Audio Chord Estimation Based on Meter Modeling and Two-Stage Decoding

Alessio Degani, Marco Dalai, Riccardo Leonardi and Pierangelo Migliorati
University of Brescia
DII, Signals and Communication Lab
Via Branze, 38, 25123, Brescia, ITALY

***Abstract*—In Music Information Retrieval (MIR) different approaches in modeling the meter structure of a song have been proposed and have been proved to be beneficial for the task of Audio Chord Estimation (ACE). In this paper we propose a novel approach that integrates the meter and beat information into the Hidden Markov Model (HMM) used for Audio Chord Estimation. In addition to the proposed meter model, we introduce also a modification in the inference procedure of the aforementioned Hidden Markov Model, in order to better capture the temporal correlation between chords progression. Experimental results show that the proposed approach is effective as the classical approaches in modeling the meter structure, but with a substantially reduced model complexity. Moreover, the proposed two-stage decoding procedure produces a significant improvement in the chords estimation accuracy.**

Keywords*—Audio Chord Estimation, Hidden Markov Model, Music Information Retrieval.

I. Introduction

The use of HMMs [1] is a common procedure for the task of Audio Chord Estimation (discussed in Section II) [2].

The basic structure of the HMM we have used for modeling a chord sequence, was proposed in [3], and is described in Sec. II. Despite the fact that this HMM is relatively simple, it has been proved to be effective for the task of Audio Chord Estimation (ACE) [3]. Recent advances incorporate different musical facets in a single model, such as the meter, beat and musical key in order to estimate the progressions [4]–[8]. The exploitation of the musical context knowledge, instead of the chord sequence only, has also been proved to be beneficial for ACE [2].

In order to effectively aggregate different musical facets in a single model, two main approaches have been proposed in the related literature. In the first approach, the additional information is included in a single HMM by extending the state space of the model, as proposed, for example in [4], where the meter of the song is jointly modeled with the chord progression using conditional probabilities and a bigger transition probability matrix. Another approach consists in using Dynamic Bayesian Networks (DBN) [5], [6], that can be considered a generalization of HMMs. DBNs can be seen as a multi-stream HMM, where each stream models a different musical facet, and the conditional probabilities define the relation between them. Practically, any DBN can be equivalently modeled as an HMM with a very large state space [2]. However, this strategy implies a slower decoding stage and a less intuitive model representation.

An HMM is a probabilistic generative model for a sequence of observed variables $O = (O_1, O_2, \dots O_T)$. These observations, in our case the Pitch Class Profile (PCP) vectors, are generated from a hidden (i.e., not directly measurable) state sequence $Q = (q_1, q_2, \dots q_T)$ (i.e., in our case, the played chords). Each state at a generic time index t is $q_t \in \mathcal{S}$, where $\mathcal{S} = \{S_1, S_2, \dots\}$ is the finite set of all possible states. The Pitch Class Profile or Chroma Vectors, is a 12 bins, octave-independent measure of the strength of all possible notes in the considered audio signal. Each bin represents the energy of a given Pitch Class (i.e., the 12 notes of the western chromatic scale) at a given time instant (frame). The PCP vectors used in this paper are calculated using the *Loudness Based Chromagram*[1] [6], without beat-dependent smoothing. The *hop-size* between adjacent frames was set to 93 ms.

In this paper, we propose an extension to the basic model in order to integrate the beat/meter information without appending an additional layer to the basic HMM, and without modeling additional conditional probabilities in the transition matrix. Furthermore, we propose, a two step approach in the estimation stage which includes, first, an estimation of the a-posteriori likelihood of each single chord and, then, a maximum likelihood estimation of the chord sequence using the Viterbi algorithm.

The paper is organized as follows. In Sec. II we describe the details about the HMM, whereas in Sec. III we introduce the proposed model. Experimental setup, simulation results and conclusions are given in Sec. IV, Sec. V, and Sec. VI, respectively.

II. The Basic Model

In this model, the set of all possible states includes 12 major chords, 12 minor chords and a *no-chord* symbol, so that the cardinality of the state set is $|\mathcal{S}| = 25$. Each observation O_t is related to the "real" underlying state by the *emission probability* $b_j(O_t) = P(O_t|q_t = S_j)$ that measures the probability that O_t is observed at a given time t, when the actual state is $q_t = S_j$. It is a common practice for ACE, to model the emission probability by a 12-dimensional multivariate Gaussian probability distribution [2] (the PCP has 12 dimensions).

