POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria e Design del suono
Dipartimento di Elettronica e Informazione

Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony

Supervisor: Prof. Augusto Sarti
Assistant supervisor: Dr. Massimiliano Zanoni

Master graduation thesis by:
Bruno Di Giorgi, ID 740696

Academic Year 2011-2012

POLITECNICO DI MILANO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria e Design del suono
Dipartimento di Elettronica e Informazione

Riconoscimento automatico di accordi basato sul modello probabilistico dell'armonia modale diatonica

Relatore: Prof. Augusto Sarti
Correlatore: Dr. Massimiliano Zanoni

Tesi di Laurea di:
Bruno Di Giorgi, matricola 740696

Anno Accademico 2011-2012

Abstract

One of the distinctive traits of our society in the last decade is the availability and consequent fruition of multimedia content in digital format. The Internet, the growing density of storage systems and the increasing quality of compressed file formats, played the main roles in this revolution. Nowadays, audio and video contents are easily created, stored and shared by million of people.

This huge amount of data has to be efficiently organized and archived, to ease the fruition of large databases as online shops (as Amazon, iTunes) and content sharing website (Youtube, Soundcloud). The task of extraction of meaningful descriptors from digital audio and classification of musical content are addressed by a new research field named Music Information Retrieval (MIR). Among the descriptors that MIR aims at extracting from audio are rhythm, harmony, melody. These descriptors are meaningful for a musician and can find many applications as computer aided music learning, and automated transcription. The high demand for reliable automated transcriptions comes from the hobby musicians too. Official transcriptions are not always published and often the information about chords is enough to reproduce the song.

This thesis propose a system that performs the two tasks of beat tracking and chord recognition. The beat-tracking subsystem exploits a novel technique in finding the beat instants, based on the simultaneous tracking of many possible paths. This algorithm provide a useful self-evaluation property that can be exploited to achieve better accuracy. The downbeat is extracted by the same algorithm, proving the validity of the same approach at higher metrical level. The chord-recognition system proposed contemplates all the four most used key modes in western pop music (previously only major and minor modes are considered). Two novel parametric probabilistic models of keys and chords are proposed, where each parameter has a musical meaning. The performances of the two parts of the system exceed those taken as state-of-art reference. Finally the information gathered by our

system is exploited to compute a set of three novel harmony-based features.

Sommario

Uno dei tratti distintivi della nostra societá nell'ultimo decennio é la disponibilitá, e conseguente fruizione, di contenuti multimediali in formato digitale. Internet, la crescente densitá dei sistemi di storage e l'aumento della qualitá dei formati di file compressi, sono i protagonisti di questa rivoluzione. Al giorno d'oggi, contenuti audio e video sono facilmente creati, immagazzinati e condivisi da milioni di persone.

É necessario che questa enorme quantitá di dati sia efficientemente organizzata e archiviata, per facilitare la fruizione di grandi basi di dati come negozi online (Amazone, iTunes) e siti di condivisione di contenuti (Youtube, Soundcloud). Il compito di estrazione di descrittori significativi da file audio e la classificazione del contenuto musicale sono affrontati da una nuova area di ricerca chiamata Music Information Retrieval (MIR). Tra i descrittori che MIR mira ad estrarre dall'audio ci sono il ritmo, l'armonia, la melodia. Questi descrittori sono significativi per un musicista e possono trovare molte applicazioni, ad esempio nello studio della musica con il computer, o per la trascrizione automatica di brani musicali. La grande richiesta per sistemi affidabili di trascrizione automatica viene anche dai musicisti non professionisti, in quanto non sempre vengono pubblicate trascrizioni ufficiali dei brani e la progressione di accordi é abbastanza per suonare il brano desiderato.

Questa tesi propone un sistema che esegue i due compiti di tracciamento del beat e riconoscimento degli accordi. Il sotto-sistema di tracciamento del beat sfrutta una nuova tecnica per trovare gli istanti di beat, basata sul tracciamento simultaneo di piú sentieri. Questo algoritmo fornisce un'utile proprietá di auto-valutazione che puó essere sfruttata per migliorarne l'accuratezza. I primi beat delle misure sono estratti mediante lo stesso algoritmo, provando cosí la validitá dello stesso approccio al livello gerarchico superiore. Il sotto-sistema di riconoscimento degli accordi proposto considera tutti e quattro i modi piú usati nel pop occidentale (precedentemente solo i modi maggiore e minore erano stati considerati). Due nuovi modelli probabilistici parametrici per gli accordi e le tonalitá sono proposti, dove ogni parametro ha un preciso significato musicale. Le prestazioni delle due parti del sistema superano quelle considerate come riferimento dello stato dell'arte. Infine, le informazioni raccolte dal nostro sistema sono sfruttate per calcolare tre nuove descrittori emotivi basati sull'armonia.

Contents

Abstract **I**

Sommario **V**

1 Introduction **3**

1.1 Beat Tracking and applications 4

1.2 Chord Recognition 5

1.3 Thesis Outline 6

2 State Of Art **7**

2.1 State Of Art in Beat Tracking 7

2.1.1 Onset detection function 7

2.1.2 Tempo estimation 8

2.1.3 Beat detection 8

2.1.4 Time signature and downbeat detection 9

2.2 State Of Art in Chord Recognition 9

2.2.1 Chromagram extraction from audio domain 9

2.2.2 Chromagram enhancement 10

2.2.3 Chord Profiles 10

2.2.4 Chord sequences 11

2.2.5 Musical contexts 12

2.2.6 Key Extraction 12

3 Theoretical Background **15**

3.1 Musical background 15

3.1.1 Pitch and pitch classes 15

3.1.2 Chords 17

3.1.3 Tonal music and keys 18

3.2 Audio features analysis 21

3.2.1 Short-Time Fourier Transform 21

3.2.2 Chromagram 22

3.3 Dynamic programming 26
3.4 Hidden Markov Models 29
3.5 Viterbi decoding algorithm 31
3.6 Dynamic Bayesian networks 32

4 System Overview 35
4.1 Beat tracking system 35
4.1.1 Onset strength envelope 35
4.1.2 Tempo estimation 39
4.1.3 States reduction 42
4.1.4 Beat tracking 44
4.1.5 Downbeat tracking 49
4.2 Chord recognition 53
4.2.1 Chord Salience matrix 55
4.2.2 Beat-synchronization 56
4.2.3 DBN Model for chord recognition 57
4.2.4 Key and chord symbols 58
4.2.5 Key node 59
4.2.6 Chord node 61
4.2.7 Chord Salience node 62
4.3 Feature extraction 62

5 Experimental Results 69
5.1 Beat Tracking 69
5.1.1 Evaluation 69
5.1.2 Dataset 71
5.1.3 Results 71
5.2 Chord Recognition 72
5.2.1 Evaluation 72
5.2.2 Dataset 73
5.2.3 Results 73
5.2.4 Harmony-related features 74

6 Conclusions and future work 77
6.1 Future works 78

A Detailed results 81

Bibliography 89

List of Figures

3.1 C major scale 17

3.2 Harmonization of C major scale. Using only notes from the scale, we obtain a sequence of triads of different types. 18

3.3 Most used diatonic modes and their harmonization. 20

3.4 Steps of the Chromagram calculation 23

3.5 Normalized spectrum $A(r, k)$. r is the time index and k is the frequency bin. In figure, for clarity, are showed only the first 256 k of 4096. Only the first half (2048) of frequency bins are meaningful, as the transform of a real signal is symmetric. . . 24

3.6 Simple and complex pitch profiles matrices. k is the frequency bin index and m is the pitch index. 25

3.7 Windows for bass and treble range chromagrams 27

3.8 Treble and bass chromagrams 27

3.9 State transition graph representation of hidden Markov Models. Nodes are states and allowed transitions are represented as arrows. 30

3.10 Hidden Markov Models are often represented by a sequence of temporal slices to highlight how q_t, the state variable at time t, depends only on the previous one q_{t-1} and the observation symbol at time t depends only on the current state q_t. The standard convention uses white nodes for hidden variables or states and shaded nodes for observed variables. Arrows between nodes means dependence. 31

3.11 A directed acyclic graph 32

3.12 Construction of a simple Dynamic Bayesian network, starting from the prior slice and tiling the 2TBN over time. 34

4.1 The blocks that constitutes our system. Three types of information are extracted starting from digital audio. Rhythmic information is extracted by our beat tracker. Harmony-related information is extracted by our chord-recognition subsystem. These information are middle level features and are addressed to musically trained users. Harmony-related features are then extracted starting from key and chord information, and is addressed to a generic user (no training needed). 36
4.2 Beat tracking system 37
4.3 Signal $s(n)$ and detection function $\eta(m)$. The two functions have different temporal resolutions but corresponding indices n and m are aligned in the figure for better understanding. . 39
4.4 Tempo estimation stage 39
4.5 Comb filter matrix F 41
4.6 Rayleigh distribution function with $\beta = 43$ compared to the sample distribution of periods in our dataset 41
4.7 Rhythmogram matrix $y_i(\tau)$. τ is the beat period expressed in number of samples of the ODF. 42
4.8 Period transition distribution extracted from the dataset . . . 43
4.9 Rhythmogram. Chosen peaks and final path are showed respectively as light and dark x.τ is the beat period expressed in number of samples of the ODF. 43
4.10 The performance of the beat tracking algorithm is influenced by N_{peaks} 44
4.11 Beat candidates c_j are the highest peaks within each region, marked with a cross. $\eta(m)$ is the onset detection function and m is its time-intex 45
4.12 Beat tracking stage 45
4.13 Join and split rules applied to 6 path at the first 3 iterations . 47
4.14 Increasing the number of paths ($N_p = 2, 6, 10$ in the figures) influences the coverage of the space 48
4.15 The performance of the beat tracking algorithm is influenced by N_p 48
4.16 Precision and recall parameters plotted against the threshold t_v 49
4.17 m is the sample index of tempo tracking songs and $\tau(m)$ is the estimated periodicity, expressed in number of ODF samples. At each iteration the algorithm smooths the tempo path in segments with least voted beats (most voted one are marked with an X). This way it can effectively correct the two tempo peaks, mistakenly detected by the earlier tempo tracking stage. 50

4.18 Performance of the algorithm slightly increase with the number of iterations 50

4.19 The spectral difference function for downbeat detection has one value per beat. n is the audio sample index and m is the beat index. The two functions have different temporal resolutions but corresponding indices n and m are aligned in the figure for better understanding. 51

4.20 The chroma variation function 53

4.21 Performances of downbeat tracking stage are maximized for a regularity parameter $\alpha = 0.5$ 54

4.22 Chord recognition scheme 55

4.23 Beat-synchronized bass and treble chromagrams. 57

4.24 The two-slice temporal Bayes net of our chord tracking system. Where shaded nodes are observed and white node are the hidden states nodes. L stands for beat label, and it represents beat labels assigned during downbeat detection. K is the Key node, C the Chords node, B the bass note node, C^B is the Bass beat-synchronized Chromagram and S the chord salience node. 58

4.25 The pitch profile $k^*(p)$ obtained from the dataset. 61

4.26 Norwegian Wood is a great example to show the power of the Modal Envelope descriptor. Verses of the song are in Mixolydian mode and Choruses are in Dorian mode. The descriptor perfectly and precisely follows the shift in mood of the song. In this particular case it also achieve optimal structural segmentation 63

4.27 Finer modal envelope feature obtained by a post-processing of chords. The song is "Angel" by Robbie Williams. Spikes from major mode to mixolydian mode reveal where the composer made use of modal interchange. The longer zones are instead temporary modulations (key change) to the parallel mixolydian mode. 64

4.28 This function is a time window used to compute the majMinFeature. The rationale is simple: just listened events have a stronger influence than those happened before. 65

4.29 The figure shows an example of the majMinFeature on the song "Hard Day's Night" by The Beatles. The deepest valleys fall exactly at the choruses, where most minor chords are found. 66

4.30 The figure shows an example of the Harmonic Rhythm feature on the song "Here Comes the Sun" by The Beatles. The longest peaks represents parts of the song where chords changes are very frequent. 67

5.1 Continuously correct segments are showed as grey rectangles, the longest is darker. $CML_t = 0.951$ and $CML_c = 0.582$ . . . 70

List of Tables

3.1 Pitch classes and their names. ♯ and ♭ symbols respectively rise and lower the pitch class by a semitone. 16

3.2 Pitches are related to their frequency using the standard reference frequency $A4 = 440Hz$. 17

3.3 Sequence of tones and semitones in the major scale 17

3.4 The four triad types. Intervals are specified in number of semitones. 18

3.5 Relationships of major scale notes with the tonic 18

3.6 Diatonic modes can be viewed as built sliding towards right a window over the major scale pattern. 19

3.7 Diatonic modes 19

3.8 Examples of representative songs for the four main diatonic modes 19

3.9 Diatonic modes 21

3.10 Diatonic modes 21

4.1 Performances of sum of squared differences (SSD) and KL Divergences (KLD) 52

4.2 Signature distribution and transition probabilities 53

4.3 Moods consistently associated with major and minor triads. . 64

5.1 Performances on the Beatles Dataset 71

5.2 Performances on the Robbie Williams Dataset 71

5.3 Performances of downbeat tracking on The Beatles Dataset . 72

5.4 Performances of downbeat tracking on the Robbie Williams Dataset 72

5.5 Performances on The Beatles dataset. krmRCO is computed mapping key mode sequences of our system to only major and minor modes, as they only are present in the ground truth annotations 73

5.6 Performances on the Robbie Williams dataset. All four the key modes are present in the annotations so we can meaningfully evaluate only our system 74
5.7 Results of the ρ index shows that in 5 of the 6 cases, there is a direct proportionality between our feature and the annotations. ME stands for Modal Envelope, Mm for MajMinRatio, HR for Harmonic Rhythm 75

A.1 Album IDs 88

Chapter 1

Introduction

The advent and the diffusion of the Internet and its increasing bandwidth availability, the growing density of newer storage systems, led to a new era for digital multimedia content availability and consumption. Music is representative of this revolution thanks to audio digital format such as mp3, aac, flac and new affordable music production tools. Online music stores as *iTunes* and *Amazon*, social platforms as *last.fm* and *soundcloud* are facing a crucial need to efficiently store and organize music content in huge databases. This involves the creation of meaningful descriptors to perform media search, classification and suggestion. This task has initially been accomplished by manually tagging songs with high-level symbolic descriptors (context-based approach). This approach is not suited for dealing with massive and ever increasing collections and, by definition, lacks of objectivity. The need of objective and automated paradigms to extract information directly from music signals (content-based approach) contributed to the birth of a new research area, named *Music Information Retrieval* (MIR), a branch of Multimedia Information Retrieval. MIR is a broad field of research that takes advantage of signal processing techniques, machine learning, probabilistic modelling, musicological and psychoacoustic theories. The fundamental layer for MIR applications is the extraction of features able to describe several characteristics of musical content. These are generally categorized in three levels [15]. Low-level features (LLF) are directly extracted from the audio signal using signal processing techniques. Mid-Level features (MLF) make use of LLF and musicological knowledge to infer descriptors such as *Melody*, *Harmony* and *Rhythm* [1]. High-level features have a higher degree of abstraction and are easily understandable by humans, like affective descriptors - emotional

[1] These tasks and others are the object of *Music Information Retrieval Evaluation eXchange* (MIREX) annual evaluation campaign.

tags or continuous spaces as arousal, valence, dominance - and non-affective descriptors as genre, *dynamicity*, *roughness*, etc.

In this thesis we will focus on the Mid level. Particularly we will address the problem of automatic beat tracking and chord recognition. Furthermore we propose three novel features computed from the extracted chord-progression by exploiting musicological background. We will then correlate these features to the emotion variation perceived in a song.

1.1 Beat Tracking and applications

Beat tracking is one of the most challenging task in the context of MIR field. Beat is a steady succession of pulses of that humans tend to follow when listening to a song.

Rhythm, as almost all aspects of music, is a hierarchical structure. It's common to consider three metrical levels. *Tatum* is the lowest level of this hierarchy. The next level is the beat level or *tactus*, the period at which most humans would tap their foot or snap their fingers. The last and highest level is the measure level. Measure is a segment of time defined by a given number of beats. Downbeats are the first beats of each measure.

Extracting beat from audio is very useful for many applications. Beat information, for example, can be exploited in subsequent beat-synchronous analysis (sampling informations using the time-grid given by beats), in score alignment and chromagram synchronization for chord recognition, as we will see later on. Beat-synchronous processing can have applications in time-stretching. Professional DJ softwares make use of beats position and tempo information to help the user making rhythmically smooth cross-fades between songs. In the music production field, music engineers can take great benefit from automatic slicing a track based on auto-detected beat instants, and then quantizing them to obtain a version with a steadier rhythm. A sequencer can vice versa adjust the tempo grid according to a track. Tempo, the period of the beats, can also be useful for automatic song library tagging. One of the id3 tags is in fact named *beat per minute* (bpm), and it indicates the average tempo of a song.

The downbeat extraction task has also many applications. Rhythmic pattern analysis can greatly benefit having a predefined grid over which to apply pattern recognition techniques. Downbeat positions can be also exploited as most likely temporal boundaries for structural audio segmentation.

Main techniques for beat-tracking work on an *Onset Detection Function* (ODF), extracted from the audio signal. This function is tailored as to highlight the transients and the start of new notes. Periodicities of this function represent the tempo. The ODF is then scanned to find a regular pattern of peaks spaced by the tempo. The search of downbeat is carried out by finding regular patterns among the beats, with periodicities of three or four beats.

In this thesis we propose a new beat sequence technique based on a new multipath tracking algorithm based on dynamic programming, aware of tempo changes. This novel technique increases accuracy in beat tracking by exploiting an iterative self evaluation. The same algorithm has been applied to downbeat detection with dynamic time signature tracking.

1.2 Chord Recognition

Automatic chord recognition task aims at generating chord transcriptions as similar as possible to those of highly trained musicians. Unlike beat-tracking, this isn't an easy task for hobby musicians too. However, chords are important for modern pop music, given they provide alone enough information to allow musicians of any level to perform a recognizable version of a song. This is confirmed by a great demand for chord transcriptions on the Internet, where some web sites provide archives of home-made transcriptions submitted by users.

Aside from automatic transcription, the chord-recognition task encompass similarity-based applications like score synchronization and cover identification, and is used for genre classification as well. The harmony of a song is also connected to mood. Many psychoacoustic researches demonstrate how sensitive humans are with respect to harmonic structure. Chord progressions can influence mood in many ways, mainly by exploiting specific patterns linked to known emotional responses in the listeners. Sloboda [40] showed the bounds between harmonic patterns such as cycle of fifths, unprepared harmony or cadences, with responses as *tears*, *shivers* and *racing heart*. This is exactly what the composer does while writing a song, he searches the right balance to achieve a precise emotional meaning, often in tune with other layers as lyrics, arrangement or melody.

Harmony is not an exception in being a hierarchical structure, as rhythm is. Above the chords level is the key level. In *tonal* music, as is the vast majority of the music, one note, called *tonic* or key root, has greater importance than others. Other notes and chords have meaning in relation to the tonic, that is consequently said to provide a context. The relationships between notes and the key root, as we will see in chapter 3, form the key mode, one of the main aspects that induce mood in the listener.

Main techniques are based on the comparison between the chromagram and a series of chord templates. Temporal correlation of chord sequence is addressed by creating probabilistic models or by filtering the chromagram in the time direction.

The goal of this thesis is to exploit diatonic modal harmony theory in order to improve chord transcription. We provide a novel parametric probabilistic modelling of chords and keys. Our model include all the four main key modes and not only the major and minor modes. Finally we exploit key mode and chord structure to extract harmony-related feature.

1.3 Thesis Outline

In chapter 2 we present some related works, representative of the state of the art in beat tracking and chord recognition techniques. Chapter 3 provides the theoretical background of algorithms and probabilistic models we use throughout our system. In chapter 4 attention is drawn to our system and we fully review each stage of beat and chord detection. Experimental results and comparison to existing systems is presented in chapter 5.

Chapter 2

State Of Art

In this chapter we will give an overview of the main existing approaches of beat tracking and chord recognition. The analysis will be subdivided in successive steps representing the common procedures in performing these tasks.

2.1 State Of Art in Beat Tracking

In this section we review the main existing approaches on the beat-tracking task. We split the analysis following the order of the building blocks of a standard beat-tracking system. Generally beat tracking task is divided in these successive steps:

- An *Onset Detection Function* (ODF) is generated from the input signal
- Periodicities in the ODF are highlighted in the Rhythmogram
- Beat positions are detected starting from the ODF and the Rhythmogram
- Downbeats are found between beats

2.1.1 Onset detection function

Most of the beat tracking algorithms are based on a mono-dimensional feature called Onset Detection Function (ODF) [4]. ODF quantifies the time-varying *transientness* of the signal where transients are defined as short intervals during which the signal evolves quickly in a relatively unpredictable way. More exaustive explanation of ODF will be given in Chapter 4.

Human ears cannot distinguish between two transients less than 10 ms apart, so that interval is used as the sampling period for ODFs. The process of transforming the audio signal (44100 samples/s) to ODF (100 samples/s) is called *reduction*. Many approaches have been proposed for reduction. Some make use of temporal features as envelope [38] or energy. Others take into account the spectral structure, exploiting weighted frequency magnitude. In[28], for example, a linear weighting $W_k = |k|$ is applied to emphasize high frequencies. Different strategies have advantages with different types of musical signals. We choose the spectral difference detection function proposed by [2] as the state of art for pop songs.

2.1.2 Tempo estimation

Periodicities in the ODF represent beat period or tempo of the song and are searched using methods as auto-correlation, comb-filter resonator or short-time Fourier Transform (STFT). A spectrogram-like representation of such periodicities is called Rhythmogram (Fig. 4.7). This task, concerning the beat rate instead of beat positions, takes the name of tempo estimation. In [12] was proposed a very effective way to find periodicities using a shift-invariant comb filter-bank. Tempo generally varies along the piece of music. The analysis is, therefore, applied at windowed frames of 512 ODF samples, with 75% overlap. One of the main problem, at this level is the trade off between responsiveness and continuity. In [12] this problem was assessed using a two state model, in which the "General State" takes care of responsiveness and the other, called "Context-Dependent State", try to maintain continuity.

2.1.3 Beat detection

The beat detection phase addresses the problem of finding the positions of beat events in the ODF. A simple peak picking algorithm would not be sufficient as there are many energy peaks that are not directly related to beats. Human perception as a matter of fact tends to smooth out inter-beat-intervals to achieve a steady tempo. This can be modelled, as proposed in [13], by an objective function that combines both goals: correspondence to ODF and interval regularity. Inter-beat-interval is the tempo, so it is derived from an earlier tempo detection stage. An effective search of an optimal beat sequence $\{t_i\}$ can be done in a simple neat way by assuming tempo as given and using a dynamic programming algorithm technique [1].

Irregularities in the detected tempo path are one of the main sources of error. We propose a novel beat-tracking technique that track simultaneously more likely beat sequences. In doing so it manages to identify and correct some of the errors carried on from earlier stages, mainly the tempo estimation stage.

2.1.4 Time signature and downbeat detection

Downbeat detection stage focuses on the highest level of rhythmic hierarchy. Outputs of this stage are the set of the first beats of each measure. As inter-beat-intervals sequence constitute the tempo, inter-downbeat-intervals, expressed in beats per measure, represents the time signature. Common time signatures are 4/4 and 3/4 meaning respectively four beats per measure and three beats per measure. In [17] a chord-change probability function is exploited in making decisions on higher level beat structure. In [16], bass drum and snare drum onsets are detected by a frequency analysis stage. Patterns formed by these onsets and their repetitions are used as cues for detecting downbeats. The algorithm used as the state of art is described in [11]. The input audio is down-sampled and a spectrum is calculated for every beat interval. A spectral difference function D is then obtained by Kullback-Leibler divergence between successive spectra. This function gives the probability that a beat is also a downbeat. Downbeat phase is then found by maximizing the correlation of D with a shifting train of impulses.

Our model exploits the same multipath algorithm to track the sequence of downbeats among beats. It exploits, as the downbeat's ODF, a combination of an energy based feature and a chroma variation function.

2.2 State Of Art in Chord Recognition

In this section we review the existing approaches in Chords and Keys extraction. Again, we split the analysis following the major steps undertaken by a standard algorithm starting from the audio signal.

2.2.1 Chromagram extraction from audio domain

Most of the chord-recognition algorithms are based on a vectorial feature called Chromagram, which will be described in detail in chapter 3. Chromagram is a pitch class versus time representation of the audio signal [43]. It is computed starting from Spectrogram by applying a mapping from linear to log-frequency. This procedure is most often accomplished by the constant-Q transform [5].

