Studying the effects of bass estimation for chord segmentation in pop-rock music

Urbez Capablo Riazuelo

MASTER THESIS UPF / 2014

Master in Sound and Music Computing

Master thesis supervisor:
Dr. Perfecto Herrera
Department of Information and Communication Technologies
Universitat Pompeu Fabra, Barcelona

UNIVERSITAT POMPEU FABRA

Abstract

In this thesis we present the research work we have carried out on bass line extraction and transcription and audio segmentation for chord estimation. First, we introduce the task and we define important musical concepts to be taken into account. Next, we review the scientific background to our work, both in the chord estimation and the bass extraction/transcription tasks. We then present a set of modifications to the Essentia's predominant melody algorithm to adapt it for bass line estimation. Bass information along with beat positions are then used to propose a novel type of audio segmentation for chromagram smoothing related to the chord estimation problem. Next, we present the evaluation methodology, music collections and metrics used in our research, followed by the evaluation results.

The results show a considerable improvement in the bass extraction task by using our approach and promising results in the bass transcription task. They also show very promising results regarding our novel audio segmentation method for chromagram smoothing, compared to the beat-synchronous chromagram approach used by current state-of-the-art algorithms.

The thesis concludes with the contributions of our dissertation, the challenges found during the research process and the future work.

Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to thank my tutor, Perfecto Herrera, for his valuable guidance and patience. I also would like to thank Emilia Gómez and Agustín Martorell for their advice in the field of music analysis. Thank you Sergi Jordà for believing in me and Xavier Serra, for allowing me to discover this awesome world of sound and music technology. Thanks also to Justin Salamon and Matti Ryynänen for their mail collaboration and to Nicolas for being english. I'm sure that I'm forgetting people, my apologies in advance.

Finally, I would like to thank the people who love me and have helped me in this long path. Thank you for your energy and support.

CHAPTER 1

Introduction

1.1 Motivations

Music has a special interest in our society: not only in the consumerism aspect (almost every person listens to music), but also in the learning process and obviously in the creation process. With the growth of new technologies, even musically untrained people are able to create music using computer tools such as sequencers. They are also capable of discovering new music by using recommendation technologies which appear already everywhere. Society is even interested in aspects related to music theory. An example of this is the number of tutorials on the web, specially focused on piano and guitar, to teach how to play chords. Many of the musical aspects that people are interested in are related with music analysis and in the sound and music computing community they are studied in order to understand them from a computational point of view. By doing so, we can create tools to democratize the music. In the special case of harmony, a chord estimator could help people to reproduce any popular song. But why is the automatic chord estimation field important and more precisely, why words such as automatic and chords could be interesting for researchers?

1.1.1 Automated tasks

The transcription of certain musical aspects such as the chords or the melody of a song involve a great effort for a human being. Moreover, these kinds of tasks cannot be done by an untrained person since a deep musical training is needed to accomplish them. Sometimes, the ambiguity or the complexity of a piece can even lead to different results in the analysis done by two different people. Finally, the amount of musical works grows day by day and music collections are increasingly larger.

Therefore, the automation of transcription tasks can be very beneficial. The main issue about this is that the estimations have to be reliable or at least have a reasonable degree of reliability. A big effort is being made to this end in the research community, reflected by the number of publications on this topic, by trying to develop new algorithms that are more robust and with a better accuracy.

1.1.2 Harmonic description

The importance of harmony in music can be understood by looking at two very specific musical examples.

The first one is related to jazz music, where the standards represent the main repertoire of this genre. It's very interesting to see which kind of musical information is included in the score: only the melody and the chord symbols. Musicians who have never met before are capable of playing together a jazz standard in a jam session with only that little piece of information.

Figure 1.1: A typical jazz standard score

The second example is related to a very popular format on the internet for describing a song. It is normally used for pop-rock music and addressed to guitar and piano players. Only the lyrics and the chord labels are included and the musician is supposed to be able to play the song just with that information.

These two examples show how powerful is the harmonic description and especially the chord progressions.

1.1.3 Mid-level descriptors in Music Information Retrieval

Music Information Retrieval (MIR) is a very important field in audio and music computing. Its main objectives are to analyze and extract musical content or context related to music to perform various tasks such as recommendation, categorization, search or transcription, among others. MIR systems are able to extract information directly from the audio signal which is used afterwards in many tasks. For instance, in recommendation tasks, the mel-frequency cepstral coefficients (MFCCs) have been used widely [52, 60]. However these types of low level features are not interpretable or, in other words, they have no meaning for a human being. On the other hand, higher level features such as pulse or tonality are very well understood by us. The latter features are being used more and more in MIR with success. Harmonic transcription due to its intrinsic characteristics can offer a good

framework for many of those tasks such as genre classification [1] or cover song detection [2].

1.2 Thesis goals

With this thesis, we expect to fulfill the following goals:

- Provide a scientific background and a literature review in the field of automatic chord estimation and bass line extraction.
- Study the role of bass line and beats in the automatic chord estimation problem.
- Develop a new method for audio segmentation to enhance audio chord estimation based on bass notes and beat positions.
- Modify Essentia's predominant melody algorithm to improve its performance in the bass transcription task.
- Provide comparative evaluation of our approaches with respect to other algorithms

In chapter 2, we review perceptual and musical aspects and definitions which are important to work on the chord estimation problem. Important vocabulary and concepts are explained to better understand the content of this thesis.

Chapter 3 presents the scientific background and literature review in the fields of automatic chord estimation and bass line extraction. We describe the general architecture of current state of the art systems and we identity possible lines of research.

In chapter 4 we describe the methodology we have followed during the process of the thesis. First we present how the bass lines extraction and transcription was

carried out using part of the Essentia's predominant melody algorithm: the parameter choice and further modifications. Then we describe different approaches to segment the audio based on bass notes and beat and finally we discuss the evaluation methodology: music collections, evaluation metrics and the data preparation.

Chapter 5 contains the results of the evaluation of the bass line algorithm and the audio segmentation tool with a concluding commentary.

The last chapter ends the thesis by noting the contributions accomplished, the challenges encountered and ideas for the future work.

CHAPTER 2

Musical Definitions

2.1 Introduction

A basic understanding of music theory and human perception is necessary to work in automatic harmonic description. In this chapter, some knowledge about pitch, chords and musical context will be given as an introductory stage for automatic chord estimation. Indeed, understanding how music works should be useful to improve or develop new algorithms in this field.

2.2 Pitch, pitch class and note

Pitch is a perceptual property which allows the ordering of sounds on a frequency-related scale extending from low to high [45]. In other words, it is approximately proportional to log-frequency. A note corresponds to the musical representation of the pitch.

The fundamental frequencies of the notes of a chromatic scale in equal temperament, which divides the octave equally in twelve, can be defined as

$$f_p = 2^{1/12} f_{p-1}$$

where f_p is the fundamental of a note and f_{p-1} is the fundamental frequency of the previous note. The octave is an interval between two notes (the distance between them) which has a frequency ratio of 2:1.

The work by Shepard related to pitch perception is very interesting. He conducted a perceptual study with humans and found out that "human beings are able to separately consider pitch class, which refers to note names with no octave information, and pitch height, which is proportional to the fundamental frequency of a note. In particular they are able to perceive notes that are in octave relation as equivalent, a phenomenon called octave equivalence" [38].

The perceptual concept of octave equivalence has its own analogy in the use of chords in music theory: in terms of chord label all the combinations of notes that have the same pitch classes are considered equivalent, with the exception of the position of the bass note.

2.3 Harmony and chords

The new Grove dictionary of music and musicians provides the following definition about chords: a chord is "the simultaneous sounding of two or more notes. Chords are usually described or named by the intervals they comprise, reckoned either between adjacent notes or from the lowest".

Indeed, chords are normally constructed by three notes with different names (i.e E, G and B) which are called triads although they can be composed by other combinations. It is possible to construct more complex chords with four (tetrads) or even more notes. In the other hand, two notes sounding simultaneously can also be

considered as a chord, but sometimes it can be difficult to determine its name or its function and normally some context is needed to do that. Figure 2 shows a chord progression in the key C major. Letters represent the label of the chords and Roman numerals represent the function of the chord inside the key.

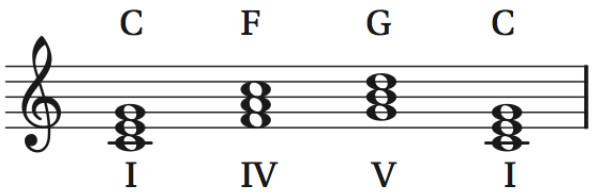


Figure 2.1: chord progression in C major

The easiest and the most common way to determine the label of a chord is to sort its notes by intervals of thirds and check what the intervalic relationships are between them. Occasionally, the notes are not sorted in this way because the chord is not in the root position (i.e. the main note of the chord, also called the tonic, is in the bass position) but it's inverted (another note of the chord is in the bass position). The later case, as it will be seen in the next section, can make the chord identification task much harder.

Although chords are very informative about harmonic content, the harmonic information can also be inferred by other means such as melodic structures which evolve in time. The most common is called an arpeggio: a sequence of close notes in pitch which are perceived with certain continuity and transmit harmonic content depending on their intervalic relationship. Indeed, a human being is able to integrate perceptually sequences of notes which are interpreted as intervals and consequently as chords [21].

The harmonic content description and particularly chord transcription can be sometimes complicated for different reasons. For instance, chords are not necessarily represented in the low-level content of a signal and a certain level of abstraction is needed to complete the task. This is the case for notes which do not

belong to a chord and they can be a source of confusion when labeling manually or automatically that particular chord. In other cases, critical information is missing (i.e. the third note of the chord which determines if it's a major or minor) and a deeper analysis is needed (context analysis, for instance).

Chord transcription is not only about transcribing notes but to understanding and interpreting the musical content and context and generating a higher level meaning.

2.4 Musical context

The different musical characteristics such as tonality, rhythm, harmony, structure or texture are not isolated elements with no interaction between them. What is normally called a composition is the art of using and combining those musical aspects to transmit emotions, sensations or even a message. It has been proved that they can also be used in a random way (i.e. *Music of changes* by John Cage) or in a very different way conceived within the occidental music tradition (i.e. *Mode de valeurs et d'intensités* by Olivier Messiaen or total serialism) but it's well known that tradition is very strongly rooted in popular music.

Popular music is generally under the parameters of tonality. Music theory says that within a tonality, chords have specific functions and follow certain rules (figure 2) and some chords are more likely to occur than others. Lerdahl [20] discusses harmony extensively. He relates chords to keys and describes theoretically their relationships and dependencies. Moreover, research in music perception has shown that those relationships in music theory have a parallelism in human perception and cognition of chords progressions. Harmonic priming studies show that human perception of chords is more accurate if they are harmonically close to their context [46].

Thompson [47] also conducted perceptual experiments and showed that chords are perceived at the same time and globally with key and melody, in a hierarchical way, in which the three qualities are connected by expectation.

Harmony and chord progressions are also strongly related to rhythm. Ledahl and Jackendoff discussed widely rhythm in music and metrical structure in [48]. They described and formalized what is called in music theory the harmonic rhythm. The harmonic rhythm represents how fast the harmonic changes occur and it is closely related with the pulse of a musical piece.

2.4.1 Bass and harmony

The bass line deserves a separate section for itself. Bass lines are very connected to chord changes. In music theory, there is the concept that chord is always built on top of the bass note, which is the most important note of the harmony. This is reflected in musical sight reading manuals [74]. Sight reading in music means that the performer plays a score that he has never seen before. Obviously, it implies an extreme difficulty but there are general rules to help the musicians. One of them shows the importance of the bass in harmony: it is permissible to skip notes if the score is very difficult, but the bass note always has to be played.

Bass lines in popular music are also very important and closely related with chord changes. Indeed, in this type of music the bass note is almost always present on the first beat of a chord and it is very unlikely that a chord changes without a bass note not being played. This is confirmed by one of the most popular bass player tutorials [49] which describes 207 example bass patterns covering styles such as Blues & R'n'B, Soul, Funk, and Rock and showing only 20 which do not start with the bass pitch class.

CHAPTER 3

Scientific Background

3.1 Introduction

In the following sections, we will provide the scientific background underlying the work carried out in this research. The chapter is divided in two main sections: the chord estimation article review and the bass line extraction and transcription article review. The later section is important since the bass information is an essential feature for the work we are doing in the thesis. The chapter finishes with a discussion about the studied field and the description of the goals of our research.

3.2 Automatic chord estimation (ACE)

In this section, we review the articles related to automatic chord estimation and we describe their approaches to the problem. In general, all the algorithms follow a general structure. The main differences between them is how they solve the different steps involved in the process. First, the systems need to extract a representative harmonic feature from the audio for every frame analyzed. Then, a

profile matching is performed to estimate the chances that a chord is present in every frame and finally a mid-level transition model is used to smooth and define the final estimated chord. Figure 3.1 shows the most common structure of a chord detection algorithm, which will be reviewed in the following sections.

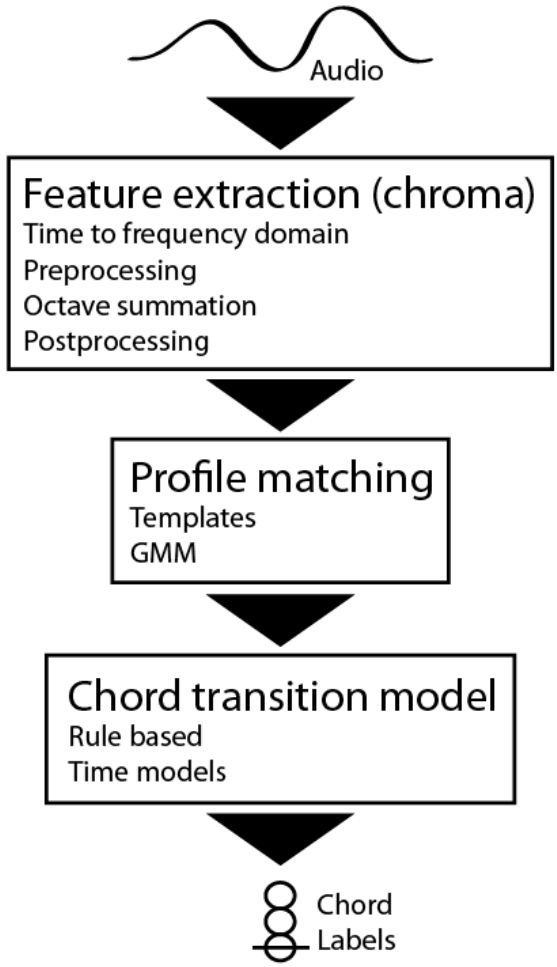


Figure 3.1: A typical chord estimation algorithm structure

3.2.1 Feature extraction

Analyzing music from a score requires high amount of musical training but has an obvious advantage: the elements for defining musical aspects of the piece are normally written down (with the exception of artistic performance modifications).

