Simultaneous Estimation of Chords and Musical Context From Audio

Matthias Mauch, *Student Member, IEEE*, and Simon Dixon, *Member, IEEE*

***Abstract*—Chord labels provide a concise description of musical harmony. In pop and jazz music, a sequence of chord labels is often the only written record of a song, and forms the basis of so-called lead sheets. We devise a fully automatic method to simultaneously estimate from an audio waveform the chord sequence including bass notes, the metric positions of chords, and the key. The core of the method is a six-layered dynamic Bayesian network, in which the four hidden source layers jointly model metric position, key, chord, and bass pitch class, while the two observed layers model low-level audio features corresponding to bass and treble tonal content. Using 109 different chords our method provides substantially more harmonic detail than previous approaches while maintaining a high level of accuracy. We show that with 71% correctly classified chords our method significantly exceeds the state of the art when tested against manually annotated ground truth transcriptions on the 176 audio tracks from the MIREX 2008 Chord Detection Task. We introduce a measure of segmentation quality and show that bass and meter modeling are especially beneficial for obtaining the correct level of granularity.**

***Index Terms*—Chord transcription, dynamic Bayesian networks (DBNs), music signal processing.**

I. Introduction

A chord is defined as the simultaneous sounding of two or more different notes. Accompaniment of jazz and popular music is based on progressions of chords and is rarely written out as complete sheet music. Instead, musicians usually rely on *lead sheets* [1]. A lead sheet typically contains the melody written on traditional staves with time and key signature, along with chord symbols over the staves and the nominal bass note for the chord (if different from the root note), as illustrated in Fig. 1. The chords found in lead sheets are an abstraction of what is actually played in a performance of the song, since often a precise replication of the original is unnecessary, or even unwanted. In recent years, the popularity of lead sheets has been underpinned by the success of the commercial software *Band in a Box*[1] and its noncommercial contender *MMA*[2], both designed to generate musical accompaniment from a representation very similar to a traditional lead sheet.

Manuscript received December 31, 2008; revised August 24, 2009. First published September 22, 2009; current version published July 14, 2010. This work was supported by the OMRAS2 Project (EPSRC Grant EP/E017614/1). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vesa Välimäki.

The authors are with the Queen Mary University of London, Centre for Digital Music, School of Electronic Engineering and Computer Science, E1 4NS London, U.K. (e-mail: matthias.mauch@elec.qmul.ac.uk; simon.dixon@elec.qmul.ac.uk).

Digital Object Identifier 10.1109/TASL.2009.2032947

[1]http://www.pgmusic.com/

[2]http://www.mellowood.ca/mma/

The underlying motivation of our research is to use automatic chord recognition to produce lead sheets. In the remainder of this section, we motivate our design choices derived from this aim, and provide a summary of previous approaches. Fig. 2 shows an overview of our system, and the details of the method are given in the two following sections: Section II explains how we extract bass and treble chroma features from audio, while Section III details the topology and parameter settings of the novel dynamic Bayesian network. Section IV provides comparative evaluations of our methods, followed in Section V by conclusions and a discussion.

A. Objectives of This Work

Our aim is that eventually musicians will be able to use automatically generated lead sheets in the same way as they have been using the traditional, hand-annotated variant. The first requirement derived from this motivation is to provide transcriptions of the musical parameters chord, key, bass, and metric position. Second, similar to human music listening, the interdependence of these musical parameters should be modeled, and inference on them should be simultaneous. For example, chords are interpreted according to the key, while at the same time the key can be understood as a product of the chords. Raphael calls this the "chicken and egg problem" [3, p. 659], and strongly argues for the simultaneous estimation of parameters for cases in which such interdependence arises. Finally, to do justice to the actual complexity of music, more specific chord labels are needed than have been used in previous automatic chord transcriptions. The choice of level of detail is difficult. On the one hand, the MIREX Chord Detection task [4] features only the two chord types major and minor. On the other hand, the chords actually used in pop songs are often much more complex (the software MMA has more than 100 chord types, i.e., 1200 chords). Our choice of 109 chords as detailed in Section III is by no means definitive, but much broader than has previously been attempted.

B. Previous Work

The foundation for a large majority of current methods for chord extraction is a low-level feature called the chroma vector (also, pitch class profile). The chroma vector is a 12-dimensional vector of real numbers representing the energy or salience of the twelve pitch classes $(C, \ldots, B)$, which amounts to considering pitch while suppressing the height dimension [5, p. 159]. Much like a spectrogram describes the spectral content of a signal over time, the chromagram is a sequence of chroma vectors that describes the pitch class content of an audio signal over time. Since its first use for chord extraction [6] the chromagram has also been used for

1558-7916/$26.00 © 2010 IEEE

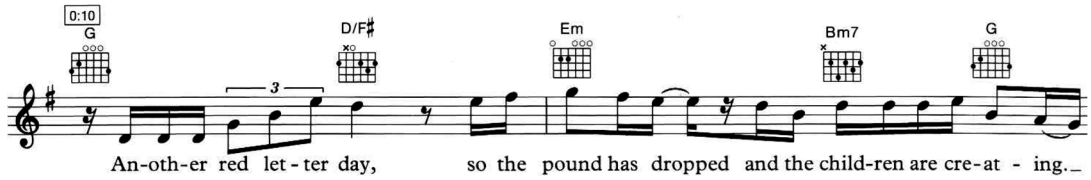

Fig. 1. Pop music lead-sheet: Excerpt of *Friends Will Be Friends* (Deacon/Mercury) taken from [2]. Chords are represented both by chord labels and the corresponding guitar fingering. The number in a box denotes the physical time. The bass is represented only implicitly in the chord labels.

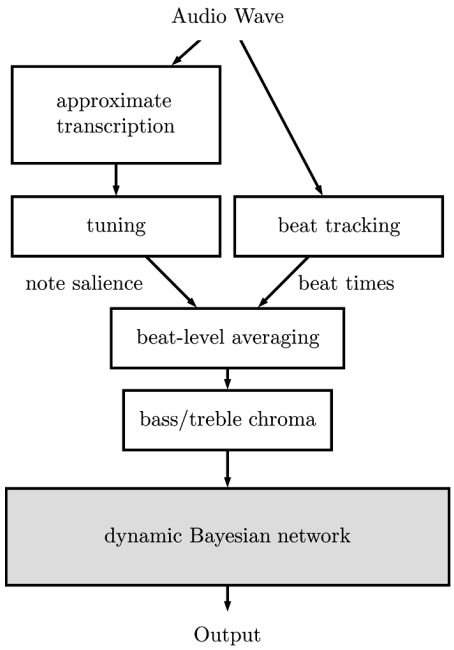

Fig. 2. Schematic overview of our method (see Sections II and III). White boxes represent the chroma extraction sub-methods.

other applications including key finding [7] and audio-to-score synchronization [8]. There are many ways to calculate chromagrams, for an introductory overview see [9]. For chord estimation, the quality of the chromagram has been improved by (automatic) tuning to the reference frequency [10] and median smoothing [11], removal of harmonics [12], and noise attenuation [13]. We combine some of these approaches in our own chromagram extraction algorithm (Section II-A).

