prior to the chroma mapping, for improved chord recognition. We call the resulting chroma feature *NNLS chroma*. To obtain these chroma representations, we first calculate a log-frequency spectrogram (Subsection 2.2), pre-process it (Subsection 2.3) and perform approximate transcription using the NNLS algorithm (Subsection 2.4). This transcription is then wrapped to chromagrams and beat-synchronised (Section 2.5). Firstly, however, let us briefly consider the high-level musical model which takes as input the chroma features, and which we use to test the effect of different chromagrams on chord transcription accuracy.

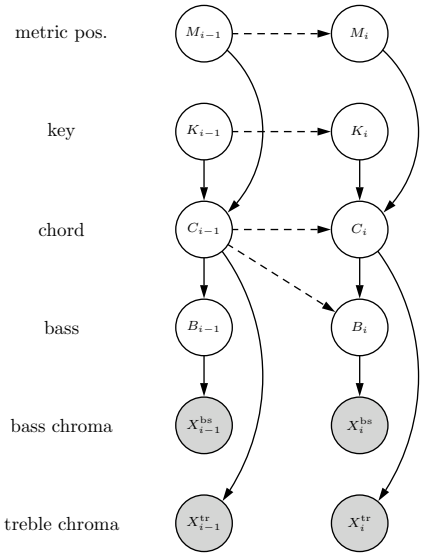


Figure 1: High-level dynamic Bayesian network, represented as two slices corresponding to two generic consecutive beats. Random variables are shown as nodes, of which those shaded grey are observed, and the arrows represent direct dependencies (inter-slice arrows are dashed).

2.1 High-level Probabilistic Model

We use a modification of a dynamic Bayesian network (DBN) for chord recognition proposed in [10], which integrates in a single probabilistic model the hidden states of metric position, key, chord, and bass note, as well as two observed variables: chroma and bass chroma. It is an expert model whose structure is motivated by musical considerations; for example, it enables to model the tendency of the bass note to be present on the first beat of a chord, and the tendency of the chord to change on a strong beat. The chord node distinguishes 121 different states: 12 for each of 10 chord types (major, minor, major in first inversion, major in second inversion, major 6th, dominant 7th, major 7th, minor 7th, diminished and augmented) and one "no chord" state. With respect to the original method, we have made some slight changes in the *no chord* model and the metric position model [3]. The DBN is implemented using Murphy's BNT Toolbox [14], and we infer the jointly most likely state sequence in the Viterbi sense.

2.2 Log-frequency Spectrum

We use the discrete Fourier transform with a frame length of 4096 samples on audio downsampled to 11025 Hz. The DFT length is the shortest that can resolve a full tone in the bass region around MIDI note 44 [4], while using a Ham-

nusoid cancellation.

[2] For example, a major third will create some energy at the major 7th through its third partial.

[3] The *no chord* model has been modified by halving the means of the multivariate Gaussian used to model its chroma, and the metric position model is now fully connected, i.e. the same low probability of 0.0167 is assigned to missing 1, 2 or three beats.

[4] Smaller musical intervals in the bass region occur extremely rarely.

ming window. We generate a spectrogram with a hop size of 2048 frames (≈0.05s).

We map the magnitude spectrum onto bins whose centres are linearly-spaced in log frequency, i.e. they correspond to pitch (e.g. [17]), with bins spaced a third of a semitone apart. The mapping is effectuated using cosine interpolation on both the linear and logarithmic scales: first, the DFT spectrum is upsampled to a highly oversampled frequency representation, and then this intermediate representation is mapped to the desired log-frequency representation. The two operations can be performed as a single matrix multiplication. This calculation is done separately on all frames of a spectrogram, yielding a log-frequency spectrogram $Y = (Y_{k,m})$.

Assuming equal temperament, the global tuning of the piece is now estimated from the spectrogram. Rather than adjusting the dictionary matrix we then update the log-frequency spectrogram via linear interpolation, such that the centre bin of every semitone corresponds to the correct frequency with respect to the estimated tuning [10]. The updated log-frequency spectrogram Y has 256 $1/3$-semitone bins (about 7 octaves), and is hence much smaller than the original spectrogram. The reduced size enables us to model it efficiently as a sum of idealised notes, as will be explained in Subsection 2.4.