For a first order Markov Chain, the transition between two consecutive states (i.e., from q_t and q_{t+1}) is modeled using a *transition probability* matrix. The emission probability, the

[1] https://patterns.enm.bris.ac.uk/hpa-software-package

transition probability matrix and the initial state distribution form the parameter set λ of the HMM, which is denoted as

$$\lambda = (\boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (1)$$

$\boldsymbol{A} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is a matrix whose elements

$$a_{i,j} = P(q_t = S_j | q_{t-1} = S_i)$$

represents the probability that a chord S_i is followed by chord S_j. The parameter $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{S}|}$ is the initial distribution, or *prior*, $\boldsymbol{\mu} \in \mathbb{R}^{12\times|\mathcal{S}|}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{12\times12\times|\mathcal{S}|}$ are respectively the mean vectors and the covariance matrices for a multivariate (12-dimensional) Gaussian distribution that models the emission probabilities. Since the process that generates the chord sequence is assumed to be a stationary Markov Chain, the transition probabilities $a_{i,j}$ do not depend on the current time index t. Further details can be found in [1].

A. Setting the model parameters

Several methods to adjust the model parameters $(\boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ have been proposed in the literature [1]. Each one of these methods try to maximize a given optimality criterion.

For ACE systems, there are two main classes of techniques, the *Machine Learning* approach, (*trained*, or *Data-Driven* systems), and the *Expert* approach. The former uses a training set of annotated songs and adjusts the model parameters, whereas the latter uses the musical knowledge of an human expert to "manually" tune the system parameters. A combination of data-driven and expert systems is also possible. An overview of the commonly used techniques for parameters tuning in ACE method is provided in [2].

As mentioned in the previous section, the emission probabilities are modeled with a multivariate Gaussian probability distribution. The mean vectors $\boldsymbol{\mu}$ and the covariance matrices $\boldsymbol{\Sigma}$ are trained using a subset of the dataset described in Sec. IV. More precisely, a 12-dimensional mean vector is calculated for each of the 25 chord template by averaging the PCP vectors that belong to the same chord type. Similarly, the covariance matrices for each chord template, are also estimated from the PCP vectors. The transition probability matrix $\boldsymbol{A}$ is calculated by counting the number of transitions between each chord type, and then is normalized in order to have the sum of the elements of each row equal to 1. The initial distribution $\boldsymbol{\pi}$ is calculated by counting the occurrences of each chord type and then is normalized to obtain a probability distribution.

B. The Decoding Step

The aim of the decoding step (or inference step) of an HMM is to find the optimal (according to a given criterion) sequence of states q_t that best "explains" the observation O_t for $t = 1 \dots T$. The decoding can be achieved in different ways [1].

The commonly used decoding procedure in the HMM based ACE systems is the well known Viterbi algorithm [1], [9]. Without going into the implementation details, the Viterbi algorithm finds the best state sequence Q that maximizes $P(Q|O, \lambda)$ (or equivalently $P(Q, O|\lambda)$). The result of the Viterbi algorithm is the maximum likelihood *path* that is the maximum likelihood chord sequence given the observation sequence O and the model parameters λ.

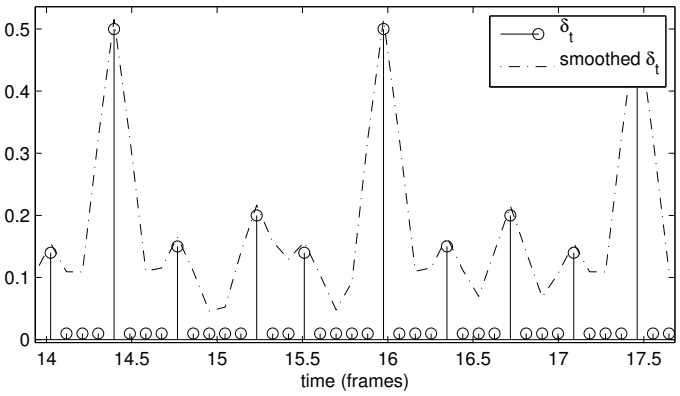

Fig. 1. An example of δ and its smoothed version $\widetilde{\delta}$.

III. The Proposed Model

As already said in Sec. I, the well performing ACE systems aggregate several musical facets as beat/metric information or musical key knowledge in a single model in order to better characterize the harmonic progression of a musical piece and gain estimation accuracy.