To robustly infer chords from a chromagram, several temporal smoothing algorithms have been proposed to suppress short-term deviations from the chord. Examples are median filtering [10], hypothesis search [14], and hidden Markov models (HMMs) [15].

Statistical time series modeling in the music computing community has often been restricted to HMMs. Other approaches include graphical modeling for chord transcription from symbolic data [16], and conditional random fields [17]. There are numerous examples in which HMMs have been used to model and estimate the context of chords. Lee and Slaney [18] perform several HMM inference runs with different, key-dependent chord transition probabilities to implicitly determine the key of the piece in addition to the chords. A different HMM for key estimation from existing chord progressions has been proposed by Noland and Sandler [19]. Previously we integrated bass features into an HMM [20]. In pieces of music for which a beat-segmentation is known, an HMM can be used to perform a simultaneous estimation of the metric position of the beats and chords [21].

For more semantic flexibility than HMMs natively model (i.e., one hidden random variable and one observed random variable per time step), Leistikov [22] proposed the use of dynamic Bayesian networks as a way of modeling notes and their context in symbolic data. This allows for a more intuitive modeling process and an increase in inference efficiency. In the audio domain however DBNs have been used only in melody-tracking [23]; we are not aware of any previous applications of DBNs for the estimation of higher-level features such as chords.

The mentioned chord detection papers have in common the use of a very limited number of different chord types. For example, Lee and Slaney [18] choose to model three chord types (major, minor, diminished), leading to 36 different chords.

The novelty of the present work is that it integrates in a single graphical model pieces of musical context that had previously been assessed only separately. Keys, chords, metric position and bass pitch class can now be estimated *simultaneously* using the efficient inference techniques available for DBNs. We also increase the amount of output detail with respect to existing models, in particular, we increase the number of output chord types.

II. Chromagram Calculation

The aim of low-level processing in our case is to transform the audio input data into a representation which the high-level "musical" model can process. This representation consists of two different beat-synchronous chromagrams, one for the bass frequencies, and one for the treble frequencies, motivated by the importance of the bass note in harmony (see also Section III-D). In this section, we explain how we obtain a note salience representation (or approximate transcription), how it is tuned and wrapped to chromagrams, and how it is finally averaged over beats.

A. Note Salience

Since the desired robust note transcription from complex audio remains an unsolved problem, we attempt an "approximate" transcription, which we refer to as note salience. The input files are monophonic wave files, low-pass filtered and downsampled to $f_s = 11025$ Hz. We calculate the amplitude spectra χ of the wave using a Hamming window (length 2048 samples, i.e., ≈ 0.19 s) with a hop size of $\Delta_h = 0.05$ s.

The salience representation is based on a dictionary of complex tones covering the notes D1 (MIDI note 25, $f_0 \approx 37$ Hz) to C6 (MIDI note 84, $f_0 \approx 1047$ Hz) in 1/3 semitone steps. We

synthesize the mth tone with frequency f_0^m as the weighted sum of its first four harmonics

$$y_m(t) = \sum_{k=1}^{4} r^{k-1} \sin\left(2\pi \cdot \frac{t}{f_s} \cdot k f_0^m\right), \quad t = 1, \ldots, 2048. \tag{1}$$

We adopted a harmonic roll-off parameter $r = 0.6$ in (1) from Gomez [7]. The amplitude spectra M_m^c of these complex tones are obtained from $y_m(t)$ in the same way as those of the input files and appear as rows in the pattern matrix M^c. If χ_j denotes the amplitude spectrum of frame j, the product

$$S_j^c = M^c \cdot \chi_j \tag{2}$$

can be interpreted as the salience of the complex tones at frame j. In order to attenuate the salience at subharmonics introduced by using the complex tone pattern approach, we require that the energy at the fundamental frequency of the mth tone be high. To that end we calculate a second dictionary matrix M^s of simple tones using only the first term in the sum (1). The corresponding salience matrix S^s is obtained analogously to S^c in (2) and subsequently convolved with a Laplacian kernel $(-1, -1, 4, -1, -1)$ to amplify spectral peaks. Negative values are set to zero. The element-wise product

$$S = S^c \otimes S^s \tag{3}$$

combines the two matrices and yields a salience description for every note at every time frame.

B. Tuning and Chroma Mapping

Having three note salience values per semitone enables us to detect the tuning of a song. This is relevant because songs are not always recorded in standard 440-Hz tuning. We assume that the tuning frequency remains the same throughout each song. We use a tuning technique similar to the one used by Dressler and Streich [24]. The tuning is interpreted as an angle $\tau \in (-\pi, \pi]$, which corresponds to a tuning of

$$2^{\tau/(12 \cdot 2\pi)} \cdot 440 \text{ Hz}.$$

Hence, the three salience values pertaining to each semitone represent tunings

$$\tau_k = \frac{2\pi k}{3}, \quad k \in \{-1, 0, 1\}.$$

We add the respective salience values over time, and over the note range

$$E_k = \sum_{j=1}^{T} \sum_{(m-k) \bmod 3 = 0} S_{mj}, \quad k \in \{-1, 0, 1\} \tag{4}$$

and retrieve an estimate of the tuning by calculating the angle

$$\hat{\tau} = \angle \left(\sum_{k=-1}^{1} E_k \cdot \exp\left\{\tau_k \sqrt{-1}\right\} \right). \tag{5}$$

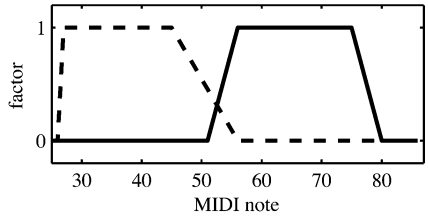

Fig. 3. Treble (solid line) and bass (dashed) templates g. These are used in (7) when calculating chromagrams from the note salience values.

We update S by linear interpolation so that the center bin of a semitone corresponds to $\hat{\tau}$ and then sum the three tone saliences pertaining to the same semitone to obtain the semitone-spaced salience matrix

$$S'_{kj} = \sum_{\lceil m/3 \rceil = k} S_{mj}. \tag{6}$$

The matrix S' is subsequently median-filtered [11] in the time direction with a filter length of nine frames (0.45 s). To obtain the treble chromagram x^*, the note salience S' is "wrapped", i.e., note saliences that belong to the same pitch class are summed

$$x^*_{ij} = \sum_{(k-i) \bmod 12 = 0} S'_{kj} \cdot g_k, \quad i = 1, \ldots, 12 \tag{7}$$

weighted by the template g illustrated in Fig. 3, which discards bass and very high treble notes. The bass chromagram is analogously obtained using different weights g to discard notes in the treble range.