2.3 Pre-processing the Log-frequency Spectrum

We use three different kinds of pre-processing on the log-frequency spectrum:

o : original – no pre-processing,

sub : subtraction of the background spectrum [3], and

std : standardisation: subtraction of the background spectrum and division by the running standard deviation.

To estimate the background spectrum we use the running mean $\mu_{k,m}$, which is the mean of a Hamming-windowed, octave-wide neighbourhood (from bin $k-18$ to $k+18$). The values at the edges of the spectrogram, where the full window is not available, are set to the value at the closest bin that is covered. Then, $\mu_{k,m}$ is subtracted from $Y_{k,m}$, and negative values are discarded (method *sub*). Additionally dividing by the respective running standard deviation $\sigma_{k,m}$, leads to a running standardisation (method *std*). This is similar to spectral whitening (e.g. [6]) and serves to discard timbre information. The resulting log-frequency spectrum of both pre-processing methods can be calculated as

$$Y^{\rho}_{k,m} = \begin{cases} \frac{Y_{k,m}-\mu_{k,m}}{\sigma^{\rho}_{k,m}} & \text{if } Y_{k,m}-\mu_{k,m} > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where $\rho = 0$ or $\rho = 1$ for the cases *sub* and *std*, respectively.

2.4 Note Dictionary and Non-Negative Least Squares

In order to decompose a log-frequency spectral frame into the notes it has been generated from, we need two basic ingredients: a note dictionary E, describing the assumed profile of (idealised) notes, and an inference procedure to determine the note activation patterns that result in the closest match to the spectral frame.

We generate a dictionary of idealised note profiles in the log-frequency domain using a model with geometrically declining overtone amplitudes [5],

$$a_k = s^{k-1} \quad (3)$$

where the parameter $s \in (0,1)$ influences the spectral shape: the smaller the value of s, the weaker the higher partials. Gomez [5] favours the parameter $s = 0.6$ for her chroma generation, in [13] $s = 0.9$ was used. We will test both possibilities, and add a third possibility, where s is linearly spaced (LS) between $s = 0.9$ for the lowest note and $s = 0.6$ for the highest note. This is motivated by the fact that resonant frequencies of musical instruments are fixed, and hence partials of notes with higher fundamental frequency are less likely to correspond to a resonance. In each of the three cases, we create tone patterns over seven octaves, with twelve tones per octave: a set of 84 tone profiles. The fundamental frequencies of these tones range from A0 (at 27.5 Hz) to G♯6 (at approximately 3322 Hz). Every note profile is normalised such that the sum over all the bins equals unity. Together they form a matrix E, in which every column corresponds to one tone.

We assume now that—like in Eqn. (1)—the individual frames of the log-frequency spectrogram Y are generated approximately as a linear combination $Y_{\cdot,m} \approx Ex$ of the 84 tone profiles. The problem is to find a tone activation pattern x that minimises the Euclidian distance

$$||Y_{\cdot,m} - Ex|| \quad (4)$$

between the linear combination and the data, with the constraint $x \geq 0$, i.e. all activations must be non-negative. This is a well-known mathematical problem called the non-negative least squares (NNLS) problem. Lawson and Hanson [7] have proposed an algorithm to find a solution, and since (in our case) the matrix E has full rank and more rows than columns, the solution is also unique. We use MATLAB's implementation of this algorithm. Again, all frames are processed separately, and we finally obtain an NNLS transcription spectrum S in which every column corresponds to one audio frame, and every row to one semitone. Alternatively, we can choose to omit the approximate transcription step and copy the centre bin of every semitone in Y to the corresponding bin of S [17].