For the task of chord-recognition, Chromagram is needed to show the relative importance of pitch classes of notes played by instruments. The

Spectrogram, however, contains noise coming from percussive transients and contains harmonics (tones at integer multiples of the fundamental frequency of a note). Furthermore the overall tuning reference frequency may not be the same for all songs. It's therefore necessary to develop strategies and work-arounds to cope with these problems.

2.2.2 Chromagram enhancement

The basic approach in reducing percussive and transient noise is to apply a FIR low pass or a median filter to the Chromagram in the time axis. The same result is achieved as a side-effect of beat-synchronization, that consists in averaging Chroma vectors inside every beat interval. Beat-synchronization is usually performed in chord-recognition as proposed in [3]. Other methods include spectral peak picking ([18]) and time-frequency reassignment ([24]).

The Harmonics contribute to characterize the timbre of instruments but are not perceived as notes and have no role in chord perception. For the chord-recognition task therefore, their contribute is undesirable. To address this issue, in [18], spectral peaks found in the spectrogram contribute also to sub-harmonic frequencies, with exponentially decreasing weight. In [29] each spectrogram frame is compared to a collection of tone profiles containing harmonics.

For historical reasons the frequencies of the notes in our tuning system, the twelve-tone equal temperament, are tuned starting from the standard reference frequency of a specific note: $A4 = 440Hz$. This frequency in some songs vary in the interval between 415 Hz and 445 Hz, then it is necessary to determine it to obtain a reliable chromagram. The approach generally used is to generate a log-frequency representation of the Spectrogram with frequency resolution higher than the pitch resolution. In [21] 36 bins per octave are extracted. The same resolution is achieved with pitch-profiles collection matrices in [30]. Since our temperament has 12 pitch classes per octave, we obtain 3 bins per pitch. Circular statistics or parabolic interpolation allow us to find the shift of the peak from the centre bin, hence the shift of the reference frequency.

2.2.3 Chord Profiles

Chord recognition is achieved by minimizing a distance or maximizing a similarity measure between the time slices of the Chromagram and a set of 12-dimensional pitch class templates of chords. Chord theory and derivation of pitch class templates is treated in full detail in the next chapter. Inner product is used as a similarity measure in [21]. In [34] the use of Kullback-Leibler divergence as a distance measure is proposed. In [3], chords template vectors are centre-points of 12-dimensional Gaussian distribution with hand tuned covariance matrices.

2.2.4 Chord sequences

Finding a chord for each slice of the Chromagram would result in a messy and chaotic transcription, useless from any musical point of view. This is caused by 2 main factors: the percussive transients that results in a wide non-harmonic spectrum, and the melody notes and other non-chord passing notes that can make the automatic choice of the right chord an hard task. To obtain a reliable and musically meaningful chord transcription we must account for the connections and hierarchies of different musical aspects.

Chords are stable in a time-interval of several seconds. It is then necessary to find a strategy to exploit this evidence and find smooth chords progressions over time. A segmentation algorithm proposed in [14] uses a "chord change sensing" procedure that computes chord changes instants by applying a distance measure between successive chroma vectors. In [21] a low pass filtering of chroma frames and then a median filtering of frame-wise chord labels is performed. In [34] the smoothing is applied not to the labels but on the frame-wise score of each chord. The majority of chord transcription algorithm use probabilistic models as *Hidden Markov Models* (HMMs), explained in chapter 3, which are particularly suited for this task as they model sequences of events in a discrete temporal grid. In HMMs for chord recognition task, chords are the states and chroma vectors are the observations. The model parameters as the chord transitions and the chroma distribution for each chord express musically relevant patterns.

Chroma distributions are mainly based on chord profiles. One of the most important parameters is the self-transition, which models the probability that a chord remains stable. Between approaches exploiting HMMs, [3] is notable as the chord transition matrix is updated for each song, starting from a common base, that model the *a-priori* common intuition of a human listener. Another probabilistic model recently used [27] in the MIR field is the Dynamic Bayesian Network (DBN) [32], reviewed in chapter 3. DBN can be seen as a generalization of HMM that allows to model, besides chord transition patterns, any other type of musical context in a network of hidden and observed variables.

2.2.5 Musical contexts

Other musical context used along with chord transition patterns are Key, Metric position, and Bass note contexts.

The fundamental importance of Key in human perception of harmonic relationships is highlighted in chapter 3. This has been exploited successfully in many chord recognition systems. Some of them [39] use the Key information to correct the extracted chord sequence, others [6] try to extract the Key simultaneously to the chord sequence. The Key changes, or modulations, in a song are addressed only by some of the existing approaches, while the majority of them assumes the Key to remain constant throughout the song. The key modes addressed by these systems are major and minor modes.

Bass note (the lowest note of a chord) can be estimated by creating a separated Bass-range Chromagram that include only the low frequency pitches. The pitch class of the Bass note is likely to be a note of the chord. This assumption is exploited in [31] by creating a CPD of chords given a bass-range chroma vector.

Metric position can also be used as a context, exploiting the fact that chord changes are likely to be found at downbeat positions, as done in [35].

We propose a novel probabilistic model of keys that include, besides major and minor modes, the Mixolydian and the Dorian mode. The parameters of this model express meaningful events as different types of modulations. Furthermore, we propose a new conditional probability model of chords, given the key context. This model assigns three different parameters to different group of chords, based on the key mode and the relationship with the tonic.

2.2.6 Key Extraction

Key extraction is usually done by comparing Chroma vectors with a set of key templates. Correlation is used as similarity measure as in [18]. HMMs are used to track the evolution of key in a song. The best known key templates (the concept of key and tonality is reviewed in the chapter 3) are the Krumhansl's key profiles ([26]). They contains 12 values that show how pitch classes fit a particular key. This profiles were obtained by musicological tests and, as expected, agree with music theory. Krumhansl's key profiles are available for major and minor keys. Other key profiles are automatically extracted in [7] from a manually annotated dataset of folk songs.

To compute the keys we propose a hybrid system. It first weights our a-priori probability model by a vector of key root saliences, obtained by correlating the chromagram with a set of key root profiles. Then the keys sequence is extracted together with chord sequence by viterbi inference for the Dynamic Bayesian Network.

Chapter 3

Theoretical Background

In this chapter we will review the theoretical background and tools used in our technique. We will begin with the basic musical background needed to understand chords, keys and key modes. Then, we will introduce two low level signal processing tools, the *Short Time Fourier Transform* (STFT) and the Chromagram. Successively, we will explain the main concepts of the probabilistic models we used in the beat tracking system: the *Hidden Markov Models* (HMM). Finally we will review a generalization of HMM, the *Dynamic Bayesian Network* (DBN): the probabilistic model that will be used in the chord recognition system to model a number of hidden state variables and their dependencies.

3.1 Musical background

In this section we review some basic concepts of music theory. In particular we cover what is pitch and pitch classes, how chords are formed and their relation to key and modes. For a comprehensive reference we remind the reader to [44].

3.1.1 Pitch and pitch classes

Pitch is a perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from low to high ([25]). Pitch is proportional to log-frequency. In the Equal temperament it divides each octave (a doubling of frequency) in 12 parts:

$$f_p = 2^{\frac{1}{12}} f_{p-1}. \tag{3.1}$$

where f_p is the frequency of a note . In this study pitch and *note* terms are used as synonyms from now on. The distance between two notes is called *interval*, and is defined by a frequency ratio. The smallest interval, called *semitone*, is defined by the ratio

$$\frac{f_p}{f_{p-1}} = 2^{\frac{1}{12}}. \tag{3.2}$$

An interval of n semitones is therefore defined by $2^{\frac{n}{12}}$. The interval of 2 semitones is called *tone*.

Human are able to perceive as equivalent pitches that are in octave relation. This phenomenon is called *octave equivalence.* Pitch classes are equivalence classes that include all the notes in octave relation. Note names indicates, in fact, pitch classes (Table 3.1).

note name	pitch class #
C	1
C♯/D♭	2
D	3
D♯/E♭	4
E	5
F	6
F♯/G♭	7
G	8
G♯/A♭	9
A	10
A♯/B♭	11
B	12

Table 3.1: Pitch classes and their names. ♯ and ♭ symbols respectively rise and lower the pitch class by a semitone.

Octave is indicated by a number after the pitch class. In Table 3.2 pitches are related to their frequency in the the Equal temperament, tuned relative to the standard reference: $A4 = 440Hz$.

	Octave				
Note	2	3	4	5	6
C	66 Hz	131 Hz	262 Hz	523 Hz	1046 Hz
C♯/D♭	70 Hz	139 Hz	277 Hz	554 Hz	1109 Hz
D	74 Hz	147 Hz	294 Hz	587 Hz	1175 Hz
D♯/E♭	78 Hz	156 Hz	311 Hz	622 Hz	1245 Hz
E	83 Hz	165 Hz	330 Hz	659 Hz	1319 Hz
F	88 Hz	175 Hz	349 Hz	698 Hz	1397 Hz
F♯/G♭	93 Hz	185 Hz	370 Hz	740 Hz	1480 Hz
G	98 Hz	196 Hz	392 Hz	784 Hz	1568 Hz
G♯/A♭	104 Hz	208 Hz	415 Hz	831 Hz	1661 Hz
A	110 Hz	220 Hz	440 Hz	880 Hz	1760 Hz
A♯/B♭	117 Hz	233 Hz	466 Hz	932 Hz	1865 Hz
B	124 Hz	247 Hz	494 Hz	988 Hz	1976 Hz

Table 3.2: Pitches are related to their frequency using the standard reference frequency $A4 = 440Hz$.

Scales are sequences of notes that cover the range of an octave. Scales are classified based on the intervals between successive notes. The particular sequence of semitone and tone intervals depicted in the Table 3.3 compose the *major scale* (Fig. 3.1).

T	T	S	T	T	T	S

Table 3.3: Sequence of tones and semitones in the major scale

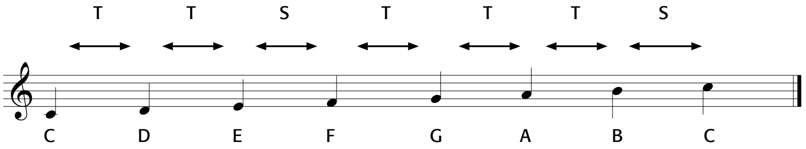


Figure 3.1: C major scale

3.1.2 Chords

Chords are the combination of two or more intervals of simultaneous sounding notes. Chords are classified by their number of notes and the intervals between them.

The most used chord type in western pop music is the *triad*. Triads are three note chords, and divide in 4 types depending on the intervals between their notes (Table 3.4).

type	major	minor	augmented	diminished
label	maj	min	aug	dim
interval 2	3	4	4	3
interval 1	4	3	4	3

Table 3.4: The four triad types. Intervals are specified in number of semitones.

We can build a triad on each note of the major scale, using only scale notes. This process is called *harmonization* of the major scale. We obtain the series of triads showed in Fig. 3.2.

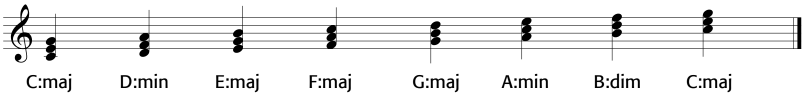


Figure 3.2: Harmonization of C major scale. Using only notes from the scale, we obtain a sequence of triads of different types.

3.1.3 Tonal music and keys

Music is *tonal* when a pitch class more important than others can be outlined. This pitch acts as centre of gravity and is called *tonic* or *key root*. The tonic is the most stable pitch class where to end a melody, to obtain a final resolution (think of any western national anthem). The triad built on the tonic is the most likely chord where to end a song. Most of the western music is tonal.

The concept of key *mode* relates to the particular choice of other notes in relation with the key root. A mode correspond to a scale in the sense that notes are taken from a particular scale of the key root. For example, given the intervals pattern of the major scale (Table 3.3), the pattern of intervals of scale notes with the key root is

T	T	S	T	T	T	S
2	4	5	7	9	11	12

Table 3.5: Relationships of major scale notes with the tonic

Most used modes are taken from a set of scales called the *diatonic* scales that include the circular shifts of the major scale. They are therefore called *diatonic modes.*

... T T S T T T S $\boxed{\text{T T S T T T S}}$ T T S T T T S ...

Table 3.6: Diatonic modes can be viewed as built sliding towards right a window over the major scale pattern.

Ionian (Major)	T T S T T T S
Dorian	T S T T T S T
Frigian	S T T T S T T
Lydian	T T T S T T S
Mixolidian	T T S T T S T
Eolian (Minor)	T S T T S T T
Locrian	S T T S T T T

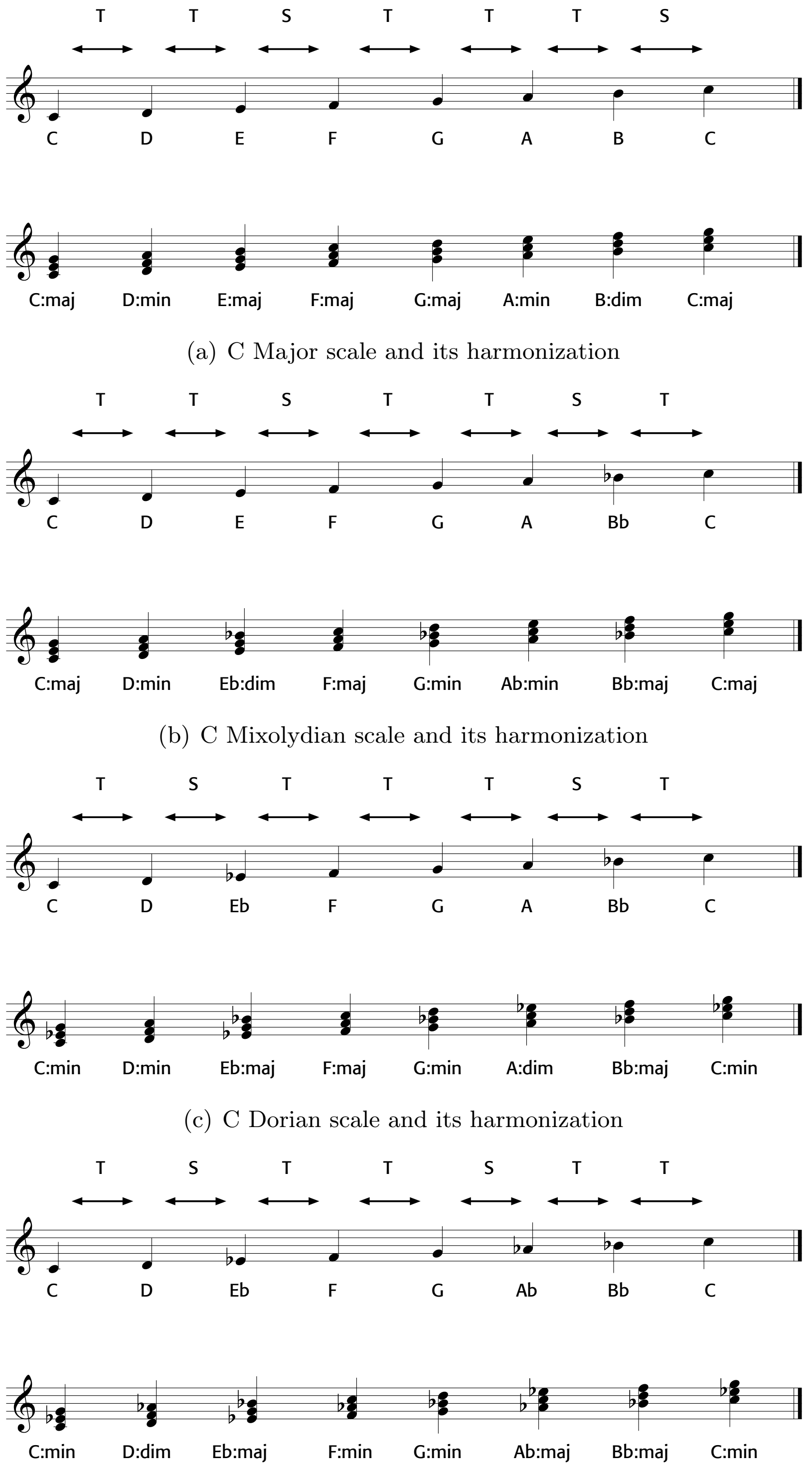
Table 3.7: Diatonic modes

Most used modes in western music (Table 3.8) are Major, Mixolydian, Dorian and Minor. Their scale and set of triads, in the key root of C, are showed in Fig. 3.3.

Major	Mixolydian
Imagine (John Lennon)	Sweet Child Of Mine (Guns 'n' Roses)
Blue Moon (Rodgers, Hart)	Don't Tell Me (Madonna)
We are golden (Mika)	Teardrop (Massive Attack)
Something Stupid (Robbie Williams)	Millennium (Robbie Williams)
Dorian	Minor
I Wish (Steve Wonder)	Losing My Religion (REM)
Oye Como Va (Santana)	Rolling In The Deep (Adele)
Great Gig In The Sky (Pink Floyd)	Have a Nice Day (Bon Jovi)
Mad World (Gary Jules)	I Belong To You (Lenny Kravitz)

Table 3.8: Examples of representative songs for the four main diatonic modes

In western music, modes have been shown to be linked with emotions, as for instance minor modes are related to sadness and major to happiness ([22]). If we evidence the intervals with key root for each diatonic mode, we can order them by the number of risen and lowered notes (Table 3.9).


T T S T T T S
C D E F G A B C
C:maj D:min E:maj F:maj G:maj A:min B:dim C:maj
T T S T T S T
C D E F G A Bb C
C:maj D:min Eb:dim F:maj G:min Ab:min Bb:maj C:maj
T S T T T S T
C D Eb F G A Bb C
C:min D:min Eb:maj F:maj G:min A:dim Bb:maj C:min
T S T T S T T
C D Eb F G Ab Bb C
C:min D:dim Eb:maj F:min G:min Ab:maj Bb:maj C:min

(a) C Major scale and its harmonization

(b) C Mixolydian scale and its harmonization

(c) C Dorian scale and its harmonization

(d) C Minor scale and its harmonization

Figure 3.3: Most used diatonic modes and their harmonization.

Lydian	2 4 **6** 7 9 11 12	1 risen note	brightest
Ionian (major)	2 4 5 7 9 11 12	-	
Mixolidian	2 4 5 7 9 **10** 12	1 lowered note	
Dorian	2 **3** 5 7 9 10 12	2 lowered notes	
Eolian (minor)	2 3 5 7 **8** 10 12	3 lowered notes	
Frigian	**1** 3 5 7 8 10 12	4 lowered notes	
Locrian	1 3 5 **6** 8 10 12	5 lowered notes	darkest

Table 3.9: Diatonic modes

This ordering, not the circular shift one, is relevant from an emotional point of view. We believe it is consistent with a direction in the emotional meaning of the modes. We can notice how the four most used modes are in central positions. A collection of keywords used to describe emotions in these four modes is provided in Table 3.10 [23].

Major	Mixolydian	Dorian	Minor
happiness	bluesy	soulful	sadness
brightness	smooth	hopeful	darkness
confidence	funky	holy	defeat
victory	folky	moon	tragedy

Table 3.10: Diatonic modes

3.2 Audio features analysis

In this section we will give a review of basic signal processing tools we will need in the next chapter. These tools are needed to extract low level features directly from the audio samples domain. *Short-Time Fourier Transform* (STFT) is a Fourier-related transform that computes a frequency-time representation of the signal, needed to calculate the Onset Detection Function in the beat-tracking system. Chromagram is a pitch-class versus time representation of the signal. It is a prerequisite for chord recognition and is obtained from the STFT.

3.2.1 Short-Time Fourier Transform

Short-Time Fourier Transform (STFT) is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal $s(n)$ as it changes over time (n is the sample index). It is computed by dividing the signal into frames by multiplication by a sliding window $w(n)$

($w(n) \neq 0$ for $0 \leq n \leq L_w - 1$) and Fourier-transforming each frame. Three parameters are specified: transform-size N_{FT}, window length $L_w < N_{FT}$ and hop-size H. The equation is

$$S_r(\omega_k) = \sum_{n=0}^{N_{FT}-1} (s(n - rH)w(n))e^{-j\omega_k n} \qquad k = 0, ..., N_{FT}, \tag{3.3}$$

where $\omega_k = 2\pi k/N_{FT}$ is the frequency specified in [rad/s], r is frame index, k is the frequency bin. Usually $L_w = N_{FT} = 2^\eta$, because the Fourier transform is computed by the fast implementation called *Fast Fourier Transform* (FFT), that has maximized performances for power of 2 transform sizes. For real valued signals the Fourier transform is symmetric, then we consider only the first half of the spectrum ($k = 0, ..., N_{FT}/2 - 1$). Sampling in the time axis is controlled by the hop-size parameter H. The parameter N_{FT} is linked to frequency resolution $\Delta f = f_s/N_{FT}$ where f_s is the sampling frequency of the $s(n)$. Frequency resolution however must account also for the effect of convolution with the Transform of the window function $W(\omega_k)$ due to multiplication in the time-domain. We will need to account this concept when choosing N_{FT} large enough to obtain the resolution needed to distinguish two tones at certain frequency. Sometimes only the magnitude information is needed so a matrix representation called Spectrogram, $P(r, k) = |S_r(\omega_k)|^2$, is used instead.

3.2.2 Chromagram

Chromagram describes how much the pitch-classes are present in a frame of the audio, then is a pitch-class versus time representation. In our chord-tracking system it will be compared with chord templates to find which chord is playing at a given time.

To compute the Chromagram we follow a series of steps (Fig. 3.4). First we extract the STFT from audio, then we compare the magnitude spectrum with a series of pitch profiles. We obtain the pitch salience representation that indicates how much the pitches are present in the audio frame. We then perform noise reduction to retain only significant peaks and discard the noise due, for example, to percussive transients. Successively we perform tuning to compensate the potential offset from the standard $A4 = 440Hz$ reference. Finally bass and treble Chromagram are separated to later exploit the importance of the bass note in the chord-recognition system.

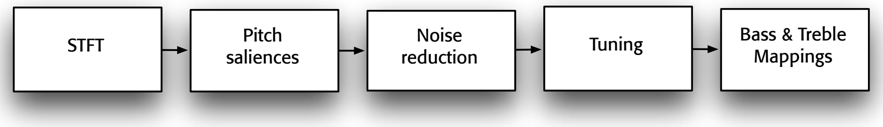


Figure 3.4: Steps of the Chromagram calculation

Frequency-Domain Transform

First of all we must compute the STFT of the signal. Fundamentals of notes from C0 to C7 lie in the range of $\sim [30, 2000]$Hz then we can down-sample the audio signal to $f_s = 11025$ Hz without loss of information. Furthermore we must ensure that we achieve the desired frequency resolution. The lower limit of N_{FT} is given by

$$N_{FT} > K \frac{f_s}{|f_2 - f_1|}, \tag{3.4}$$

where K is a parameter that depends on the window function used, for Hamming window $K = 4$. We want the frequencies of A3 and G♯3 to be discernible, so N_{FT} has to satisfy

$$N_{FT} > 4 \frac{11025}{|220 - 208|} = 3675. \tag{3.5}$$

The minimum (to save computation time) power of 2 that satisfy this requirement is $2^{12} = 4096$. So we compute the STFT of the signal with $N_{FT} = 4096$ and normalize each time slice with respect to L_2 norm (Fig.):

$$A(r, k) = \frac{|S_r(\omega_k)|}{(\sum_q |S_r(\omega_q)|^2)^{1/2}} \tag{3.6}$$

Pitch salience

To construct the time-pitch representation and simultaneously account for harmonics in timbre of instruments we adopt the approach proposed in [18] and modified in [29]. We construct a pitch profile matrix M^c (Fig. 3.6), similar to a constant-Q transform kernel, where each row is the Fourier transform of a Hamming windowed [41] complex tone, containing four harmonics in geometric amplitude sequence.

$$M^c(m, k) = FFT_{N_{FT}}(w(n) \sum_{h=1}^{4} \alpha^h \cos(2 * \pi h f_0(m) n)) \tag{3.7}$$

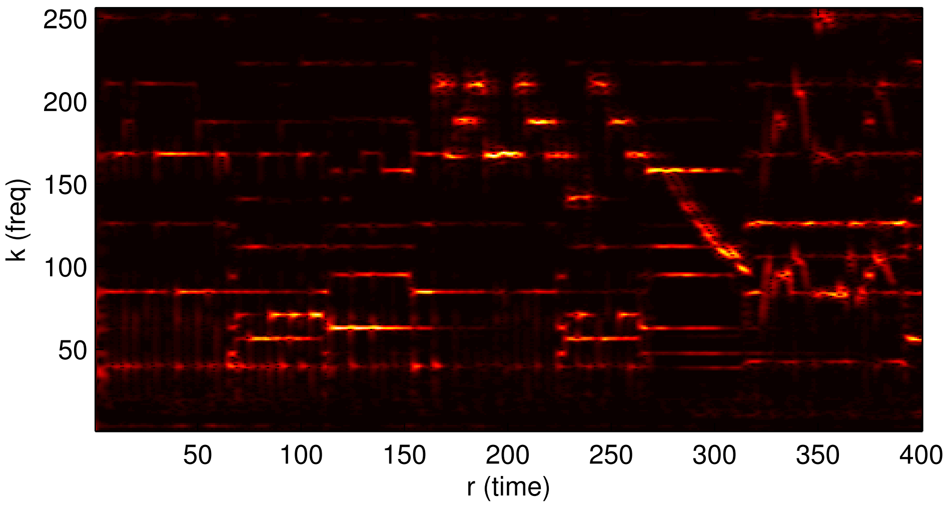


Figure 3.5: Normalized spectrum $A(r,k)$. r is the time index and k is the frequency bin. In figure, for clarity, are showed only the first 256 k of 4096. Only the first half (2048) of frequency bins are meaningful, as the transform of a real signal is symmetric.

where $\alpha = 0.9$, $w(n)$ is the Hamming window and the fundamentals starts from the fundamental frequency of A0 (27.5 Hz) and are

$$f_0(m) = 27.5 \times 2^{\frac{m-2}{36}} \qquad m = 1, ..., 6 \times 36, \tag{3.8}$$

where m is the pitch index. This means that our pitches span 6 octaves, from A0 to G♯6, with the resolution of 1/3 semitone. This fine resolution will allow us to perform tuning at a later stage.