C. Averaging Over Beats and Normalization

Beat, or "tactus," represents the main regular pulse in a piece of music [25, p. 71]. In order to segment the audio into musically meaningful chunks, we use an automatic beat-tracking algorithm [26]. The system extracts beat times $0 < t_0 < \cdots < t_N$. We take the median (over time) of the chromagram frames within each beat

$$x_{ij} = \underset{t_j \le (j' \cdot \Delta_h) < t_{j+1}}{\text{median}} x^*_{ij'}. \tag{8}$$

A measure of chroma flatness is computed to express the salience of "no bass note" and becomes a 13th dimension to the bass chromagram

$$x_{13,j} = \left(\frac{12 \cdot \max_i x_{ij}}{\sum_{i'=1}^{12} x_{i'j}} \right)^{-2} \in \left[\frac{1}{144}, 1 \right]. \tag{9}$$

Both beat-quantized chromagrams—including the additional bass bin—are subsequently normalized according to the maximum norm [7, p. 79], i.e., every bin value is given relative to the most salient bin of the same frame, see Fig. 4.

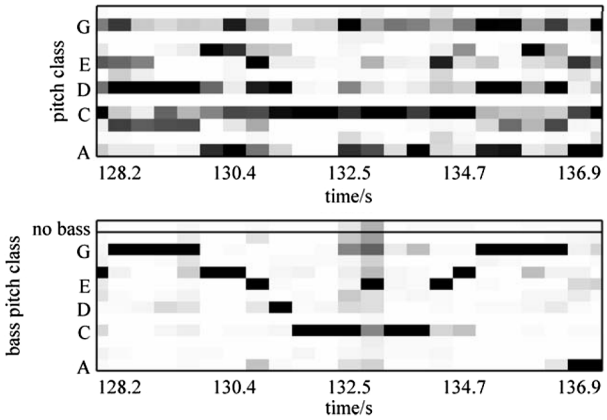

Fig. 4. Example treble and bass chromagrams generated from the song *Let It Be* (Lennon/McCartney).

III. Network Model

A Bayesian network (BN) is a joint distribution of several random variables. It is called a "network" because its dependency structure can be represented using a directed acyclic graph. Every node represents one random variable[3]. A directed edge represents a direct dependency; it points at the node that directly depends on the node from which the edge originates. This duality of the graph and the joint distribution allows very intuitive modeling as detailed in this section. The requirement of the graph to be acyclic means that there is no dependency "short circuit," so a random variable is never its own descendent.

To model time series with BNs, *dynamic* Bayesian networks (DBNs) are used [27]. A DBN can be thought of as a succession of simple BNs. The succession is assumed to be Markovian, and time-invariant, i.e., the model can be described recursively by defining only two slices [28]: one "initial state" slice and one "recursive" slice. Such models are also called *2-slice temporal Bayesian networks* (2-TBN). Note that any DBN could equivalently be modeled as an HMM, comprising the different state variables of the DBN in a single (very large) state variable. As a result, modeling of the adequate HMM is less intuitive and inference can be much slower [27].

In the DBN topology as shown in Fig. 5, discrete nodes model the states of metric position, key, chord, and bass pitch class, and continuous nodes model bass and treble chroma. Our DBN is a generative model, i.e., some state configuration sequence of the hidden source nodes is assumed to have generated the observed data (chromagrams). This assumption allows us to use Bayesian reasoning to infer the state sequence from the data [22, p. 96]. We use the Bayes Net Toolbox [29], which implements diverse inference and learning methods, to model the data and perform the inference.

To complete the definition of the network the conditional probability distributions (CPD) of the random variables need to be specified, providing a good approximation of how beats, keys, chords, and bass interact. Since we do not have any preconception of the initial metric position, key, chord, or bass pitch class of a piece, all initial nodes are set to a uniform distribution. The following subsections will detail the CPDs of the recursive nodes on the right-hand side of the 2-TBN depicted in Fig. 5. Like Leistikow [22] we choose to map expert musical knowledge onto a probabilistic framework, rather than learning parameters from a specific data set. In a complex model such as the one presented in this section, the decisions regarding parameter binding during learning, and even the choice of the parameters to be learned pose challenging research questions, which we plan to address in future work, while focusing here on the definition and evaluation of the expert model.

[3]We will use the two expressions node and random variable interchangeably.

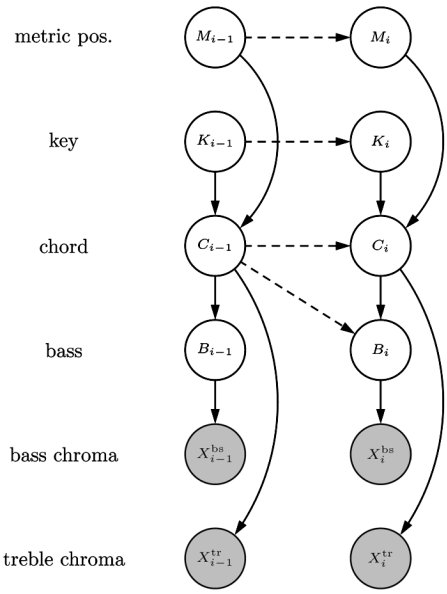

Fig. 5. Our network model topology, represented as a 2-TBN with two slices and six layers. The clear nodes represent random variables, while the observed ones are shaded gray. The directed edges represent the dependency structure. Intra-slice dependency edges are drawn solid, inter-slice dependency edges are dashed.

A. Metric Position

Western music is usually grouped in measures, each containing a number of beats. In much popular music, there are four beats per measure throughout a piece, and our model assumes this case. The first beat (metric position 1) in a measure is followed by the second (metric position 2), and so on, until after the fourth the next measure starts on metric position 1. Hence, the node M_i has four states to represent the metric position of the current beat. We use pieces of music in which occasional beat tracking errors or compositional irregularities in the music are frequent, hence we have to allow for the small probability $\varepsilon = 0.05$ of deviation from the normal succession of beats. Since node M_i depends only on node M_{i-1}, the conditional distribution $P(M_i|M_{i-1})$ can be represented as a transition matrix with two dimensions

$$\begin{pmatrix} \frac{\varepsilon}{2} & 1-\varepsilon & \frac{\varepsilon}{2} & 0 \\ 0 & \frac{\varepsilon}{2} & 1-\varepsilon & \frac{\varepsilon}{2} \\ \frac{\varepsilon}{2} & 0 & \frac{\varepsilon}{2} & 1-\varepsilon \\ 1-\varepsilon & \frac{\varepsilon}{2} & 0 & \frac{\varepsilon}{2} \end{pmatrix}.$$

Fig. 6. Key: C major/A minor key. Pitch classes in shaded squares are the ones belonging to the key. To obtain the other keys, the pitch classes are "rolled" accordingly (circular shift).

Fig. 7. Chord examples: `Cmaj7` and `Cmin` chords. The shaded squares denote the pitch classes belonging to the chord. To obtain the same chord type with a different root, the chord is "rolled" (circular shift).