A. Time-variant transition probability matrix

In order to include the beat/meter information, we first have to estimate the tuning frequency [10], the bar and beat positions. For that purpose we use the method proposed in [11]–[13] and freely available as *VAMP* plug-in[2] for *Sonic Visualizer*[3]. Assuming a standard meter signature $\left(\frac{4}{4}\right)$, the beat tracker provides the beats position (time-stamp) and the metric information of each beat within a measure (i.e., the first beat of a measure is marked with "1", the second with "2" and so on until the last beat of the measure, that is "4").

The output of the beat/meter tracker is used to calculate the meter index function $M_t \in \{0, 1, 2, 3, 4\}$, where the index "0" means that no beat is detected at the time t. Based on the assumption that a chord change is more likely to happen at the beginning of a measure ("1"), less likely on "3", even less likely on "2" and "4" and very unlikely during off-beat frames (i.e., meter marked as "0") [5], we can recompute $\boldsymbol{A}$ at each time instant t in order to take into account this information.

The elements on the main diagonal of $\boldsymbol{A}$ indicate the probability to have no transition (i.e., $q_t = q_{t-1}$), and the off-diagonal elements of $\boldsymbol{A}$ give the probability to have a transition ($q_t \neq q_{t-1}$). The basic idea is to modulate the balance within the elements on the main diagonal and the rest of the matrix in order to modulate the probability of having (or not) a chord transition between time $t-1$ and t, using the beat/meter information provided by the beat tracker. In that way, the transition matrix is no more time-invariant, but depends on the meter information at each time instant. Hence, our model becomes:

[2]http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html
[3]http://www.sonicvisualiser.org/

$$\lambda = (\boldsymbol{A}_t, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{2}$$

At this point we define the probability density function of the transitions given the meter index m_i as:

$$P(\text{chord\_change}|m_i) = \begin{cases} 0.01 & \text{if } m_i = 0 \\ 0.5 & \text{if } m_i = 1 \\ 0.15 & \text{if } m_i = 2 \\ 0.20 & \text{if } m_i = 3 \\ 0.14 & \text{if } m_i = 4 \end{cases} \tag{3}$$

The values in (3) are estimated by counting the chord transition on each meter index using 50% of the data set (discussed in Sec. IV). For each meter index, a chord transition is considered as detected if there is a transition within a grace time of ± 2 time frames in order to mitigate the effect of the errors in the beat tracker stage.

In order to calculate $\boldsymbol{A}_t$ that will be used in the decoding stage, we have to evaluate the weighting coefficient δ_t at each time instant:

$$\delta_t = P(\text{chord\_change}|m_i = M_t). \tag{4}$$

Again, in order to take into account the errors in the beat tracker, we smooth δ using a Gaussian kernel with a standard deviation of 2 frames, obtaining the smoothed weighting coefficients $\widetilde{\delta}$. An example of δ and $\widetilde{\delta}$ is illustrated in Fig. 1.

Now we can calculate $\boldsymbol{A}_t$ by first computing the matrix:

$$\boldsymbol{W}_t = (1 - \widetilde{\delta}_t) D(\boldsymbol{A}) + \widetilde{\delta}_t \bar{D}(\boldsymbol{A}), \quad \forall t = 1 \dots T, \tag{5}$$

where

$$D(\boldsymbol{A}) = \begin{pmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{i,i} \end{pmatrix} \tag{6}$$

and $\bar{D}(\boldsymbol{A}) = \boldsymbol{A} - D(\boldsymbol{A})$.

In this way we modulate the ratio between the main diagonal (i.e., probabilities of no transition) and the rest of the matrix (i.e., probabilities of transition).

Furthermore, we have to normalize $\boldsymbol{W}_t$ in order to obtain again a transition probability matrix $\boldsymbol{A}_t$ in the following way:

$$a_{i,j,t} = \frac{w_{i,j,t}}{\sum_j w_{i,j,t}}, \quad \forall i, \forall t, \tag{7}$$

where $w_{i,j,t}$ are the elements of $\boldsymbol{W}_t$.

See Fig. 2 for an example of two different $\boldsymbol{A}_t$.

Another possible strategy is to estimate 5 different matrices from the dataset, one for each meter index, and chose the right one for each time instant during the decoding stage.