The key can be inferred from the chords and the notes, and the chords can be defined by the notes and their disposition and so on.

When dealing with audio, another approach has to be taken since musical notes can't be inferred directly from the audio signal. Pitch detection from polyphonic signals, onset detection and pitch duration estimation are active research fields and unsolved problems [59]. The main difficulty of this task remains in the spectrogram analysis, which represents not only the fundamental frequencies of the notes but also the related frequencies (upper partials) and also to assign every spectral peak to the correct note. Moreover, percussive sounds generate a big amount of noise in the frequency domain and even audio signals can be imperfect and noisy.

Therefore, to avoid this transcription stage, the most common used representation of the audio in automatic chord estimation is the chromagram. The first author mentioning the chroma representation was Babbitt [68]. Then, Shepard stated that two dimensions could help in understanding how human auditory system works [38]: tone high and chroma (pitch class). In Music Information Retrieval, the chromagrams can be computed in different ways but generally speaking, all of them describe the salience of every pitch class over time.

The first author expressing concerns with chord transcriptions from real audio was Fujishima. In his work [10], he used a feature called pitch class profiles (PCP): a twelve dimensional vector created by wrapping the whole spectrum to a unique octave. Indeed, the PCP has been used as a chromatic representation of the harmonic content of the audio by Gómez [11], Bello [3], Harte [16], among others.

3.2.1.1 Chromagram computation

Time to frequency domain transformation

The waveform of an audio file is not directly very informative about the harmonic content of the signal. A more adequate representation of the sound is needed to perform harmonic analysis tasks. It is well known that the human auditory system performs a transformation at the cochlea level to the frequency domain. Similarly, the sound and music computing community has used the Fourier transform to analyze harmonic content from audio.

The most common transformation in automatic chord estimation, especially during the first years of research in this field, was the Short Time Fourier Transform (STFT). Since the interest was focused on determining local harmonic variations, it seemed more appropriate to compute the frequency magnitudes using a sliding window across the signal. One limitation of this technique is that it uses a fixed-length window, which involves a trade-off between temporal and frequency resolution [54].

Another popular time to frequency transformation is the constant Q transform which has been used increasingly in the past years. This transformation is a spectral analysis where frequency-domain bins are not linearly spaced, as in DFT-based analysis, but logarithmically spaced, and consequently closely resembling the frequency resolution of the human ear [3].

As small amount of papers have used other types of transformations. Wavelet transform has similar properties to constant Q, giving a better resolution for both low and high frequencies [39] and enhanced autocorrelation offers a good trade-off between complexity and quality [44].

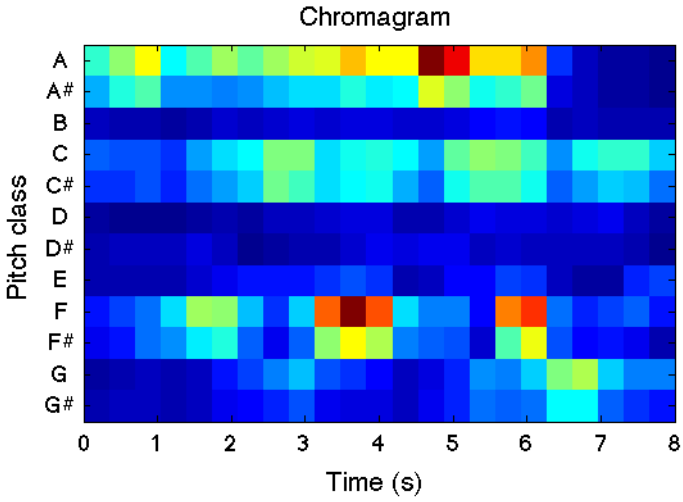


Figure 3.2: The twelve-dimensional chromagram shows the evolution over time of the energy corresponding to the different pitch classes.

Preprocessing techniques

The audio signal can be composed by a variety of instruments, sounds and even noises. It's obvious that not all the information in the spectrogram has interest for describing harmonic content. For instance, a percussive sound can generate energy along the entire spectrum and maybe this energy should be filtered.

As a general rule of thumb, the frame rate of the chroma analysis has to be faster than the rate of chord changes in a piece of music [9]. When using short windows, several disadvantages can arise. For instance, the frames of the resulting chromagram could respond to changes too locally and therefore become too sensitive to noise or transients. Therefore, the most common approach is to use a low pass filter for smoothing purposes [29].

Filtering background noise

Another goal of preprocessing the spectrogram is to deal with the background spectrum (parts of it which are not useful for a harmonic description). A typical filter for this purpose is the median filter, which has been used extensively in chord

description [22], [35]. Its use is mainly to discard outliers.

A particular case of useless background spectrum is energy from percussive sounds. Some efforts were made in this direction by trying to remove parts of the spectrum which were generated by percussion [26]. This method is known as Harmonic Percussive Source Separation (HPSS). It considers the spectrum to be a sum of two spectra -the harmonic and the percussive- and performs the chroma computation only from the harmonic one.

Another noise removal technique was researched by Khadkevich et al. related to chord estimation: the time-frequency reassignment [19]. The main idea behind TFR is to remap spectral energy of each spectrogram cell into another cell that is the closest to the true region of support of the analyzed signal: "blurred" spectral representation becomes "sharper" which increases frequency resolution.

Harmonics

During the preprocessing stage, it's also worth to mention the numerous techniques to deal with harmonics in automatic chord estimation. For musical instruments, not only the f0 is played but also a series of harmonics (partials, for inharmonic instruments). Those harmonics can sometimes confuse the feature extraction and therefore some researchers have tried to filter them.

One method designed to only take into account meaningful harmonics is called harmonic pitch class profiles (HPCP). This method takes into account high energy frequencies which are considered to be harmonics of sub-fundamentals [11].Other methods used for this purpose are based on multi-pitch tracking techniques. Different proposals have been made related to this [43], [36] but the main idea behind it is to detect meaningful pitches in the spectrogram. A more recent work proposed to approximate the spectrum by a linear combination of note spectra [21], namely each of them being a f0 frequency plus four harmonics.

Timbre invariance

The exact two notes, played in the same way by two different instruments have a different spectrum: the energy of each of the harmonics can vary dramatically. Therefore, the effect of the timbre can affect the chromagram computation, depending of the content of the audio signal. The HPCP method has a preprocessing stage called spectral whitening [11] aimed to reduce the impact of timbre. A more recent proposal is to use the envelope information given by the MFCCs to normalize the chroma-vector [24].

Perceptual loudness

The last approach in the preprocessing stage to improve the pitch salience detection is to approximate the frequency information from the spectrogram to the human auditory system. This has be done by weighting the spectrum by an arc-tangent function in the context of key detection [32]. The loudness-based chromagrams also try to simulate the perception of loudness by humans. This has been approximated by doing a logarithmic compression [24] and also by doing a A-weighting of the spectrum [26].

Tuning

The International Organization for Standardization adopted A4 = 440Hz to be the standard tuning in 1955. Nevertheless, some musics for different reasons don't use this kind of tuning (i.e. baroque music sometimes uses A415).

In popular music, this was noticed by Sheh and Ellis when developing an algorithm for chord detection and segmentation [37]. They computed a spectrogram with a higher frequency resolution (half semi-tone resolution) to be able to adapt the tuning.

This method was then improved by Harte [17], by computing a finer spectrogram with a three band resolution per semi-tone, and looking for the energy maximization depending of the tuning selected. His technique has been used by ulterior research works [3], [18].

Octave summation

The next step for processing the chromagram is to sum the energies or pitch saliences corresponding to every pitch class, followed by a normalization. This stage is commonly known as wrapping the spectrum to an octave. Indeed, in doing so, the octave information is rejected, which in chord estimation is normally seen as irrelevant.

The resulting feature is a twelve dimensional vector (one for each pitch class) which represents the harmonic content of a frame and the chromagram is a matrix containing those vectors over time (each column being a frame).

The first approach in the automatic chord estimation field was to warp the spectrum in only one chroma [10], [37]. Since 2008, there has been an increasing trend of computing two different chromagrams to improve the estimation [36]: one for the bass and the other for the treble. This is justified because the root of a chord can give additional information and lead to an increase of precision when identifying chords [21].

Post-filtering

Post-filtering techniques are usually used for smoothing purposes. Since chord changes are more likely to happen on beats [13], a very common practice is using a beat tracker to determine beat positions and smooth the chromagram between

them by using the mean [3] or the median [22]. These chromagrams are called beat-synchronous. Recently, Bello proposed another post-processing method based on the use of recurrence plots [7].

3.2.1.2 Another feature proposal: Tonal centroid

The most serious feature presented as an alternative of traditional chromagrams was proposed by Harte et al., notably a transformation of the chromagram known as the Tonal Centroid feature [18]. The main idea behind this feature was to design a representation where harmonic relationships such as perfect fifths and major and minor thirds had a closer Euclidean distance than in the traditional chromagram. The hypothesis was that this feature could lead to an improvement in detecting harmonic change. Therefore, they mapped the twelve-dimensional chroma onto a six dimensional hypertorus which corresponds to Chew's spiral array model [6]. This representation has been used in posterior works as a unique feature and also combined with the traditional chromagram [5].

3.2.2 Profile matching

Once the chromagram is computed, the next step consists in estimating which chord could be sounding in each frame. To this end and to our knowledge, most of the chord detection algorithms use two different systems. The first one is known as template matching and the second one is normally based on a machine learning model called Gaussian Mixture Models (GMM) which uses labeled data (in this case, labeled chroma vectors) to learn and determine the parameters of the model.

3.2.2.1 Template matching

Estimating the similarity between an analyzed chroma vector and a twelve-dimensional reference template is called template matching. These templates describe a chord according to the importance of each pitch profile and can be generated in different ways. Moreover, in the matching process, many formulas can be used to calculate the distance or similarity between the chroma and the template. Presumably, the chord template closer to the analyzed chroma feature should correspond to the right chord. For several reasons, this is not always the case and, in fact, a common procedure is to consider several candidates.

Templates

In the automatic chord estimation literature, multiple chord templates are described and used. Binary templates [10] are twelve-dimensional vectors where there is a one in the positions of the chord notes and zeros in the other positions. For instance, as figure 3.3 shows, a C major binary template would be [1,0,0,0,1,0,0,1,0,0,0,0].

Other works have considered and tested different chord templates. The most common approach consists in including a certain weight to the harmonics of the main notes of the chord [28]. It's assumed an exponentially decreasing spectral profile for the amplitudes of each partial: an amplitude of 0.6^{i-1} is added for the i^{th} harmonic of every note in the chord [11].

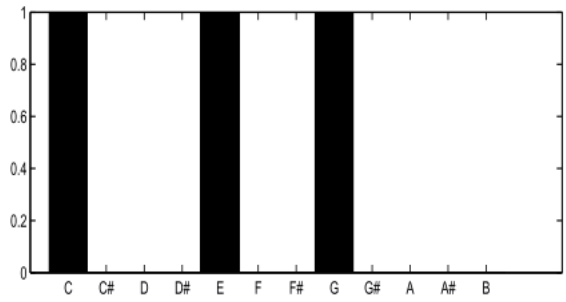


Figure 3.3: C major Binary template

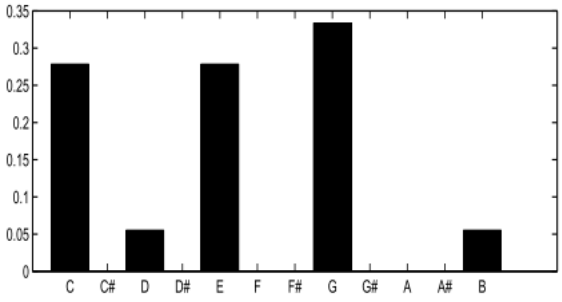


Figure 3.4: Template considering 4 harmonics

Distances

In the literature, the similarity or distance between chroma and templates has been measured in many different ways using different formulas. The most commonly used methods are inner product, euclidean distance and correlation value. Other measures have been less preferred such as Mahalanobis distance [41] and the Dirichlet linear distance [34].

There's a work in the literature which does a comparative between the Kullback-Leibler divergence, the euclidean distance and the Itakura-Saito divergence in the automatic chord estimation context and it gets the best results with the former one [27] but there is no study that considers and analyzes all of them with a clear conclusion about their performance.

3.2.2.2 Gaussian Mixture Models (GMM)

The Gaussian Mixture Model is used commonly in automatic chord estimation to determine what is the probability of an observed chroma vector to represent a certain chord.

A Gaussian Mixture Model is a parametric probability density function

represented as a weighted sum of Gaussian component densities [67]. GMMs are commonly used as a probabilistic distribution of features. This model is normally used in probabilistic frameworks as Hidden Markov Models to determine the probabilities of the chroma representing a certain chord [28].

3.2.2.3 Least common profile matching

In the early days of automatic chord estimation, Pardo used a rule-based method to do the profile matching [30]. Chords were represented as pitch classes, not pitches. The pitch class C was represented as 0, and the G major chord was then represented as <7 11 3> (for G, B, D). A score for a chord template over a segment was determined by the number of coinciding notes minus the number of notes that are false positives or false negatives. In the case of equal scores between chord templates, a tie break rule based on the chord root was applied.

In literature, there are some examples of the use of neural networks for chord recognition [39]. In those cases, the profile is done by the weight of the weights of the network nodes.

Finally, an interesting approach was taken by Chen et al. [5] by approximating the computed spectrum to a 24-dimensional vector using linear regression.

3.2.3 Chord Transition Modeling

3.2.3.1 No-musical context algorithms

Template based

There are some algorithms, especially in the early days of automatic chord estimation, which don't present a higher-level consideration for chord transition than the frame level. The most common approach in these cases is to analyze frame by frame the audio signal, perform the template matching and finally do some type of smoothing to deal with possible "noisy" chords [10], [39].

Transition modeling

As it has been said in the pre-processing chapter, analyzing real audio frame by frame is prone to detecting chord changes too often since the scope is set too locally. In general, harmony has a degree of stability which is much higher than the frame rate and therefore some type of smoothing is needed to correct very short term chord changes. To this end, most of the automatic chord estimation algorithms use probabilistic time series models.