Each row represents a state of M_{i-1}, every column a state of M_i. The same information can be written as a conditional probability distribution

$$P(m_i|m_{i-1}) = \begin{cases} 1-\varepsilon, & \text{if } (m_i - m_{i-1}) \bmod 4 = 1, \\ \varepsilon/2, & \text{if } (m_i - m_{i-1}) \bmod 4 \in \{0, 2\}, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

B. Key

The node K_i represents the key state. The knowledge of the key and key changes can have two benefits: improving the stability of the chord estimation by making off-key chords less probable, and providing a means of setting the key signature in a score. We choose to model 12 keys, each of which corresponds to a major/relative minor key pair, which is enough to cover all key signatures, since any major and the corresponding relative minor key share a key signature. Relative to the root pitch class, every key has a diatonic profile; an example is depicted in Fig. 6.

To model the key we only need to express that the key is expected to remain the same with a high probability of 0.98, i.e., we assume that at any beat the key changes with a probability of 0.02

$$P(k_i|k_{i-1}) = \begin{cases} 0.98 & \text{if } k_{i-1} = k_i, \\ \frac{1-0.98}{11} & \text{otherwise.} \end{cases} \tag{11}$$

The behavior of the key node only describes the rate of change of keys. The way in which the key acts upon the chord is coded into the chord CPD as detailed in the following subsection.

C. Chord and Treble Chroma

The chord nodes C_{i-1} and C_i together with the respective treble chroma nodes X_i and X_{i-1}, take a central place in our model. We use a pool of $N_C = 109$ chords:

- 7×12 in root position: major (shorthand[4]: `maj`), minor (`min`), major 7th (`maj7`), major with a minor 7th (`7`), major 6th (`maj6`), diminished (`dim`), augmented (`aug`);
- 2×12 major chords in first and second inversion (`maj/3` and `maj/5`);
- 1 "no chord" (N).

[4]We use the shorthand notation as proposed in [30], but omit the colon as in `C:maj7`.

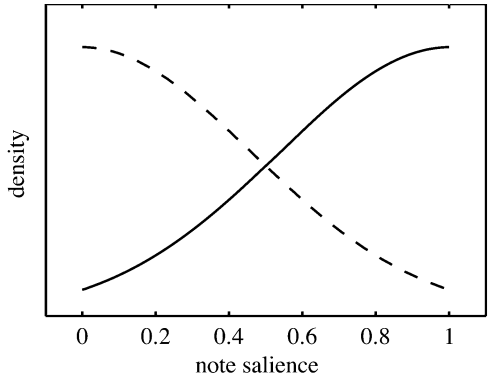

Fig. 8. Treble chroma node: distribution of single elements of the 12-dimensional Gaussian, monotonically increasing curve for chord pitch classes, monotonically decreasing curve (dashed) for non-chord pitch classes.

To keep calculations feasible and prevent over-specification we have refrained from including yet more chords, for example `sus4` and `min7`. However, we believe that this choice, in particular the `7` chord (suggesting a functional difference to `maj`), and the inversions offer a great increase of information when compared to a smaller set of `maj`, `min` , `dim`, and `aug` chords.

First, let us consider the treble chroma node X_i (Fig. 5). Following Harte's chord definitions [30], the quality of a chord is expressed by the pitch classes it contains (see Fig. 7). This should be reflected in the treble chroma the chord generates. As has been explained in Section II, the chroma features $x_i \in [0, 1]$ are normalized by the maximum norm, so high values will be close to one, and—ideally—low values close to zero.

The probability density $P(X_i|c_i)$ of the chroma node given a chord should monotonically *increase* with any of the chord pitch class saliences increasing. It should monotonically *decrease* with any of the non-chord pitch class saliences increasing. We model this behavior as a 12-dimensional Gaussian random variable in which the mean vector has zeros at the elements representing non-chord pitch classes and ones at elements representing the chord pitch classes, see Fig. 8. We choose a diagonal covariance matrix in which all diagonal elements are set to $\sigma^2 = 0.2$. A rigorous estimation of variance values is left to future work. Note that due to the chroma normalization, a flat chroma vector will contain only ones. Therefore, we define `N` (no chord) as including all pitch classes.

We have described the treble chroma node, which depends only on the chord node. The chord node itself, C_i, depends on the previous chord node C_{i-1} as well as the current metric position node M_i and the current key node K_i. This configuration allows us to model that:

- a chord change is likely at the beginning of a measure (metric position 1), less likely in the middle of a measure (position 3), and even less likely at the remaining metric positions 2 and 4;
- a chord is more likely the fewer non-key pitch classes it contains.

Accordingly, we factorize the probability as

$$P(c_i|c_{i-1}, m_i, k_i) = P(c_i|c_{i-1}, m_i) \cdot P(c_i|k_i) \tag{12}$$

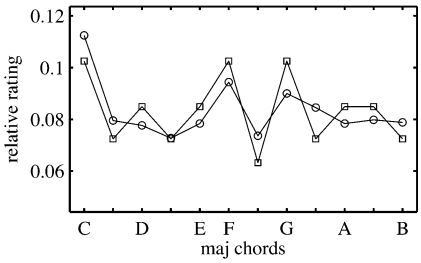

Fig. 9. Our f chord-context ratings (denoted by $\square$) for major chords in a C major key context, compared to the Krumhansl profiles ($\circ$), both normalized by the L_1 norm.

in which the first factor describes the dependency of a chord change on the metric position. Let the vector

$$a = (0.5, 0.1, 0.4, 0.1) \tag{13}$$

contain the probabilities of a chord change at metric positions 1 to 4, then

$$P(c_i|c_{i-1}, m_i) = \begin{cases} \frac{a_{m_i}}{N_{\mathrm{C}}-1,} & \text{if } c_{i-1} \neq c_i \\ 1 - a_{m_i}, & \text{otherwise} \end{cases} \tag{14}$$

where N_{C} is the number of chords. The second factor in (12) describes how likely a chord is, conditional on the key. Perceptual chord ratings in a key context are available for `maj`, `min`, and `dim` chords [31], but not for the more complex chords we consider. We introduce an expert function

$$f(c_i, k_i) = \frac{1}{\#\{\text{non-key chord notes}\} + c} \tag{15}$$

that can express a rating for any kind of chord. To determine the smoothing parameter c we use as a reference the `maj` chord subset of the mentioned chord ratings (Fig. 9), interpreted as probabilities [19]. To obtain a function f that approximates the ratings best, we minimize with respect to c the Jensen–Shannon divergence between the chord ratings and the corresponding ones obtained from the function f. The resulting value of $c = 4.83$ is then used for all chords. For instance, the `Cmaj7` chord depicted in Fig. 7, in the key of C major has $f(\texttt{Cmaj7}, \mathrm{C\ major}) = 1/(0+c) \approx 0.21$, whereas for the `Cmin` chord in the same figure, $f(\texttt{Cmin}|\mathrm{C\ major}) = 1/(1+c) \approx 0.17$ because $E\flat$ is not part of the C major key. The f values are then normalized by a constant κ such that

$$P(c_i|k_i) = \kappa \cdot f(c_i, k_i) \tag{16}$$

is a conditional probability distribution, i.e., for a fixed k_i the probabilities sum to unity.