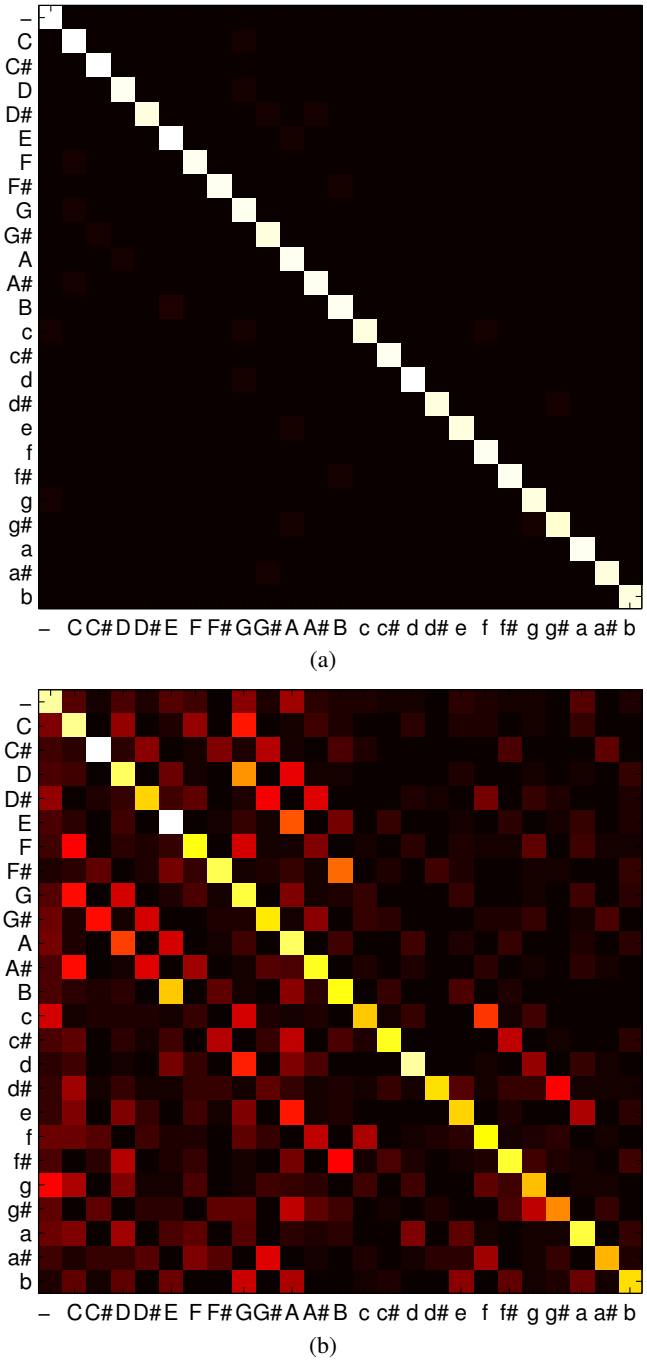

Fig. 2. Example of transition probability matrix $\boldsymbol{A}_t$: (a) the chord transition is very unlikely; (b) the chord transition is very likely (note that the values outside the main diagonal has been equally rescaled for visualization purposes).

The main disadvantage of this approach is that the amount of data needed for the storage becomes larger as the number of possible chords increases. Furthermore, our approach uses a smoothly-changing transition probability matrix that mitigates possible estimation errors of the beat tracker. In the following sub-section, we describe the proposed modifications in the decoding stage.

B. Two-stage Decoding

The classical decoding stage for the HMMs used in ACE systems follows the method illustrated in Sec. II-B, that is a maximum likelihood estimation of the chord sequence, or path, given the observations. The Viterbi algorithm uses the emission probabilities $b_j(O_t)$ for each state S_j at a given time t to calculate the *forward* coefficient

$$\alpha_t(j) = P(O_1 \dots O_t, q_t = S_j|\lambda)$$

that are used for the computation of the maximum likelihood state sequence path [1].

Furthermore, a *Maximum-A-Posteriori* (MAP) estimation of the chord sequence can also be performed [1]. The coefficients $\gamma_t(j)$ maximize $P(q_t = S_j|O, \lambda)$, that is the probability of choosing the states q_t that are *individually* most likely. The $\gamma_t(j)$ are calculated using the *forward-backward* procedure [1] as follows:

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\sum_j \alpha_t(j)\beta_t(j)}, \tag{8}$$

where $\beta_t(j) = P(O_{t+1} \dots O_T|q_t = S_j, \lambda)$ are the *backward* coefficients. In terms of emission and transition probability, $\alpha_t(j)$ and $\beta_t(j)$ are recursively calculated, as in [1], using the following equations:

$$\alpha_{t+1}(j) = \left[\sum_i \alpha_t(i)a_{i,j,t}\right] b_j(O_{t+1}), \tag{9}$$

with $\alpha_1(j) = \pi_j b_j(O_1)$ and

$$\beta_t(i) = \sum_j a_{i,j,t} b_j(O_{t+1})\beta_{t+1}(j), \tag{10}$$

with $\beta_T(i) = 1, \quad \forall i$. In our case, the coefficients $a_{i,j}$ depend on t, since $\boldsymbol{A}$ is time-variant.