We obtain a pitch salience matrix S^c by multiplying A by M^c. This however leads to a problem because also sub-harmonics (pitches at $f = f_0/n$) have high values. This is addressed by constructing another pitch profile matrix M^s similar to M^c but considering only a simple tone with no harmonics. Pitch salience matrices are obtained by

$$S^c(m,r) = M^c(m,k)A(k,r) \qquad S^s(m,r) = M^s(m,k)A(k,r), \tag{3.9}$$

and passed to the next stage where they will be filtered and combined.

Broadband noise reduction

To lower broadband noise we have to retain only peaks in both salience matrices. We threshold each time-slice and retain only values higher than the local mean plus the local standard deviation. This two statistics are computed considering an interval of half an octave. Thresholded S^c and S^c are then combined by element-wise product.

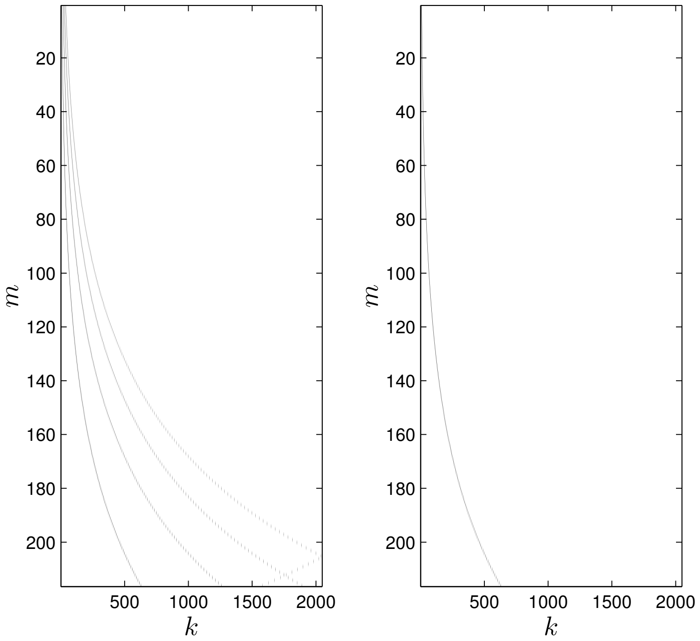


Figure 3.6: Simple and complex pitch profiles matrices. k is the frequency bin index and m is the pitch index.

$$S^{pre}_{m,t} = \begin{cases} S^s_{m,t} S^c_{m,t} & \text{if } S^s_{m,t} > \mu^s_{m,t} + \sigma^s_{m,t} \\ & \text{and } S^c_{m,t} > \mu^c_{m,t} + \sigma^c_{m,t} \\ 0 & 0 \end{cases} \tag{3.10}$$

Tuning

Having recovered the pitch salience matrix with three times the resolution needed, we can compensate for tuning shifts from the standard reference of 440 Hz, by performing circular statistics. To achieve a more robust tuning we exploit the fact that the tuning do not change within a song, so we can average all the temporal slices

$$\bar{S} = \frac{1}{T} \sum t S^{pre}_{m,t}. \tag{3.11}$$

To find the tuning offset find the phase of the complex number obtained by

$$c = \sum_m \bar{S}(m) e^{j \frac{2\pi}{3}(m-1)}, \tag{3.12}$$

and divide it by 2π to obtain the tuning shift in semitones:

$$t = \frac{phase(c)}{2\pi}, \tag{3.13}$$

With this information we can interpolate S^{pre} so that the middle bin of each semitone corresponds to the pitch estimated. Now the extra resolution of 1/36 semitone is not needed any more, so we sum the three bins for each semitone:

$$S_{n,t} = \sum_{m=3n-2}^{3n} S^{pre}_{m,t}. \tag{3.14}$$

Bass and treble Chroma

As said, we need two chromagram representation for the different ranges. Given the importance of the bass note in the harmony, we will exploit the bass chromagram to increase the accuracy of chord detection.

The bass and treble chromagrams (Fig. 3.8) are obtained by multiplying $S_{n,t}$ by two windows functions $w_t(n)$ and $w_b(n)$ (Fig. 3.7), that satisfy 2 constraints:

- they sum to 1 in the interval from A1 to G♯3

$$w_t(n) + w_b(n) = 1 \qquad 13 < n < 48. \tag{3.15}$$

- they give the a constant total weight to all the pitch classes

$$\begin{aligned} &\textstyle\sum_{k=0}^{5}(w_t(12k + pc)) = \rho_t \\ &\textstyle\sum_{k=0}^{5}(w_b(12k + pc)) = \rho_b, \end{aligned} \tag{3.16}$$

The two chromagrams are obtained by:

$$\begin{aligned} C^T_{p,t} &= \textstyle\sum_{k=0}^{5} w_t(12k + p)S_{12k+p,t} \\ C^B_{p,t} &= \textstyle\sum_{k=0}^{5} w_b(12k + p)S_{12k+p,t} \end{aligned} \tag{3.17}$$

A third version of the chromagram that we will use for creating the chord salience matrix, called wide chromagram C^W, is obtained by summing the two.

$$C^W_{p,t} = C^T_{p,t} + C^B_{p,t}. \tag{3.18}$$

3.3 Dynamic programming

Dynamic programming (DP) is a technique for the design of efficient algorithms. It can be applied to optimization problems that generate subproblems of the same form. DP solves problems by combining the solutions

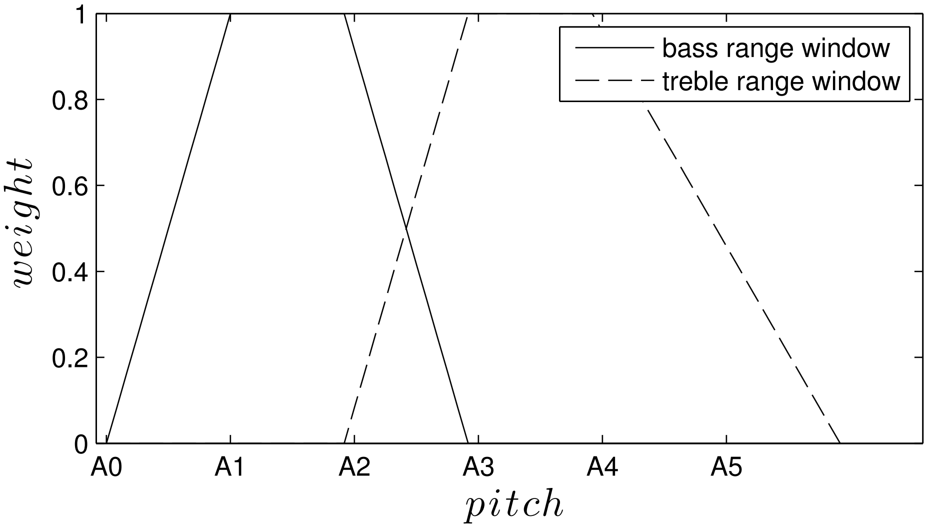


Figure 3.7: Windows for bass and treble range chromagrams

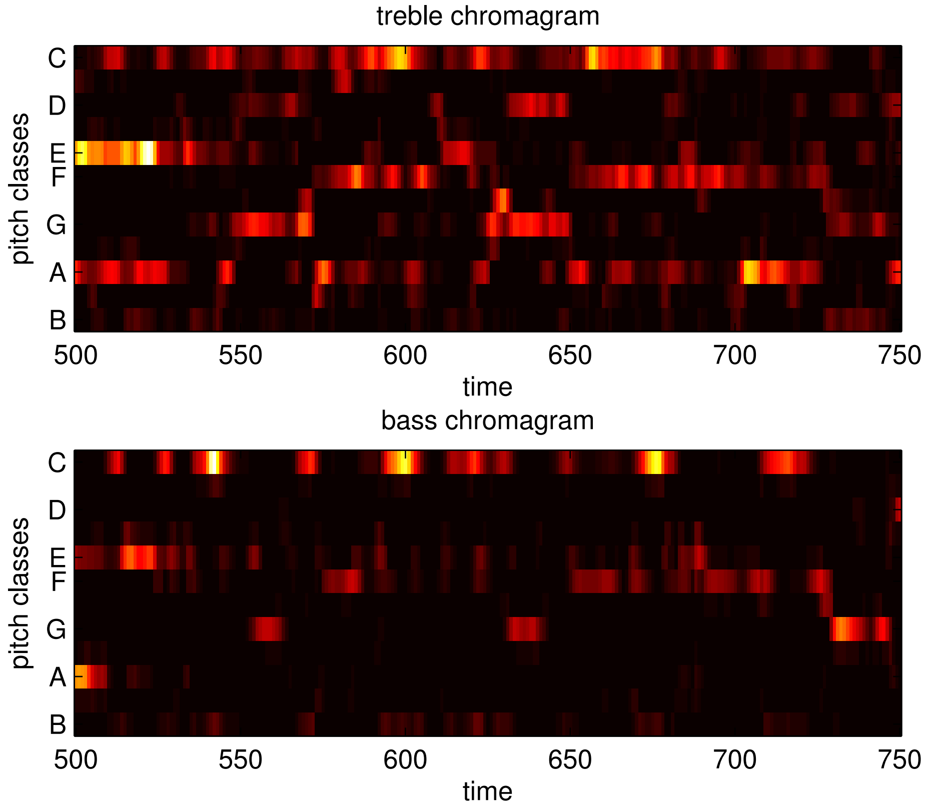


Figure 3.8: Treble and bass chromagrams

to sub-problems. It can be applied when the sub-problems are not independent, i.e. when sub-problems share sub-sub-problems. The key technique is to store the solution to each sub-problem in a table in case it should reappear. The development of a dynamic-programming algorithm can be splitted into a sequence of four steps.

1. Characterize the structure of an optimal solution.

2. Recursively define the *value* of an optimal solution.

3. Compute the *value* of an optimal solution in a bottom-up fashion.

4. Construct an optimal solution from computed information.

A simple example is the assembly-line scheduling problem proposed in [8], which shares many similarities with beat-tracking algorithm in [13]. Let's focus on beat tracking and see how DP can be applied to our problem.

Given a sequence of candidate beat instants $t(i)$ with $i = 1, ..., N$, two specific functions can be formulated: $O(i)$ and $T(i)$. $O(i)$ is an onset detection function and says how much a beat candidate is a good choice based on local acoustic properties. $T(i)$ is the tempo estimation that describe the ideal time interval between successive beat instants. We search a optimal beat sequence $t(p(m))$ with $m = 1, ..., M$, such that onset strengths and correspondence to the tempo estimation is maximized.

As the first step let's characterize the structure of the optimal beat sequence that ends with the beat candidate $t(i_{end})$. To obtain it, we must evaluate all the J sequences that end with $t_{i_{end}}$, that we represent as $t(p_j(m))$ with $p_j(M) = i_{end}$, and choose the best one $t(p_{best}(m))$. $t(p_{best}(m))$ will surely contain the best sequence up to $t(p_j(M-1))$. The key is to realize that the optimal solution to a problem contains optimal solutions to subproblems of the same kind.

In the second step we have to recursively define the *value* of an optimal solution. The optimal beat sequence solution $t(p_{best}(m))$ will have to both maximize $\sum O(p_{best}(m))$ and the probabilities of all the transitions. Let's define a single objective function C that combines both of these goals. C evaluates a sequence $p(m)$ and returns a score:

$$C(p) = \sum_{m=1}^{M} O(p(m)) + \sum_{m=2}^{M} F(t(p(m)) - t(p(m-1)), T(p(m))) \qquad (3.19)$$

where F is a score function that assign a score to the time interval Δt between two beats, given an estimation T of the beat period. F is given by this equation:

$$F(\Delta t, T) = -(log\frac{\Delta t}{T})^2. \tag{3.20}$$

The key property of this objective function is that the score of the best beat sequence up to the beat i can be assembled recursively. This recursive formulation C^* is given by:

$$C^*(i) = O(i) + \max_{prev=1,...,i-1}(F(t(i) - t(prev), T(i)) + C^*(prev)) \tag{3.21}$$

The third step is another key point in dynamic algorithms. If we base a recursive algorithm on equation 3.21 its running time will be exponential in N, the number of beats in the sequence. By computing and storing $C^*(p(m))$ in order of increasing beat times, we're able to compute the value of the optimal solution in $\Omega(N)$ time.

Fourth and last step regards the actual solution. For this purpose, while calculating C^*, we also record the ideal preceding beat $P^*(i)$:

$$P^*(i) = \arg\max_{prev=1,...,i-1}(F(t(i) - t(prev), T(i)) + C^*(prev)) \tag{3.22}$$

Once the procedure is complete, P^* allows us to retrieve the ideal preceding beat $P^*(i)$ for each beat i. We can now *backtrace* from the final beat time to the beginning of the signal to find the optimal beat sequence.

3.4 Hidden Markov Models

In the MIR field is typical to characterize signals as statistical models. They are particularly useful for the recognition of sequence of patterns. One example are the *hidden Markov Models*(HMMs) [37]. Within beat tracking task, HMM can be used, for example, to find the tempo-path, given the Rhythmogram.

Markov models describe a system that may be in one of N distinct states, $S_1, S_2, ..., S_N$. After a regular specific quantum of time it changes the state, according to a set of probabilities associated with the current state. Let's denote the actual state at time t as q_t. The probability of being in the state $q_t = S_j$ given the previous state $q_{t-1} = S_i$ is given by

$$p(q_t = S_j | q_{t-1} = S_i). \tag{3.23}$$

In eq. 3.23, p is independent by the time, then we can gather those probabilities in a state-transition matrix with elements:

$$a_{ij} = p(q_t = S_j | q_{t-1} = S_i), \qquad 1 \leq i, j \leq N, \tag{3.24}$$

where

$$a_{ij} \geq 0, \qquad \sum_{j=0}^{N} a_{ij} = 1, \forall i. \tag{3.25}$$

Markov models where each state S_j corresponds to an observable event v_j are called *observable Markov models* or simply Markov Models. The three states model of the weather proposed in [37] is an example. The only parameters required to specify the model are the state-transition matrix $A = \{a_{ij}\}$ and the initial state probabilities $\pi_i = p(q_1 = S_i)$.

In contrast to Markov models, in *hidden Markov Models*, observations symbols v_k do not correspond to a state, but depend to the state, following a series of *conditional probability distributions* CPDs b_{kj}:

$$b_{kj} = p(O_t = v_k | q_t = S_j), \qquad 1 \leq j \leq N, 1 \leq k \leq M, \tag{3.26}$$

where O_t is the symbol observed at time t. An easy example is the tossing of N coins, differently biased, where each coin is a state S_j with $j = 1, ..., N$ and outcomes are the observations v_k with $k = 0, 1$. HMM are fully specified by the state-transition matrix A, the initial state probabilities π and the observation symbol CPDs $B = \{b_{ij}\}$. The complete parameter set is then indicated by $\lambda = (A, B, \pi)$.

There are two main graphic representations of HMM. The first, called state-transition diagram, is a graph where node represent states and arrows are allowable transitions between them (Fig. 3.9). The second is a directed graphical model that shows variables in a sequence of temporal slices, highlighting time dependencies (Fig. 3.10).

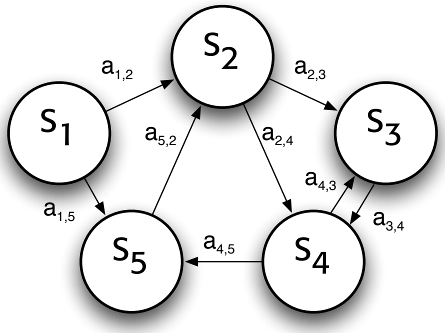


Figure 3.9: State transition graph representation of hidden Markov Models. Nodes are states and allowed transitions are represented as arrows.

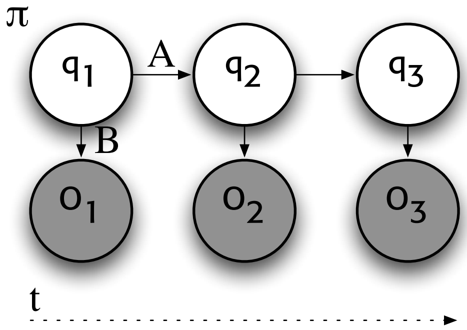


Figure 3.10: Hidden Markov Models are often represented by a sequence of temporal slices to highlight how q_t, the state variable at time t, depends only on the previous one q_{t-1} and the observation symbol at time t depends only on the current state q_t. The standard convention uses white nodes for hidden variables or states and shaded nodes for observed variables. Arrows between nodes means dependence.

3.5 Viterbi decoding algorithm

The Viterbi algorithm is a formal technique, based on the dynamic programming method that finds the single best state sequence $Q = \{q_1, q_2, ..., q_T\}$ that maximize $p(Q|O, \lambda)$, where O is the sequence of observations $O = \{O_1, O_2, ..., O_T\}$. First we define in a recursive fashion the value or *score* of a solution: we define the best score that ends to the state S_i at time t as

$$\delta_t(i) = \max_{q_1,q_2,\cdots,q_{t-1}} p(q_1, q_2, ..., q_t = S_i|O_1, O_2, ..., O_t, \lambda) \tag{3.27}$$

And find the recursive relationship

$$\delta_{t+1}(j) = \max_i [\delta_t(i)a_{ij}]b_j(O_{t+1}) \tag{3.28}$$

Then we compute this value from the start of the sequence, keeping track along the way of both sub-solution score $\delta_t(i)$ and best previous state $\psi_t(i)$ (needed for backtracking). These are computed for every time slice and for every state. The full procedure follows these four steps. Initialization

$$\delta_1(i) = \pi_i b_i(O_1) \qquad 1 \leq i \leq N \tag{3.29}$$

$$\psi_1(i) = 0 \tag{3.30}$$

Recursion

$$\delta_t(j) = \max_i [\delta_{t-1}(i)a_{ij}]b_j(O_t) \qquad 1 \leq j \leq N, 2 \leq t \leq T \tag{3.31}$$

$$\psi_t(j) = \arg\max_i[\delta_{t-1}(i)a_{ij}] \qquad 1 \leq j \leq N, 2 \leq t \leq T \tag{3.32}$$

Termination

$$score = \max_i[\delta_T(i)] \tag{3.33}$$

$$q_T^* = \arg\max_i[\delta_T(i)] \tag{3.34}$$

Backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}) \qquad t = T-1, T-2, ..., 1 \tag{3.35}$$

3.6 Dynamic Bayesian networks

Dynamic Bayesian Networks (DBN) are a generalization of HMMs that allows the state space to be represented in factored form, instead of a single discrete random variable.

A *Directed Acyclic Graph* (DAG Fig. 3.11) is a set of nodes and edges $G = (Nodes, Edges)$, where Nodes are vertices and Edges are connections between them. Edges are directed if they imply a non-symmetric relationship, in this case the parent→son relationship. A graph is *directed* if all its edges are directed. *Acyclic* means that it is impossible to follow a path from $Node_i$ that arrives back at $Node_i$, as to say that $Node_i$ is an ancestor of itself.

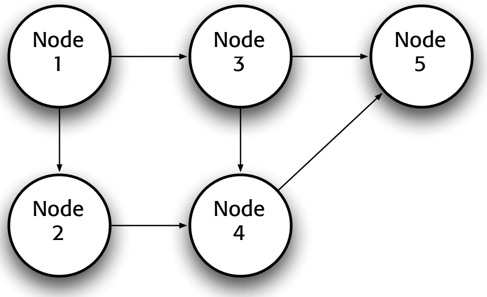


Figure 3.11: A directed acyclic graph

A Bayesian network (BN) is a directed acyclic graph whose nodes represent a set of random variables $\{X_{1:N}\}$, where N is the number of nodes, and whose edge structure encodes a set of conditional independence assumptions about the distribution $P(X_{1:N})$:

$$(X_i \perp \text{NonDescend}(X_i) \,|\, \text{Parents}(X_i)). \tag{3.36}$$

Under these assumptions, and if the set $\{X_{1:N}\}$ is topologically ordered with parents preceding their children, $P(X_{1:N})$ can be factored as the product of local probabilistic models $P(X_i|Parents(X_i))$:

$$P(X_1, ..., X_N) = \prod_{i=1}^{N} P(X_i|Parents(X_i)). \tag{3.37}$$

For example, the DAG in Fig. 3.11 is already ordered, then its joint probability is:

$$P(X_1, ..., X_N) = P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_3, X_2)P(X_5|X_3, X_4). \tag{3.38}$$

This ability to divide a complex system into smaller ones is what renders it a great modelling tool. For each variable in a system we simply add a node and connect with other nodes, where there are direct dependencies. Let's now introduce the time axis and temporal sequences.

First order Markov Models perform a prediction of type $P(N_i|N_{i-1})$ that can be seen as a special case of a general inference query $P(Attribute_i|Context_i)$. In the chord-recognition task, attribute and context variables can range from chord, key, meter, and any other variables of interest. DBNs extend the capability of first-order Markov Models increasing the number of musical relationships and hierarchies that can be tracked. This renders it particularly suited to model musical systems that include lots of interconnected variables.

A DBN is defined to be a pair, $(B_1, B_{\rightarrow})$, where B_1 is a BN which defines the prior $P(Z_1)$, and $B_{\rightarrow}$ is a two-slice temporal Bayes net (2TBN) which defines $P(Z_i|Z_{i-1})$ by means of a DAG that factors as follows:

$$P(Z_i|Z_{i-1}) = \prod_{i=1}^{K} P(Z_i^j|Parents(Z_i^j)). \tag{3.39}$$

where Z_i^j indicates the Z^j variable at time index i. The resulting joint distribution is given by:

$$P(Z_{1:K}) = \prod_{i=1}^{K} \prod_{j=1}^{J} P(Z_i^j|Parents(Z_i^j)). \tag{3.40}$$

Building a DBN that include unobserved hidden states is straightforward. We only need to add the nodes and create the dependencies. Model specification is accomplished by specifying probability distributions in B_1 and $B_{\rightarrow}$, as in HMMs.

There are many standard inference types [32]. The one we will use is called *offline smoothing* because our system will have access to all the observations.