D. Bass Pitch Class and Bass Chroma

The bass pitch class plays a crucial role in the recognition of chords. Being at the bottom of the frequency range, it "anchors" the chord and makes the rest of the notes more easily interpretable. For instance, knowing whether the bass note is C or E can help disambiguate the chords `Cmaj7` and `Emin`, which have very similar pitch class sets (namely, C, E, G, B and E, G, B).

A bass pitch class can be determined for every chord on a lead sheet. In chords written without further bass information, the bass pitch class is the same as the root note, otherwise the *slash notation* of the bass pitch class determines the bass pitch class. In Harte's syntax [30], an `Fmaj` chord has the bass note `F`, but the bass pitch class of its first inversion `Fmaj/3` is `A`, where `/3` means that the bass note is the third above the root.

The bass chroma is modeled in much the same way as the treble chroma, by a Gaussian vector. Its number of dimensions is $13 = 12 + 1$, with 12 dimensions representing the bass pitch classes `C` through `B`, and the 13th representing "no bass note." Since the bass is defined by just one note, every profile has only one element (rather than 3 or 4 in the case of chords) for which the mean value is set to 1, while the others are set to 0. Usually only one bass note is played at any time, which implies that the pitch class played will more often have a normalized salience of 1, and the other pitch classes will have saliences close to zero. Accordingly, we choose a lower variance value of $\sigma^2 = 0.1$.

Bass lines tend to include many different consecutive notes and pitch classes. The role of the chord bass pitch class becomes clear if one observes that in popular music the bass note is almost always present on the first beat of a chord. One popular bass player tutorial [32] confirms this: among the 207 example bass patterns covering styles Blues & R'n'B, Soul, Motown/Atlantic Records, Funk, and Rock only 20 do *not* start with the bass pitch class. Allowing for some more variation than given in these examples, we estimate that the played and the chord bass note coincide on the first beat of the chord 80% of the time. To model this behavior, we set the probabilities to

$$P(b_i|c_{i-1} \neq c_i) = \begin{cases} 0.8, & \text{if bass is chord bass} \\ \frac{0.2}{12,} & \text{otherwise.} \end{cases} \tag{17}$$

As the chord continues, we still expect the "nominal" bass pitch class as the most likely option, but other pitch classes may be used as a bass note too, so we set the probabilities as follows:

$$P(b_i|c_{i-1} = c_i) = \begin{cases} 0.4, & \text{if bass is chord bass} \\ \frac{0.6}{12,} & \text{otherwise.} \end{cases} \tag{18}$$

Note that while modeling essential properties of popular music in 4/4 time, the CPDs described in this section do not explicitly suppress or encourage particular key, chord or bass note transitions.

IV. Evaluation

Since chord labeling is not a well-defined classification task even for human musicians, the evaluation of automatic chord transcription is difficult. It has been common practice to use the relative correct overlap with respect to a ground truth annotation as an accuracy measure [4]. We would also like to stress that chord extraction from audio is a segmentation task as much as a classification task, and the similarity of ground truth and automatic segmentation should be taken into account. Both kinds of measures will be explained in this section, followed by the corresponding results.

A. Performance Measures

A segmentation of a song is a vector B of one or more contiguous, non-overlapping intervals $B_1, \ldots, B_{N_B}$ such that $\bigcup B_i$ covers the whole song, and $T = |\bigcup B_i|$ is the length of the song, where vertical lines $|\cdot|$ denote the length of an interval. Let $B^0 = (B_1^0, \ldots, B_{N_0}^0)$ be the given (ground truth) segmentation, and B that obtained from an automatic algorithm. Similarly, let $L^0 = (l_1^0, \ldots, l_{N_0}^0)$ be the ground truth class labels corresponding to B^0, and $L = (l_1, \ldots, l_N)$ those corresponding to B.

1) Relative Correct Overlap and MIREX Score: Rather than dealing with thousands of possible chords directly we break up the chords into classes, resulting in a partition $\mathcal{L}$. If the chord labels l_1 and l_2 are in the same class, they are called $\mathcal{L}$-equivalent, $l_1 \sim l_2$. In the MIREX task, the chord labels are partitioned into $|\mathcal{L}| = 25$ different classes: 12 `min` classes (each class comprises the chords whose labels contain `min` and which have identical roots, for example, `Fmin` $\sim$ `Fmin7`), and 12 `maj` classes (each class comprises the chords whose labels do not contain `min` or `N` and which have identical roots), as well as the "no chord" class `N`. We use the Iverson bracket as follows:

$$[l_1 \sim l_2] = \begin{cases} 1, & \text{if } l_1,\ l_2 \text{ are } \mathcal{L}\text{-equivalent} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The relative correct overlap for one song is then defined as

$$\mathcal{O}_{\mathcal{L}} = \frac{1}{T} \sum_{i=1}^{N_0} \sum_{j=1}^{N} |B_i^0 \cap B_j| \times [l_i^0 \sim l_j]. \tag{20}$$

The MIREX score is the mean of $\mathcal{O}_{\mathcal{L}}$ over all songs. The choice of MIREX chord classes is very coarse, and for the further results we use a different $\mathcal{L}$ to differentiate the $|\mathcal{L}| = 109$ chord classes that map most closely to the chords in the DBN.

The measure described above is necessarily biased towards the chord type that occupies most of the duration in a song or collection. In the case of the Beatles' music this is the `maj` chord type. To assess the method's performance on a specific chord subset $\mathcal{S} \subset \mathcal{L}$ the formula (20) changes to

$$\frac{\sum_{l_i^0 \in \mathcal{S}} \sum_{j=1}^{N} |B_i^0 \cap B_j| \times [l_i^0 \sim l_j]}{\sum_{l_i^0 \in \mathcal{S}} |B_i^0|}. \tag{21}$$

2) Segmentation Quality: The segmentation quality of a transcription with respect to the ground truth can be evaluated without taking chord labels into account. This is desirable because such a measure is less likely to suffer from the necessarily subjective chord interpretation of the ground truth annotator.