2.5 Chroma, Bass Chroma and Beat-synchronisation

The DBN we use to estimate the chord sequence requires two different kinds of chromagram: one general-purpose chromagram that covers all pitches, and one bass-specific chromagram that is restricted to the lower frequencies. We emphasise the respective regions of the semitone spectrum by multiplying by the pitch-domain windows shown in Figure 2, and then map to the twelve pitch classes by summing the values of the respective pitches.

log-freq. spectrum	no NNLS	NNLS $s = 0.6$	NNLS $s = 0.9$	NNLS LS
o	38.6	43.9	43.1	47.5
sub	74.5	74.8	71.5	73.8
std	79.0	**80.0**	76.5	78.6

(a) MIREX metric – correct overlap in %

log-freq. spectrum	no NNLS	NNLS $s = 0.6$	NNLS $s = 0.9$	NNLS LS
o	31.0	35.1	33.9	37.4
sub	58.1	58.2	56.1	57.3
std	61.3	62.7	62.0	**63.3**

(b) metric using all chord types – correct overlap in %

Table 1: Results of the twelve methods in terms of the percentage of correct overlap. Table (a) shows the MIREX metric, which distinguishes only 24 chords and a "no chord" state, Table (b) is shows a finer metric that distinguishes 120 chords and a "no chord" state.

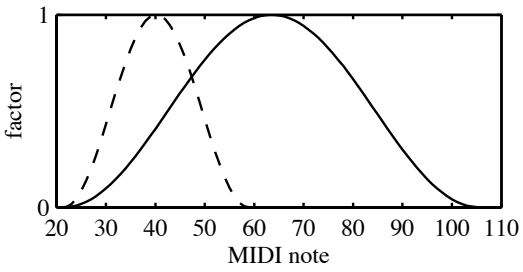


Figure 2: Profiles applied to the log-frequency spectrum before the mapping to the main chroma (solid) and bass chroma (dashed).

Beat-synchronisation is the process of summarising frame-wise features that occur between two beats. We use the beat-tracking algorithm developed by Davies [4], and obtain a single chroma vector for each beat by taking the median (in the time direction) over all the chroma frames between two consecutive beat times. This procedure is applied to both chromagrams, for details refer to [10]. Finally, each beat-synchronous chroma vector is normalised by dividing it by its maximum norm. The chromagrams can now be used as observations in the DBN described in Section 2.1.

3. EXPERIMENTS AND RESULTS

Our test data collection consists of the 210 songs used in the 2009 MIREX Chord Detection task, together with the corresponding ground truth annotations [11]. We run 12 experiments varying two parameters: the preprocessing type (*o*, *sub* or *std*, see Section 2.3), and the kind of NNLS setup used ($s = 0.6$, $s = 0.9$, LS, or direct chroma mapping, see Section 2.4).

3.1 Overall Accuracy

The overall accuracy of the 12 methods in terms of the percentage of correct overlap

$$\frac{\text{duration of correctly annotated chords}}{\text{total duration}} \times 100\%$$

is displayed in Table 1: Table 1a shows results using the MIREX metric which distinguishes only two chord types and the "no chord" label, and 1b shows results using a finer evaluation metric that distinguishes all 121 chord states that the DBN can model; see also [10, Chapter 4].

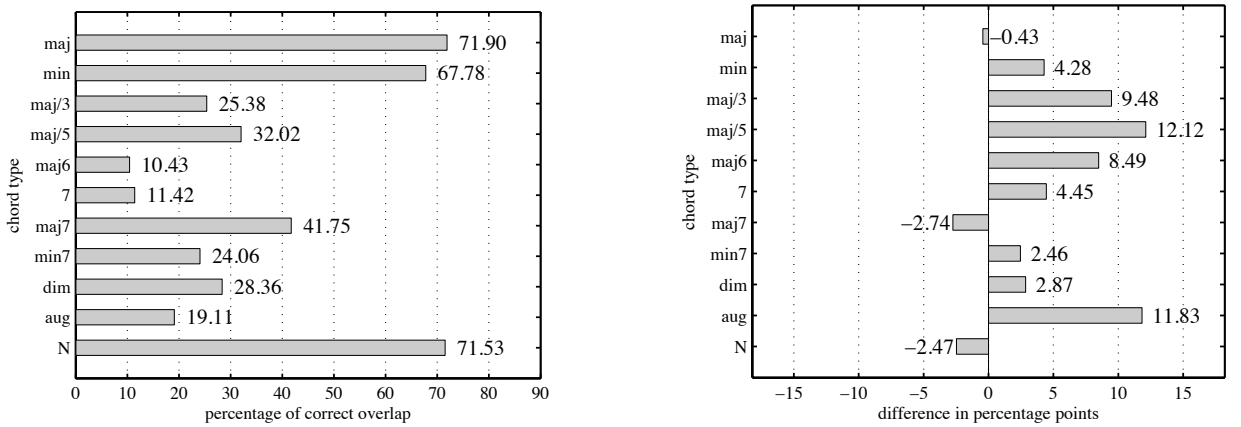
When considering the MIREX metric in Table 1a it is immediately clear that one of the decisive factors has been the spectral standardisation: all four *std* methods clearly outperform the respective analogues with *sub* preprocessing or no preprocessing. We performed a 95% Friedman multiple comparison analysis on the song-wise results of the *std* methods: except for the difference between no NNLS and LS all differences are significant, and in particular the NNLS method using $s = 0.6$ significantly outperforms all other methods, achieving 80% accuracy. With a p-value of 10^{-10} in the Friedman test, this is also a highly significant increase of nearly 6 percentage points over the 74% accuracy achieved by the highest scoring method [20] in the 2009 MIREX tasks.