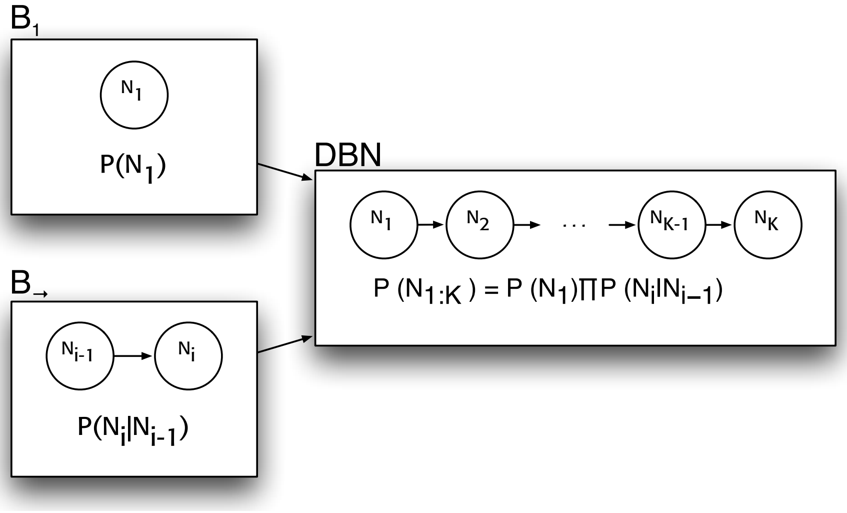


Figure 3.12: Construction of a simple Dynamic Bayesian network, starting from the prior slice and tiling the 2TBN over time.

It is based on a 2 pass forward filtering-backward smoothing strategy, which is not discussed here because it is beyond the scope of this thesis.

Chapter 4

System Overview

In this chapter we will review the main components of our system. The first section is devoted to the beat-tracker subsystem. In the second section we will give an overview of the chord-recognition subsystem. Finally we will explain the three novel harmony-based features and the ideas behind them.

4.1 Beat tracking system

In this section we review in full detail every building block of the beat tracking system (Fig. 4.2).

Starting from the audio signal $s(n)$, we first compute the STFT and extract the onset detection function $\eta(m)$. Periodicities in $\eta(m)$ are estimated and stored in $\tau(m)$ by the tempo estimation block. Starting from $\eta(m)$ and $\tau(m)$ the beat tracking stage extract the beat times $b(i)$. These are exploited by the downbeat extraction stages, which extract downbeats $db(k)$ among beats.

4.1.1 Onset strength envelope

The first step in computing the onset strength envelope is taking the short-Time Fourier Transform (STFT) of the signal. Starting from a stereo signal we first average the two channels to achieve a mono signal $s(n)$, then the STFT is computed with window length of $N = 1024$ samples and overlap of 50% ($h = 512$) obtaining a time-resolution of 11.6 ms:

$$S_k(m) = \sum_{n=-\infty}^{\infty} s(n)w(mh-n)e^{-j2\pi k/N} \tag{4.1}$$

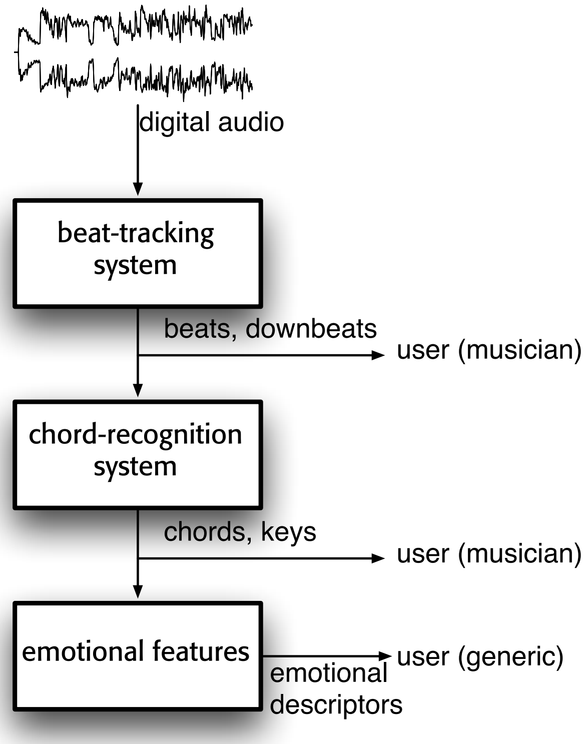


Figure 4.1: The blocks that constitutes our system. Three types of information are extracted starting from digital audio. Rhythmic information is extracted by our beat tracker. Harmony-related information is extracted by our chord-recognition subsystem. These information are middle level features and are addressed to musically trained users. Harmony-related features are then extracted starting from key and chord information, and is addressed to a generic user (no training needed).

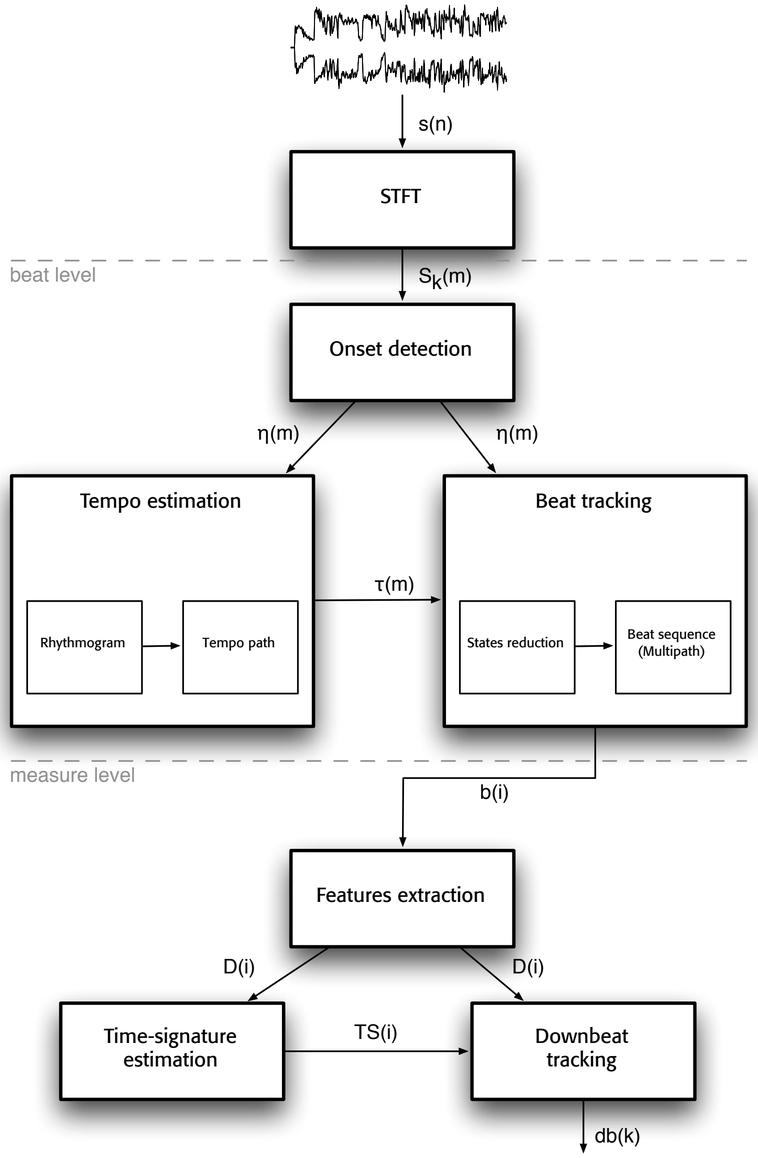


Figure 4.2: Beat tracking system

A basic energy-based onset detection function can be constructed by differentiating consecutive magnitude frames:

$$\delta S_k(m) = \sum_{k=1}^{N} |S_k(m)| - |S_k(m-1)| \tag{4.2}$$

This exploits the fact that energy bursts are often related to transients. Neglecting the phase information, however, results in great performance losses when dealing with non-percussive signals, where soft "tonal" onsets are related to abrupt phase shifts in a frequency component. To track phase related onsets, we consider the phase shift difference between successive frames. This consists in taking the second derivative of unwrapped phase $\psi_k(m)$ and then wrapping again into the $[-\pi, \pi]$ interval:

$$d_{\psi,k}(m) = \mathrm{wrap}[(\psi_k(m) - \psi_k(m-1)) - (\psi_k(m-1) - \psi_k(m-2))]. \tag{4.3}$$

As said, the onset detection computed through a combination of energy-based and phase-based approaches performs better than using only one of them. Simultaneous analysis of both energy and phase-based approach is obtained through a computation in the complex domain. The combined equation is:

$$\begin{aligned}\Gamma_k(m) = \{&|S_k(m-1)|^2 + |S_k(m)|^2 - \\ &- 2|S_k(m-1)||S_k(m-1)| \cos d_{\psi,k}(m)\}^{1/2}\end{aligned} \tag{4.4}$$

And the detection function $\eta(m)$ (Fig. 4.3) is obtained by summing over the frequency bins.

$$\eta(m) = \sum_{k=1}^{K} \Gamma_k(m) \tag{4.5}$$

$\eta(m)$ is equal to the energy based onset function when $d_{\psi,k}(m) = 0$.

$$\begin{aligned}\eta(m) = \sum_{k=1}^{K} \{|S_k(m-1)|^2 + |S_k(m)|^2 - 2|S_k(m-1)||S_k(m-1)|\}^{1/2} = \\ = \sum_{k=1}^{K} |S_k(m)| - |S_k(m-1)|\end{aligned} \tag{4.6}$$

Before proceeding to next stage we remove leading and trailing zeros from $\eta(m)$.

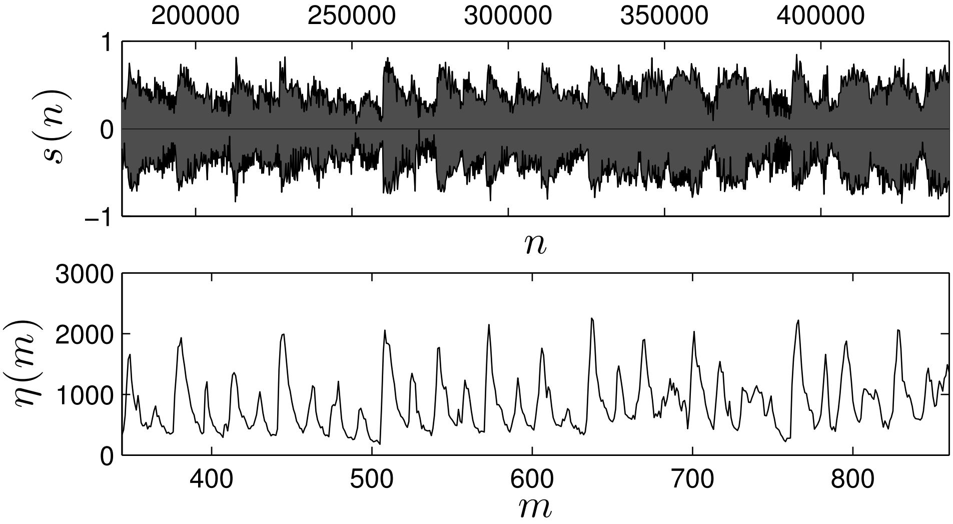


Figure 4.3: Signal $s(n)$ and detection function $\eta(m)$. The two functions have different temporal resolutions but corresponding indices n and m are aligned in the figure for better understanding.

4.1.2 Tempo estimation

The tempo estimation stage takes the onset detection function as input $\eta(m)$ and is divided into two successive stages: computation of the Rhythmogram $y_i(\tau)$, extraction of the tempo path $\tau(i)$ and linear interpolation to obtain the same time resolution of $\eta(m)$ (Fig. 4.4).

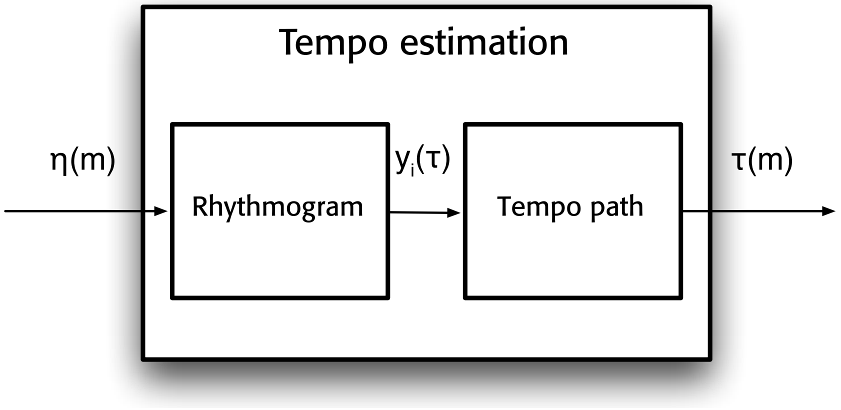


Figure 4.4: Tempo estimation stage

Rhythmogram

For beat period detection we first divide $\eta(m)$ into analysis frames $\eta_i(m)$ of $B_f = 512$ samples with $B_h = 128$ samples hop-size, corresponding to one 6 s frame every 1.5 s:

$$\eta_i(k) = \eta(k - (i-1)B_h) \qquad 1 \leq k \leq B_f \tag{4.7}$$

Each frame is pre-processed, to discard the least significant peaks. Preprocessing consists in moving average threshold and successive half-wave rectify:

$$\begin{aligned} \bar{\eta}_i(m) &= \mathrm{mean}(\eta_i(q)) m - Q <= q <= m + Q \\ \tilde{\eta}_i(m) &= \mathrm{HWR}(\eta_i(m) - \bar{\eta}_i(m)), \end{aligned} \tag{4.8}$$

where we choose the half length of the moving average filter $Q = 7$. Then we compute the autocorrelation, with lag bias compensation.

$$A_i(l) = \frac{\sum_{m=1}^{B_f} \tilde{\eta}_i(m)\tilde{\eta}_i(m-l)}{|l - B_f|} \qquad l = 1, ..., B_f \tag{4.9}$$

To compute the rhythmogram we multiply $A_i(l)$ with a shift invariant comb filter-bank, represented as the matrix $F(l, \tau)$

$$y_i(\tau) = w(\tau) \sum_{l=1}^{B_f} A_i(l) F(l, \tau), \tag{4.10}$$

where every column of the matrix F (Fig. 4.5) is a comb filter where impulses are spaced of period $\tau = 1, ..., 128$.

$$F(l, \tau) = \sum_{p=1}^{4} \sum_{v=1-p}^{p-1} \frac{\delta(l - \tau p + v)}{2p - 1} \qquad l = 1, ..., B_f \tag{4.11}$$

$w(\tau)$ is an a-priori probability distribution on the periods, to attenuate the probabilities of slowest and fastest tempi. The curve that best fit the sample is the Rayleigh distribution function (Fig. 4.6).

$$w(\tau) = \frac{\tau}{\beta^2} e^{\frac{\tau}{2\beta^2}} \qquad \tau = 1, ..., 128 \tag{4.12}$$

where we assign $\beta = 43$, which is the parameter that indicates the peak of the distribution.

The outputs $y_i(\tau)$ (Fig. 4.7) are the probabilities that beats in the $i - th$ analysis frame have period equal to τ.

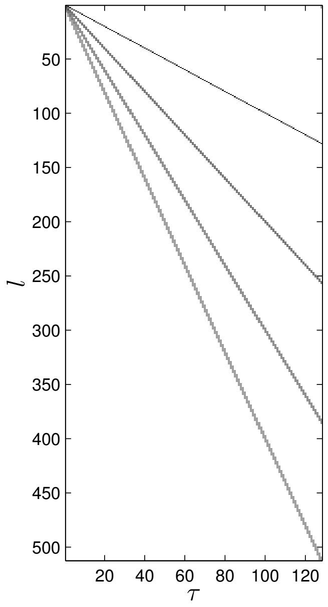


Figure 4.5: Comb filter matrix F

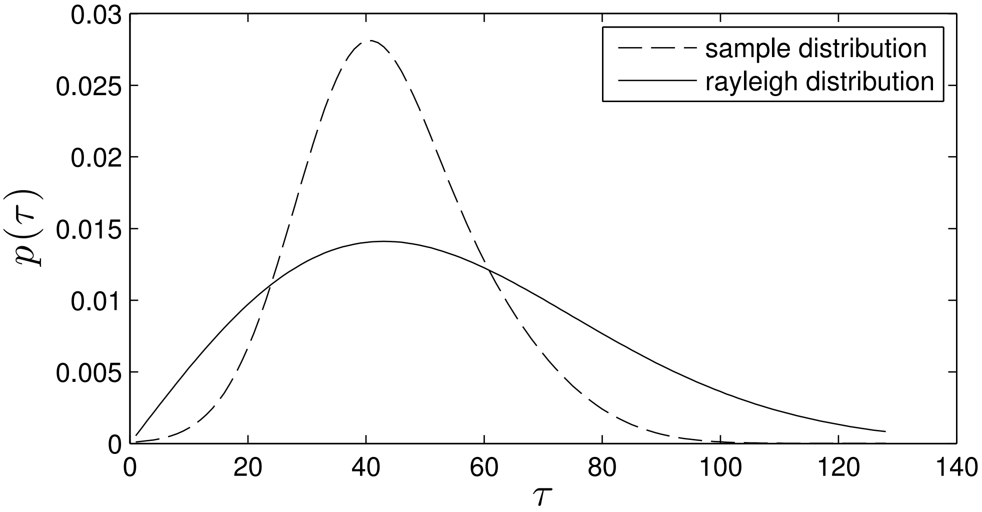


Figure 4.6: Rayleigh distribution function with $\beta = 43$ compared to the sample distribution of periods in our dataset

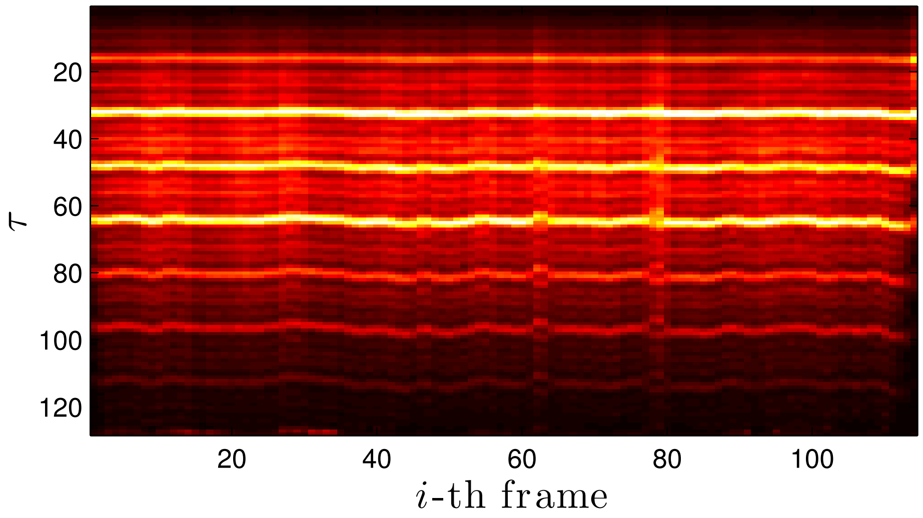


Figure 4.7: Rhythmogram matrix $y_i(\tau)$. τ is the beat period expressed in number of samples of the ODF.

Tempo path

To find the best path of periods throughout the song, we first discard periods under 20 samples of η (20 × 11.6 ms). This is reasonable as that period correspond to an extremely fast tempo, about 258 bpm, that can be safely used as a higher bound. Then we extract $N_{peaks} = 20$ peaks (Fig. 4.10) $\{t_{i,k}\}$ from each i-th frame of $y_i(\tau)$ and we use the Viterbi algorithm to track the best path among them. We assign an observation probability

$$p(\tau_i = t_{i,k}) = \frac{y_i(t_{i,k})}{\sum_{k=1}^{N_{peaks}} y_i(t_{i,k})} \tag{4.13}$$

and a transition probability

$$p(\tau_i|\tau_{i-1}) = d(\log(tau_i) - \log(\tau_{i-1})) \tag{4.14}$$

where $d(x)$ is the period-transition distribution (Fig. 4.8), previously extracted from the dataset.

The Viterbi algorithm finds the best period sequence (Fig. 4.9) that maximizes the probability $p(\{\tau_i\}|y_i(\tau), \lambda)$ where λ is our model.

In the successive stages we will need a beat-period value for every sample of η, so we linearly interpolate the sequence to obtain $\tau(m)$.

4.1.3 States reduction

The goal of the states reduction stage is to reduce the number of beat candidates, keeping only the most probable ones. This will reduce the memory

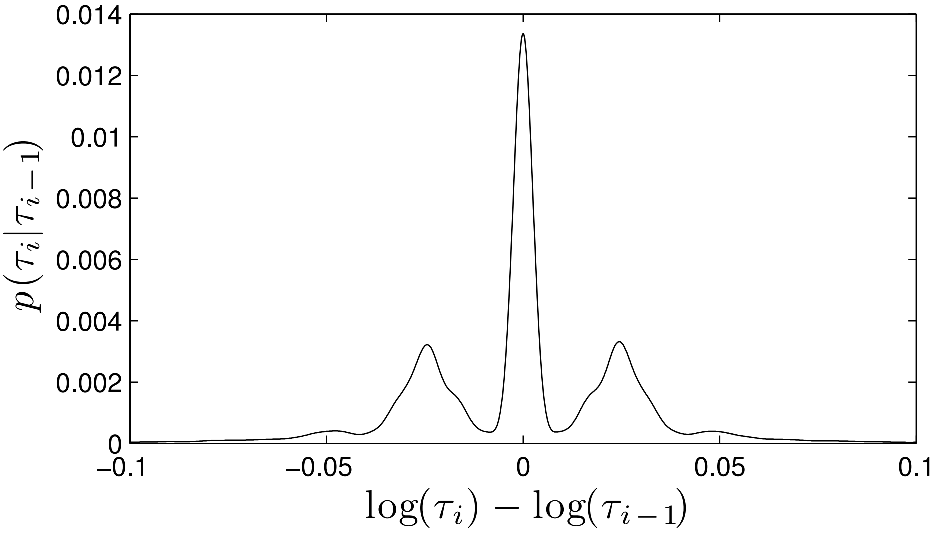


Figure 4.8: Period transition distribution extracted from the dataset

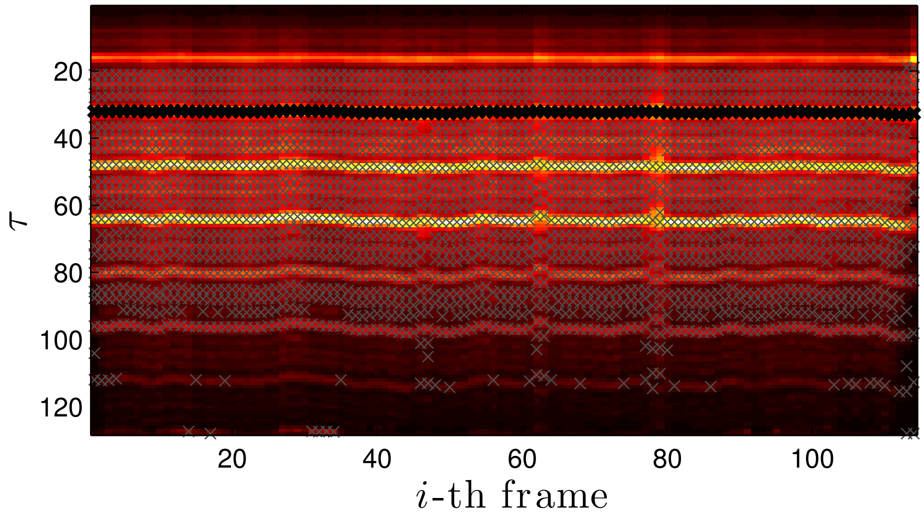


Figure 4.9: Rhythmogram. Chosen peaks and final path are showed respectively as light and dark x.τ is the beat period expressed in number of samples of the ODF.

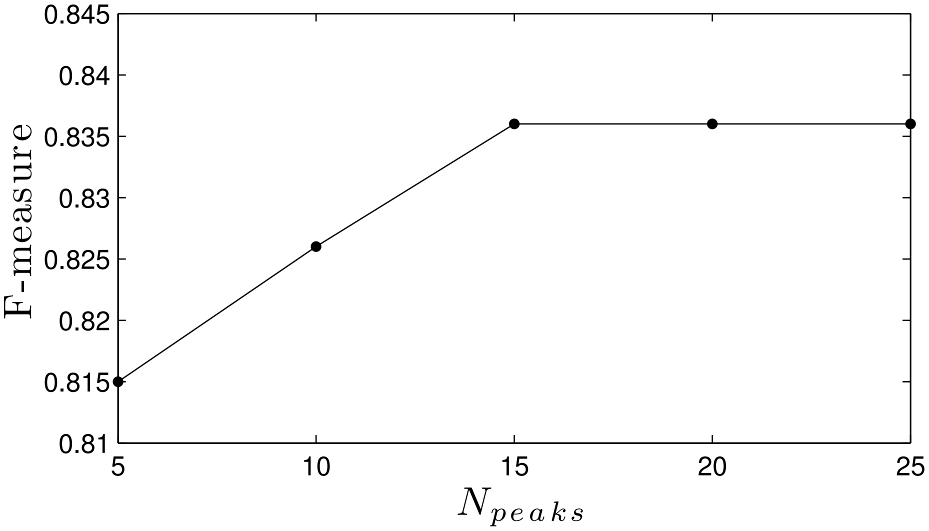


Figure 4.10: The performance of the beat tracking algorithm is influenced by N_{peaks}

requirement and the time of computation of the beat tracking stage.