Hidden Markov Models (HMM)

The most common model used is the hidden Markov model (HMM), which has been used widely in speech recognition. The main reason of using HMM in chord detection is because it models contiguous and non-overlapping events over time [21].

There are numerous examples in the automatic chord estimation literature which use hidden Markov Models without considering other musical aspects. Some of them train all the parameters of the HHM from the data [37], [28]. In other cases,

some musical experience is included in the system. This can easily be done at different levels. At the observation probabilities level, it has been done by constructing the multivariate Gaussian models using music theory [28]. At the transition matrix level, the probabilities to move to a chord from another one can be set up by also following music theory. One way of doing it is using the circle of fifths for a consonance measure [3].

Multi-stream HMM

This approach was presented in the MIREX 2013 edition [8]. In the next section, two-stream HMM will be discussed but basically every stream represents a succession of observable variables: one for bass sounds in particular and another one for treble sounds. The peculiarity of this work is the use of four simultaneous observation variables: a six octave spectrum is divided in four frequency bands and four chromagrams are computed and used as observations. No musical knowledge used by the algorithm is included in the report.

Other approaches

There are other strategies to model time series. Conditional random fields (CRFs) are a class of statistical modeling method often applied in pattern recognition and machine learning. This approach has been used in literature to try to model chord progressions [4]. Linear chain CRFs differ from the HMMs in that each hidden state depends not just on the current observation but on the complete observation sequence, which seems logical since chord changes in harmonic progressions are not only dependent on the last chord.

Other dynamic modeling strategies are found in the literature and in most of the cases are very specifically designed. Some of them have a perspective very similar to HMM by the use of bi-grams, also very used in speech recognition [36].

3.2.3.2 Musical Context Aware Algorithms

Algorithms without a machine learning approach

Hypothesis-based system

An interesting approach is given by Yoshioka [41] et al. with his hypothesis-based method. Their algorithm estimates chord labels and boundaries by generating dynamically hypothesis based on tuples of chords and keys. At every time unity (eighth-note level calculated with a beat tracker), a finite number of hypothesis are generated or expanded from last time position. To avoid an exponential increase of them, pruning is performed by evaluating them and keeping the most probable ones. Cues for calculating the likeliness of the hypothesis are chroma features, bass sounds and chord progression patterns, which introduce penalization factors for mismatching situations.

Voting-based model

This bottom-up approach by Sailer [50] et al. uses a voting system to decide which chord candidate fits the best in a chord progression. First of all, it performs a key estimation based on Krumhansl profiles. For each candidate from each frame, a triple voting is done: the first is related to the amplitude of the notes of a candidate (in the chroma) with respect to the maximum score, the second one is the temporal duration of the chord (its presence in the adjacent frames) and the final one is its fitness to the key. Finally, a filtering of short chords (less than 80ms) is performed.

Rule-based model

This system was introduced by Shenoy et al. [51] in 2006. The system uses rhythm and key information to improve chord estimation. First, a beat detection is performed, followed by an initial estimation of the chords. The chords are used afterwards to estimate the key in a symbolic way. Beat information and key is used

then to perform what the authors call chord accuracy enhancement which correspond basically to a filtering of the initial chord estimation by rejecting chords which don't fit in the context. Moreover, based on the double assumption that the most common measure is 4/4 and chord changes are more likely on measure changes [13], they detect the measure boundaries with chord information and perform an intra-measure chord check to filter the first estimation even more.

Chord/Key model

This method by Zenz [44] is a modular algorithm composed by a beat tracker, a key estimator, a chord detector and a chord sequence smoother. The beat tracker is used to segment the audio in big non-overlapping frames considering that the chord remains stable during one beat. The key detection performs a filtering of the least likely chords obtained from the template matching: it only considers diatonic chords, secondary dominants and secondary subdominants. The chord sequence smoother is not described properly in the paper but it is supposed to select the chord sequence with high probabilities for each single chord and few chord changes.

Lerdahl's distance-based model

Rocher [35] proposes a method based on concurrent key and chord detection. Two chromagrams with different time scope are computed: a short one to find chord candidates and a longer one to find key candidates. For each frame, the most likely chord and key candidates are selected and combined into a general harmonic candidate: a tuple of chord and key. To compute the transition cost between chords, the Lerdahl's distance is used, which takes into account the circle of fifths and the common notes of the chords [20]. The last step consists in finding the most likely path by minimizing the total sum of weights along the path leading to each candidate.

MPTrEE and HarmTrace

The MPTrEE model for automatic chord estimation is a very interesting system which uses as its cornerstone a parsing technology called HarmTrace [14]. The HarmTrace harmony model is explicitly designed for modeling the relations between chords over a long time span. It's is very interesting that it takes into account functional harmony, not only the name of chords. Actually, it performs a real harmonic analysis.

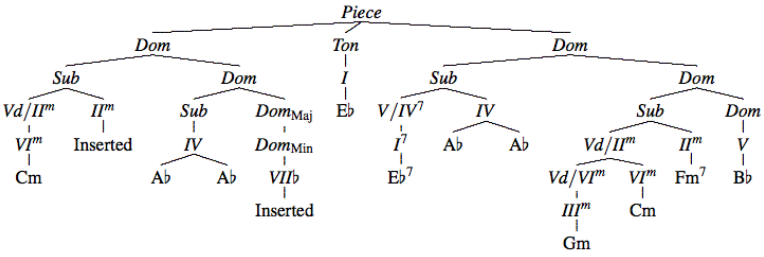


Figure 3.5: An excerpt of HarmTrace analysis, taken from [14]

This work uses a beat-synchronous chroma to determine fragments with stable keys, using a dynamic programming algorithm, and for every frame a possible list of chords. The matching is done using Krumhansl profiles for keys and binary templates for chords. The next step consists in grouping the frames to reduce the possible combinations. This is done by merging lists with chords in common. Moreover, the song is segmented where a tonic or a dominant is recognized because in the harmonic model, subtrees are rooted by a tonic or a dominant.

Once the context is settled, the chord candidates and the key are introduced in the HarmTrace harmony model: the selected sequence is the one having the lowest error-ratio.

Musical Probabilistic Time Series Models

Probabilistic hypothesis

Sumi et al. [40] resumes Yoshioka's idea by reformulating the hypothesis reliability using a probabilistic framework. Indeed, the relationship between musical facets, formerly treated with penalties, is now modeled with a probabilistic function. This function also integrates acoustic features (based on the likelihood of GMM), progression patterns (based on transition probabilities obtained from bi-gram models) and bass information (obtained using PreFest method [12]). The hypothesis-based model, with the pruning and the hypothesis expansion, remains the same.

Double-state HMM

In the 2.3.1.2 section, the HMM model was reviewed as a possible option to model chord transitions. If more musical aspects must be included in the model for a concurrent estimation, more hidden variables are needed. This is the case in the work by Papadopoulos et al. [29] where they wanted to estimate downbeats and chords at the same time from real audio.

The two hidden variables in this work are the chord label and the position of the chord in the measure. The observation probabilities depend only on the template matching and the musical information is all coded in the transition matrix. Again, the segmentation is done by beat-tracking at two levels: quarter and octave-note. The main assumption (inferred from the musical analysis of their corpus) is that chords are more likely to change on downbeat positions. It is also interesting that they consider imperfect beat detection (or beat addition or deletion) by building two global transition matrices: one for 3/4 measures allowing a fourth additional beat and another for 4/4 measures, allowing the deletion of the last one.

Explicit-duration HMM

The explicit-duration HMM used by Chen et al. [5] is similar to the one order HMM described in the previous section but includes another probability distribution in the formula: the duration distribution. It represents the probability of a state (a chord label) spanning a certain number of beats. In this case, smoothing and segmentation is also done at the beat level. The duration distribution is trained from data and it's considered to be global (there is not a particular distribution for each chord label). When computing the probabilities for each state, the algorithm keeps track of N previous states (in this case, N = 20) to compute the right duration probability. The final step consists of finding the most probable chord sequence using Viterbi's algorithm. In this work, not only one duration distribution is done: using clustering techniques, three possible distributions were found and used to improve the estimation.

Dynamic Bayesian Networks

To our knowledge, the work by Mauch [22] is the most complete in the literature if we talk about using synergies from multiple musical facets. Indeed, he uses a Dynamic Bayesian Network with discrete hidden nodes representing metric position of the chord, key, chord label and bass note. The continuous nodes model bass and treble chroma.

The conditional probability distributions (CPD) of random variables are used to define the interaction between keys, beats, bass and chords. For instance, as it can be seen in figure 3.6, the chord label depends on its metric position, the key and the previous chord. The metric positions are defined by the use of a beat tracker and the chromagrams are smoothed beat-wise. The model is built with musical knowledge and observations from musical repertoire. This knowledge is mapped in the probabilistic framework and doesn't have any training stage. It is also worth

noting that the algorithm assumes 4/4 measures but has a small consideration of possible deviations.

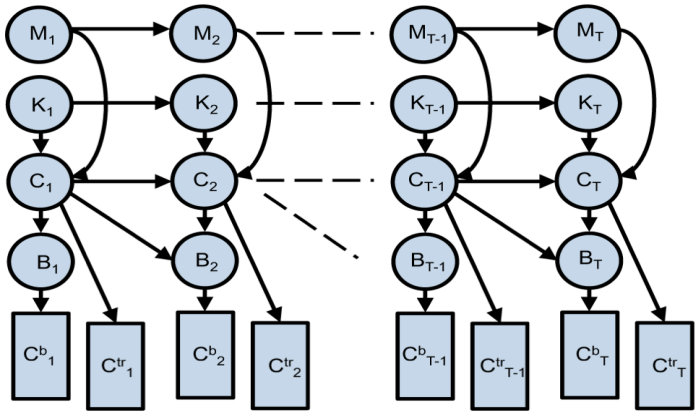


Figure 3.6: Mauch's Dynamic Bayesian Network. Hidden nodes M, K, C and B represent metric position, key, chord and bass. C^b and C^{tr} represent bass and treble chromagrams, taken from [23]

This algorithm is know as *Chordino* in the Sonic Visualizer community and it is available in the Isophonics web page (http://www.isophonics.net).

Harmonic progression analyzer

A harmonic progression HMM topology was proposed by McVicar et al. [23] containing three hidden and two observed variables. This method could be considered close to the one proposed by Mauch since it tries to model the context, even if the relationship between the variables is not exact. The hidden variables are the key, the chord labels and the bass.

The chord is actually decomposed into two aspects: chord label and bass note. The observed variables correspond to the bass chroma and the treble chroma, extracted with Mauch's system.

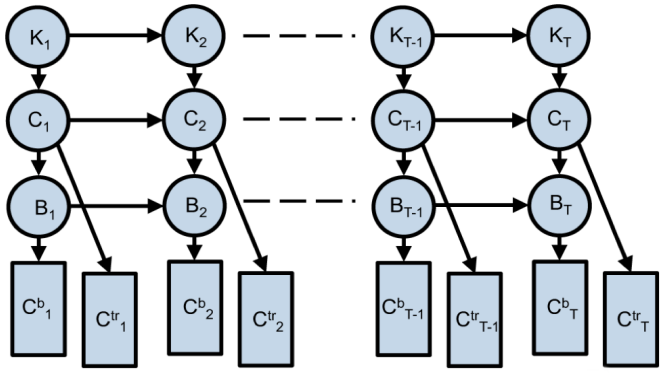


Figure 3.7: Harmony progression analyzer. Hidden nodes K, C and B represent key, chord and bass. C^b and C^{tr} represent bass and treble chromagrams, taken from [23]

As it can be seen in the figure 3.7, the current key depends on the previous key, the chord label transitions depend on the key and the previous chord and the bass transitions depend on the chord label and on the previous bass note.

To estimate concurrently chords, bass notes and keys, the most probable path has to be decoded by the Viterbi's algorithm. To improve computing performance, a search space reduction is done before running the whole algorithm: key, bass notes and chords alphabet are reduced using musical knowledge.

3.2.4 Chord estimation literature discussion

As it as been showed in the literature review, almost all the algorithms have the same general structure. Starting with a feature extraction, the chroma is then compared with precomputed profiles (profile matching) to approximate the chord label and finally a higher level model works on the transition of the chords. We have also seen that the strategies have evolved to the inclusion of musical

characteristics in the algorithms. The main assumption for that is that having musical knowledge would help the chord estimators to take better decisions when labeling the audio.

We have identified one aspect that is common to almost of the state-of-the-art algorithms which is the beat-synchronous chromagram. Synchronizing the chromagram with the beat is also called in the literature as smoothing it between beat positions. Basically, all the chroma frames within two beats are merged together with a function (mean and median are the most common). By doing this, noise and abrupt changes in the feature are avoided. Moreover, this idea is also supported by the fact that in pop-rock music, it is very unlikely that a chord changes between beats [29].

Our thesis will work on this direction. We will try to segment the audio in an adaptive way to optimized the effect of the chromagram smoothing. Therefore, we need additional information to work with, besides the beat positions, to create the segments in a coherent way. As we explain in section 2.3, the musical context in chord estimation is very important, especially the bass line. We need to review the bass estimation literature to follow our approach.

3.3 Bass estimation

In this section, we review the articles related to bass line extraction and transcription and we describe their approaches to the problem. It is important to note that the main difference between extraction and transcription is that the former tries to guess the correct note (or f0 in Hz) at the frame level and the latter transcribes the notes in terms of onset time, height and duration.

3.3.1 Salient function and Melodia

3.3.1.1 Salient function

One of the most interesting approaches to bass extraction is done by Salamon in his master thesis [63]. In his thesis, he designs a salient function for the melody and the bass line estimation using chroma features. It is constructed by adapting the Harmonic Pitch Class Profile (HPCP) [11] and used to extract a mid-representation which uses pitch class instead of absolute frequencies.

For the bass line extraction, the algorithm adopts the frequency range between 32.7Hz (1200 cent) and 261.Hz (4800 cent) to compute the HPCPs. The two other important parameters that he considers are the bin resolution and the window size. In order to detect subtleties in the analysis, the bin resolution is set to 120 bins. For the window size, since the frequencies to be analyzed are very low, he uses a 186 ms window (8192 frames for 44100 sample rate). Given the salience function, the bass line is selected as the highest peak of the function at every given frame. Moreover, no further post processing is performed.