The measure we propose is based on the *directional Hamming divergence*[5], which has been used in the context of image segmentation [33] and musical song segmentation [34]. For each interval in a segmentation, the directional Hamming divergence measures how much of it is *not* overlapped by the maximally overlapping segment of the other segmentation. Then the values over all intervals are summed. In mathematical terms, given two segmentations B^0, B we define the directional Hamming divergence as

$$h(B\|B^0) = \sum_{i=1}^{N_B} \left(|B_i^0| - \max_j |B_i^0 \cap B_j| \right). \tag{22}$$

It describes how fragmented B is with respect to B^0. If we swap the two segmentations in (22), we obtain what has been called the *inverse* directional Hamming distance, a measure of how fragmented B^0 is with respect to B. The arithmetic mean of both, normalized by the length of the song is a symmetric measure for the dissimilarity of the two segmentations:

$$H(B, B^0) = \frac{h(B\|B^0) + h(B^0\|B)}{2T}. \tag{23}$$

It is desirable that an automatic transcription B have low $H(B, B^0)$ against a ground truth segmentation B^0.

B. Results

We use Beatles chord transcriptions [30] as ground truth, and extract chromagrams from the corresponding original Beatles recordings. Several experiments are conducted to investigate the influence of choice of chord set, metric position, bass note, and key in our model. We choose among three different chord sets, namely:

full	the full chord set, consisting of all 109 chords introduced in Section III-C;
maj-min	only the two chord classes `maj` and `min`, and the `N` class (25 chords);
inv	the set which extends the *maj-min* set by adding the first and second inversion major chords `maj/3` and `maj/5` (49 chords).

We also consider four different DBN configurations by enabling only specific nodes.

plain	In the *plain* model, the metric position, key, and bass pitch class modeling is disabled, chord duration is modeled as a negative binomial distribution[6] [20] with shape parameter 2, and scale parameter 1/3, corresponding to an expected chord duration of 4 beats.
M	In the *metric* model (*M*), metric position is fully modeled as described in III; bass and key are disabled.
MB	In the *metric-bass* model (*MB*), the bass pitch class node is additionally enabled.
MBK	The *metric-bass-key* model (*MBK*) is the entire model as described in Section III .

We infer the most likely state sequence for the enabled discrete nodes using the Viterbi algorithm. Inference in the most complex model, the *MBK* model with *full* chord set, is very memory-intensive, since the chord node would have to deal with $109 \times 4 \times 12 \times 109 = 570288$ states. We perform a preprocessing step to discard the 59 chords that appear least often among the

[5]also called *directional Hamming distance*

[6]the discrete analog of a gamma distribution

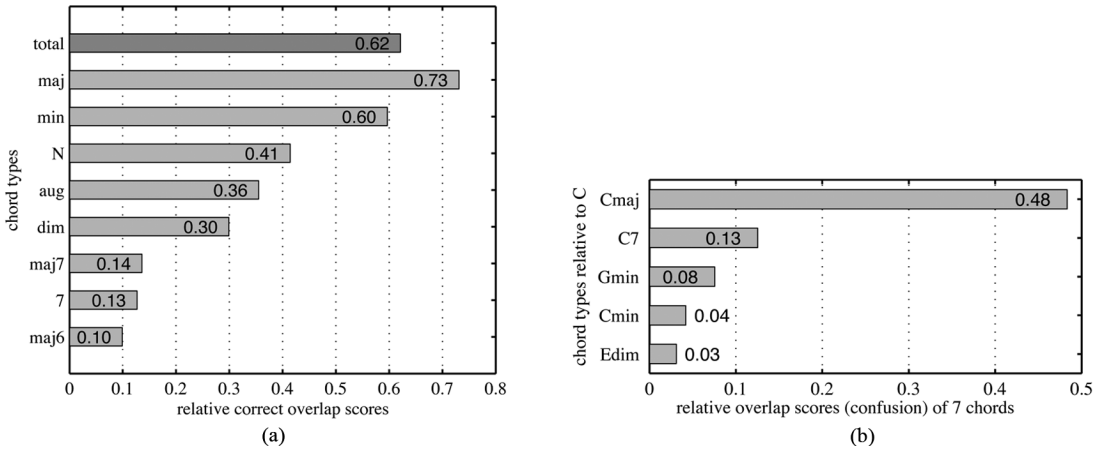

Fig. 10. Full chords *MBK*: relative correct overlap scores for different chord types. Chords are considered correct if root note and chord class match, inversions of the same chord are considered equivalent. (a) shows chord type overlap. (b) details the most common matches for chord class 7, with the correct chord itself ranking second.

locally best-fitting ten chords at every beat, leaving us with 50 chords, which are still fully connected. Processing time does not exceed the song play time, for example, inference on the Beatles song *You Won't See Me* (Lennon/McCartney) with a play time of 202 s takes 104 s using the *full-MBK* model.

1) MIREX-Style Results: The MIREX score as defined in (20) is a good benchmark for comparing our algorithm to others', since the song-wise original MIREX task results are freely available.[7] To comply with the MIREX format, we have to map all our chords to the 24 `maj` and `min` labels, plus one `N` label. All chords with a major third degree are mapped to the respective `maj` chord, all chords with a minor third degree to the respective `min` chord. Several versions of our algorithm (Table I) have a mean relative correct overlap of over 0.70, i.e., they perform better than the best performances in the 2008 MIREX pretrained Audio Chord Detection task (Bello and Pickens [15] scored 0.66). To assess if the MIREX score difference between our best-performing model, *inv-MBK*, and Bello and Pickens's model are significant, we perform a one-way ANOVA analysis. The resulting p-value of 0.006 is very low, and hence we can be confident that our *inv-MBK* model performs significantly better. A further Tukey–Kramer multiple comparison test between all our models and Bello and Pickens's at 95% confidence level based on the Friedman analysis of variance (see, e.g., [35]) confirms that the *MBK* models all perform significantly better than Bello and Pickens's. To assess which of the variants of our model have a significant influence on the MIREX score, we consider only the *full* chords versions and perform a Tukey–Kramer multiple comparison based on the Friedman analysis of variance. We find that with 95% confidence, each additionally added node achieves a significant improvement. We conclude that meter, bass, and key modeling all significantly contribute to better chord labeling in our model.

2) Segmentation Quality: We evaluate the segmentation quality according to the H measure given in (23) on all *full* chord versions (see Table II). The more complex models yield lower, i.e., better, segmentation scores. In fact, according to the Tukey–Kramer multiple comparison with a confidence level of 95%, segmentation significantly improves by adding meter modeling to the *plain* model. Additionally adding bass modeling to the *M* model brings about another significant improvement. Meter and bass modeling provide means of finding chord change positions at a level of granularity more closely related to manual annotations.