In order to achieve this result, we pre-analyze the detection function η and pick $N_c = 20$ beat candidates (Fig. 4.11) per beat-period. The number $N_c = 20$ is chosen experimentally, as it maximizes performances over the dataset. We scan the full sequence $\eta(m)$ iteratively, starting from $m = j = 1$

$$c_j = \arg\max(\eta(i)) \qquad m \leq i < m + \frac{\tau(m)}{N_c} \tag{4.15}$$

$$m = m + \frac{\tau(m)}{N_c} \tag{4.16}$$

$$j = j + 1 \tag{4.17}$$

4.1.4 Beat tracking

The beat tracking stage (Fig. 4.12) exploits the onset detection function $\eta(m)$ and the estimated periodicities $\tau(m)$ to extract the beat instants $b(i)$. This stage considers as potential beat instants only the candidates c_j.

The basic sequence tracking algorithm is based on a forward looking search, which returns a beat sequence. To better explore the solution space we keep track concurrently of more possible sequences and, at the end, chose the best according to a score function. A welcomed side-effect of this strategy is that the number of paths passing by a beat candidate c_j is proportional to the reliability of c_j being a beat. We exploit this fact proposing an iterative

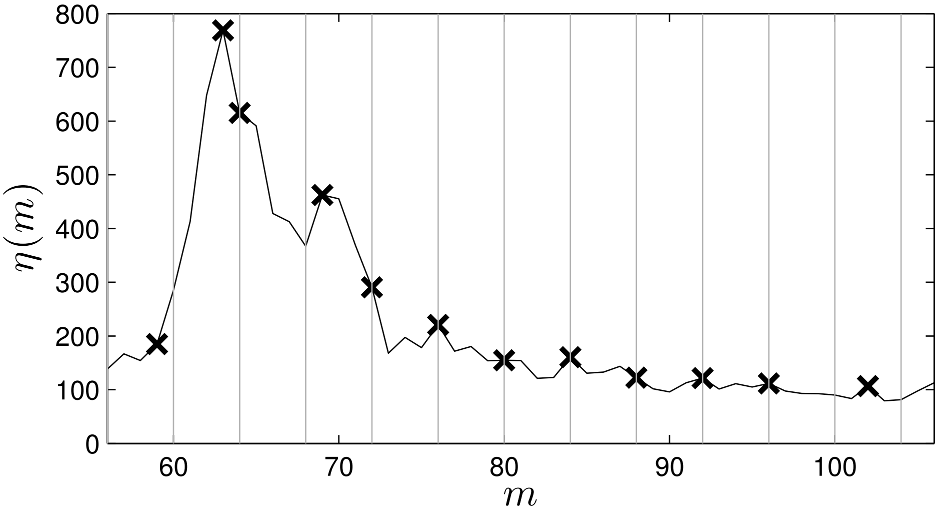


Figure 4.11: Beat candidates c_j are the highest peaks within each region, marked with a cross. $\eta(m)$ is the onset detection function and m is its time-intex

morphing of $\tau(m)$, in the less reliably tracked segments of the song, aimed at correcting mistakes in the estimation of $\tau(m)$.

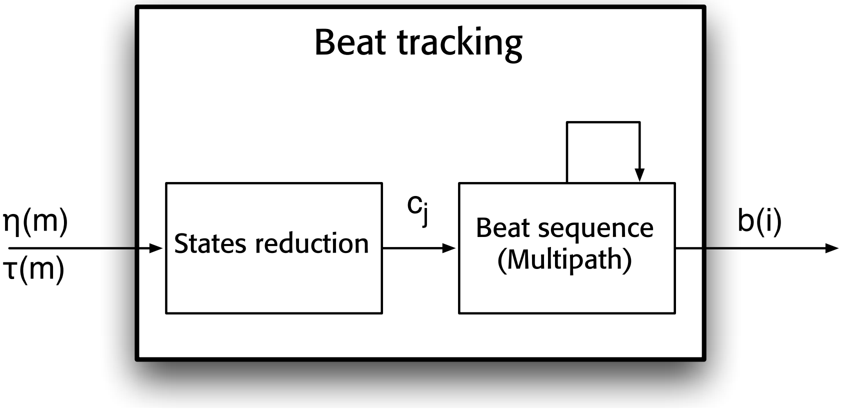


Figure 4.12: Beat tracking stage

Sequence tracking

Beat sequence tracking takes as input the onset detection function $\eta(m)$ and the beat-period sequence $\tau(m)$. If any mistake has been produced at earlier stages, it will propagates to the final beat-sequence. For this reason, we propose a path-finding algorithm aimed at correcting possible mistakes in the tempo-tracking stage. The algorithm starts from the assumption that the best successor b_{i+1} of a beat b_i is not too far away. We search it among the beat-candidates $\{c_j\}$ within an interval $I_{i,Q}$ such that

$$I_{i,Q} = [(b_i + \tau(b_i)) - Q, (b_i + \tau(b_i)) + Q] \tag{4.18}$$

And we select the next beat as the one that maximizes the score function

$$b_{i+1} = \arg\max(score_i(\{c_j\} \in I_{i,Q})) \tag{4.19}$$

$$score_i(b) = p_{I_i}(b) + T(b - b_i, \tau(b_i)) \tag{4.20}$$

$$p_I(b) = \frac{\eta(b)}{\sum_{b \in I} \eta(b)} \tag{4.21}$$

$$T(d, \tau) = e^{-\frac{(d-\tau)^2}{2\sigma^2}} \tag{4.22}$$

where we chose the variance of the Gaussian distance $\sigma = \tau/8$ because it maximizes the final performance. The iteration of this method is not suited to find the best total beat-sequence, because the search of the best successor depends only on the current beat, and this leads to a sub-optimal solution. However, this forward-looking strategy has the advantage that the sequence doesn't need to be backtracked and only requires a 1-dimensional data structure $B(i)$ to store the chosen beat-instants.

Multipath sequence tracking

This light and fast technique allows to implement an iteration strategy that keeps track of a number of paths N_p at the same time. This way we better explore the solutions space (Fig. 4.14). The number of N_p is chosen to maximize the overall performance (Fig. 4.15). We need to store the beat-instants $\{b_i\}$ in a N_p-dimensional data structure $B_p(i)$, with $p = 1, ..., N_p$. We also need to generate and update a score function $score_p$ for each path, to choose the best at the end. As the iteration progresses, several joining and splitting of paths are performed, following a predefined set of rules:

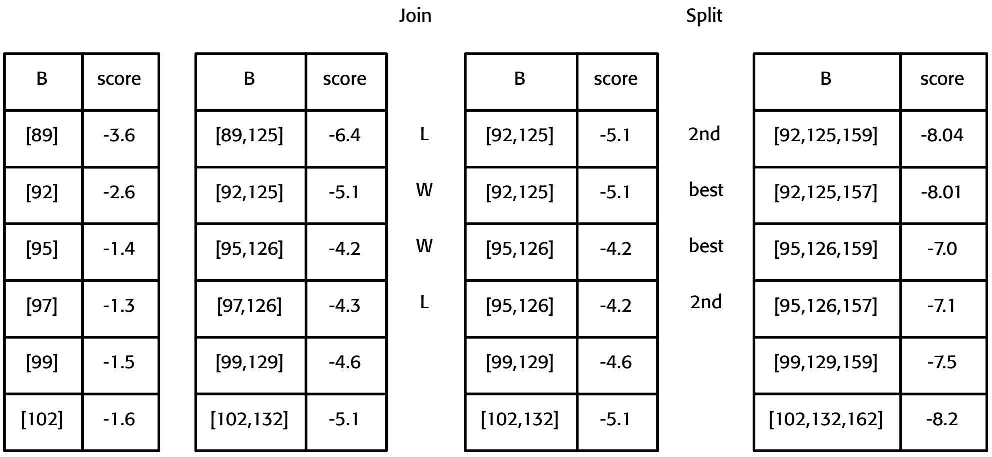
- **Initialization rule**: every path is initialized with near beat-candidates, e.g. $B_1(1) = c_1$, $B_2(1) = c_2$ and so on.
- **Joining rule**: whenever two paths fall on the same beat-candidate, the one with lower score *loses* and becomes a copy of the *winner*: if $B_w(i)$ is equal to $B_l(i)$ and $score_w \geq score_l$, then we assign $B_l = B_w$ and $score_l = score_w$ (Fig. 4.13).
- **Splitting rule**: after a joining, to avoid that the two paths continue following the same beat sequence, we perturb one. The *loser* path will

continue not with the best, but with the second-best beat successor (Fig. 4.13).

- **Termination rule**: the iteration is terminated when all the paths reach the end of the beat-candidates sequence $\{c_j\}$

The score function for each path is the cumulative and log version of the 4.20.

$$score_p = score_p + \log p_{I_i}(b_{i+1}) + \log T(b_{i+1} - b_i, \tau(b_i)) \tag{4.23}$$



B	score	B	score	Join	B	score	Split	B	score
[89]	-3.6	[89,125]	-6.4	L	[92,125]	-5.1	2nd	[92,125,159]	-8.04
[92]	-2.6	[92,125]	-5.1	W	[92,125]	-5.1	best	[92,125,157]	-8.01
[95]	-1.4	[95,126]	-4.2	W	[95,126]	-4.2	best	[95,126,159]	-7.0
[97]	-1.3	[97,126]	-4.3	L	[95,126]	-4.2	2nd	[95,126,157]	-7.1
[99]	-1.5	[99,129]	-4.6		[99,129]	-4.6		[99,129,159]	-7.5
[102]	-1.6	[102,132]	-5.1		[102,132]	-5.1		[102,132,162]	-8.2

Figure 4.13: Join and split rules applied to 6 path at the first 3 iterations

Iterative tempo path morphing

When we study the behaviour of the paths, we notice that the number of paths that converge on the same beat-candidate is correlated to the reliance in the expectation of that beat. To confirm this we stored the number of paths converged on, let's say *vote*, each beat during the multipath-tracking stage. Than we applied a threshold t_v on the final path, keeping only the beats with $\geq t_v$ votes ($\geq t_v$ converged on that beat). The relationship of the precision and recall parameters with this threshold t_v demonstrate our suspect. Precision increases as only "higher quality" beats are selected, and recall decreases as we create holes in the beat sequence (Fig. 4.16). This is a very pleasant side-effect because it allows the algorithm to give a measure of confidence for each beat of the final beat-sequence. We exploited this

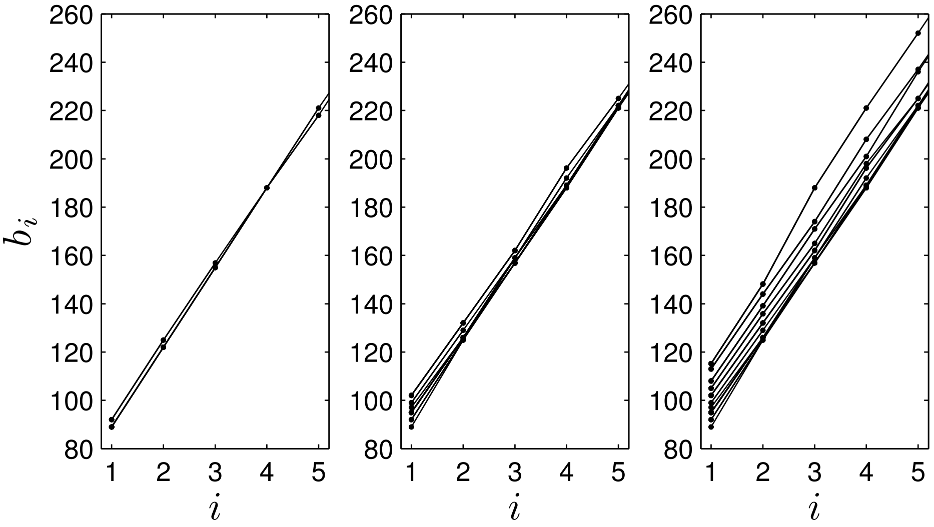


Figure 4.14: Increasing the number of paths ($N_p = 2, 6, 10$ in the figures) influences the coverage of the space

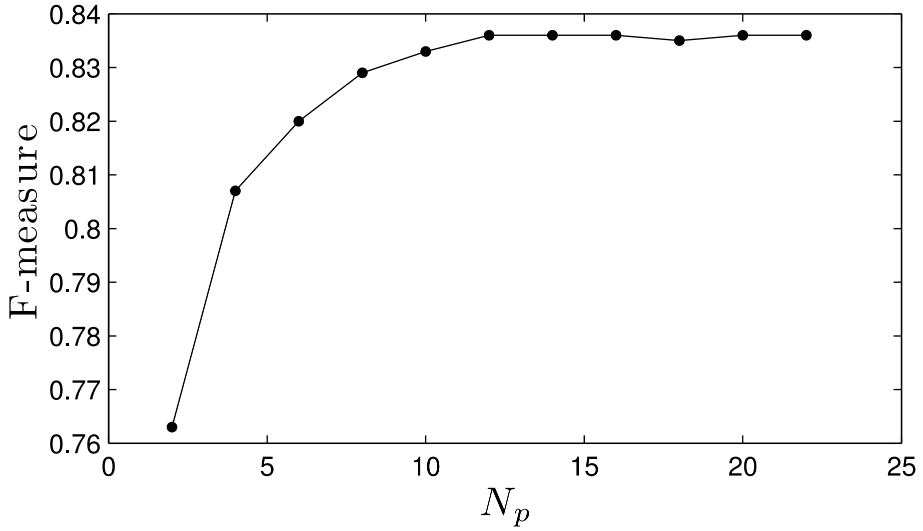


Figure 4.15: The performance of the beat tracking algorithm is influenced by N_p

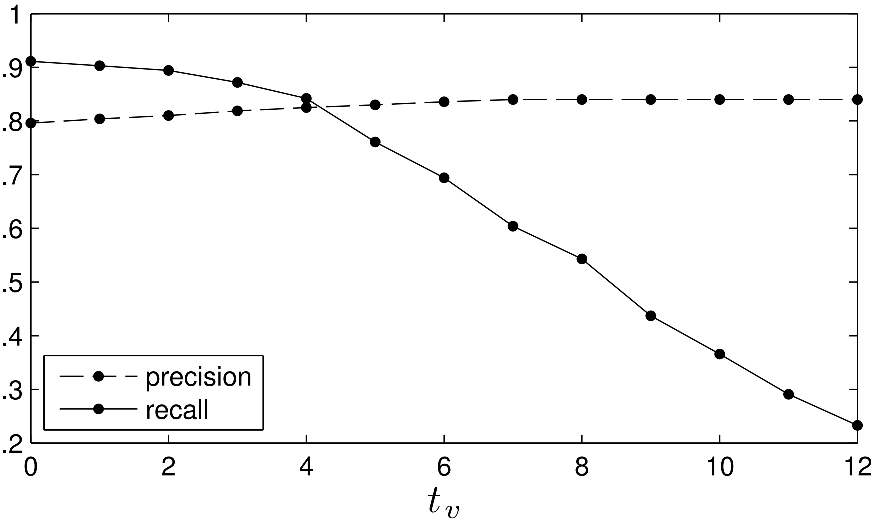


Figure 4.16: Precision and recall parameters plotted against the threshold t_v

effect by iteratively running the path finding algorithm. At each iteration we threshold the beat sequence using votes and to morph the tempo path in the segments with least voted beats (Fig. 4.17). As expected we note a performance increase when increasing the number of iterations (Fig. 4.18).

4.1.5 Downbeat tracking

Downbeat tracking stage shares many similarities with beat tracking stage (Fig. 4.2). We now analyze the song at measure level and try to find the downbeats (the first beats of each measure) among the beats $b(i)$. This is accomplished reusing the same strategy that allowed us to find beat instants. First we compute a detection function $D(i)$, then we search for periodicities in $D(i)$, called time signatures $TS(i)$. Finally we track the downbeat sequence $db(k)$, with the same multipath algorithm used before to find the beat sequence.

Features

For downbeat tracking we use a combination of two features. The first one is a *spectral difference function* $SD(i)$ based on the spectrum, the second is a chroma variation function $CV(i)$ based on the chromagram. The two features well cover the two assumptions that downbeats are stronger in energy than most beats and are the most likely instants where harmonic changes

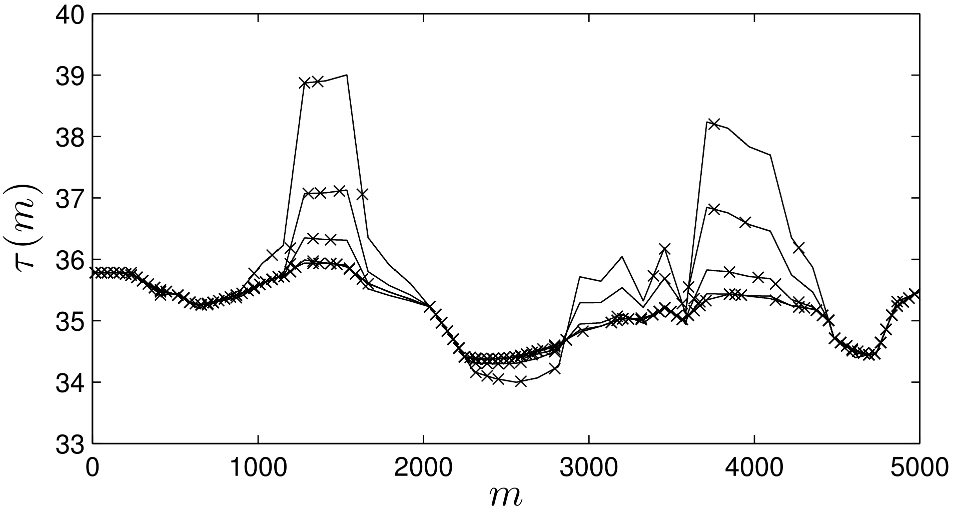


Figure 4.17: m is the sample index of tempo tracking songs and $\tau(m)$ is the estimated periodicity, expressed in number of ODF samples. At each iteration the algorithm smooths the tempo path in segments with least voted beats (most voted one are marked with an X). This way it can effectively correct the two tempo peaks, mistakenly detected by the earlier tempo tracking stage.

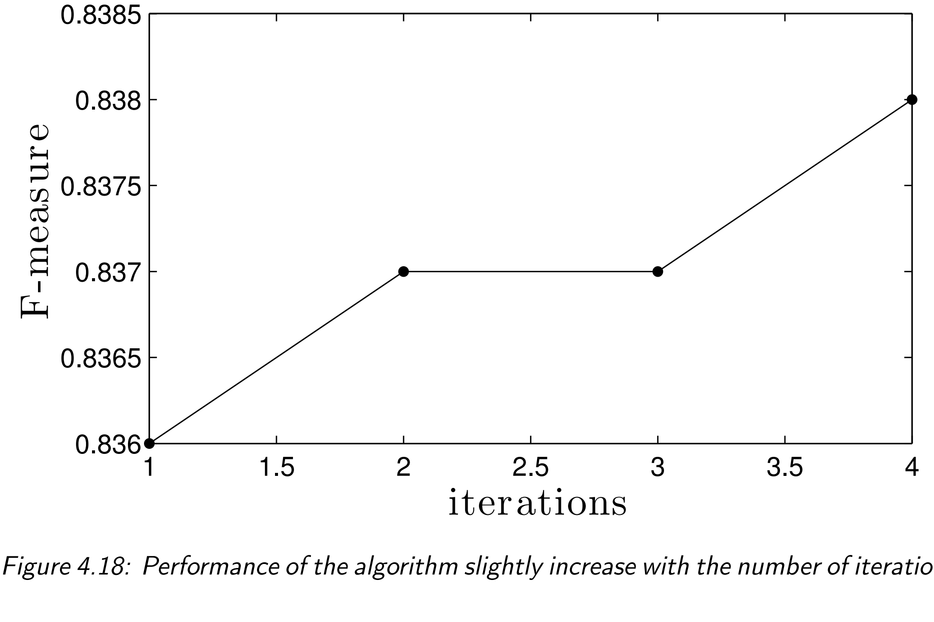


Figure 4.18: Performance of the algorithm slightly increase with the number of iterations

can be found. We obtain a detection function $D(i)$ as the sum of the two:

$$D(i) = SD(i) + CV(i) \tag{4.24}$$

.

To compute the spectral difference feature SD (Fig. 4.19) we first subsample the audio signal by a factor of 16. This retains the part of the spectrum ($<$ 2.8 kHz) that contains most of the energy of the signal. Then we compute the STFT. Then, we beat-synchronize the STFT, averaging the spectrum slices within every inter-beat interval, to obtain one spectrum slice per beat $S_i(\omega)$. To emphasize the most prominent peaks we pre-process these slices one by one, applying a moving average threshold and then half-wave rectifying. The spectral difference is then achieved by using the Kullback-Leibler (K-L) divergence between successive slices:

$$SD(i) = \sum_{\omega=1}^{N/2} \hat{S}_i(\omega) \ln \frac{\hat{S}_i(\omega)}{\hat{S}_{i+1}(\omega)} \tag{4.25}$$

.

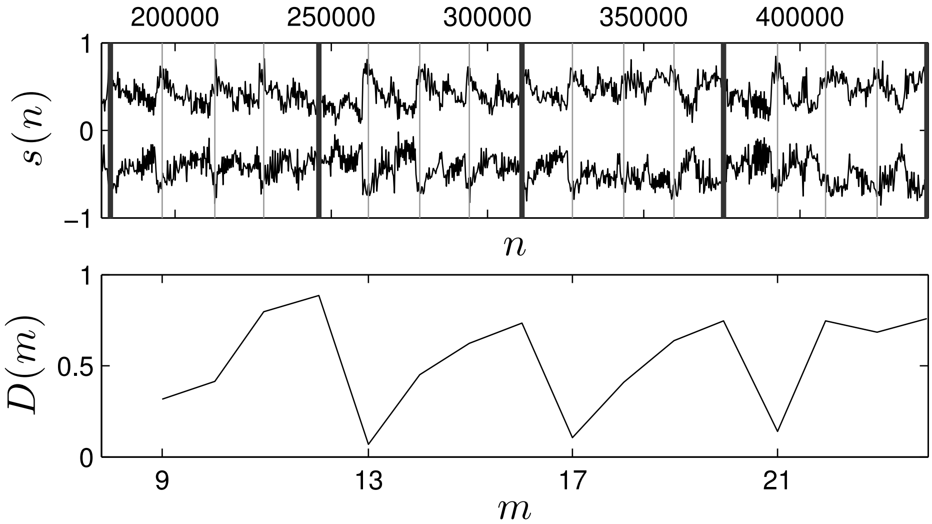


Figure 4.19: The spectral difference function for downbeat detection has one value per beat. n is the audio sample index and m is the beat index. The two functions have different temporal resolutions but corresponding indices n and m are aligned in the figure for better understanding.

To compute the chroma variation function (Fig. 4.20) we first extract the chromagram $C(p,t)$ from the audio signal. Then for every beat b_i we compute a left-context and a right-context chroma vector.

$$C_l(p,i) = \text{mean}(C(p,t)) \qquad b_{i-2} < t < b_i - \frac{\tau(b_i)}{2} \tag{4.26}$$

$$C_r(p,i) = \text{mean}(C(p,t)) \qquad b_i < t < b_{i+2} \tag{4.27}$$

where the $\tau(b_i)/2$ offset accounts for syncopated chord changes. For every beat a chroma variation function $CV(i)$ is extracted using a distance measure between the right and the left context for each beat. As distance measure we compared the sum of squared differences,

$$CV(i) = \sum_{p=1}^{12} (C_r(p,i) - C_l(p,i))^2 \tag{4.28}$$

, and the generalized Kullback-Leibler divergence,

$$CV(i) = \sum_{p=1}^{12} (C_l(p,i) \log \frac{C_l(p,i)}{C_r(p,i)} + C_r(p,i) - C_l(p,i)) \tag{4.29}$$

. The latter outperformed the former (Table 4.1).