It is important to note that two main improvements for future work are proposed for this method:

- A post-processing step for selecting the bass line out of the potential candidates (peaks of the salient function)
- A voicing detection method to determine when the bass line is present.

These improvements, among others, were included in the following algorithm, called *Melodia*. It represent the evolution of this primary work

3.3.1.2 Melodia

Melodia is the name of the melody extraction Vamp plugin which is based on the predominant melody algorithm [65] developed by Salamon during his doctoral thesis at the MTG. Even if its final version is designed for the melody extraction task, it was used by [66] as a bass extraction algorithm, with several modifications. Positive results were reported by the authors of the article. The predominant melody detection algorithm from Essentia [69] follows *Melodia*'s approach. Its main overall strategy is based on the salience function of the previous section but with a wide range of modifications and improvements (figure 3.8):

Sinusoid extraction and salience function

These two stages correspond basically to the salience function presented in the previous section. They replace the HPCP computation process with an important difference: the spectrum is not folded into one octave. This means that the salience of a given frequency is computed as the sum of the energies found at integer multiples (harmonics).

An important feature is also added to this module of the algorithm: the bin contribution of the peaks. Indeed, every peak not only contributes to the bin which corresponds to its harmonic but also to the neighbor bins. In doing so, tuning and possible frequency deviations are taken into account.

Pitch contour creation

In this stage, least salient bins are filtered and most salient bins are kept for the peak streaming. The goal of this module is to group salient bins into contours which potentially could represent melodic or bass lines. This process is done using heuristics based on auditory cues. The most salient peak is selected and added to a new pitch contour. Then, it tracks forward in time for a salient peak located at the following time frame which is within 80 cents from the previously found peak. The

process stops when there are no more salient peaks.

Melody selection

The melody selection process is thought as a filtering contour problem. Pitch contours are characterized in different features: pitch mean, pitch, deviation, pitch total length, among others. Using those features, the non-melodic contours are removed.

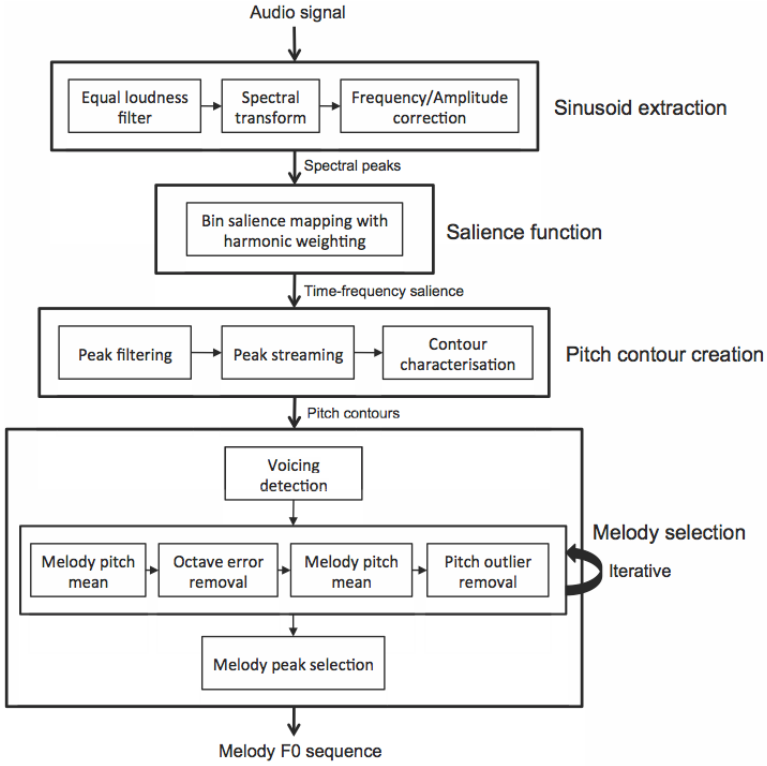


Figure 3.8: Block diagram of the *Melodia's* four main blocks

3.3.2 Other systems

There are two interesting approaches in bass estimation that we review in this section. The first one is by Goto [57] who uses a probabilistic frame work and the Expectation Maximization algorithm [73] to track the bass line. The second one is proposed by Klapuri, who uses multiple f0 estimation by harmonic amplitudes

summation [72] and machine learning techniques to transcribe the bass into notes [62].

3.3.2.1 Probabilistic bass line modeling

The PreFEst (Predominant F0 Estimation Method) system was created by Goto. He was the first to demonstrate successful melody and bass line extraction from real world audio signals. Figure 3.9 shows its general architecture.

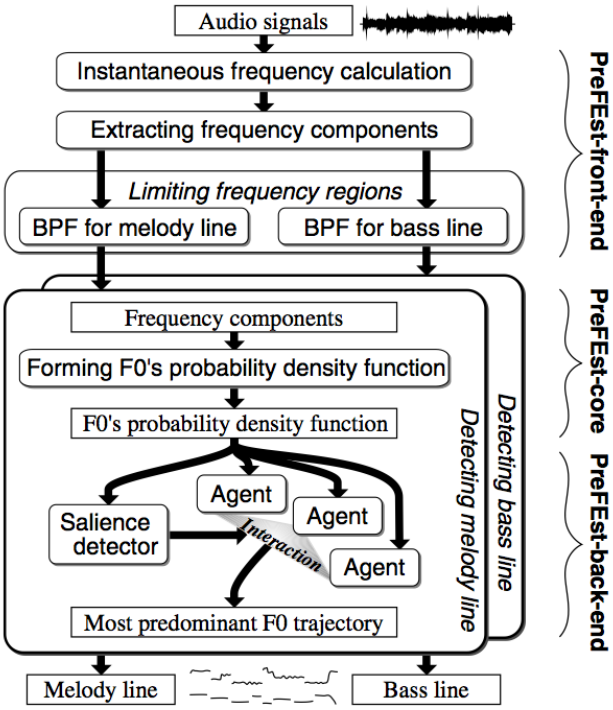


Figure 3.9: Block diagram of the PreFEst architecture [57].

The PreFEst system is divided in three main modules: the front-end, the core and the back-end:

- Front-end. It is the one responsible for the spectral analysis, using limited

frequency ranges depending on the type of task (melody or bass extraction). It produces the frequency components for the algorithm analysis.

- Core. This module represents the typical multi-pitch algorithm, very common in melody extraction strategies. It regards the observed frequency components as a weighted mixture of all possible harmonic-structure tone models and estimates the weights for the frequency components using the Expectation Maximization (EM) algorithm. The maximum weight model is considered as the most predominant harmonic series and f0 is inferred from it. For each frame, it outputs a set of f0 candidates by taking the top weighted models.
- Back-end. Considering the different f0 candidates for each frame, the most dominant and stable temporal trajectory is chosen and returned as the melody or bass line. This process is carried out by a multiple-agent architecture which performs the f0 tracking.

3.3.2.2 Bass transcription

This algorithm was proposed by Ryynänen and Klapuri [62] and was part of a larger system which also included key and chord estimation. The strategy that we review in this section can be used for both melody and bass transcription. Figure 3.10 shows the overall structure of the algorithm. Is uses multiple f0 estimation and a complex machine learning structure to model the notes and the notes transition.

Klapuri already used multiple f0 estimation in [72] for a f0 estimation algorithm which worked with candidate periods rather than with candidate f0s.

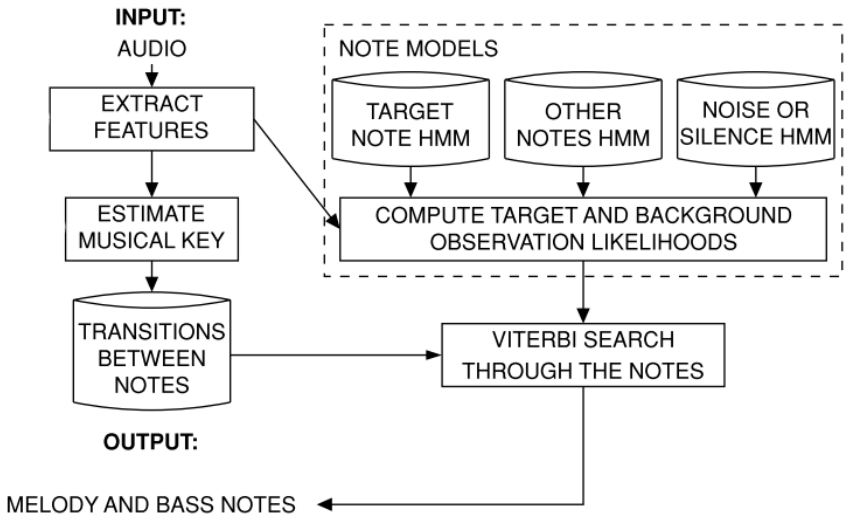


Figure 3.10: Block diagram of the bass transcription strategy using a machine learning approach [62].

In this case, he and Ryynänen propose a bass transcription algorithm which is much more complicated to achieve. Transcription implies onset detection and note duration. Their method can be divided in three main sections:

- Feature extraction. The pitch salience salience of each f0 candidate is calculated as a weighted sum of the amplitudes of its harmonic partials in a spectrally whitened signal frame.

- Acoustic modeling of notes. The idea behind this model is that all possible note pitches at all times are classified either as target notes (from bass line), notes from other instruments or as noise or silence. Therefore, three different acoustic models are trained. The target and the other notes are modeled with three-state left-to-right HMM (simulating the attack, sustain and release states). The training of the model is done using the RWC database.

- Musicological model for note transitions. The model takes into account the key of the musical segments to define the note bigrams. They determine the

probability of making a transition between notes or between notes and silence. For the bass lines, the note bigram estimation is done from a collection of over 1300 midi files including bass lines. They do not take into account the absolute pitch of the notes but only the interval between them.

3.3.3 Bass estimation literature discussion

As we have seen in the literature review, there are two types of well defined tasks: bass extraction and transcription. For our purposes, our final bass estimation algorithm should be able to transcribe bass notes since we want to use this information for segmentation purposes. Moreover, the Essentia's predominant melody algorithm is adequate for our approach. We have complete access to the code and it shows great performance in melody extraction. With some modifications, it should be a good starting point for our work.

CHAPTER 4

Methodology

In this chapter, we describe the methodology we have followed during our research. First of all, we introduce the external tools we have used to complement our work. Then, we introduce our chord detection algorithm, which includes a segmentation module based on bass information. We expose our strategies to improve bass estimation and then we describe different segmentation approaches in the chord estimation frame. Finally, we explain our evaluation methodology, including materials, evaluation metrics and experiments.

4.1 External tools

Before starting to describe the work we have carried out in this thesis, we want to mention the tools created by other researchers that we have used in our different strategies. Without them, most of our work would not exist.

4.1.1 Essentia

Essentia [69] is an open source library for audio analysis and audio-based music information retrieval create by the MTG group from University Pompeu Fabra. We have used two algorithms for the development of our research. The first one is the predominant melody algorithm which is based on the work *Melodia* by Salamon [65]. The second one is the HPCP algorithm by Gómez [11] which extracts the chroma features from audio.

4.1.2 Beat tracking

Extracting the beat positions of a song is a very important step in our segmentation algorithm. We obtained a list of the beat positions of a song using the Queen Mary, University of London, Bar and Beat tracker plugin [70]. We chose this plugin among others because, besides beat timestamps, this beat tracker also outputs the position of the beat inside the bar. Indeed, we are also interested in knowing the locations of the downbeats.

4.1.3 Key estimation

One of the segmentation strategies uses key information to filter possible harmonic relationships. We use the Queen Mary, University of London, Key Detector plugin [71] to extract the key from the different sections of the songs and therefore we can give a tonal context to our approach.

4.1.4 Librosa

Librosa is a python package for music and audio analysis (more information can be found at http://bmcfee.github.io/librosa/index.html) created by D. Liang, B. McFee, M. MacVicar and C. Raffel. We use one algorithm from this library called decomposition-hpss which performs a median harmonic percussive source separation. The idea is to apply our strategies to a harmonic spectrum to avoid noise when extracting the bass line.

4.2. Our chord estimation algorithm overview

The chord estimation algorithm that we are working on is somehow very similar to the first chord estimators (see section 3.2.3.1), where there was not a mid-level transition model. They have in common the analysis phase, with the chroma representation, and also the template matching stage. In our case, we also use a simple model of binary templates and we only consider major and minor chords. The main difference of our chord detector algorithm with respect to the most primitive ones is the smoothing of the chromagram.

The smoothing is a technique introduced in automatic chord estimation when researchers started to take into account other musical aspects, and more precisely the beat locations. This technique was supported by the fact that in popular music, a chord was very unlikely to change in between two beats [29]. Therefore, it was justified to smooth the chromagram between beats. Smoothing means basically unifying several chroma frames using a function such as the mean or the median. It makes the analysis more robust and less prone to noise and changes at a very local level.

As it has been shown in the literature review, this is a very common practice in

state-of-the-art algorithms for chord detection and almost all of them use the beat level smoothing [14, 26, 29]. Our algorithm, however, proposes to push the limit of the beat smoothing much further: we consider that the beat level as the minimum size of the segment to be smoothed but larger sizes can also be considered. Using music theory, pop music knowledge and mainly bass information we can consider larger fragments with harmonic coherence.

Figure 4.1 shows a general schema of our chord estimation algorithm with our main contribution highlighted in red.