[7]http://www.music-ir.org/mirex/2008/results/chord/task1_results/ACD.task1.results.overlapScores.csv

TABLE I
MIREX $\mathcal{O}$ SCORE RESULTS: MEAN RELATIVE CORRECT OVERLAP, AVERAGED OVER THE 176 SONGS USED IN THE 2008 MIREX TASK. FOR SIGNIFICANCE TESTS SEE SECTION IV-B1. BP MIREX [15] IS THE BEST PERFORMING ALGORITHM IN THE ORIGINAL TASK

	MIREX score			
chord set	*plain*	*M*	*MB*	*MBK*
maj-min	0.663	0.674	0.703	0.709
inv	n/a	n/a	0.716	0.712
full	0.654	0.662	0.704	0.709
BP MIREX		0.661		

TABLE II
MEAN SEGMENTATION DIVERGENCE (23) OF DIFFERENT MODELS, USING *FULL* CHORDS. LOWER VALUES ARE BETTER. IMPROVEMENTS FROM *PLAIN* TO *M* AND FROM *M* TO *MB* ARE SIGNIFICANT (SEE SECTION IV-B2)

plain	*M*	*MB*	*MBK*
0.186	0.176	0.167	0.166

3) Chord Confusion: For the rest of our evaluation we will consider the *full-MBK* model as described in Section III and confine the analysis to 155 Beatles songs that do not explicitly violate the time signature assumption we made in our model. To investigate the method's performance on less common chords we use 109 specific chord classes (instead of the coarser MIREX classes), 12 for each of the chord types used in the model (see Section III-C), as well as `N`.

Fig. 10 shows that the mean relative overlap score remains at a high overall level of 0.62 even with the much finer class partitioning. The `maj` and `min` chords are recognized most reliably, followed by "no chord" and `aug` and `dim` chords. The worse performance of `maj7`, `7`, and `maj6` chords is not surprising, since

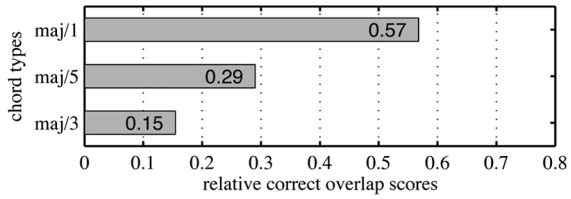

Fig. 11. Full chords *MBK*: `maj` chord relative correct overlap (root note, chord class, and bass note match) for the different inversions.

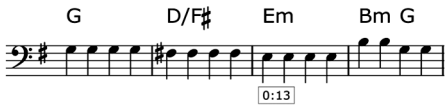

Fig. 12. Excerpt of an automatic output of our algorithm using the *inv-MBK* model for *Friends Will be Friends* (compare to Fig. 1), music engraving by LilyPond. Physical time is displayed in the box. In the second measure, the `Dmaj` chord is correctly identified as being in first inversion. The key signature of G major is also correct. The notes in the staves represent the bass pitch class.

these are `maj` chords with added notes. For instance, taking a closer look at the `7` chord reveals that the `7` chord is most frequently classified as the ordinary `maj` chord on the same root [Fig. 10(b)]. Since reversely very few `maj` chords are incorrectly recognized as `7` chords (relative overlap is 0.026), the successfully recognized `7` chords add a new level of detail to chord recognition. The other three among the top five confusions in Fig. 10(b) are easily explained too, since they all share two or three pitch classes with the `7` chord.

The inversions of `maj` chords are of particular interest, since they show the impact of the bass note. Fig. 11 shows that `maj` chords in root position score highest. Chords in first inversion are recognized as such only 15% of the time, but can still provide information that was not available with previous approaches, e.g., see Fig. 12. Second inversion chords have an overlap score of 29%.

4) Key Signature: We model only the key signature, i.e., 12 different major/minor pairs. For a given piece, we retrieve the main key signature (see Section III-B) that is active most frequently. Our method correctly recognizes 63% of the main key signatures, which is not very high compared to state of the art key extraction algorithms [36], but acceptable since we do not explicitly model minor keys. Since the additional key information does provide an increased performance in the MIREX score (see Section IV-B1), we expect that future work on key modeling will result in further improvements.

5) Examples: Our system automatically generates LilyPond[8] source files and *Sonic Visualiser*[9] XML files. The lead sheet depicted in Fig. 12 is compiled from a LilyPond source file. Key, chord inversion, and the metric information provide a detailed notation that matches the official version from [2] depicted in Fig. 1.

In Fig. 13, an excerpt of the song *Something* (Lennon/McCartney), is displayed as loaded from an automatically created XML file into Sonic Visualizer (gray). For comparison, we have additionally loaded the ground truth annotations (black). Note that while the ground truth correctly annotates the first

[8]http://lilypond.org/web/

[9]http://www.sonicvisualiser.org/

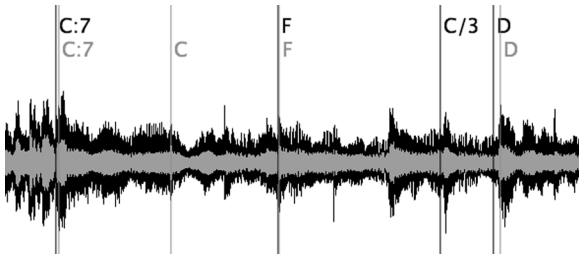

Fig. 13. Excerpt of *Something* (Lennon/McCartney), displayed in the free software *Sonic Visualizer*. The first (black) line of chords is the ground truth transcription, the lines below (gray chord symbols) are our automatic transcription, using full chords, metric position, bass, and key.

two full measures of the example as `C7`, our method switches back to `Cmaj` in the second measure. This happens because in the second measure the flat seventh that turns a `Cmaj` chord into a `C7` is not present, but still assumed to continue by the annotator. This gives a qualitative explanation for the confusion of the `7` chord discussed in Section IV-B3.

V. Conclusion and Future Directions

We have presented a musically-informed dynamic Bayesian network for the automatic extraction of chord transcriptions from musical audio. The main novelty of this approach is *simultaneous* inference of metric position, key, chord, and bass pitch class, which reflects the natural interdependence of these entities. With 109 chord classes, the model provides a higher level of detail than previous approaches.

The method presented achieves a mean correct overlap score of 71%, and significantly outperforms all systems tested in the 2008 MIREX task for pretrained chord detection. We compared ten different variants of our algorithm and show that each additional musical parameter significantly improves the method's performance. The greatest enhancement is achieved by additional bass modeling. While aiding the correct identification of chords, the key estimation itself has performed slightly worse than anticipated. The high number of chords provides new musical information, without decreasing the performance of the method.

As a complement to the correct overlap evaluation method, we have introduced a measure of chord segmentation quality which provides a measure of how well the locations and granularity of chord changes resemble those of the ground truth. Our results show a significant improvement in segmentation quality due to modeling of metric position and bass.

Taking the present expert system as a point of departure, we believe that careful probabilistic learning could yield even better results, despite inevitably being specific to the music collection on which it is trained. A model with parameters learned from data could shed light on the flaws of the present key model as well as making the chroma models easily adapt to changes in the audio front-end. This may be especially useful when applying the basic model structure in different domains, e.g., chord extraction from MIDI, or figured bass extraction from Baroque recordings. We would like to extend our approach further and work towards a more complete model of music listening which includes beat detection, form, melody, and time signature.

REFERENCES

[1] R. Rawlins and N. E. Bahha, *Jazzology*. Indianapolis, IN: Hal Leonard, 2005.