SSD	KLD
0.6537	0.6602

Table 4.1: Performances of sum of squared differences (SSD) and KL Divergences (KLD)

Time signature and downbeat

To find the time signatures and downbeat of the song, and consequently label every beat with its number, we apply a simple rationale. The downbeat stands to the beat as the time signature stands to the beat-period. Then, we can feed the previous algorithm with $D(i)$ as onset detection function and a new probabilistic model to obtain the downbeat position.

We subdivide $D(i)$ in frames of 48 samples with hop-size of 4 samples. Pre-processing and autocorrelation is done for each frame similarly to what has been done with η in tempo-tracking. This time we consider three possible periodicities that correspond to 2, 3 and 4-beat measures. The overall distribution and transition probabilities are again extracted from the dataset (Table 4.2).

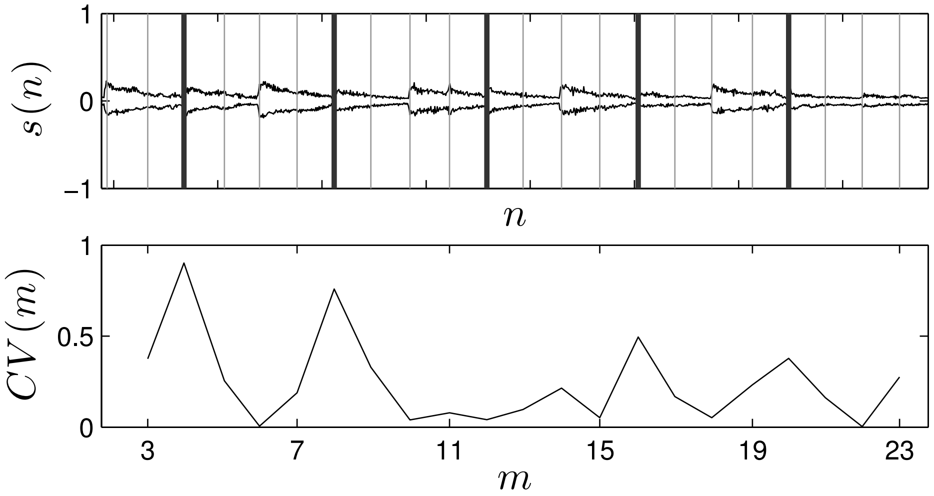


Figure 4.20: The chroma variation function

2/4	3/4	4/4
0.0375	0.0500	0.9125

	2/4	3/4	4/4
2/4	0.8879	0.0087	0.1033
3/4	0.0099	0.9418	0.0483
4/4	0.0065	0.0043	0.9892

Table 4.2: Signature distribution and transition probabilities

The time-signature estimation stage returns a time-signature value $TS(i)$ for every beat.

Indices of downbeats $db(k)$ are then tracked by the multipath algorithm discussed above. This time we add a parameter α in the score function called *regularity.* The eq. 4.20 becomes

$$score_k(db) = p_{I_k}(db) + \alpha T(db - db_k, TS(db_k)) \tag{4.30}$$

. The value of the parameter α that maximizes the overall performance is shown to be 0.5 (Fig. 4.21).

Downbeat are labelled "1" being the first beat of a measure. The other labels are assigned incrementally.

4.2 Chord recognition

In this section we present our system for chord recognition (Fig. **??**). Our main goal is to provide musicians with a meaningful chord transcription. The

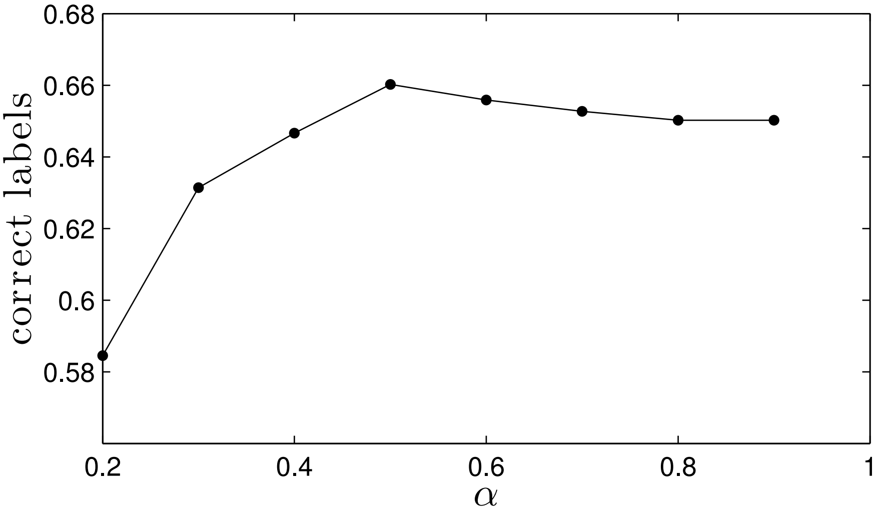


Figure 4.21: Performances of downbeat tracking stage are maximized for a regularity parameter $\alpha = 0.5$

novel approach consists in accounting for all the four most used diatonic key modes: Major, Mixolydian, Dorian and Minor (see Chap. 3.1.3). This way we better fit the key context in songs written on Mixolydian and Dorian key modes and, as a side-effect, we can extract emotional features based on key mode at a later stage.

The model is based on a Dynamic Bayesian Network (described in Section 4.2.3) in which temporal slices correspond to beat instants. Nodes represent musical aspects that are connected with chords. Inputs to the model are the beat synchronized bass chroma vectors and chord salience vectors, the beat labels and other parameters used in *Conditional Probability Distributions* (CPDs) of the nodes. We first give an overview on the creation of the chord salience matrix, and the beat-synchronization technique. CPDs of the nodes are then addressed in the successive sections.

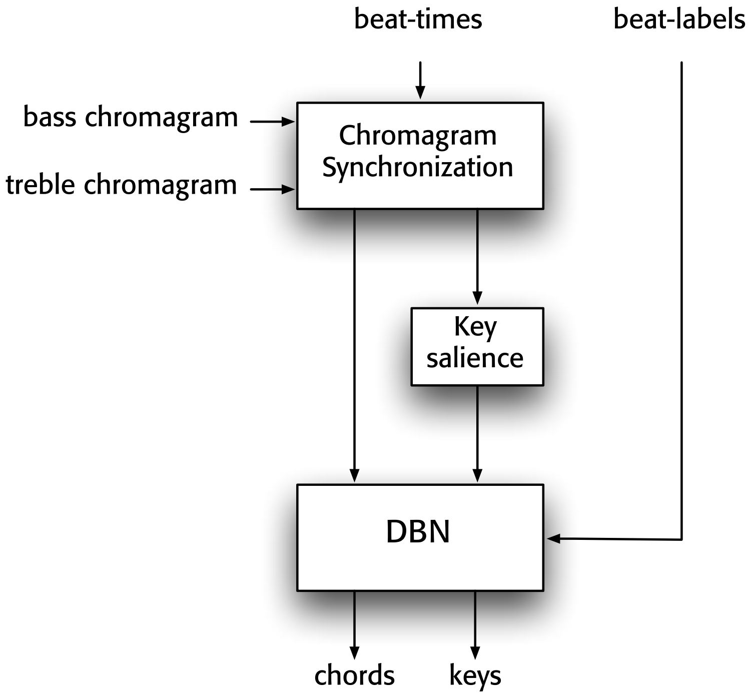


Figure 4.22: Chord recognition scheme

4.2.1 Chord Salience matrix

The chord salience matrix tells how much a chord is likely to be the generator of each chromagram vector (of the wide chromagram, see 3.2.2). For doing this we need a vocabulary of chords templates and a distance measure between chords and chromagram vectors.

As the set of chords templates $\{c_{p,k}\}$, where p is the pitch-class and k is the chord, we used the binary 12-dimensional vectors in which every bin represents the presence or absence of a pitch-class in the chord. For example, in the D major chord, only the pitch classes of D, F♯ and A are present, and are given a value of 1 in the template for the chord (Fig.). Other pitch-classes are set to 0. Only major and minor triads are considered, so we obtain a set of 24 12-dimensional chord templates.

To compare chords templates to the chroma vectors $C^W_{p,t}$ the generalized Kullback-Leibler distance has been proven to be most effective [34]. So we compute a matrix of distances $D(k,t)$ as:

$$D(k,t) = \sum_{p=1}^{12} c_{p,k} \log \frac{c_{p,k}}{C^W_{p,t}} + C^W_{p,t} - c_{p,k}. \tag{4.31}$$

We then invert it and normalize to the maximum norm, to obtain the chord salience matrix $S(k,t)$:

$$S(k,t) = \frac{\min_t(D(k,t))}{D(k,t)}. \tag{4.32}$$

Finally $S(k,t)$ is filtered with a median filter of length 15.

4.2.2 Beat-synchronization

Chords are likely to change in beat instants. It is then sufficient to consider only one chroma vector and one chord salience vector per beat interval. We therefore take the median of bass chroma vectors (see Section 3.2.2) and salience matrix vectors inside every inter-beat interval (Fig. 4.23). This process, called beat-synchronization, achieves a chromagram representation more suited to our task and reduces the sensibility to noise. Beat instants $b(i)$ are provided by our beat-tracking algorithm.

$$\begin{aligned} S_{sync}(p,i) &= \text{median}(S_{p,t}) \qquad t \in \{B_i\} \\ C^b_{sync}(p,i) &= \text{median}(C^b_{p,t}) \qquad t \in \{B_i\}, \end{aligned} \tag{4.33}$$

where $\{B_i\}$ is the set of all the time indices t for which the chromagram frame is inside the inter-beat interval $[b_i, b_{i+1}]$. Finally, each Chromagram frame is normalized by the L_∞ norm.

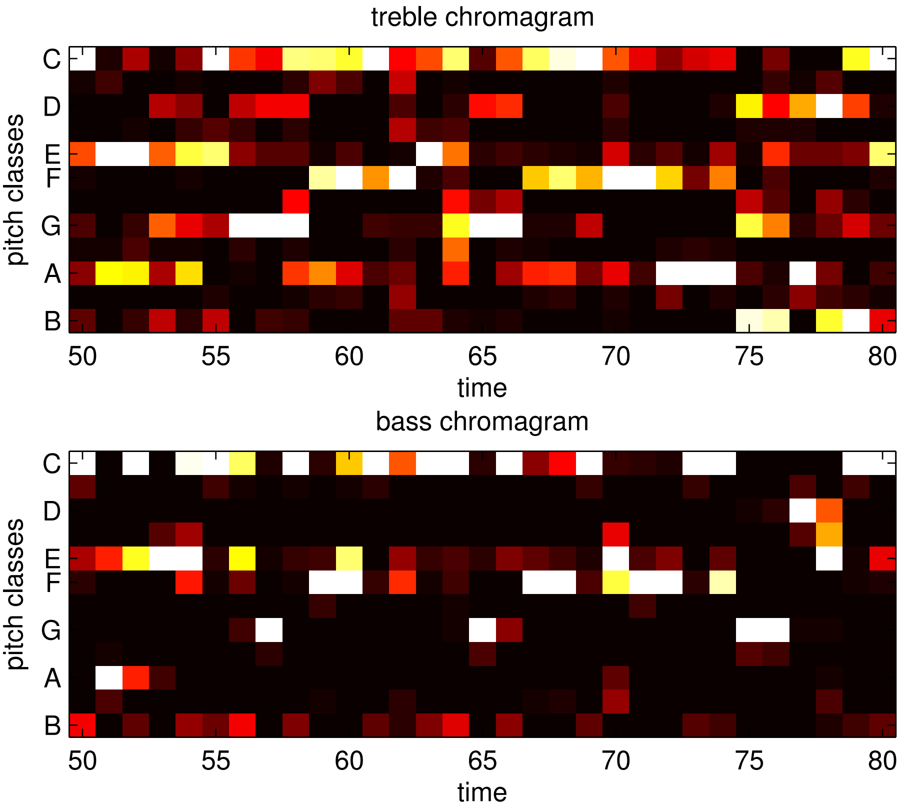


Figure 4.23: Beat-synchronized bass and treble chromagrams.

4.2.3 DBN Model for chord recognition

To model the relationships between different aspects of music that relate to chords, we make use of the Dynamic Bayesian Network model with the topology depicted in Fig. 4.24. Musical aspects included in the model are: chords, keys, bass note, beat label. The dependencies are represented by the arrows and are explained in detail in the next sections. The only continuous nodes are S and C^B, the others are discrete.

To specify the evidences (the sequences of observations) and to perform inference, we used the Bayes Net Toolbox [32] for MATLAB.

Our system takes the graph of the DBN from the model adopted in [29]. We then for three main aspects:

- the beat labels, output of our beat tracker, are given as observed evidence

- the model of the key node, that includes all the four most used key modes. The key transition model we adopt is based on a purely theoretical approach, biased by a-priori analysis of the chromagram of the whole song.

- the model of the chord node, which include a trained distribution of chord changes given the beat labels, and a theoretical based parametric distribution of chords given key.
- the model of the chord salience node

The probabilistic models of K,C and S are discussed in the next sections.

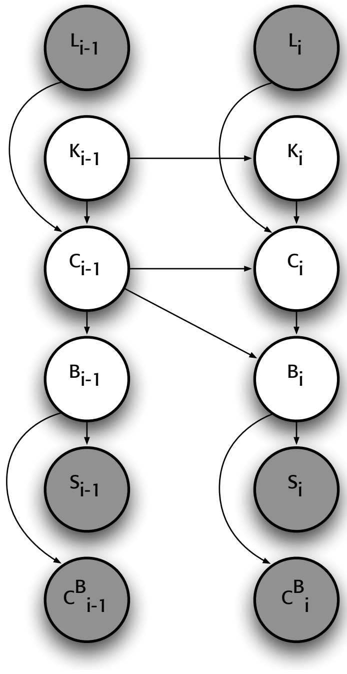


Figure 4.24: The two-slice temporal Bayes net of our chord tracking system. Where shaded nodes are observed and white node are the hidden states nodes. L stands for beat label, and it represents beat labels assigned during downbeat detection. K is the Key node, C the Chords node, B the bass note node, C^B is the Bass beat-synchronized Chromagram and S the chord salience node.

4.2.4 Key and chord symbols

For the key node, we used a set of 48 symbols to model the 4 modes that we are interested in (Major, Mixolydian, Dorian and Minor), in any of the 12 key roots. Lets use the following conventions:

- $M(\kappa)$ is the mode of a key κ and can assume values from 1 to 4.

- $R(\kappa)$ is the root of a key κ and can assume values from 1 to 12 (12 pitch classes).

For the chord node we used 24 symbols to model major and minor triads.

- $T(\xi)$ is the type of a triad ξ and can assume values from 1 to 2 indicating major and minor.
- $R(\xi)$ is the pitch class of the root note of a chord ξ (12 pitch classes).

4.2.5 Key node

As said in Chapter 3, key encloses two concepts: the *key root* or tonic is the most important and stable pitch class, and the *key mode* is the set of other notes in relation to the key root. Key node is a discrete random variable that describes the 48 keys. Key is modelled as generating chords, in fact, the seven chords that are present in the harmonization of a key (see Fig. 3.3) happen more likely than the others in that key context.

In our system the Key node K_i depends only on the predecessor K_{i-1} so we only need to model key transitions. In doing that we first model a parametric distribution based on musicological considerations. Then we multiply it by a key salience vector to exploit information about the Key root coming from a time-coarse analysis of the chromagram.

Key transitions using musicological cues

The values that indicate the probability of the transition are stored in a matrix K_{trans} that is built following some steps.

First we can see that a measure of similarity between keys given by the number of common notes. If $C(\kappa_1, \kappa_2)$ is the number of common notes between two keys κ_1 and κ_2, we say that the probability of transition between them is:

$$\begin{aligned} &\text{if}(\kappa_1 \neq \kappa_2) \quad K_{trans}(\kappa_1, \kappa_2) = \gamma_c^{((7-c)+1)}, \\ &\text{else} \qquad\qquad K_{trans}(\kappa_1, \kappa_2) = 1 \end{aligned} \tag{4.34}$$

where the value of coefficient γ_c that performs best is $\gamma_c = 0.3$.

We then adjust the probability in two specific cases. Parallel keys are the pair of keys for which $R(\kappa_1) = R(\kappa_2)$ but $M(\kappa_1) \neq M(\kappa_2)$. For example, C Major and C Dorian are parallel keys. We raise the probability of modulation between parallel keys by a factor γ_p:

$$\begin{aligned} &\text{if}(R(\kappa_1) = R(\kappa_2)\,\text{AND}\,M(\kappa_1) \neq M(\kappa_2)) \\ &\qquad\qquad K_{trans}(\kappa_1, \kappa_2) = K_{trans}(\kappa_1, \kappa_2) \times \gamma_p, \end{aligned} \tag{4.35}$$

where $\gamma_p = 4$ This is to account for a technique much used in harmony, called *modal interchange*, that consists in borrowing a chord from a parallel key. So temporary transition between parallel keys are frequent.

Diatonic keys are the pair of keys that have all the notes in common $C(\kappa_1, \kappa_2) = 7$ and different root $R(\kappa_1) \neq R(\kappa_2)$. For example, C Major and G Mixolydian are diatonic keys. These kind of transitions are often mistakenly detected by the system so we lower their probability by a factor γ_d.

$$\begin{aligned}&\text{if}(R(\kappa_1) \neq R(\kappa_2)\,\text{AND}\,C(\kappa_1, \kappa_2) = 7)\\ &\qquad K_{trans}(\kappa_1, \kappa_2) = K_{trans}(\kappa_1, \kappa_2) \times \gamma_d,\end{aligned} \tag{4.36}$$

where we found experimentally $\gamma_d = 0.15$ to be the best value for this parameter.

We then normalize each row of K_{trans} to obtain stochastic row vectors.

$$\bar{K}_{trans}(i,j) = \frac{K_{trans}(i,j)}{\sum_{q=1}^{48} K_{trans}(i,q)} \tag{4.37}$$

This is not yet the transition matrix we feed to the system. To account for the information about key root provided by the chromagram, we perform a pre-analysis to find a key root salience vector $k_s(p)$.

Key root salience vector

$k_s(p)$ says how likely the pitch class p is the key root in a song. To compute $k_s(p)$, we find the correlation of the averaged Chromagram with the 12 circular shifts of a key profile $k^*(p)$. Let P_n be the permutation matrix that represent the circular shift operation,

$$\begin{gathered}(p) = \sum_{t=1}^{t=T} C^t(p,t)\\ k_s(n) = \text{corr}(P_n k^*(p), c(p))\end{gathered} \tag{4.38}$$

The Key Profile is obtained from treble chromagrams of the Robbie Williams Dataset, which we have manually annotated all the key roots and modes. Let K_t be the key annotation at time t, the key profiles of the four modes are

$$k^*(p, M(K_t)) = k^*(p, M(K_t)) + P_{R(K_t)} C^t(p,t) \tag{4.39}$$

for every time instant t of every song in the dataset. Then the profiles of the four modes are normalized and then combined into one vector $k^*(p)$ (Fig. 4.25).

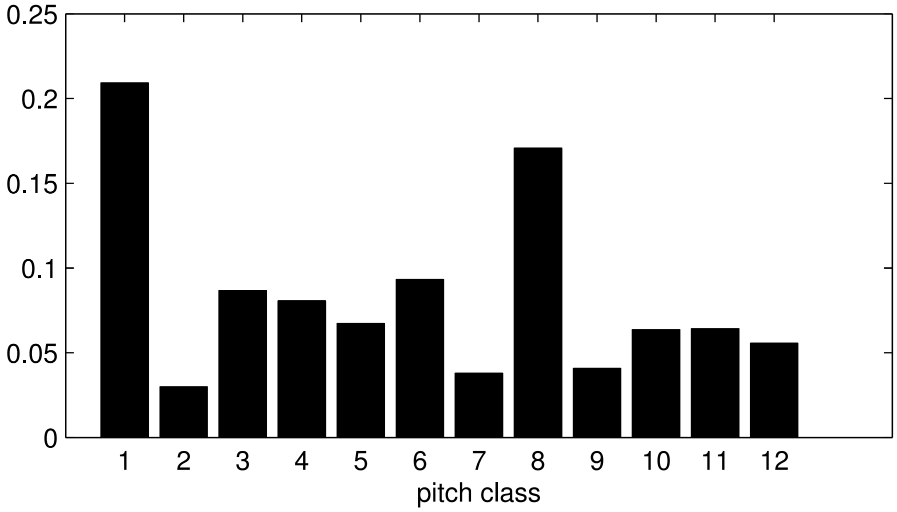


Figure 4.25: The pitch profile $k^*(p)$ obtained from the dataset.

$k^*(p)$ is then used to compute a new version of the key transition matrix:

$$\bar{K}_{trans}(\kappa_1, \kappa_2) = \bar{K}_{trans}(\kappa_1, \kappa_2) k^*(\kappa_2), \tag{4.40}$$

which is then re-normalized

$$K_{trans}(i, j) = \frac{\bar{K}(i, j)}{\sum_{q=1}^{48} \bar{K}(i, q)}. \tag{4.41}$$

4.2.6 Chord node

As can be seen in Fig. 4.24, the chord node depends on current beat label (the position within the measure), on current key and on previous chord.

To understand the dependence on beat label, let's notice that a chord change is more likely to occur at the start of a new measure, when the current beat is a downbeat (the beat label $L_i = 1$) than within the measure. We model the probability that, given the beat label, a chord change will occur into the function $f_l(C_i \neq C_{i-1}, C_{i-1}, L_i)$. It varies a lot and it is very different for different time signatures. We trained from the dataset the values of $f_l(C_i \neq C_{i-1}, C_{i-1}, L_i)$ for time signatures of 4/4 and 3/4.

The dependence on the key is given by the assumption that chord of the harmonization of a key are more likely than chord not present in that harmonization. To model this concept we use a parametric chord profile for every key mode. In this model we assign $f_k(C_i, K_i) = \lambda_1$ to chord present in the key, $f_k(C_i, K_i) = \lambda_2$ to chords not in the harmonization, and $f_k(C_i, K_i) = \lambda_3$ to one characteristic major chord and one characteristic minor chord for key mode. We select this two chords as those that, alternated to the tonic triad, give more the feeling of a key mode [36]. The values that gave us the best result are $\lambda_1 = 1$, $\lambda_2 = 0.7$, $\lambda_3 = 1.2$.

The CPD $f_k(C_i, K_i)$ is then normalized to obtain a matrix with stochastic row vectors.

The two probabilities are then combined

$$p(C_i, L_i, K_i, C_{i-1}) = f_l(C_i, L_i, C_{i-1}) f_k(C_i, K_i), \tag{4.42}$$

and finally the 4-dimensional $p(C_i, L_i, K_i, C_{i-1})$ is normalized multiplying by a normalization constant k_{norm} such that:

$$\sum_{C_i=1}^{N_chords} k_{norm} p(C_i | L_i, K_i, C_{i-1}) = 1. \tag{4.43}$$

4.2.7 Chord Salience node

The chord salience node maps each slice of the chord salience matrix $S(k, t)$ to a chord k. The most likely chord for every instant will have a value of 1 and will be the maximum, due to our normalization of $S(k, t)$. Given the structure of S this is a continuous node, so we create a Normal distribution with the following parameters:

$$\begin{aligned} \mu &= I_{24} \\ \sigma_k &= 0.2 \times I_{24}, \qquad k = 1, ..., 24 \end{aligned} \tag{4.44}$$

where μ is the mean and σ is the covariance.

4.3 Feature extraction

In this section we provide an overview of three novel harmony-based features extracted from key mode and chord sequence. The three features are called Modal Envelope, MajMin Ratio and Harmonic Rhythm. After seeing their structure, we suggest a possible mood interpretation for each one.

Modal Envelope

Modal envelope is represented by the temporal evolution of the key mode. The feature is then:

$$ME(t) = M(\kappa(t)), \tag{4.45}$$

where t is the time index, $\kappa(t)$ is the key at time t as resulted from our beat tracking system, and $M(\kappa(t))$ is the mode of that key (see Section 4.2.4).