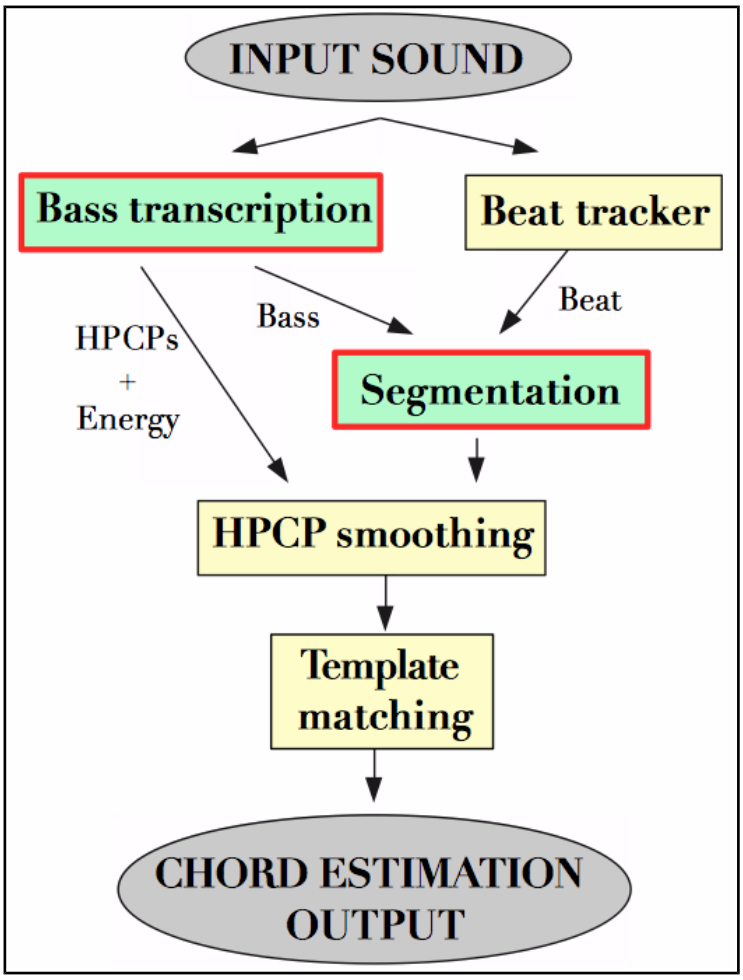


Figure 4.1 Our automatic chord estimation algorithm

Generally speaking, the algorithm uses the bass transcription and the beat positions to decide where to segment the chromagram and perform the HPCP smoothing. Unlike the state-of-the-art algorithms, which smooth the chromagram using constant segments, our approach selects fragments from different sizes depending on melodic and harmonic information extracted from the bass part.

In the next sections, we describe in a precise way the different approaches we have taken for the bass transcription process and also how we have used the bass and the beat information to segment the audio.

4.3. Bass line extraction and transcription

4.3.1 Introduction

In this section, we explain the approach we have followed to extract and transcribe a bass line, which will be used to segment the audio for the chord estimation task. Our main approach uses several steps of the Essentia's predominant melody extractor algorithm which is based on the *Melodia* plugin by Salamon [65]. Using this algorithm for the bass line extraction problem is based on the assumption that the bass line will tend to have a continuous contour, similar to a main melodic line. We have studied the effects of the analysis parameters, the effects of the parameter selection, we have modified its salience function and finally we have added filtering techniques to improve our algorithm performance.

4.3.2 Analysis parameters

In this sub-section, we describe the effect of window size and hop size in the Fourier analysis.

4.3.2.1 Window size

The window size is a key parameter in the Fourier analysis: it has a direct impact on which range of frequencies we are able to detect There is a trade-off between the time and the frequency resolution of the analysis, depending on the window size [54]. A small window allows us to detect fast changes in the spectrum, which means that we have a good temporal resolution but it implies that we cannot separate close partials or harmonics (i.e. one or more semitones apart) in the low octaves. On the contrary, a large window allows a good low frequency resolution but fast spectrum changes remain undetected. In the figure 4.2 we can see the difference between two different analysis:

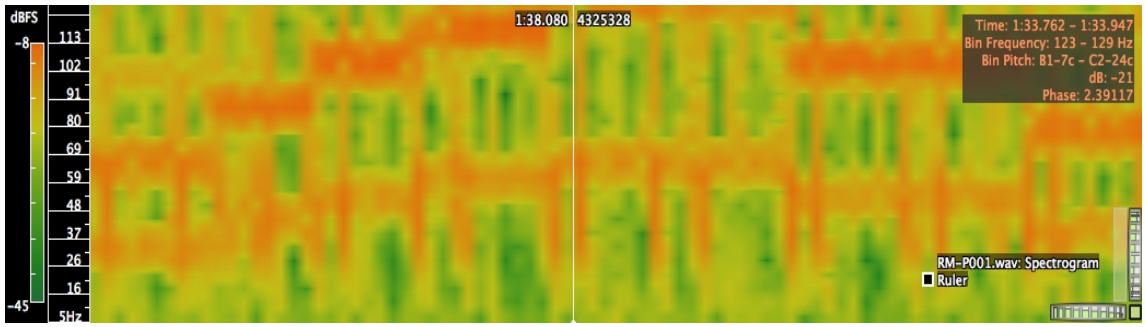
- 4.2a Shows a spectrogram obtained with a window size of 8142 samples. We can observe several thick lines which represent the bass line. They represent the energies in different frequencies. We can also see several vertical lines which represent percussive sounds.

- 4.2b Shows the same spectrogram but with a window size of 16384 samples. We can clearly observe that the lines representing the bass are thinner, which means that the frequency analysis is much more accurate. However, the vertical lines which were clear before are now blurred.

In our approach, we have to talk about another trade-off. Using a smaller window can detect subtleties in the bass line but also there could also be single frames where it is not the most salient line, resulting in noise. This is especially the case with percussive sounds which constantly interfere in the bass estimation. Since bass lines tend to be quite stable we opted to favor larger windows.

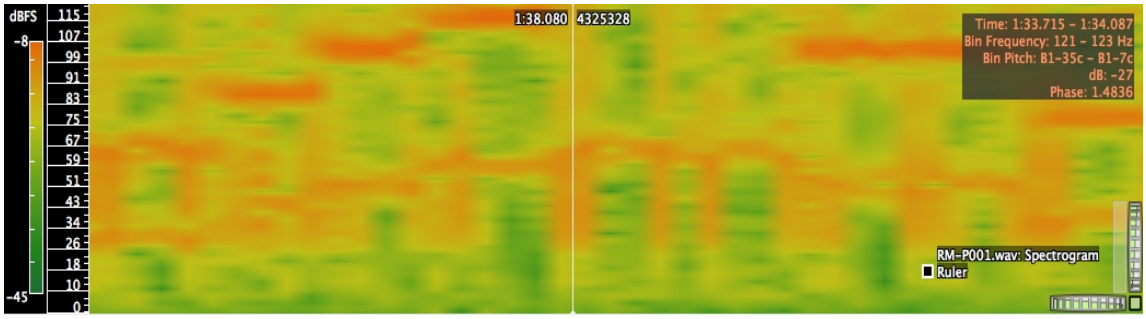
Following experiments using different window sizes we empirically set the window size to 16384 samples (372 ms) for a 44100 sampling rate.

Window Size: 8142. Overlap: 50%



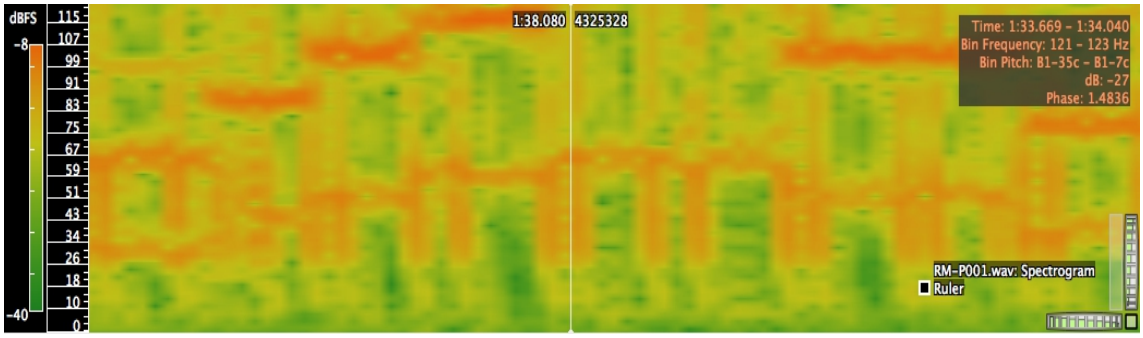
a

Window Size: 16384. Overlap: 50%



b

Window Size: 16384. Overlap: 75%



c

Figure 4.2: Effect of window size and hop size in spectrogram computation. X and y axis represent time and frequency respectively.

4.3.2.2 Hop size

The hop size parameter can be used to compensate the frequency-time trade-off explained in the previous section. If we use a large window size, a minimum requirement is that we update our analysis frequently to try to detect fast changes in the spectrogram. In figure 4.2 we can also see the effect of the hop size when extracting the frequency content of a signal. 4.2b and 4.2c analysis used the same window size (16384 samples). The difference is that b used an overlap of 50% and c an overlap of 75%. A larger overlap means a smaller hop size. We can observe that the spectrogram in 4.2b is much more blurred than in the spectrogram 4,2c. This means that there is higher time resolution when the hop size is small. Since we are using a large window for our analysis, we need to use a small hop size to at least try to obtain a better time resolution.

Following experiments using different hop sizes we empirically set it to 512 samples (96% overlap). Our other best hop size candidate was 256 samples but the benefits of using it were insignificant and the computational cost was much higher.

4.3.3 Essentia's melody extractor step selection

The original algorithm is meant to detect the predominant melody of a polyphonic audio. However, it can be used and configured to extract melodic lines in general, not only the most prominent. It is necessary to use only few steps of the original algorithm since some of them were designed to enhance melodic lines in mid-range frequencies.

The functions that we have used for our purposes are the following:

- Spectral Peaks
- Pitch Salience Function
- Pitch Salience Function Peaks
- Pitch Contours

The two main stages which are missing are the equal loudness filter to enhance the frequencies to which humans are more sensitive (which are usually perceived with more difficulty) and the final step of the algorithm, which is mainly focused on predominant melody detection with voice detection emphasis.

4.3.4 Essentia's melody extractor parameter selection

The Essentia's algorithm that we are mainly using for bass estimation has several functions and many parameters to adjust. Sometimes, it is not obvious what the exact impact of one small parameter change is and at other times the selection is trivial.

In general, the approach we have followed to select the parameters has two steps. The first one is evaluating the characteristics of what we have to analyze. For instance, in this case, the bass is located in the low frequency range so frequency

range considerations have to be taken into account. These first steps help us to approximate the values of the parameters. The second step has more to do with fine tuning and heuristics. The following table shows the main decision we had to take when adjusting the algorithm and with which we obtained the best results for bass extraction:

FUNCTION	PARAMETERS
Spectral Peaks	minFrequency: 20.0
	maxFrequency: 1100.0
Pitch Salience Function	referenceFrequency: 27.5
	harmonicWeight: 0.8
	numberHarmonics: 10
Pitch Salience Function Peaks	referenceFrequency: 27.5
	minFrequency: 27.5
	maxFrequency: 110.0
Pitch Contours	minDuration = 200.0
	timeContinuity = 100.0
	pitchContinuity = 27.5

Table 4.1: Optimized parameters for bass estimation

The decisions we made are related mainly with the range of the analyzed frequencies and the contour creation. The frequency range, number of harmonics and weighting schemes have a big impact in pitch class errors whilst pitch contours tuning has an important consequence when dealing with percussive sounds.

4.3.5 Essentia's melody extraction algorithm modifications

In order to try different hypothesis about the bass extraction task using the Essentia's algorithm, we had to modify two steps.

The first one belongs to the preprocessing stage where normally an equal loudness filter is applied to the signal. For obvious reasons, we modified this filter and replaced it with a low pass filter, which was recommended in [12, 66].

The second step that we modified is the salience function. As it's explained in [65], "the salience computation [...] is based on harmonic summation, [...] where the salience of a given frequency is computed as the sum of the weighted energies found at integer multiples. [...] Only spectral peaks are used in the summation". Moreover, in the salience function "each peak contributes not just to a single bin of the salience function but also to the bins around it (with $\cos^2$ weighting)". The amount of contribution to the closest bins is normally fixed in the algorithm but we found that it was interesting to modify it to study its impact on the bass estimation. Specially because our pitch class confusion matrix in our results analysis showed that many mistakes were made with adjacent pitch class notes. We have added a new parameter to the function to control the range of contribution to the closest bins of every peak.

4.3.6 Filtering

The filtering stage is one of the most important parts of our bass estimation algorithm. It is performed at three different levels: in the frequency bin representation, in the contour representation and at the final estimation level. The following sections describe the work we have done to improve the performance of the algorithm by filtering non-relevant information.

4.3.6.1 Contour selection

Once the contours are created, the challenge remains in identifying which of the contours belong to the bass line. This can be an easy task when the low frequency range of the spectrum is clean but when a song has a lot of percussion or another instrument is playing in the same range(i.e. a piano), it can become very hard. Moreover, Essentia's predominant melody detection is not a perfect algorithm and contours can contain information other than bass lines.

Contour characterization

Characterize contours can help to decide when the contour selection is not obvious. In figure 4.3, we can observe two different scenarios when a contour has to be selected. 4.3a represents a very easy situation but the contour selection in 4.3b is not obvious.

Contours can be characterized by the following parameters [65]:

- Pitch mean: the mean pitch height of the contours
- Pitch deviation: the standard deviation of the contour pitch
- Contour mean salience: the mean salience of all peaks comprising the contour
- Contour total salience: the sum of the salience of all peaks
- Contour salience deviation: the standard deviation of the salience of all peaks comprising the contour
- Length: the length in time of the contour

Using these characteristics, we can establish ways or decision rules to choose a contour before another one.

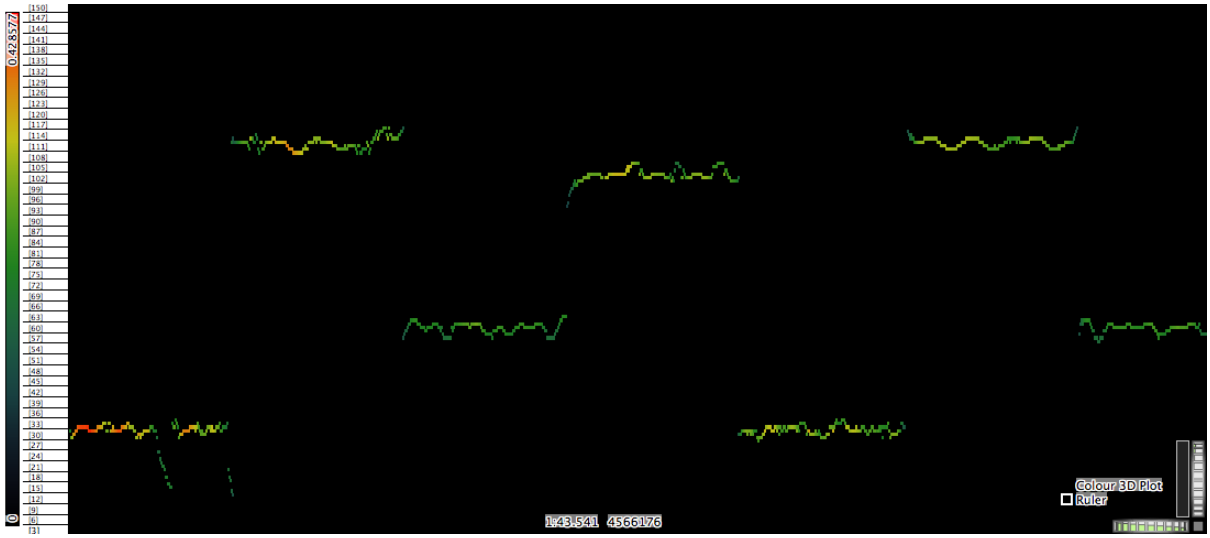
Final contour selection

There are several parameters of a contour that can be useful for our approach. Length is the first one and we use it to discard contours which are too short. A short contour probably represents noise or a percussive sound. The other two features which are interesting are contour mean salience and contour total salience.