In Table 1b the results are naturally lower, because a much finer metric is used. Again, the *std* variants perform best, but this time the NNLS chroma with the linearly spaced s has the edge, with 63% accuracy. (Note that this is still higher than three of the scores in the MIREX task evaluated with the MIREX metric.) According to a 95% Friedman multiple comparison test, the difference between the methods *std*-LS and *std*-0.6 is not significant. However, both perform significantly better than the method without NNLS for this evaluation metric which more strongly emphasises the correct transcription of difficult chords.

The reason for the very low performance of the *o* methods without preprocessing is the updated model of the "no chord" state in the DBN. As a result, many chords in noisier songs are transcribed as "no chord". However, this problem does not arise in the *sub* and *std* methods, where the removal of the background spectrum suppresses the noise. In these methods the new, more sensitive "no chord" model enables very good "no chord" detection, as we will see in the following subsection.

3.2 Performance of Individual Chords

Recall that our main goal, as stated in the introduction, is to show an improvement in those chords that have the problem of bass-note induced partials whose frequencies do not coincide with those of the chord notes. Since these chords are rare compared to the most frequent chord type, major, differences in the mean accuracy are relatively small (compare the *std* methods with NNLS, $s = 0.6$, and without in Table 1a). For a good transcription, however, all



(a) *std* method without NNLS

(b) improvement of *std* with NNLS chroma ($s = 0.6$) over baseline *std* method.

Figure 3: Percentage of correct overlap of individual chord types.

chords are important, and not only those that are most frequently used. First of all we want to show that the problem does indeed exist and is likely to be attributed to the presence of harmonics. As a baseline method we choose the best-performing method without NNLS chroma (*std*), whose performance on individual chords is illustrated in Figure 3a. As expected, it performs best on major chords, achieving a recognition rate of 72%. This is rivalled only by the "no chord" label `N` (also 72%), and the minor chords (68%). All other chords perform considerably worse. This difference in performance may of course have reasons other than the bass note harmonics, be it an implicit bias in the model towards simpler chords, or differences in usage between chords. There is, however, compelling evidence for attributing lower performance to the bass note partials, and it can be found in the chords that differ from the major chord in only one detail: the bass note. These are the major chord inversions (denoted `maj/3`, and `maj/5`): while the chord model remains the same otherwise, performance for these chords is around 40 percentage points worse than for the same chord type in root position.

To find out whether the NNLS methods suffer less from this phenomenon, we compare the baseline method discussed above to an NNLS method (*std*, with the chord dictionary parameter $s = 0.6$). The results of the comparison between the baseline method and this NNLS method can be seen in Figure 3b. Recognition rates for almost all chords have improved by a large margin, and we would like to highlight the fact that the recognition of major chords in second inversion (`maj/5`) has increased by 12 percentage points. Other substantial improvements can be found for augmented chords (also 12 percentage points), and major chords in first inversion (9 percentage points). These are all chords in which even the third harmonic of the bass note does not coincide with the chord notes (the first two always do), which further assures us that our hypothesis was correct. Note that, conversely, the recognition of major chords has remained almost stable, and only two chords, major 7th and the "no chord" label, show a slight performance decrease (less than 3 percentage points).