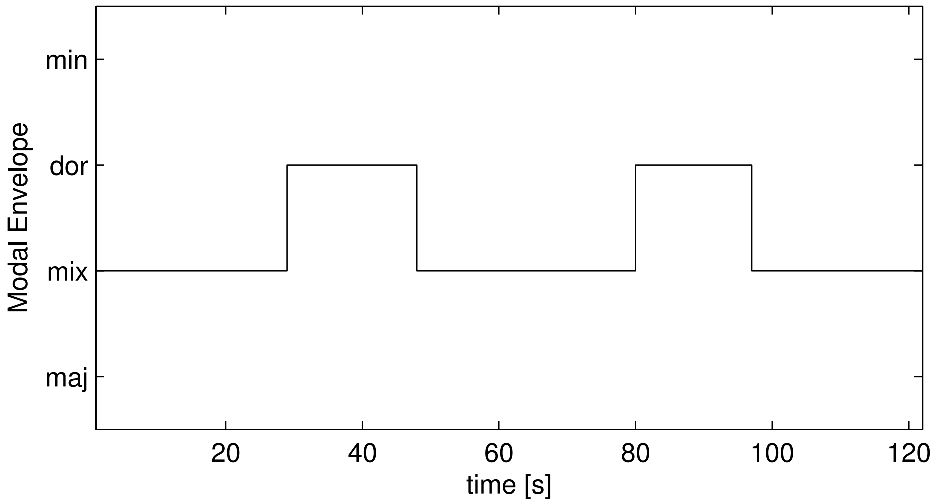


Figure 4.26: Norwegian Wood is a great example to show the power of the Modal Envelope descriptor. Verses of the song are in Mixolydian mode and Choruses are in Dorian mode. The descriptor perfectly and precisely follows the shift in mood of the song. In this particular case it also achieve optimal structural segmentation

The rationale is that the four main key modes are emotionally ordered from the brightest to the darkest. We can exploit this fact and create an envelope that follows these kind of mood changes within a song. To do so we map the four key modes to the numbers $[-1, -0.33, 0.33, 1]$ and plot the envelope versus time. A great example of the descriptive power of this feature is showed in Fig. 4.26.

Moreover, as said, modal interchange is a technique that allows the composer to borrow chords from a parallel key mode. In doing so, a little drift in mood envelope is obtained. This is very desirable from the composer standpoint because it generates a point of interest in the chord sequence. To track these faster shifts of mood we post-process the chord sequence to find all the chords that belong to a parallel key. With this information we compute a new key mode envelope, that results more fragmented, as expected (Fig. 4.27).

MajMin Ratio

MajMinRatio feature represents the ratio of major triads over minor triads within a time window. The importance of this feature follows from the fact that major and minor triads in western music are linked with opposite set of emotions as those showed in Table 4.3 ([23]).

To model this quality of chords we retain only the information about the

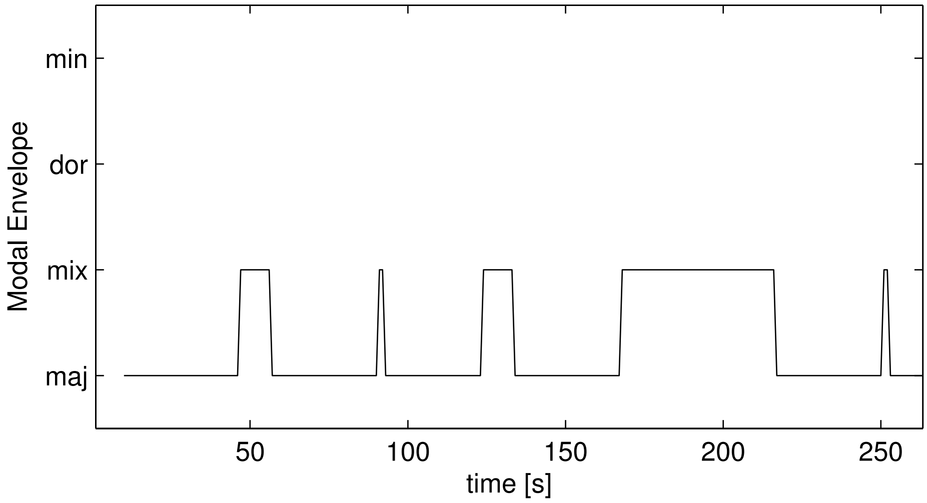


Figure 4.27: Finer modal envelope feature obtained by a post-processing of chords. The song is "Angel" by Robbie Williams. Spikes from major mode to mixolydian mode reveal where the composer made use of modal interchange. The longer zones are instead temporary modulations (key change) to the parallel mixolydian mode.

Major triad	Minor triad
Happy	Introspective
Optimistic	Dramatic
Bright	Sad
Cheerful	Melancholy
Satisfying	Serious
Light	Longing

Table 4.3: Moods consistently associated with major and minor triads.

type of the chord: $T(c(t))$, where $c(t)$ is the chord at time t. $T(c(t))$ is 1 if the chord is major and 0 if the chord is minor.

For this feature we exploited a time window $w(t)$,

$$w(t) = \begin{cases} 0 & t <= -\tau \\ (t-\tau)\frac{2}{\tau^2} & -\tau < t <= 0 \\ 0 & t > 0 \end{cases}, \tag{4.46}$$

that models the influence that music events have on the listener. The window is half triangular, its area is 1 and the duration is $\tau = 10$ seconds (Fig. 4.28).

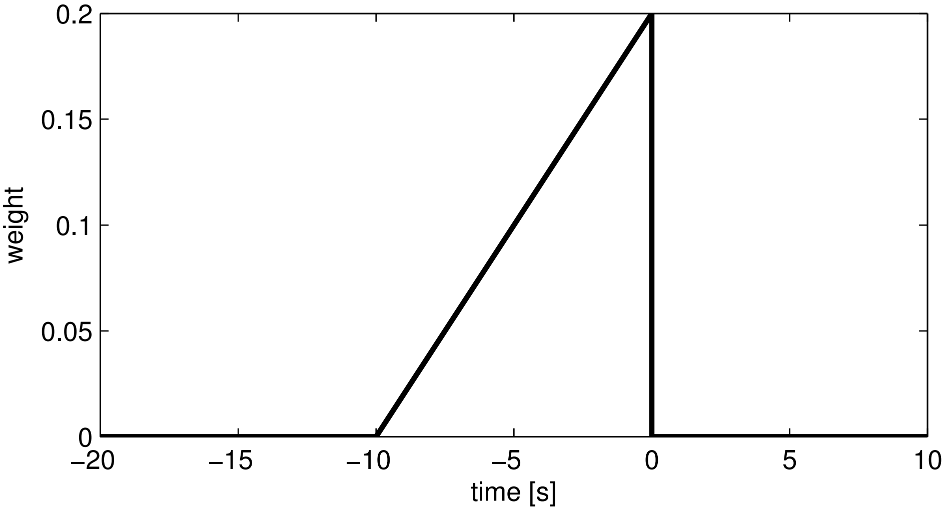


Figure 4.28: This function is a time window used to compute the majMinFeature. The rationale is simple: just listened events have a stronger influence than those happened before.

The feature is obtained as:

$$Mm(t) = \sum_{t'=t-\tau}^{t} T(c(t'))w(t'-t). \tag{4.47}$$

A good example of the descriptive power of this feature is showed in Fig. 4.29.

Harmonic Rhythm

Harmonic rhythm feature represents the rate of harmonic changes detected in a certain time frame of the audio. The instants of the changes are represented by a function

$$Hc(t) = \begin{cases} 1 & c(t^-) \neq c(t^+) \\ 0 & c(t^-) = c(t^+) \end{cases}, \tag{4.48}$$

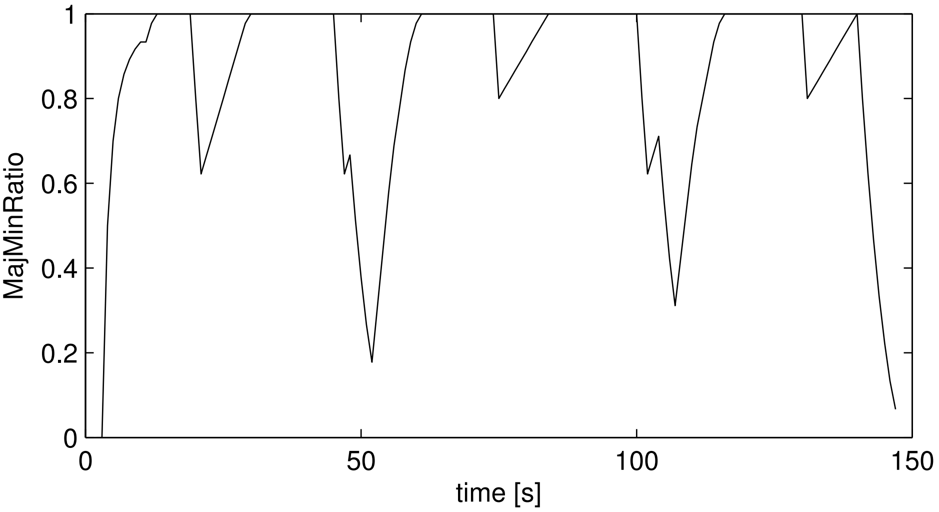


Figure 4.29: The figure shows an example of the majMinFeature on the song "Hard Day's Night" by The Beatles. The deepest valleys fall exactly at the choruses, where most minor chords are found.

where t^- and t^+ indicate respectively the left and right neighbourhood of t. For this feature also we use $w(t)$ to smooth the temporal distribution in a consistent way.

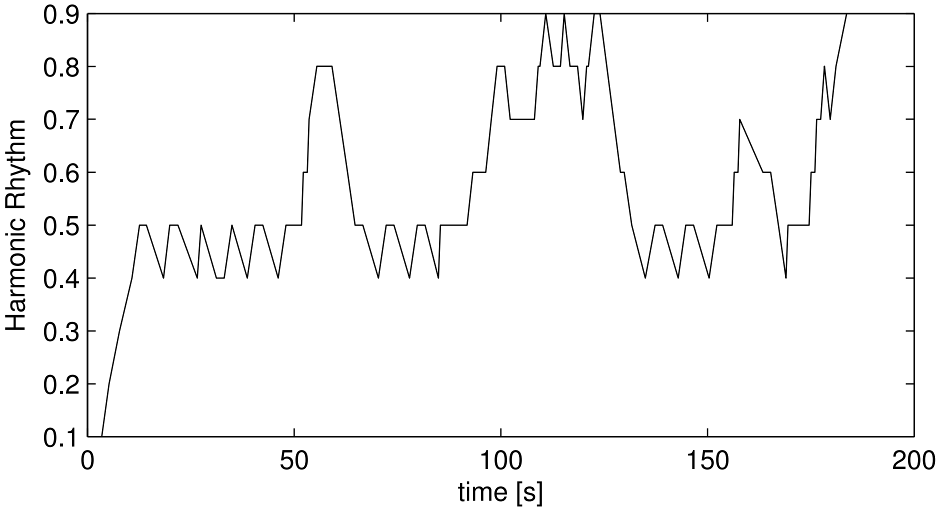


Figure 4.30: The figure shows an example of the Harmonic Rhythm feature on the song "Here Comes the Sun" by The Beatles. The longest peaks represents parts of the song where chords changes are very frequent.

Chapter 5

Experimental Results

In this chapter we will focus on experimental results of our beat-tracking and chord-recognition system, and approach the problem of evaluating the three proposed emotion-related features. Before presenting the numeric results an overview of the used evaluation metrics will be provided.

5.1 Beat Tracking

In this section we review $F_{measure}$ and CML (acronym of Correct Metrical Level), the two evaluation metrics used for beat-tracking. Then we present the annotated dataset used and numeric results, compared with other reference algorithms the state of art.

5.1.1 Evaluation

The most used metrics for the evaluation of a beat sequence $\{b_i\}$ against ground truth annotations $\{a_j\}$ are the F-measure and the two correct metrical level (CML) measures, as described in [10]. The two sequences have different indices because they might be asynchronous. For example if the algorithm misses the second beat from annotation, we obtain that b_1 is synchronous with a_1 and b_2 synchronous with a_3.

F-measure is obtained from three basic parameters: the number of correct detections or true positives c, the number of incorrect detections or false positives f_+ and the number of missed detections or false negatives f_-. A beat b_i is correct if falls within a tolerance window of an annotation a_j, where the used tolerance is $\pm 70ms$. c, f_+ and f_- are used to compute two intermediate parameters: precision and recall.

$$p = \frac{c}{c + f_+} \tag{5.1}$$

$$r = \frac{c}{c + f_-} \tag{5.2}$$

They are combined with harmonic mean to provide the F-measure

$$f = \frac{2pr}{p + r} \tag{5.3}$$

If the beat sequence $\{b_i\}$ falls on the offbeats F-measure is zero. If the $\{b_i\}$ has double or half beat-period of the annotations, respectively $f_- = c$ or $f_+ = c$, then F-measure falls accordingly. CML_c and CML_t are two continuity-based evaluation methods, used in [19] and [25]. Beats $\{b_i\}$ are compared against ground truth annotations $\{a_j\}$ with a set of three rules:

1. $a_j - \theta\Delta_{a_j} < b_i < a_j + \theta\Delta_{a_j}$
2. $a_{j-1} - \theta\Delta_{a_{j-1}} < b_{i-1} < a_{j-1} + \theta\Delta_{a_{j-1}}$
3. $(1 - \theta)\Delta_{a_j} < \Delta_b < (1 + \theta)\Delta_{a_j}$

where Δ are the inter-beat intervals, e.g. $\Delta_{a_j} = a_{j+1} - a_j$, and θ is a tolerance parameter, fixed to $\theta = 0.175$ [10]. If these rules are satisfied, the beat b_i is correct and we can extract the number of continuously correct segment M, and the number of beats in each of them Υ_m. The less restrictive measure CML_t sums the contribution of all the M continuously correct segments.

$$CML_t = \frac{\sum_{m=1}^{M} \Upsilon_m}{J} \tag{5.4}$$

where J is the total number of annotations. The more restrictive measure CML_c uses only the contribution of the longest continuously correct segment

$$CML_c = \frac{\max(\Upsilon_m)}{J} \tag{5.5}$$

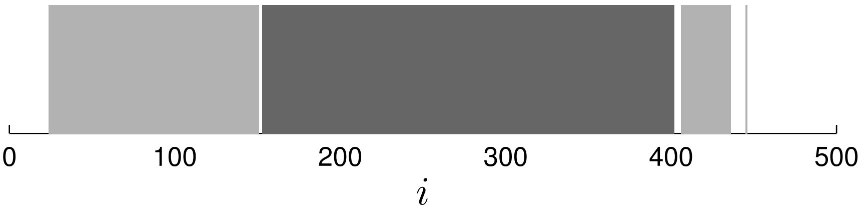


Figure 5.1: Continuously correct segments are showed as grey rectangles, the longest is darker. $CML_t = 0.951$ *and* $CML_c = 0.582$

For the evaluation of beat labels we used the percentage of correct labels over the total number of beats in a song. Since labels' correctness can be evaluated only for correct beats, this measure and F-measure are correlated. This however seems to be the best strategy since dividing only by the correct beats would result in an unreliable measure.

5.1.2 Dataset

The dataset we used to test our system is composed by manual annotations of the Beatles discography and of the first two albums of Robbie Williams. Beatles annotations are provided by Davies in [10] and available online[1]. Robbie Williams annotation work has been carried on by us to extend the sample towards a more modern style, and will be available online in the near future.

5.1.3 Results

We tested our system (named "MultipathBT") and other three reference systems over the Beatles dataset and over the Robbie Williams dataset. The other systems we tested are the beat-tracker plug-in [9] from Sonic Visualizer[2], the beat-tracker from Harmony Progression Analyzer Toolbox [33] and a basic implementation of a dynamic programming beat-tracker with varying tempo.

	F-m	CML_c	CML_t
MultipathBT	0.8386	0.6250	0.7086
Sonic Visualizer PlugIn	0.8042	0.6160	0.7084
HPA Beat-Tracker	0.6538	0.2196	0.4003
DP Based Beat-Tracker	0.6380	0.4303	0.5203

Table 5.1: Performances on the Beatles Dataset

	F-m	CML_c	CML_t
MultipathBT	0.8527	0.6078	0.7353
Sonic Visualizer PlugIn	0.7792	0.6857	0.7561
HPA Beat-Tracker	0.7324	0.2295	0.4518
DP Based Beat-Tracker	0.6630	0.4511	0.5599

Table 5.2: Performances on the Robbie Williams Dataset

[1] http://isophonics.net/content/reference-annotations

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

	Correct Labels
MultipathBT	0.6484
Sonic Visualizer PlugIn	0.5791

Table 5.3: Performances of downbeat tracking on The Beatles Dataset

	Correct Labels
MultipathBT	0.7605
Sonic Visualizer PlugIn	0.6342

Table 5.4: Performances of downbeat tracking on the Robbie Williams Dataset

5.2 Chord Recognition

In this section we describe the two evaluation metrics adopted to score the performance of our chord-recognition algorithm: *Relative Correct Overlap* (RCO) and *segmentation quality* (SQ). Then, we address the problem of evaluation of key sequences. Finally we present the annotated dataset used for evaluation and the results of our algorithm according to the chosen metrics.

5.2.1 Evaluation

Relative Correct Overlap evaluates a chord transcription against a ground truth annotation. Chords are a segmentation of the song. So we have basically to compare two different segmentations. RCO compute the total duration of overlapping segments having the same label, and then divides it by the duration of the song.

$$RCO = \frac{\text{correct overlap time}}{\text{total duration}} \tag{5.6}$$

This measure can be used to evaluate performance on each song in the Dataset. If we want a single measure we have to weight the RCO by the song duration. This measure is called *weighted average overlap ratio* (WAOR). Overlap based measures however don't address the fact that a very fragmented transcription can achieve a high value, while being totally useless from a musical standpoint.

Segmentation quality measure is based on the *Directional Hamming Divergence*, a metric used in the field of image segmentation. It measures the diversity of a segmentation $S = \{S_i\}$ respect to another taken as reference

$S^0 = \{S_i^0\}$, where S_i and S_i^0 are segments. The word directional indicate that this measure is not symmetric.

$$h(S||S^0) = \sum_{i=1}^{N_S}(|S_i^0| - \max_j |S_i^0 \cap S_j|) \tag{5.7}$$

, where $|S_i|$ is the duration of a segment. The ideal segmentation $S = S^0$ results in $h(S||S^0) = T$, where T is the duration of the song. As proposed in [20], a measure of the similarity of the segmentation is given by:

$$h(S||S^0) = 1 - \frac{1}{T}\max\{h(S||S^0), h(S^0||S)\} \in [0, 1] \tag{5.8}$$

,

To evaluate the key sequences we used two different overlap-based metrics and one segmentation-based metric *kSegmQ*. Of the two overlap based metrics, *krRCO* considers only key roots, and *krmRCO* accounts for both key root and key mode.

5.2.2 Dataset

The dataset that we used for chord recognition task is the same used before, including the Beatles discography[3] and of the first two albums of Robbie Williams. The annotations of keys and chords are provided in [20]. The annotations for Robbie Williams's song is provided by us. The Beatles annotations however omit systematic transcription of Mixolidian and Dorian modes. To fully evaluate our key mode transcription we used only our dataset.

5.2.3 Results

We tested our system (named "CTM") against an implementation of the algorithm ("MD") described in [29] (Table **??**).

	RCO	segmQ	krRCO	krmRCO	kSegmQ
CTM	0.720	0.757	0.754	0.68	0.864
MD	0.696	0.757	0.743	0.66	0.901

Table 5.5: Performances on The Beatles dataset. krmRCO is computed mapping key mode sequences of our system to only major and minor modes, as they only are present in the ground truth annotations

[3] The Beatles corpus was used for the chord recognition task of MIREX 2008.

	RCO	segmQ	krRCO	krmRCO	kSegmQ
CTM	0.707	0.797	0.769	0.49	0.485
MD	0.697	0.785	0.697	-	0.330

Table 5.6: Performances on the Robbie Williams dataset. All four the key modes are present in the annotations so we can meaningfully evaluate only our system

5.2.4 Harmony-related features

Harmony is strictly related to the emotions perceived in a song, for this reason, in order to evaluate the effectiveness of the defined harmony-related features, we correlate them to mood variation. The evaluation of the proposed harmony-related features is not an easy task, due to the lack of a proper corpus of reliable annotations. We used the *msLiteTurk* dataset [42] to execute some measures.

The dataset provides Arousal and Valence (AV) tags by 546 testers. The AV space is the most used mathematical representation of emotions. It defines emotions as points in a 2D space in terms of Valence (how positive or negative) and Arousal (how exciting or calming) [45]. These two emotion dimensions are found to be the most fundamental by psychologists.

A segment of 15 seconds for each song is annotated with time resolution of a value per second. Since our feature are meaningful at larger scale, we extracted only a value for each segment of annotations.

To achieve correspondence with the direction of moods in the valence axis, we flipped our modal envelope feature. This way we obtain darker moods on the bottom and brighter moods on the top of the scale.

We based our evaluations on the Pearson Bravais correlation coefficient $r(x, y)$

$$r(x, y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y}, \tag{5.9}$$

where x is our feature, y are the tags, σ_{xy} is their covariance and σ_x and σ_y their standard deviation.

We subdivided the msLiteTurk dataset by the IDs of tagging users and computed the correlations between their tags and our features. We computed a new index:

$$\rho = \frac{n_p^d}{n_p^i}, \tag{5.10}$$

where n_p^d is the number of users for which our feature x and tags y have $r(x, y) > p$, and n_p^i is the number of users for which $r(-x, y) > p$. Results we obtained for $p = 0.6$ are showed in Table 5.7.

	ME	Mm	HR
valence tag	3.05	6.56	7.35
arousal tag	0.37	3.81	5.10

Table 5.7: Results of the ρ index shows that in 5 of the 6 cases, there is a direct proportionality between our feature and the annotations. ME stands for Modal Envelope, Mm for MajMinRatio, HR for Harmonic Rhythm

As expected we noticed that for the majority of the users there is a direct proportionality between this pairs of variables:

- (valence, modal envelope)
- (valence, major minor ratio)
- (valence, harmonic ratio)
- (arousal, major minor ratio)
- (arousal, harmonic ratio)

This good results should however be tempered by the fact that the annotations of different users on the same songs tends to be very contrasting. Our results may outline how the perception of harmony greatly influence the way some individuals perceive music.

Chapter 6

Conclusions and future work

In this work a system for the automatic retrieval of musically relevant features from audio has been proposed. The system includes algorithms to explore a song at the rhythmic level and at the harmony level.

As far as concerning the rhythm analysis, we introduced a novel approach in finding beat instants. The sequence of beat instant is found starting from two features, the onset detection function and the rhythmogram. Usually the errors generated in the creation of these two features are propagated to the beat sequence. We addressed the issue at the later stage of beat sequence tracking. We first proposed a novel technique of path finding based on keeping track of more paths at once and perturbing them as they converge. We noticed how the different paths converged on the most reliable beat instants, because considering only them resulted in increased precision. We exploited this fact by iteratively morphing the beat-period path starting from the most reliable beats. This results in a smoother, and often more accurate beat path.

Given the hierarchical structure of beat, the same steps can be successfully applied in finding the downbeats, the first beats of the measures. This allow us to consider many different and changing time signatures. This is good not only to track songs that really have changing time-signatures, but also to adaptively follow the measure level in case of mistakenly missed, or added beats. This is yet another case in which a later stage could correct errors from earlier stages. The detection function that we used for downbeat tracking is a combination of an energy based feature and a chromagram based feature. This, in our view, well models the intuition that harmony and rhythm levels are deeply interconnected.

As far as concerning the harmony analysis focus on the chord and key structure. In particular, we focused on the importance of diatonic key modes in the probabilistic model of chord progressions in western music. The state-of-art systems only handle two key modes, Major and Minor. We showed how this two modes are the two extremes of a possible ordering of four most used modes, that includes also the Mixolydian and Dorian modes. These two central modes defines intermediate hues and are very used by pop composers to convey specific emotions. For the purpose of chord recognition, we propose new probability model for key based on transitions between parallel keys and between diatonic keys. That model specification needs a new and musically meaningful way to describe chords within keys. We propose a parametric model where each parameter represent a musical characteristic. These new specifications and the tuning of the parameters have been shown to improve the accuracy of the chord recognition, from what we considered to be the state-of-art system.

The availability of a complete range of key modes gave us the possibility to exploit this information in other ways. In particular, the used ordering of these key modes has a strong emotional interpretation. It can be linked to a scale that goes from brighter moods (Major) to darker moods (Minor). We therefore propose three features based on the sequence of chords and keys extracted by our system. The first one, called Modal Envelope is based on the sequence of key modes and exploit their link with emotion. The second, called MajMin Ratio, computes the ratio of major and minor chords, weighting their duration by a window that models the listener's temporal memory for music events. It is based on the fact that major and minor chords are associated in western music culture with defined emotions. The contribute of single chords can refine the general mood given by the key mode. The third one, Harmonic Rhythm, simply computes the number of chord changes in a time window. These features can be used for many purpose mainly audio classification, mood-based segmentation and automatic tagging based on mood.