We performed preliminary experiments about contour selection when several contours overlapped in time as only one of them should be capturing the bass line. We tried two different approaches: first, selecting a final contour using its mean salience and then using its total salience. We realized that the latter approach performed a better bass extraction.

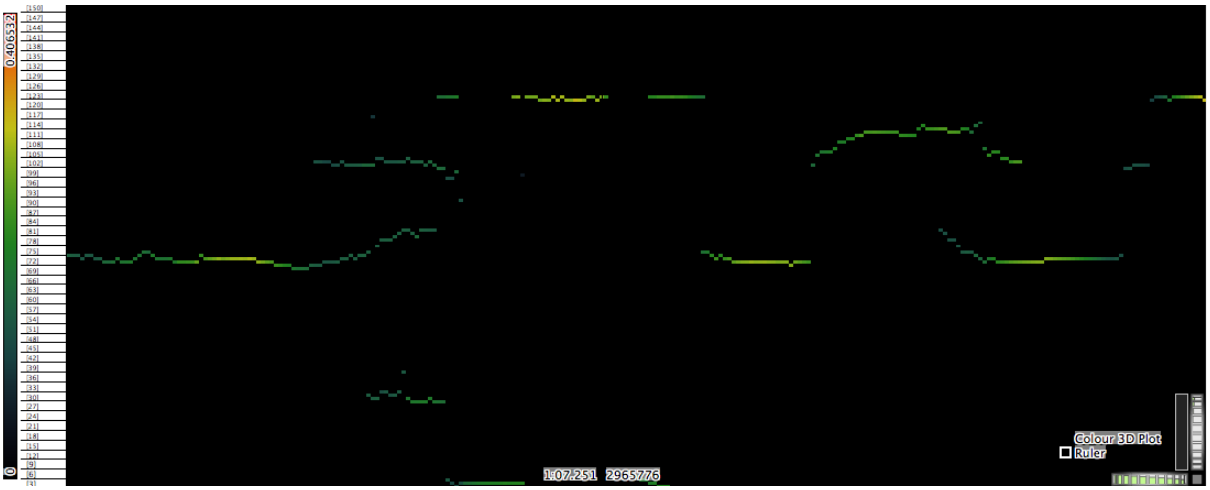
This could be explained by the fact that a percussive sound can be included in a short contour and this type of sound is characterized by being short and by having high energy. A contour like that would obtain a high mean salience. In the other hand, if we use the total salience, we are favoring long contours over the short ones.

RM-P001.wav from Popular RWC



a

RM-P001.wav from Popular RWC



b

Figure 4.3: Contours extracted with Essentia's predominant melody algorithm. X and y axis represent time and bins respectively
Bass estimation in song a is simpler than in song b

4.3.6.2 Energy-wise filtering

As we have mentioned already, one of the biggest problems in bass estimation is the percussive sounds which share frequency range with the bass, like a kick. We also want to be sure that our algorithm does not output any value when there is silence or only percussive instruments are playing.

To this end, we have developed an algorithm that discards frames where there is silence in the low frequencies or only percussive sounds. It is based on energy analysis and it assumes two things:

- If there's no bass present, the energy in the low part of the spectrum should be very low or non-existent.
- If only percussive sounds are present, the energy presence should be short in duration.

In figure 4.4 we can see an example of low frequency energy analysis where percussive sounds can be easily spotted by their short duration. Blue circles represent percussion without bass and green circles represent bass presence.

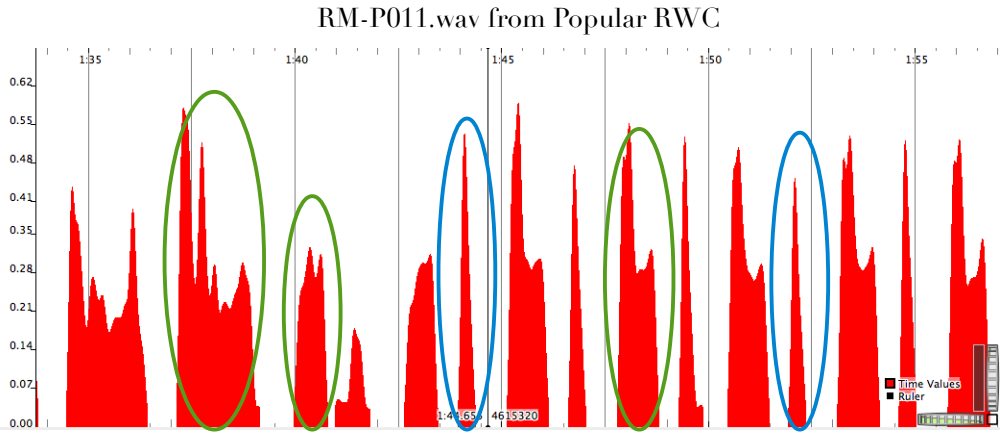


Figure 4.4: Energy in the 27-180Hz band. X axis represents time and y axis represents energy. Blue circles show percussive sounds and green circles represent bass presence

The steps followed by this filtering algorithm, performed at the final stage are the following:

- Spectrum computation.
- Energy calculation in the 27-180Hz band (bass presence).
- Frames with energy less than an energy threshold are discarded.
- Consecutive frames (having energy) shorter than a time threshold are discarded.

We chose the values of the threshold empirically:

- Energy threshold: 5% of the maximum energy during the song.
- Time threshold: 300 ms.

4.3.6.3 Short notes filtering

The final filtering stage consists of getting rid of "notes" that would be too short. The development of this filter was motivated by two facts: the final estimation vector representing the bass line needed to be smoothed and the ultimate goal of the bass transcription which is segmenting the audio for chord estimation.

In the first case, the final estimation presented sometimes changes of notes that were too abrupt. This happened because of the system we used to select the pitch class notes using the contours total salience. To be able to smooth these cases, we have used two different approaches: a median filter and also a filter developed by us. We describe it in the next paragraphs.

In the second case, we assume that very short notes in the bass line are passing notes which means that they are not important when defining the harmony: they come from longer notes which are the ones important to our approach. We are not

interested in a perfect bass transcription (firstly because it's a very difficult task) but at the minimum we want to obtain a good approximation of the bass line notes. We have to choose between two options:

- Transcribing all the notes (also the short ones) but having lots of false invented notes (or "false positives").
- Not transcribing all the notes but having a more stable transcription (missing notes but not invented notes).

We have opted for the second option since it is more adequate for our segmentation purposes. It represents the maximization of the precision of the algorithm, even if the recall diminishes.

The filter we have implemented is very simple: it checks the duration of the notes and if one note duration is less than a threshold, it changes its name to the name of the previous long note, as shown in the next table:

Frame	Original	Filtered
12003	A	A
12004	A	A
12005	A	A
12006	C	**A**
12007	C	**A**
12008	G	G
12009	G	G
12010	G	G

Table 4.2: Example of filtering short notes frame-wise

Filtering with beat information

The previous version of the filtering uses a fixed time threshold for the minimum duration of a bass note. Since in pop-rock music it is very unlikely that a harmonically important bass note is shorter than a beat, we use the beat positions to modify dynamically the threshold depending on the inter-beat interval (IBI). This means that we can adapt the filtering of the short notes for every song.

We have used binary values for the dynamic threshold since most popular music is written using a binary measure (2/4 or 4/4):

$$\frac{IBI}{n} \quad where \quad n \in \{1,2,4\}$$

4.3.6.4 Salient bins filtering using beat positions

Working with contours is a very interesting task but it has limitations: it is quite high level since the Essentia's melody algorithm has already taken many decisions when outputting the potential bass lines (contours). Those are constructed with salient bins (from the salience function) but some of them can correspond to non-bass line elements (i.e. percussion, higher harmonic, noise, etc.).

In our research, we have also tried to work at a lower level by filtering salient bins. More specifically, we have focussed on bins which could belong to percussion, namely kick sounds. The assumption which supports this filtering algorithm is that salient bins belonging to the bass line should be more stable between beats than the percussive ones. Indeed, we can use the salience of the bins between beat positions and the standard deviation to measure the stability of each bin during the inter-beat interval. As we can see in the figure 4.5, we can think that salient bins in the green zones, which belong to the bass line, will have a low standard deviation

value if we consider them between beat positions. On the contrary, the SD of other bins belonging to other unstable zones (red) should be much higher.

RM-P060.wav from Popular RWC

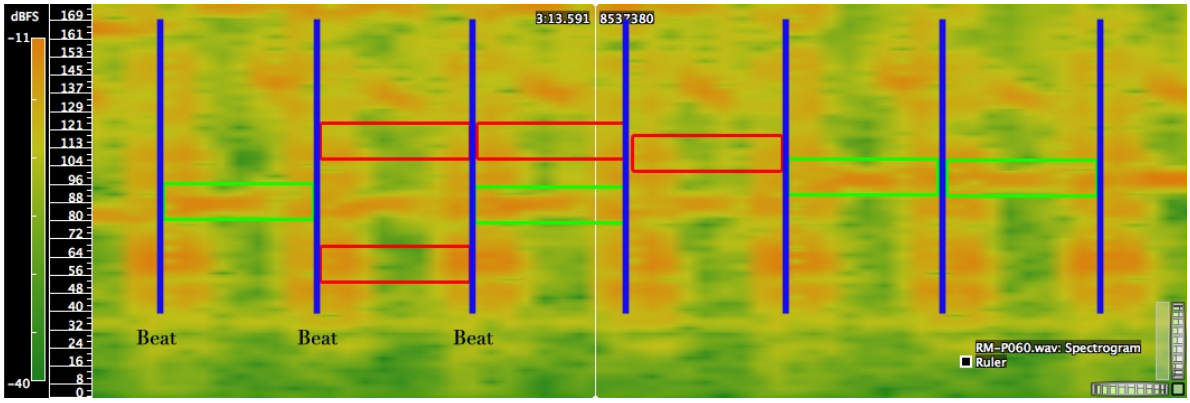


Figure 4.5: Spectrogram. Blue lines represent beat position, green represents bass line bins, red represents noisy or percussive bins.

The filtering algorithm follows these steps:

- Beat positions extraction.
- Standard deviation calculation for every frequency bin using the salience values of the frames between beat positions.
- Bins which have a standard deviation value higher than a threshold are discarded since bass notes should have a low SD.

We have also tried to filter different amounts of frames (for every bin). First, we filtered all the frames between the beats. We also tried to filter only the first half of the interval between beats where supposedly the percussive sound should be present.

4.4 Audio chord segmentation

In this section, we explain the different approaches we have followed to segment the audio for automatic chord estimation. We describe how we use the bass information and the beat positions, sometimes combined with key information, to find the most adequate segments to smooth the chromagram.

4.4.1 Segmentation based on bass information

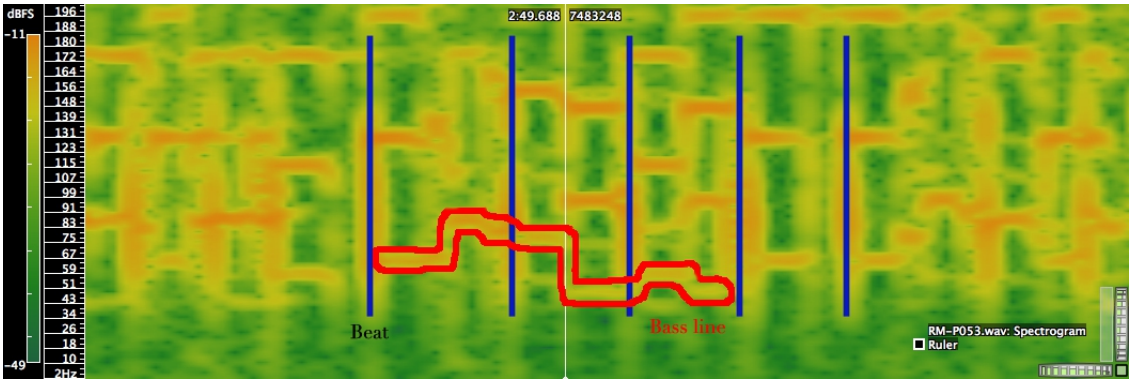
In this sub-section, we describe the strategies we have used to segment the audio files only using the information of the bass, namely, the onsets of the bass notes. We start with the most simple and we finish with the most complex, which also uses key information.

4.4.1.1 Note to note segmentation

This approach is the most direct way of segmenting an audio file based on the bass line. The simplest way to separate the audio into fragments is to consider the regions between bass onsets. Obviously, it has advantages and drawbacks.

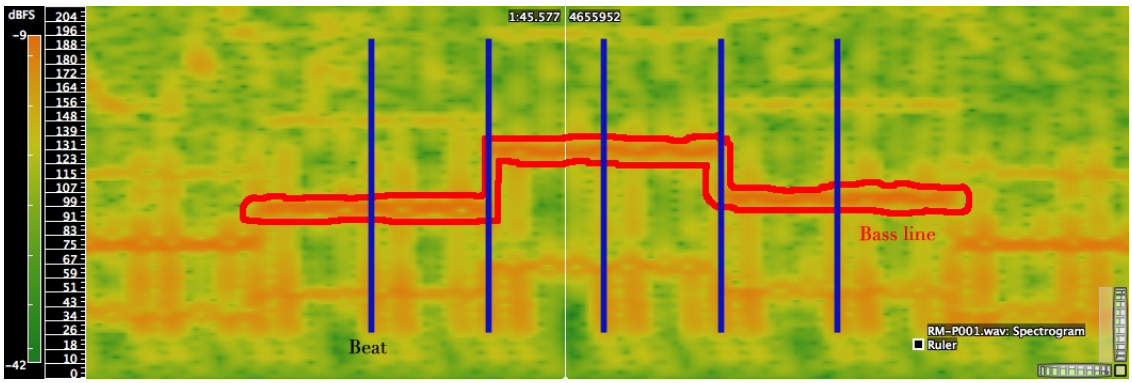
The main advantage of this strategy is that since it's very unlikely that a chord changes without a new bass note, we are almost certain that we are segmenting at the right places. Larger segments than beat duration can often be selected, as shown in figure 4.6b. It would be even possible to create larger segments but at least with this approach the fragments will hopefully correspond to only one harmony type. The main drawback of the strategy is that in some cases bass notes can change more rapidly than the beat (as shown in figure 4.6a) and our objective is to do the opposite: try to select larger fragments than the interval between beats.