4. DISCUSSION

While the better performance of the difficult chords is easily explainable by approximate transcription, there is some scope in researching why the major 7th chord performed slightly worse in the method using NNLS chroma. Our hypothesis is that the recognition of the major 7th chord actually benefits from the presence of partials: not only does the bass note emphasise the chord notes (as it does in the plain major chord), but the seventh itself is also emphasised by the third harmonic of the third; e.g. in a C major 7th chord (C, E, G, B), the E's third harmonic would emphasise the B. In future work, detailed analyses of which major 7th chords' transcriptions change due to approximate transcription could reveal whether this hypothesis is true.

Our findings provide evidence to support the intuition that the information which is lost by mapping the spectrum to a chroma vector cannot be recovered completely: therefore it seems vital to perform note transcription or calculate a note activation pattern *before* mapping the spectrum to a chroma representation (as we did in this paper) or directly use spectral features as the input to higher-level models, which ultimately may be the more principled solution.

Of course, our approximate NNLS transcription is only one way of approaching the problem. However, if an approximate transcription is known, then chord models and higher-level musical models can be built that do not mix the physical properties of the signal ("spectrum given a note") and the musical properties ("note given a musical context"). Since the components of such models will represent something that actually exists, we expect that training them will lead to a better fit and eventually to better performance.

5. CONCLUSIONS

We have presented a new chroma extraction method using a non-negative least squares (NNLS) algorithm for prior approximate note transcription. Twelve different chroma methods were tested for chord transcription accuracy on a

standard corpus of popular music, using an existing high-level probabilistic model. The NNLS chroma features achieved top results of 80% accuracy that significantly exceed the state of the art by a large margin.

We have shown that the positive influence of the approximate transcription is particularly strong on chords whose harmonic structure causes ambiguities, and whose identification is therefore difficult in approaches without prior approximate transcription. The identification of these difficult chord types was substantially increased by up to twelve percentage points in the methods using NNLS transcription.

6. ACKNOWLEDGEMENTS

This work was funded by the UK Engineering and Physical Sciences Research Council, grant EP/E017614/1.

7. REFERENCES

[1] S. A. Abdallah and M. D. Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, 2004.

[2] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-Based Music Information Retrieval: Current Directions and Future Challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[3] B. Catteau, J.-P. Martens, and M. Leman. A probabilistic framework for audio-based tonal key and chord recognition. In R. Decker and H.-J. Lenz, editors, *Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation*, pages 637–644, 2007.

[4] M. E. P. Davies, M. D. Plumbley, and D. Eck. Towards a musical beat emphasis function. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2009)*, 2009.

[5] E. Gomez. *Tonal Description of Audio Music Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2006.

[6] A. P. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 216–221, 2006.

[7] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*, chapter 23. Prentice-Hall, 1974.

[8] K. Lee and M. 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*, 16(2):291–301, February 2008.

[9] N. C. Maddage. Automatic structure detection for popular music. *IEEE Multimedia*, 13(1):65–77, 2006.

[10] M. Mauch. *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary University of London, 2010.

[11] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 metadata project 2009. In *Late-breaking session at the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, 2009.

[12] M. Mauch and S. Dixon. Simultaneous estimation of chords and musical context from audio. *to appear in IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[13] M. Mauch, K. C. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, pages 231–236, 2009.

[14] K. P. Murphy. The Bayes Net Toolbox for Matlab. *Computing Science and Statistics*, 33(2):1024–1034, 2001.

[15] L. Oudre, Y. Grenier, and C. Févotte. Template-based chord recognition: Influence of the chord types. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 153–158, 2009.

[16] H. Papadopoulos and G. Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, pages 121–124, 2008.

[17] G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[18] 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.

[19] M. Varewyck, J. Pauwels, and J.-P. Martens. A novel chroma representation of polyphonic music based on multiple pitch tracking techniques. In *Proceedings of the 16th ACM International Conference on Multimedia*, pages 667–670, 2008.

[20] A. Weller, D. Ellis, and T. Jebara. Structured prediction models for chord transcription of music audio. In *MIREX Submission Abstracts*. 2009. `http://www.cs.columbia.edu/~jebara/papers/icmla09adrian.pdf`.

[21] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno. Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 100–105, 2004.