Usually, when HMMs are used for ACE, the chord sequence is assumed to be a Markovian process. In practice a chord sequence is not, in general, a Markov Chain. That means that the Viterbi is a sub-optimal estimation for the chords sequence.

In order to improve the estimation accuracy and better capturing the temporal behaviour of a chords progression, we propose a two-stage decoding system.

In the **first stage** we maximize the probability (MAP) of choosing the states q_t that are *individually* most likely, given the sequence of the observation (i.e., maximize $P(q_t = S_j|O, \lambda)$).

TABLE I. EXPERIMENTAL RESULTS. THE ROW "IMPROVEMENT" SHOWS THE DIFFERENTIAL IMPROVEMENTS BETWEEN EACH ALGORITHM VARIANT AND THE BASIC ALGORITHM.

Score	**HMM**	**HMM+B**	**HMM+2S**	**HMM+B+2S**
Mirex2010	0.751	0.763	0.763	0.773
Improvem.	-	1.2%	1.2%	2.2%

The **second stage** is the Viterbi algorithm applied to the coefficients $\gamma_t(j)$ obtained by the previous stage, instead of the emission probabilities $b_j(O_t)$.

IV. EVALUATION

The proposed ACE algorithm has been evaluated using the same method proposed in [14] with the software *MusOOEvaluator*[4] and the pre-set called *Mirex2010*, on the same database. The *Mirex2010* evaluation metric calculates a segment-based score by considering matches between overlapping segments of the estimated chord sequence and the annotated chord sequence.

The total score for a given song is evaluated by summing the length, in seconds, of all the segments that are marked as correct match, and dividing the result by the total length of the song. The *Mirex2010* pre-set takes into account the differences between the chord vocabulary of our ACE method and the chord vocabulary of the annotations. Our method can identify 25 kinds of chords (12 major, 12 minor and 1 no-chord) while the annotated labels have a wider vocabulary. In order to have a score that is defined for each segment, *MusOOEvaluator* performs a mapping of the annotated labels to the set of the major and minor chords (i.e., the chord `C7` is mapped to `Cmaj`, and so on). Furthermore, the *Mirex2010* metric does not take into account the chord inversion. This means that if the bass note of a chord in the annotation is different from the estimation, but the chord in its original state is correct, this evaluation metric marks the estimation as a correct match (i.e., `Cmaj/G` is considered equivalent to `Cmaj`).

The ACE algorithm has been tested in four variants, namely: the basic model (**HMM**), the time variant model with the beat/meter information (**HMM+B**), the basic model with two-stage decoding (**HMM+2S**), and the combination of the beat/meter model plus the proposed two-stage decoding (**HMM+B+2S**).

During the training step of our algorithm, we have used the 60% of the dataset, and the results reported in the Tab. I are evaluated using the remaining 40% of the dataset.

The proposed algorithm has been tested on a set of 176 hand-labelled Beatles songs. The chord annotations are part of the *Isophonics*[5] dataset provided by Queen Mary University of London [15]. We have selected these songs in order to be able to compare the performance of the proposed algorithm with the state of the art on the same dataset.

V. EXPERIMENTAL RESULTS

As we can see in Tab. I, each of the proposed variants gives an improvement with respect to the base-line. As for other

[4]https://github.com/jpauwels/MusOOEvaluator

[5]http://www.isophonics.net/datasets

methods [4], [5], the meter/beat knowledge has been proven to be beneficial for ACE task. The method proposed in [4] combines the beat/meter information using an advanced model of the rhythm that considers the standard meter signature $\frac{4}{4}$ in combination with the $\frac{3}{4}$ meter at two different metrical level: the *tactus* and the *tatum*. The level called *tactus* (beat) is the most salient (in terms of psychoacoustics) metrical level, that corresponds to the foot-tapping rate. The level *tatum* is the smallest metrical subdivision (atom) of the *tactus* [4].