RM-P053.wav from Popular RWC



a

RM-P001.wav from Popular RWC



b

Figure 4.6: Spectrograms. Blue lines correspond to beat positions and red represents the bass line

4.4.1.2 Segmentation using harmonic relationships

This strategy is more elaborated than the previous one because it takes into account music theory knowledge and pop-rock music knowledge. To be able to design it, we have taken the assumption that a chord, in this type of music, is not likely to change every beat but every half bar or even every bar. We also know that normally it is very probable that in every bar more than one bass note is played. With this approach, what we seek to do is locating larger segments by grouping

more than one bass note. Those notes need to have a harmonic relationship: they have to be part of the same major or minor triad. This strategy doesn't take into account the final duration of the segment: very large segments could be selected if the bass notes contained in the chunk belong to the same chord.

We are going to describe how the algorithm works and show an example to clarify the strategy. The goal of the algorithm is to merge bass notes into chords. First, a matrix containing all the templates of the major and minor chords is created such as {0,4,7} (corresponding to a C major chord), {0,3,7} (corresponding to a C minor chord), {1,5,8} (to a Db major chord), etc. When the algorithm considers a bass note, it is added to an empty array. Then, the next note is considered and added to the same array. If the array is a subset of one of the templates, the algorithm keeps adding more notes to it. It stops when the array is not a full subset of the template.

Bass notes merging algorithm

```
1.  create templates with all major and minor chords
2.  for n = 1 to number of bass notes
3.      add the note to an array
4.      compare the array with all the templates
5.      if the array is not a subset of one of the templates
6.          empty the array
7.          store the position of the current note as a separator
8.          add the current note to the array
9.      end
10. end
```

Example

Let's consider the following bass notes with midi notation (0(C), 1(Db), 2(D), 7(G), etc.):

0(C) 1(Db) 5(F) 8(Ab) 5(F) 9(A) 3(Eb)

- First the algorithm considers 0(C). The array a = [0], which is a subset of the chords C major, C minor, A minor, F major, etc.
- The algorithm adds to the array the value 1. Now a = [0,1] which is not a subset of any template. A separator is created at this note.
- The array value at this point is a = [1]. Then the 5 is added (a = [1,5]). Now, the array is a subset of two templates: {1,5,8} and {9,1,5}. When the 8 (next note) is added, the array is only a subset of the template {1,5,8} which corresponds to the chord Db Major.
- From now on, if there is a different pitch class, a separator will be created.

The resulting four segments of this example would be:

0 / 1-5-8-5 / 9 / 3

4.4.1.3 Harmonic relationship segmentation using key information

This strategy is basically the same as the one described in the previous section but in this case we are using key information to improve the algorithm performance. Key is used as a cue to filter the chord templates. Using the key, we can consider only the most common chords of it, reducing the number of out-of-the-harmony groupings. For instance, for the C major key, the most common chords are the following: C, Dmin, Emin, F, G, Amin and Bdim.

The programming of this strategy is very simple. We use the same algorithm for

the previous section but with a few changes:

- Extract the key information of the song (with sonic annotator).
- Select the most common chords for a major and minor key and store them into two templates.
- Depending on the key of the fragment considered, before comparing the array with the templates, we do a circular shifting of the matrix containing the chord templates and select the correct indexes for the most common chords.

The main benefit of this strategy is to prevent wrong bass notes groupings containing harmonic relationships far away from the song.

4.4.2 Segmentation based on bass and beat information

In this sub-section, we describe the strategies we have used to segment the audio files by combining the information of the bass and the beat positions. Most of the algorithms are similar to the previous section but include more conditions and constraints derived from the inter-beat interval (IBI).

4.4.2.1 Segment's length limitation by number of beats

This strategy is very similar to the one described in 4.3.1.2 but taking advantage of the beat information. The main difference is the inclusion of an additional condition in the segmentation process. We use the beats to establish a temporal condition: basically, segments cannot be longer than a certain time threshold. This idea is based on the assumption that chords are more likely to change at the beginning of the bar in pop-rock music [29]. Translated to more scientific terms, this means that the segments should have a maximum length equivalent to four

beats (in a 4/4 measure, which is the most common in pop music).

Bass notes merging algorithm with temporal threshold

```
1.  create templates with all major and minor chords
2.  T = threshold    // set with the inter-beat time (IBI)
3.  for n = 1 to number of bass notes
4.          add the note to the notes array (a)
5.          add the onset time to the times array (t)
6.          compare the array with all the templates
7.          compute the difference d between the last and first value of t
8.          if the array is not a subset of one of the templates or d > T
9.                  empty the array
10.                 store the position of the current note as a separator
11.                 add the current note to the array
12.         end
13.  end
```

The algorithm acts almost as the one described in the previous section but if the difference between the onset time of the last note added to the array and the onset time of the first note of it is longer than a temporal threshold, a separator is set automatically, even if the next bass note was part of the same previous chord. The thresholds we have considered are:

$$IBI \times n \quad where \quad n \in \{1,2,4,8\}$$

This strategy can be combined both with the merging algorithm taking into account key information or the version which does not.

4.4.2.2 Segmentation using downbeat information

Conceptually, this algorithm is based on the same assumption as the previous one and seeks exactly the same objective. It uses downbeats (even more informative information than only beats) to try to merge bass notes in a more logical way. The main difference is that in this approach, at the beginning of each bar (determined by the downbeat) there is always a separator: this means that the merging process is only allowed inside a bar but never between bars.

The algorithm performs the following steps:

- Locates the downbeats among the beats (using sonic annotator).
- Locates the bass notes corresponding to the downbeats.
- Includes a new condition to the already-explained bass-note-merging algorithm which automatically creates a separator if the bass note corresponds to a downbeat.

This approach should reflect more accurately the assumption that a chord is more likely to change from bar to bar than in other cases in pop-rock music.

4.4.2.3 Bass note alignment with the beat

In the bass estimation process, due to the different filtering stages, in some cases bass notes can be displaced forwards or backwards in time. Using the beat position information, we are able to align the notes beat-wise and improve the segmentation.

The alignment is done in a very naïve way: if a bass note onset time is close to a beat time, the onset time of the beat is adopted by the bass note. The decision of how close a note onset should be is set with a fixed threshold of 100ms.

4.5 Evaluation methodology

In this sub-chapter, we discuss the evaluation methodology followed in our thesis. More precisely, we explain the steps followed on the one hand to evaluate the bass line extraction and on the other hand, to evaluate the audio segmentation tool for chord estimation. We start by describing the music collections used in our research. Then we discuss the evaluation metrics and their suitability for the tasks and finally we explain how the preparation of the reference annotations was done. The results of the evaluation are presented in the next chapter.

4.5.1 Introduction

The evaluation is an essential part of research. It is the natural way to confirm or discard an hypothesis and ensures that a work has been done and tested rigorously. However, different methodologies can be used to evaluate the same problem and this can represent a problem for the research community. To be able to compare results between researchers, it is necessary to build common criteria for the whole community. In the automatic chord estimation field, standards have been somehow defined such as chord notation [58] or Public Music Collections which are used by the researches (Isophonics or Billboard). Many of those common elements are gathered in the MIREX (Music Information Retrieval Evaluation eXchange): a competition which takes place annually and evaluates different algorithms in tasks related to music.

4.5.2 Music collections

Music collections are generally composed of a set of songs or musical pieces and

a number of reference files which describe their musical parameters. The parameters are normally annotated and coded in a certain format which has been decided and approved by the research community. In music research, we can find key, chord, beat, onset or structure annotations among others. Music collections are a very important part of research since they allow researchers to test their work in an objective way by using the same material. Therefore, reviewing, maintaining and expanding those collections is an important and necessary task. In this section, we review the principal datasets used for automatic chord estimation and we present the ones we have used for our evaluation methodology.

4.5.2.1 Chord estimation datasets

Ground truth chord data is essential for testing the accuracy of chord detection algorithms. Available datasets have grown in the past years and are becoming more and more diverse. This diversity is very important, especially when machine-learning based algorithms are becoming more and more popular, since there is the potential problem of over-fitting them. In this sub-section we describe the available dataset for the chord estimation problem.

Chord symbol-based datasets

The first dataset available for researchers was published by Harte and collaborators in 2005. It was composed by 180 pop songs by the Beatles, and later expanded to include songs by Queen, Zweieck, Carol King and Michael Jackson. They are part of The Center of Digital Music (CDM) at the University of London and it's know as the Isophonics dataset. There is also a 195 song subset of the 'USpop' dataset which was hand annotated by Cho [5] and it's also available and open. Another important dataset belongs to the McGill University Billboard Project

which has compiled a corpus of songs selected from the Billboard charts spanning 1958 through 1991. Massimiliano Zanoni has also published a corpus of Robbie Williams songs with harmonic and beat information. At present, MIREX competition uses the Isophonics and the Billboard datasets for testing the chord detection algorithms. We will also use the Isophonics collection to evaluate our work in chord detection.

These dataset projects provide manually- encoded annotations of songs, although the data in these corpora is limited primarily to harmonic and timing information.

Functional harmony-based datasets

More recently, Temperley et al. published a corpus of harmonic analysis and melodic transcriptions of 200 rock songs [42]. The most interesting aspect of this corpus, besides the fact that there is a melodic transcription of the songs for the first time, is that the harmonic information is coded not in absolute chords (i.e. Gmajor, Aminor7) but in an actual functional analysis of the songs. This means that the chords are labeled as degrees, instead of chord names. Therefore, this corpus could be used for instance to train algorithms based on functional harmony instead of pure chord labels (without taking into account the key context).

4.5.2.2 RWC

The Real World Computing Music Collection (RWC) Music Database was gathered by Goto et al. [56] to provide evaluation material for music researchers. The first version of the collection contained 215 songs in four databases: Popular Music (100 songs), Royalty-Free Music (15 songs), Classical Music (50 songs) and

Jazz Music(50 songs). In 2004, the collection was extended with a Music Genre Database (100 songs) and a Musical Instrument Sound Database (50 instruments) [55]. For our research, we have used the Popular Music database for the bass extraction evaluation because it's the only database which contains transcriptions for the bass line.

For the whole set of songs, the authors coded the transcription of the instruments present in each song in a Standard MIDI File (SMF). Most of the music was transcribed by ear and actually the first version of the SMD was not synchronized with the audio. A second synchronized version was released afterwards to facilitate the evaluation process.

Finally, what makes the Popular Music database interesting is that J.P. Bello et al. annotated in 2011 the chord information for the whole set of songs [53].

4.5.2.3 Songs and datasets

In our research, we have used two of the described collections. The following table shows the datasets and to which purposes they have been utilized.

Dataset	Number of songs	Task
RWC Popular	66	Bass extraction
RWC Popular	84	Bass transcription
Isophonics Beatles	136	Chord estimation

Table 4.3: Song and datasets used for evaluation

The complete list of songs used for every task can be found in the appendix.

4.5.3 Evaluation metrics

In this section, we describe the metrics used to evaluate the steps that we have followed in our work related to the automatic chord estimation problem. We have to take into consideration two different evaluations scenarios:

- Bass line extraction: since the information of the bass line is critical to our approach, we need to evaluate the performance of our algorithm in this respect.
- Chord estimation: in order to validate our hypothesis, we have to evaluate the efficiency of the chord transcription when using our audio segmentation tool.

4.5.3.1 Bass line evaluation

We now present the evaluation metrics used for the bass extraction problem. Our work is partially based on the master thesis by Salamon and this fact makes us dependent on its evaluation process for the bass line extraction. Since we have to compare our results to the ones obtained in his thesis, we need to use at least the same evaluation metrics.

Frame-wise metrics

The evaluation metrics used in Salamon's [63] thesis are based on the MIREX 2004 and 2005 metrics for melody extraction. Two main metrics are described in his work: the first computes the raw transcription concordance and the second computes the chroma transcription concordance:

- Raw Pitch Accuracy: the proportion of voiced frames in the estimated transcription which are correctly labeled, out of the total number of voiced frames in the reference transcription

- Raw Chroma Accuracy: the same as raw pitch accuracy but mapping both reference and estimated frequencies onto one single octave.

Since his approach only outputs an octave agnostic chroma representation, he only uses the second metric. His evaluation is also adequate to our approach because we are just interested in pitch class changes in the bass line. We just want to know when the root of a chord is changing for the chord estimation task. Therefore, the chroma accuracy is a very good metric for our purposes.

Precision, Recall, F-measure

The previous section describes two metrics which are useful to compare the work that has been done previously but for our actual goals are insufficient. The Chroma Accuracy described previously is an adequate metric for bass extraction but not for bass transcription. Bass transcription involves at least the segmentation of the bass line into notes, which means that onsets have to be detected, and quantization of the pitch into semitones has to be done. In our case, as it has been explained already, pitch is represented by pitch class notes. To be able to segment the audio using bass information, the onsets of the bass notes and therefore the bass transcription is needed rather than the bass extraction.

In order to be able to evaluate the note transcription, we have used different metrics. Following [62], we have decided to use precision, recall and F-measure to be able to do a comparative evaluation.

- Precision is defined by the number of correctly transcribed notes divided by the number of transcribed notes.
- Recall is defined by the number of correctly transcribed notes divided by the number of reference notes.

- The F-measure F is defined by the following formula:

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

Moreover, we consider a reference note correctly transcribed by a note in the transcription "if their midi note number are equal, the absolute difference between their onset times is less than 150 ms, and the transcribed note is not already associated with another reference note".

4.5.3.2 Chord estimation evaluation

In this section we describe the most common evaluation metrics used in automatic chord estimation. We also expose the evaluation metrics and methodology that we will use in our work to compare the performance of our segmentation tool to other approaches.

Chord symbol recall

The most common performance metric that is used for chord evaluation is what we call chord symbol recall, also known sometimes as the average overlap score or relative correct overlap [21]. This is a measure of what proportion of the time chords in the annotated ground truth chord sequence have been identified correctly in the machine estimated sequence.