6.1 Future works

The beat tracking system structure is significant because it suggests that exploiting musical knowledge we can operate on higher level features to correct mistakes carried by the lower level ones. The main goal might be to create an automated system that iteratively maximizes the "*structure*" (purposely generic term to describe cross and self-similarities), by concurrently extracting features from different but interconnected aspects of music. We approached the surface of this issues by exploiting a chroma variation function in the downbeat detection algorithm, and the beat labels in the chord recognition system. A deeper research in this direction might be desirable.

The chord tracking system we presented is based on probabilistic modelling based on musicological considerations. Its parameters therefore aim to be universal, at least in the context of western music. We can asymptotically approach this goal by training the parameters on a very large dataset. A clever move in this direction might be to build a musician-friendly interface to perform computer-aided transcriptions. It can take several listenings to a trained musician to transcribe a song, but with the help of a transcription system it can reduce to one. Musicians save time and we gather an huge dataset. Besides, the current accuracy of chord recognition systems is not yet sufficient to provide transcriptions that don't need further modifications. The interaction with the user is what we see as the more profitable direction of research.

The contribution of the key modes to mood is well established in composition treatises. The currently most used two dimensional space to represent mood-related reature, is the Valence-Arousal (AV) space. However a better relation model can be explored as future work. A consistent evaluation of the three mood features is difficult and would require a listening test with a selected dataset that would enhance the differences between modes.

Appendix A

Detailed results

In this appendix we report the detailed results of our measurements for every song in the dataset. Songs are taken from the albums in Table A.1. The measures that we used to evaluate the performances of our system are reviewed in chapter 5. Here we give a brief overview:

- $F_{measure}$: classic Information Retrieval measure. It is the harmonic mean of precision and recall parameters.
- CML_c: Correct Metrical Level, it is a continuity-based measure. It indicates the duration, relative to the total song duration, of the longest correctly tracked region.
- CML_t: same as CML_c but sums the correctly tracked regions.
- Lab: the number of correct beat labels divided by the total number of correct beats.
- RCO: relative correct overlap. The sum of duration of correct overlap of chords, divided by the duration of the song.
- segmQ: addresses the quality of the segmentation. It is high if there aren't neither under nor over-segmentation issues.
- krRCO: as RCO but addresses the key roots.

album	**song**	F_m	CML_c	CML_t	**Lab**	**RCO**	**segmQ**	**krRCO**
01	01	0.987	0.686	0.980	0.991	0.851	0.900	0.970
01	02	0.948	0.956	0.956	0.960	0.875	0.884	0.954
01	03	0.983	0.984	0.984	0.990	0.700	0.810	0.972

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
01	04	0.995	0.990	0.990	0.997	0.788	0.835	0.964
01	05	0.994	0.991	0.991	0.997	0.744	0.742	0.966
01	06	0.976	0.539	0.977	0.994	0.681	0.820	0.928
01	07	0.981	0.989	0.989	0.996	0.799	0.844	0.957
01	08	0.996	0.991	0.991	0.582	0.657	0.793	0.963
01	09	0.993	0.989	0.989	0.996	0.904	0.835	0.963
01	10	0.941	0.566	0.962	0.941	0.819	0.878	0.967
01	11	0.931	0.986	0.986	0.991	0.675	0.856	0.000
01	12	0.880	0.808	0.894	0.556	0.820	0.767	0.954
01	13	0.996	0.988	0.988	0.996	0.759	0.847	0.950
01	14	0.966	0.609	0.964	0.980	0.724	0.870	0.000
02	01	0.952	0.957	0.957	0.967	0.844	0.824	0.959
02	02	0.994	0.987	0.987	0.996	0.838	0.881	0.000
02	03	0.986	0.991	0.991	0.997	0.814	0.872	0.964
02	04	0.995	0.993	0.993	0.998	0.655	0.820	0.963
02	05	0.973	0.969	0.969	0.977	0.856	0.888	0.954
02	06	0.959	0.646	0.958	0.970	0.624	0.830	0.964
02	07	0.990	0.990	0.990	0.997	0.851	0.848	0.964
02	08	0.977	0.967	0.979	0.988	0.708	0.796	0.969
02	09	0.982	0.982	0.982	0.991	0.842	0.856	0.970
02	10	0.952	0.853	0.964	0.520	0.820	0.890	0.968
02	11	0.274	0.000	0.000	0.092	0.756	0.776	0.954
02	12	0.988	0.989	0.989	0.996	0.875	0.844	0.967
02	13	0.994	0.988	0.988	0.996	0.713	0.854	0.513
02	14	0.986	0.989	0.989	0.951	0.476	0.782	0.987
03	01	0.987	0.982	0.982	0.976	0.758	0.879	0.887
03	02	0.996	0.991	0.991	0.997	0.810	0.753	0.871
03	03	0.986	0.988	0.988	0.996	0.853	0.892	0.959
03	04	0.988	0.984	0.984	0.996	0.643	0.716	0.411
03	05	0.987	0.989	0.989	0.996	0.869	0.910	0.000
03	06	0.988	0.991	0.991	0.997	0.686	0.787	0.911
03	07	0.657	0.000	0.000	0.125	0.680	0.650	0.954
03	08	0.988	0.990	0.990	0.993	0.774	0.907	0.967
03	09	0.000	0.000	0.000	0.000	0.743	0.799	0.944
03	10	0.994	0.991	0.991	0.997	0.745	0.504	0.962
03	11	0.986	0.989	0.989	0.996	0.628	0.642	0.639

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
03	12	0.986	0.978	0.978	0.982	0.835	0.899	0.964
03	13	0.995	0.990	0.990	0.811	0.805	0.843	0.985
04	01	0.973	0.679	0.975	0.971	0.912	0.912	0.957
04	02	0.656	0.000	0.000	0.251	0.713	0.822	0.781
04	03	0.358	0.000	0.000	0.244	0.754	0.528	0.954
04	04	0.993	0.438	0.976	0.998	0.873	0.899	0.967
04	05	0.987	0.987	0.987	0.996	0.744	0.770	0.950
04	06	0.977	0.972	0.972	0.959	0.825	0.595	0.959
04	07	0.972	0.990	0.990	0.997	0.724	0.747	0.964
04	08	0.982	0.992	0.992	0.994	0.917	0.855	0.968
04	09	0.987	0.988	0.988	0.996	0.897	0.925	0.954
04	10	0.994	0.634	0.977	0.996	0.755	0.832	0.969
04	11	0.994	0.988	0.988	0.996	0.850	0.706	0.956
04	12	0.658	0.000	0.000	0.249	0.876	0.855	0.967
04	13	0.986	0.991	0.991	0.994	0.841	0.899	0.963
04	14	0.943	0.230	0.833	0.937	0.865	0.761	0.988
05	01	0.080	0.061	0.085	0.080	0.831	0.779	0.954
05	02	0.987	0.993	0.993	0.998	0.802	0.918	0.969
05	03	0.965	0.570	0.889	0.976	0.737	0.828	0.961
05	04	0.989	0.991	0.991	0.997	0.830	0.707	0.964
05	05	0.000	0.000	0.000	0.000	0.357	0.684	0.953
05	06	0.983	0.990	0.990	0.997	0.736	0.888	0.377
05	07	0.982	0.982	0.982	0.984	0.794	0.883	0.967
05	08	0.000	0.000	0.000	0.000	0.797	0.777	0.958
05	09	0.962	0.531	0.971	0.971	0.783	0.860	0.137
05	10	0.962	0.494	0.937	0.970	0.856	0.884	0.594
05	11	0.989	0.991	0.991	0.704	0.708	0.699	0.963
05	12	0.115	0.063	0.108	0.108	0.723	0.760	0.955
05	13	0.821	0.424	0.869	0.827	0.635	0.741	0.952
05	14	0.983	0.992	0.992	0.997	0.914	0.934	0.987
06	01	0.986	0.990	0.990	0.997	0.689	0.871	0.961
06	02	0.978	0.682	0.938	0.977	0.901	0.808	0.971
06	03	0.991	0.992	0.992	0.995	0.759	0.672	0.971
06	04	0.988	0.991	0.991	0.631	0.692	0.847	0.975
06	05	0.993	0.986	0.986	0.997	0.547	0.907	0.975
06	06	0.969	0.606	0.954	0.837	0.522	0.709	0.976

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
06	07	0.997	0.990	0.990	0.994	0.599	0.866	0.000
06	08	0.000	0.000	0.000	0.000	0.866	0.804	0.962
06	09	0.983	0.987	0.987	0.991	0.768	0.763	0.523
06	10	0.000	0.000	0.000	0.000	0.413	0.468	0.959
06	11	0.951	0.560	0.963	0.959	0.754	0.864	0.962
06	12	0.943	0.476	0.940	0.963	0.592	0.538	0.955
06	13	0.935	0.941	0.941	0.945	0.785	0.727	0.963
06	14	0.664	0.000	0.000	0.249	0.830	0.821	0.982
07	01	0.977	0.976	0.976	0.979	0.663	0.555	0.967
07	02	0.995	0.989	0.989	0.996	0.783	0.583	0.962
07	03	0.979	0.990	0.990	0.980	0.759	0.890	0.967
07	04	0.808	0.499	0.814	0.333	0.494	0.373	0.967
07	05	0.628	0.000	0.000	0.239	0.675	0.873	0.966
07	06	0.988	0.989	0.989	0.996	0.674	0.731	0.000
07	07	0.926	0.926	0.926	0.387	0.681	0.685	0.967
07	08	0.935	0.935	0.935	0.834	0.807	0.898	0.000
07	09	0.979	0.973	0.973	0.984	0.868	0.803	0.961
07	10	0.986	0.771	0.977	0.997	0.685	0.805	0.849
07	11	0.996	0.992	0.992	0.997	0.889	0.777	0.280
07	12	0.983	0.990	0.990	0.997	0.823	0.650	0.754
07	13	0.977	0.967	0.979	0.945	0.597	0.667	0.965
07	14	0.990	0.992	0.992	0.000	0.717	0.139	0.963
08	01	0.954	0.983	0.983	0.994	0.718	0.838	0.849
08	02	0.985	0.990	0.990	0.997	0.805	0.907	0.993
08	03	0.625	0.209	0.387	0.211	0.572	0.759	0.000
08	04	0.999	0.991	0.991	0.997	0.510	0.676	0.292
08	05	0.991	0.989	0.989	0.961	0.613	0.381	0.975
08	06	0.862	0.261	0.836	0.602	0.639	0.737	0.989
08	07	0.847	0.382	0.748	0.784	0.706	0.806	0.144
08	08	0.897	0.309	0.925	0.372	0.271	0.187	0.492
08	09	0.996	0.992	0.992	0.997	0.842	0.754	0.994
08	10	0.662	0.000	0.000	0.251	0.000	0.877	0.000
08	11	0.935	0.934	0.957	0.426	0.543	0.710	0.463
08	12	0.990	0.980	0.980	0.993	0.620	0.840	0.448
08	13	0.521	0.000	0.000	0.239	0.581	0.785	0.330
09	01	0.284	0.104	0.104	0.112	0.685	0.812	0.150

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
09	02	0.646	0.000	0.000	0.244	0.607	0.594	0.975
09	03	0.788	0.979	0.979	0.993	0.644	0.801	0.963
09	04	0.600	0.199	0.578	0.159	0.806	0.440	0.975
09	05	0.989	0.989	0.989	0.690	0.875	0.937	0.669
09	06	0.616	0.000	0.000	0.234	0.695	0.858	0.421
09	07	0.803	0.801	0.801	0.368	0.822	0.746	0.873
09	08	0.576	0.000	0.000	0.210	0.636	0.767	0.851
09	09	0.973	0.991	0.991	0.997	0.776	0.762	0.241
09	10	0.991	0.989	0.989	0.996	0.646	0.677	0.967
09	11	0.985	0.992	0.992	0.139	0.667	0.747	0.984
10a	01	0.954	0.991	0.991	0.638	0.785	0.869	0.895
10a	02	0.652	0.000	0.000	0.247	0.507	0.539	0.983
10a	03	0.875	0.867	0.867	0.883	0.481	0.755	0.874
10a	04	0.986	0.991	0.991	0.994	0.906	0.806	0.000
10a	05	1.000	0.962	0.962	0.253	0.001	0.652	0.000
10a	06	0.745	0.401	0.734	0.333	0.529	0.617	0.995
10a	07	0.994	0.994	0.994	0.998	0.644	0.893	0.320
10a	08	0.881	0.526	0.821	0.435	0.662	0.817	0.000
10a	09	0.991	0.986	0.986	0.406	0.811	0.807	0.000
10a	10	0.631	0.000	0.000	0.257	0.792	0.887	0.994
10a	11	0.926	0.413	0.903	0.325	0.426	0.349	0.990
10a	12	0.937	0.966	0.966	0.000	0.769	0.761	0.000
10a	13	0.657	0.000	0.000	0.250	0.777	0.900	0.031
10a	14	0.649	0.000	0.000	0.258	0.852	0.845	0.997
10a	15	0.981	0.980	0.980	0.987	0.851	0.853	0.974
10a	16	0.983	0.983	0.983	0.994	0.463	0.666	0.970
10a	17	0.651	0.000	0.000	0.497	0.705	0.803	0.000
10b	01	0.973	0.992	0.992	0.000	0.611	0.667	0.680
10b	02	0.733	0.270	0.624	0.304	0.353	0.738	0.990
10b	03	0.650	0.000	0.000	0.496	0.640	0.629	0.988
10b	04	0.975	0.922	0.961	0.254	0.534	0.665	0.673
10b	05	0.661	0.000	0.000	0.248	0.898	0.921	0.992
10b	06	0.601	0.000	0.000	0.238	0.546	0.489	0.883
10b	07	0.889	0.702	0.977	0.981	0.800	0.832	0.987
10b	08	0.996	0.993	0.993	0.572	0.763	0.677	0.000
10b	09	0.864	0.738	0.825	0.426	0.685	0.837	0.985

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
10b	10	0.973	0.397	0.960	0.003	0.530	0.682	0.000
10b	11	0.635	0.000	0.000	0.265	0.789	0.798	0.430
10b	13	0.289	0.000	0.000	0.107	0.517	0.738	0.986
11	01	0.659	0.000	0.000	0.249	0.727	0.739	0.926
11	02	0.649	0.000	0.000	0.247	0.699	0.749	0.825
11	03	0.982	0.962	0.982	0.984	0.752	0.649	0.000
11	04	0.974	0.641	0.976	0.141	0.726	0.744	0.809
11	05	0.661	0.000	0.000	0.500	0.768	0.868	0.978
11	06	0.901	0.442	0.844	0.286	0.625	0.784	0.325
11	07	0.867	0.332	0.682	0.576	0.864	0.817	0.988
11	08	0.642	0.000	0.000	0.247	0.663	0.848	0.758
11	09	0.656	0.000	0.000	0.242	0.701	0.863	0.628
11	10	0.663	0.000	0.000	0.251	0.688	0.798	0.575
11	11	0.973	0.964	0.964	0.918	0.870	0.893	0.000
11	12	1.000	0.590	0.975	0.255	0.257	0.708	0.990
11	13	0.661	0.000	0.000	0.250	0.643	0.775	0.988
11	14	0.647	0.000	0.000	0.252	0.770	0.840	0.465
11	15	0.667	0.000	0.000	0.250	0.851	0.763	0.000
11	16	0.892	0.815	0.815	0.830	0.338	0.712	0.765
11	17	0.279	0.000	0.000	0.000	0.342	0.550	0.920
12	01	0.971	0.992	0.992	0.442	0.716	0.656	0.976
12	02	0.918	0.332	0.944	0.857	0.756	0.867	0.973
12	03	0.977	0.232	0.938	0.662	0.730	0.617	0.000
12	04	0.521	0.104	0.195	0.323	0.572	0.819	0.986
12	05	0.846	0.959	0.959	0.990	0.786	0.866	0.956
12	06	0.595	0.000	0.000	0.215	0.814	0.781	0.990
12	07	0.604	0.000	0.000	0.231	0.739	0.757	0.916
12	08	0.646	0.000	0.000	0.247	0.448	0.361	0.980
12	09	0.636	0.000	0.000	0.245	0.499	0.740	0.969
12	10	0.613	0.000	0.000	0.236	0.782	0.871	0.988
12	11	0.975	0.987	0.987	0.994	0.631	0.669	0.972
12	12	0.883	0.984	0.984	0.990	0.718	0.771	0.782
13	01	0.614	0.000	0.000	0.235	0.864	0.906	0.985
13	02	0.943	0.365	0.923	0.958	0.646	0.809	0.564
13	03	0.971	0.966	0.983	0.991	0.488	0.788	0.363
13	04	0.949	0.425	0.946	0.899	0.824	0.926	0.991

Continued on next page

Table A.2 – continued from previous page

album	song	F_m	CML_c	CML_t	Lab	RCO	segmQ	krRCO
13	05	0.961	0.995	0.995	0.878	0.607	0.811	0.082
13	06	0.978	0.583	0.986	0.995	0.756	0.702	0.695
13	07	0.975	0.503	0.962	0.924	0.774	0.832	0.994
13	08	0.995	0.994	0.994	0.998	0.706	0.629	0.961
13	09	0.636	0.000	0.000	0.244	0.556	0.633	0.989
13	10	0.900	0.473	0.865	0.682	0.542	0.754	0.810
14	11	0.958	0.989	0.989	0.989	0.758	0.813	0.963
14	01	0.995	0.992	0.992	0.997	0.874	0.938	0.932
14	02	0.920	0.783	0.928	0.922	0.916	0.945	0.796
14	03	0.985	0.992	0.992	0.647	0.571	0.785	0.967
14	04	0.972	0.986	0.986	0.991	0.626	0.639	0.996
14	05	0.976	0.985	0.987	0.896	0.885	0.937	0.977
14	06	0.984	0.372	0.975	0.986	0.495	0.812	0.249
14	07	0.982	0.994	0.994	0.998	0.854	0.854	0.988
14	08	0.397	0.000	0.000	0.165	0.796	0.833	0.820
14	09	0.935	0.900	0.985	0.991	0.479	0.628	0.770
14	10	0.647	0.000	0.000	0.249	0.761	0.817	0.993
14	11	0.967	0.989	0.989	0.996	0.488	0.637	0.863
14	12	0.385	0.000	0.000	0.161	0.790	0.805	0.897
14	12	0.758	0.910	0.910	0.920	0.588	0.722	0.728
14	12	0.534	0.000	0.000	0.234	0.627	0.844	0.053

ID	Title
01	TB-Please Please Me
02	TB-With The Beatles
03	TB-Hard Day's night
04	TB-Beatles For Sale
05	TB-Help
06	TB-Rubber Soul
07	TB-Revolver
08	TB-Sgt Pepper's Lonely Hearts Club Band
09	TB-Magical Mistery Tour
10a	TB-The White Album cd1
10b	TB-The White Album cd2
11	TB-Abbey Road
12	TB-Let It Be
13	RW-Life Through A Lens
14	RW-I've Been Expecting You

Table A.1: Album IDs

Bibliography

[1] Richard Bellman and Robert E Kalaba. *Dynamic programming and modern control theory.* Academic Press New York, 1965.

[2] Juan P Bello, Chris Duxbury, Mike Davies, and Mark Sandler. On the use of phase and energy for musical onset detection in the complex domain. *Signal Processing Letters, IEEE*, 11(6):553–556, 2004.

[3] Juan P Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *Proc. ISMIR*, pages 304–311, 2005.

[4] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *Speech and Audio Processing, IEEE Transactions on*, 13(5):1035–1047, 2005.

[5] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89:425, 1991.

[6] Benoit Catteau, Jean-Pierre Martens, and Marc Leman. A probabilistic framework for audio-based tonal key and chord recognition. In *Advances in Data Analysis*, pages 637–644. Springer, 2007.

[7] Wei Chai. *Automated analysis of musical structure.* PhD thesis, Massachusetts Institute of Technology, 2005.

[8] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms.* MIT press, 2001.

[9] Matthew Davies. Mirex 2009 audio beat tracking evaluation: Davies, robertson and plumbley, 2009.

[10] Matthew EP Davies, Norberto Degara, and Mark D Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen*

Mary University, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06, 2009.

[11] Matthew EP Davies and Mark D Plumbley. A spectral difference approach to downbeat extraction in musical audio. In *Proc. EUSIPCO*. Citeseer, 2006.

[12] Matthew EP Davies and Mark D Plumbley. Context-dependent beat tracking of musical audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(3):1009–1020, 2007.

[13] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[14] Takuya Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proc. ICMC, 1999*, pages 464–467, 1999.

[15] Borko Furht. *Handbook of multimedia for digital entertainment and arts*. Springer, 2009.

[16] Masataka Goto and Yoichi Muraoka. Music understanding at the beat level: Real-time beat tracking for audio signals. *Computational Auditory Scene Analysis*, pages 157–176, 1998.

[17] Masataka Goto and Yoichi Muraoka. Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions. *Speech Communication*, 27(3):311–335, 1999.

[18] Emilia Gómez Gutiérrez. *Tonal description of music audio signals*. PhD thesis, Citeseer, 2006.

[19] Stephen Webley Hainsworth. *Techniques for the automated analysis of musical audio*. PhD thesis, University of Cambridge, 2003.

[20] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 66–71, 2005.

[21] Christopher A Harte and Mark B Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society*, 2012.

[22] Patrik N Juslin, John A Sloboda, et al. *Music and emotion*, volume 315. Oxford University Press New York, 2001.

[23] Jimmy Kachulis and Jonathan Feist. *The Songwriter's Workshop: Harmony*. Berklee PressPublications, 2005.

[24] Maksim Khadkevich. *Music signal processing for automatic extraction of harmonic and rhythmic information*. PhD thesis, University of Trento, 2011.

[25] Anssi P Klapuri, Antti J Eronen, and Jaakko T Astola. Analysis of the meter of acoustic musical signals. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):342–355, 2006.

[26] Carol L Krumhansl. *Cognitive foundations of musical pitch*, volume 17. Oxford University Press New York, 1990.

[27] Randal J Leistikow. *Bayesian modeling of musical expectations via maximum entropy stochastic grammars*. PhD thesis, Citeseer, 2006.

[28] Paul Masri. *Computer modelling of sound for transformation and synthesis of musical signals*. PhD thesis, University of Bristol, 1996.

[29] Matthias Mauch. Automatic chord transcription from audio using computational models of musical context. 2010.

[30] Matthias Mauch and Simon Dixon. Simultaneous estimation of chords and musical context from audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(6):1280–1289, 2010.

[31] Matthias Mauch, Simon Dixon, and Queen Mary. A discrete mixture model for chord labelling. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 45–50, 2008.

[32] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, 2002.

[33] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie. An end-to-end machine learning system for harmonic analysis of music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1771–1783, 2012.

[34] Laurent Oudre, Yves Grenier, and Cédric 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.

[35] Hélene Papadopoulos and Geoffroy Peeters. Simultaneous estimation of chord progression and downbeats from an audio file. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 121–124. IEEE, 2008.

[36] Vincent Persichetti. *Twentieth-century harmony: creative aspects and practice.* WW Norton New York, NY, 1961.

[37] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[38] Walter Andrew Schloss. *On the automatic transcription of percussive music: from acoustic signal to high-level analysis.* Number 27. Stanford University, 1985.

[39] Arun Shenoy and Ye Wang. Key, chord, and rhythm tracking of popular music recordings. *Computer Music Journal*, 29(3):75–86, 2005.

[40] John A Sloboda. Music structure and emotional response: Some empirical findings. *Psychology of music*, 19(2):110–120, 1991.

[41] Julius O. Smith. *Spectral Audio Signal Processing.* accessed <date>. online book.

[42] JA Speck, EM Schmidt, BG Morton, and YE Kim. A comparative study of collaborative vs. traditional annotation methods. *ISMIR, Miami, Florida*, 2011.

[43] Gregory H Wakefield. Mathematical representation of joint time-chroma distributions. In *International Symposium on Optical Science, Engineering, and Instrumentation, SPIE*, volume 99, pages 18–23, 1999.

[44] Keith Wyatt and Carl Schroeder. *Harmony and Theory: a comprehensive source for all musicians.* Musicians Inst Press, 1998.

[45] Yi-Hsuan Yang and Homer H Chen. *Music Emotion Recognition.* CRC Press, 2011.