The metrical analysis is done on a beat-averaged chromagram and the meter/beat information are integrated in the HMM obtaining a large transition probability matrix. The accuracy result of [4] without the meter information is 0.688 while the accuracy with the meter information and a *tactus* analysis is 0.704, and with *tatum* analysis is 0.728 that gives, respectively, a differential improvement of 1.6% and 4%.

The work presented in [5] models the metric position in a DBN. The chromagram used is beat-averaged. The conditional probabilities to have a chord transition given the metric position is modeled similarly to our approach (3), but instead of calculating a time-variant transition matrix, they incorporate the metric structure in a separate stream of the DBN. The accuracy result of [5] without the meter information is 0.663 while the accuracy with the meter information is 0.674 that gives an improvement of 1.1%.

The relative improvements of these two algorithms are comparable with the one obtained by our proposal, except for [4] with *tatum* analysis that showed a greater performance benefit. This suggests that incorporating the meter information can be effectively achieved using different strategies.

Moreover, the advantage of our proposed method is that it is possible to add the meter structure information to the model without increasing the number of the states of the model or using the multi-stream approach of the DBNs. Furthermore, a more accurate meter analysis that considers also different meter signatures (i.e., $\frac{3}{4}$), like in [4], would be beneficial and is subject of further developments.

The proposed two-stage decoding offers a better discriminative performance over the single Viterbi decoding for ACE that uses HMM. Intuitively, if a chord sequence is a Markov process, there should be no improvement in performances, since the Viterbi decoder is already a maximum likelihood path estimation. This suggests that in practice, a chord sequence can be effectively modeled with a HMM (or DBN), but the nature of a chord sequence is not a Markov Chain in general. For this reason a two-stage decoding procedure can help in better capturing the temporal behaviour of a chord progression.

VI. Conclusions

In this paper we have proposed a new approach for modeling the beat/meter structure of a song in the context of the Audio Chord Estimation that implies the introduction of a time-variant transition probability matrix. Furthermore, we have proposed a two-stage decoding procedure that helps in capturing the temporal behaviour of chords sequences. In addition, our approach to the metric structure modeling simplifies the training and the decoding stage with respect to other approaches that involve the use of a bigger HMM or DBN that increase the state space in order to define several conditional probabilities.

Experimental results show that the proposed framework is effective as the classical approaches in modeling the meter structure, but with a substantially reduced model complexity. Moreover, the two-stage decoding procedure produces a significant improvement in the chords estimation accuracy.

References

[1] Lawrence Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] Matt McVicar, Raúl Santos-Rodríguez, Yizhao Ni, and Tijl De Bie, "Automatic chord estimation from audio: a review of the state of the art," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 556–575, 2014.

[3] Alexander Sheh and Daniel P.W. Ellis, "Chord segmentation and recognition using EM-trained hidden markov models," *Proceedings of the International Conference on Music Information Retrieval, (ISMIR)*, 2003.

[4] Hélène Papadopoulos and Geoffroy Peeters, "Joint estimation of chords and downbeats from an audio signal," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 138–152, 2011.

[5] Matthias Mauch and Simon Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1280–1289, 2010.

[6] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie, "An end-to-end machine learning system for harmonic analysis of music," *IEEE Transaction on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1771–1782, 2012.

[7] Matthias Mauch, Katy Noland, and Simon Dixon, "Using musical structure to enhance automatic chord transcription," *Proceedings of 10th International Society for Music Information Retrieval Conference, (ISMIR)*, 2009.

[8] Alessio Degani, Marco Dalai, Riccardo Leonardi, and Pierangelo Migliorati, "Harmonic change detection for musical chords segmentation," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2015.

[9] Andrew J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[10] Alessio Degani, Marco Dalai, Riccardo Leonardi, and Pierangelo Migliorati, "Comparison of tuning frequency estimation methods," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5917–5934, 2015.

[11] Matthew E. P. Davies and Mark D. Plumbley, "Context-dependent beat tracking of musical audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1009–1020, 2007.

[12] Adam M. Stark, Matthew E. P. Davies, and Mark D. Plumbley, "Real-time beat-synchronous analysis of musical audio," *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx)*, 2009.

[13] Daniel P. W. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[14] Johan Pauwels and Geoffroy Peeters, "Evaluating automatically estimated chord sequences," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pp. 749–753, 2012.

[15] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez, "Symbolic representation of musical chords a proposed syntax for text annotations," *Proceedings of the International Conference on Music Information Retrieval, (ISMIR)*, 2005.