[2] F. Mercury, J. Deacon, B. H. May, and R. M. Taylor, *Greatest Hits II: Top Line and Chorus*, B. M. E. Ltd, Ed. London, U.K.: Queen Music Ltd./EMI Music Publishing, 1992.

[3] C. Raphael, "A graphical model for recognizing sung melodies," in *Proc. 2005 ISMIR Conf.*, London, U.K., 2005, pp. 658–663.

[4] MIREX Audio Chord Detection Subtask, Music Information Retrieval Evaluation Exchange, 2008. [Online]. Available: http://www.music-ir.org/mirex/2008/index.php/Audio_Chord_Detection.

[5] R. Shepard, "Pitch perception and measurement," in *Music, Cognition, and Computerized Sound: An Introduction to Psychoacoustics*, P. Cook, Ed. Cambridge, MA: MIT Press, 1999, pp. 149–165.

[6] T. Fujishima, "Real time chord recognition of musical sound: A system using common lisp music," in *Proc. Int. Computer Music Conf. (ICMC)*, 1999, pp. 464–467.

[7] E. Gomez, "Tonal description of audio music signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[8] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen, "Automated synchronization of scanned sheet music with audio recordings," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, Vienna, Austria, 2007, pp. 261–266.

[9] H. Papadopoulos and G. Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and HMM," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, 2007, pp. 53–60.

[10] C. Harte and M. Sandler, "Automatic chord identification using a quantised chromagram," in *Proc. 118th Conv. Audio Eng. Soc.*, 2005.

[11] G. Peeters, "Chroma-based estimation of musical key from audio-signal analysis," in *Proc. 7th Int. Conf. Music Inf. Retrieval, ISMIR 2006*, Victoria, BC, Canada, 2006.

[12] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Comput. Music J.*, vol. 32, no. 3, pp. 72–86, 2008.

[13] A. Camacho, "Detection of pitched/unpitched sound using pitch strength clustering," in *Proc. 9th Int. Conf. Music Inf. Retrieval, ISMIR 2008*, Philadelphia, PA, 2008, pp. 533–537.

[14] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic chord transcription with concurrent recognition of chord symbols and boundaries," in *Proc. 5th Int. Conf. Music Inf. Retrieval, ISMIR 2004*, Barcelona, Spain, 2004, pp. 100–105.

[15] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proc. 6th Int. Conf. Music Inf. Retrieval, ISMIR 2005*, London, U.K., 2005, pp. 304–311.

[16] J.-F. Paiement, D. Eck, and S. Bengio, "A probabilistic model for chord progressions," in *Proc. 6th Int. Conf. Music Inf. Retrieval, ISMIR 2005*, London, U.K., 2005, pp. 312–319.

[17] J. A. Burgoyne, L. Pugin, C. Kereliuk, and I. Fujinaga, "A cross-validated study of modelling strategies for automatic chord recognition in audio," in *Proc. 8th Int. Conf. Music Inf. Retrieval, ISMIR 2007*, Vienna, Austria, 2007, pp. 251–254.

[18] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 291–301, Feb. 2008.

[19] K. Noland and M. Sandler, "Key estimation using a hidden Markov model," in *Proc. 7th Int. Conf. Music Inf. Retrieval, ISMIR 2006*, Victoria, BC, Canada, 2006.

[20] M. Mauch and S. Dixon, "A discrete mixture model for chord labelling," in *Proc. 9th Int. Conf. Music Inf. Retrieval, ISMIR 2008*, Philadelphia, PA, 2008, pp. 45–50.

[21] H. Papadopoulos and G. Peeters, "Simultaneous estimation of chord progression and downbeats from an audio file," in *Proc. 2008 ICASSP Conf.*, 2008, pp. 121–124.

[22] R. J. Leistikov, "Bayesian Modeling of Musical Expectations via Maximum Entropy Stochastic Grammars," Ph.D. dissertation, Dept. of Music, Stanford Univ., Stanford, CA, 2006.

[23] H. Thornburg, R. J. Leistikow, and J. Berger, "Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1257–1272, May 2007.

[24] K. Dressler and S. Streich, "Tuning frequency estimation using circular statistics," in *Proc. 2007 ISMIR Conf.*, Vienna, Austria, 2007, pp. 357–360.

[25] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press, 1983.

[26] M. Davies, "Towards automatic rhythmic accompaniment" Ph.D. dissertation, Queen Mary Univ. of London, London, U.K., 2007.

[27] K. P. Murphy, "Dynamic Bayesian Networks: Representation, inference and learning," Ph.D. dissertation, Univ. of California, Berkeley, 2002.

[28] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," in *Proc. 14th Annual Conf. Uncertainty in Artif. Intell. (UAI-98)*, 1998, pp. 33–42.

[29] K. P. Murphy, "The Bayes net toolbox for Matlab," *Comput. Science Statist.*, vol. 33, no. 2, pp. 1024–1034, 2001.

[30] C. Harte, M. Sandler, S. A. Abdallah, and E. Gomez, "Symbolic representation of musical chords: A proposed syntax for text annotations," in *Proc. 6th Int. Conf. Music Inf. Retrieval, ISMIR 2005*, London, U.K., 2005, pp. 66–71.

[31] C. L. Krumhansl, *Cognitive Foundations of Musical Pitch*. New York: Oxford Univ. Press, 1990.

[32] P. Westwood, *Bass Bible: A World History of Styles and Techniques*. Berlin, Germany: AMA Verlag, 1997.

[33] Q. Huang and B. Dom, "Quantitative methods of evaluating image segmentation," in *Proc. Int. Conf. Image Process., 1995*, 1995, vol. 3, pp. 53–56.

[34] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes, "Theory and evaluation of a Bayesian music structure extractor," in *Proc. 6th Int. Conf. Music Inf. Retrieval, ISMIR 2005*, London, U.K., 2005, pp. 420–425.

[35] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.*, New York, 1993, pp. 329–338.

[36] K. Noland and M. Sandler, "Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio," *Comput. Music J.*, vol. 33, no. 1, 2009.

Matthias Mauch (S'08) received the Diplom degree in mathematics from the University of Rostock, Rostock, Germany, in collaboration with the Max-Planck Institute for Demographic Research. He is currently pursuing the Ph.D. degree in the Centre for Digital Music at Queen Mary University of London, London, U.K.

His research focuses on the automatic extraction of high-level musical features from audio, with an emphasis on harmonic progressions and repetitions. He is songwriter in the band Zweieck.

Simon Dixon (M'07) is a Lecturer in the Centre for Digital Music at Queen Mary University of London, London, U.K. He received the B.Sc. and Ph.D. degrees in computer science from the University of Sydney, Sydney, Australia, and AMusA and LMusA diplomas in classical guitar.

His research interests focus on the extraction and processing of musical (particularly rhythmic and harmonic) content in audio signals, including tasks such as tempo induction, beat tracking, onset detection, audio alignment, automatic transcription, and the measurement and visualization of expression in music performance.