The chord recall, which will be described in the next sections, can be calculated in two ways. One is to sample the chord sequences into uniform length chord symbol frames and calculate the frame-based chord symbol recall. The other ways is to add up the durations of the continuous sections of estimated segments that correctly match the ground truth and calculate the segment-based chord symbol

recall [16]. In our research, the evaluation approach we have followed is the frame-based chord symbol recall.

Frame-based chord symbol recall

The frame-based chord recognition recall can be defined as:

$$\mathcal{R} = \frac{N_{\mathrm{C}}}{N_{\mathrm{T}}}$$

N_C represents the number of correct frames estimated and N_T represents the total number of frames. These measure have been very popular in automatic chord estimation algorithms because most of them have been frame-based and a usual thing to do is to sample the annotated ground truth data at the same frame rate as the estimator to perform the recall evaluation.

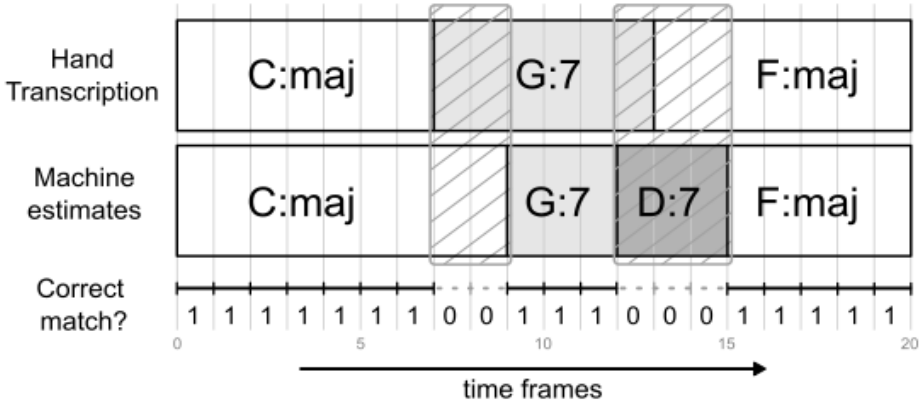


Figure 4.7: Frame based recall example, taken form [16]

Chord vocabulary and mapping

The chord vocabulary of the Isophonics and RWC databases that we are using to evaluate our work is complex. The output of our algorithm is reduced to major and minor chords but the reference files of those collections can include more complex chords such as seventh chords, augmented or diminished chords or even extended

chords and also information about the state of the chord (if it's in the root position or inverted).

In order for the evaluation to be more fair, we have performed a vocabulary reduction and also a chord mapping which is also done in the MIREX competition:

- Mapping. "A mapping exists if both the root notes and bass notes match and the structure of the output label is the largest possible subset of the input label given the vocabulary" (http://www.music-ir.org/mirex/wiki/MIREX_HOME). This means that a reference G major chord with a seventh would match a simple G major chord estimated by an algorithm which is able to detect only major and minor chords.

- Vocabulary reduction. "If a chord cannot be represented by a certain class, i.e. mapping an augmented chord or a sustained 4 chord to a major or minor, the chord is excluded from the evaluation".

4.5.4 Data preparation

In this section, we describe how the music collections and the metadata files were prepared for the evaluation process. The Isophonics dataset was easily set up for the chord detection task since it's a collection used in MIREX and the reference files are already designed to facilitate the evaluation as much as possible.

On the contrary, the preparation of the RWC database took certain amount of effort before being able to use it. Several problems arise from the fact that the format of the annotations is coded into midi files. The first version of the metadata wasn't synchronized with the audio but we had access to the new synchronized version. Even though, we faced problems and in this section we will describe the steps we followed to solve them.

It is very important to note that not all the Popular Music database from the RWC collection was used for the evaluation of the bass line extraction. In order to be able to compare our work with the previous work done by Salamon in his master thesis, we selected the same subset of songs that he used for his evaluation. More information about the songs can be found in his work [63].

4.5.4.1 Alignment

The annotations of the RWC popular database called the AIST Annotation were done by Goto and collaborators manually and by ear. They represent the beat structure, the melody line and the chorus sections. Moreover, the transcription of the instruments present in every song are included in standard midi files (SMD). There are two versions of those files: the first one wasn't synchronized with the audio and the second one used the annotated beat positions of the audio files to achieve the synchronization. Even though, we still had to take several aspects into consideration to prepare the ground truth information.

Initial offset

When extracting the note information from the reference files, we realized that the onsets of the bass weren't aligned with the onsets of the audio file. We manually checked this fact by using Sonic Visualizer: we computed the spectrogram with a window size of 16384 and a 75% of overlap to be able to distinguish extremely low frequencies with a reasonable time definition. We repeated this process for the whole set of the popular database (100 songs) and we realized that there was variable initial offset for all of them. We have manually annotated the initial offset for every song of the popular sub-dataset of the RWC music collection.

Song ending

During the process of the song alignment, we spotted two different cases in the database. On one hand, songs which had a clear and actual end (no fade out and a proper cadence) had a ground truth file with a precise annotation of the last note of the bass. On the other hand, songs with an unclear ending, normally due to a fade-out effect, had an annotation which was longer in time that the real audio. This means that, even if the annotation was correct, there was a point when the reference file was reporting non-existent notes because the audio was already finished. For this second group of songs, we had to decide where to stop the evaluation mainly for two reasons:

- The fade-out effect makes the analysis very difficult since the energy of the spectrum is close to zero at one point.
- To crop the reference to a point where its information is still correct.

We decided to stop the analysis and the ground truth files at the moment the fade-out started.

Synchronization checking

Even if the authors of the AIST Annotation claimed that the new midi files were synchronized, we thought that it was important to check the alignment of all the reference files with the audio files. After defining the initial offset, we chose for each song several points of interest (i.e. the start of a note after a long silence, the last note of the song, etc.) to check if the synchronization was correct. We found out that it was well done for the songs we used for the evaluation.

4.5.4.2 Format conversion

In this section we describe the steps we followed to convert the SMF reference files of the RWC Music Collection into a more suitable format for our evaluation process. As we described in the last section, every midi file contains the transcription of the instruments of every song, each of them stored in a different track. Therefore, for every piece of music, we have to extract the bass line and convert it into the formats that we will use to perform the evaluation. To this end, several steps have to be followed:

- Track identification
- Note on/off format matrix conversion
- Frame-wise format conversion
- Bass notes final format and note filtering

Track identification

As we have mentioned before, the popular database of the RWC Music Collection includes 100 songs. Even if they have been labeled as popular, they belong to different genres and their arrangements are very diverse. The number and the type of instruments used in every piece of music differ from one song to another. Moreover, the annotations are not consistent when assigning a track number to an instrument. For instance, the bass is normally placed in the 4^{th} track but this is not always true. In other cases, there is no bass in the song. In order to locate the bass tracks, for each song of the database, the reference file was opened in Logic and the relevant track was stored in a text file.

Synchronization text file

In order to facilitate the bass ground truth extraction process, we stored all the relevant information related to synchronization in a text file. The file contains the following four column format:

1st Column	2nd Column	3rd Column	4th Column
Song number	Bass track number	Initial offset	Song ending time

Table 4.4: Structure of synchronization file

This information is used afterwards to generate the ground truth for the evaluation process.

Note format matrix conversion

The standard midi file with its timestamp messages is a complicated format to be processed and understood by a person. It is necessary to adapt it to an easier representation, more suitable and manageable. We have used for this purpose the KARAOKE MIDI JAVA library. Written in Java and usable in Matlab, this library has very useful functions to operate with midi files. Using the readmidi_java function with a SMF returns an eight column matrix with the following format:

1st Column	2nd Column	3rd Column	4th Column	5th Column	6th Column	7th Column	8th Column
Note start in beats	Note duration in beats	Channel	Midi pitch	Velocity	Note start in seconds	Note duration in seconds	Track

Table 4.5: Midi information matrix

Bass notes final format and note filtering

The format described in the previous section is very informative but it has too much information for our purposes. Our note format needs to have only three things: the pitch class label, the onset of the note and the end of the note in seconds. Therefore, the rest of the information is discarded and the midi note is transformed to the pitch class representation by applying the modulo 12 operator. As a result, the pitch classes range is 0 to 11, the C note being represented by a 0 and B note by the 11. The final format contains the following three columns:

1st Column	2nd Column	3rd Column
Pitch class label	Bass note onset	Bass note end

Table 4.6: Final ground truth format for bass estimation

There was one final consideration that we had to make to properly prepare properly the ground truth file when evaluating the algorithm using the contours from Essentia's predominant melody algorithm. As we already explained, this algorithm, when creating the contours, uses a parameter called timeContinuity

which represents the time continuity cue: the maximum allowed gap duration for a pitch contour (in ms). This means that the algorithm can potentially include in the same contour two notes which are separated by a silence inferior to that parameter. Therefore, the ground truth should be corrected to connect the notes where the gap is inferior to the timeContinuity parameter as shown in the next table (for the default value 100ms):

Filtering	Pitch class label	Bass note onset	Bass note end
BEFORE	0	13.100	13.200
	2	13.250	13.500
	2	13.575	13.700
	7	14.000	15.500
AFTER	0	13.100	**13.250**
	2	13.250	**13.700**
	7	14.000	15.500

Table 4.7: Ground truth note correction example

The first note end time is extended to the beginning of the second note and the third note, being the same as the second one and close to it (less than 100ms), is removed. This format is used to evaluate the bass transcription task and also to generate the ground truth for the frame-wise evaluation.

Frame-wise format conversion

In the previous section, we described a reference format which is ideal to evaluate the amount of bass notes detected and, in particular, the onsets. The frame-wise format is a vector and every position corresponds to the pitch class of the bass at every frame. This means that the length of the vector is hop size

dependent: the smaller it is, the longer is the vector. The absence of bass is coded with a value equal to -1. We are using this bass representation since we want to compare our approach with the one done in [63] by Salamon. This format is not useful to evaluate the transcription of a bass line (bass notes with precise onsets) but can be helpful to detect (broadly speaking) whether the bass estimation algorithm is improving or going in the wrong way.

4.6 Experiments

In the evaluation process, we have tested the bass estimation algorithm performance and also the chord detector algorithm performance. We have also planned two experiments to see the potential of our segmentation strategy. In this sub-chapter, we explain the experiments and the goals behind them.

4.6.1 Chromagram smoothing

With this experiment, what we want to show is that segmenting in a more intelligent way could lead to a general improvement of the chord estimation algorithms.

Using a simple binary template matching approach, we are going to label songs with chord tags using different segmentation strategies. The chromagram frames of every segment will be smoothed with a median function and compared with the templates to assign a chord tag.

We will compare the following strategies:

- Smoothing at the frame level: which means no smoothing at all.
- Smoothing at the beat level: we will smooth the chromagram frames between beat positions, which correspond to a beat segmentation level.
- Smoothing using bass notes: we will smooth the chromagram frames between bass note onsets, which correspond to a dynamic segmentation (bass notes do not always have the same length).

We want to prove that smoothing the chromagram using our segmentation approach (using bass information) leads to better results than using the beat level smoothing, which is the most common in current state-of-the-art algorithms. For this evaluation experiment, we have used the chord estimation database (songs by the Beatles from Isophonics dataset).

4.6.2 "Best case" estimation

With this experiment we want to prove that, theoretically, by having the right information (about the bass notes and the beat position), it would be possible to reach good results in chord estimation by only using a template matching approach and a good segmentation technique, even without using a mid-level transition model.

We provide the algorithm with the correct bass and beat information (it does not have to estimate them). With the correct data, it performs the segmentation and the chord labeling. Finally we compare the result with the one reached by the *Chordino* plugin. For this experiment, we have only used three songs of the Isophonics dataset which were manually annotated by us: Help!, Not a second time and Please, please.

CHAPTER 5

Results

5.1 Introduction

In this chapter, we present the evaluation results for our bass line extraction and transcription approach based on the Essentia's predominant melody algorithm. We expose all the contributions step by step and we compare them with the results obtained by Salamon [63] and Ryynänen and Klapuri [62].

We also describe the results obtained in chord estimation with our segmentation tool (using a very basic template matching approach) to an almost state-of-the-art algorithm: the *Chordino*. Finally, we present the outcome of the experiments to show the potential of segmentation based on bass information.

5.2 Bass line algorithm performance

In this section, we describe the results we have obtained in the bass line extraction and transcription tasks. First we present the results frame-wise and secondly we present the performances when detecting bass notes and their onsets.

5.2.1 Frame-wise evaluation

In this sub-section, we present the evaluation results of the different algorithms and approaches we have tested in the bass extraction task. We also report the errors we have analyzed and comment on their possible explanations.

5.2.1.1 Algorithm performance

In order to evaluate our contributions to the bass extraction algorithm, we have tested our work by combining different steps of the approach. The starting point of the evaluation is the Essentia's predominant melody algorithm (omitting the last step which is centered in melodic detection, as discussed in section 4.2.3). For text convenience, we have called it EPMC (essentia's predominant melody contours). Its results represent our reference to check the level of improvement we have achieved. Table 5.1 shows the precision of the algorithm using different modules.

Algorithm	Precision
EPMAC (default parameters)	69.49%
EPMAC (optimized parameters)	71.74%
EPMAC + energy-based filtering	73.06%
EPMAC + note filtering	73.54%
EPMAC + median filtering (21)	72.78%
EPMAC + energy and note filtering	**74.64%**

Table 5.1: Frame-based evaluation for the bass extraction task

As we can see, by using the EPMAC with the double filtering (by energy and short note) we have reached a 74.64% of precision, outperforming our reference by more than five points. For this evaluation process, we have used the whole duration of the analyzed songs.

Using our best approach, we have also tested the algorithm in the conditions described by Salamon in his master thesis which has been an inspiration for the present work. He used 66 songs from the RWC Popular dataset for the bass extraction task but only by taking into account the voiced segments (were the bass was present) to test his salience function. We have used the ground truth to select those passages and we report the results in the following table.

Music Collection	**Metric**	**Salience function**	**Our approach**
RWC Pop (66 songs)	Chroma(semitone)	73.00%	78.15%

Table 5.2: Frame-based evaluation for the bass extraction task (voiced segments)

As we can see, by adjusting the algorithm and adding our filtering modules we again outperform the salience function by more than five points.

5.2.1.2 Other evaluation results

In the table 5.1, we only showed the contribution of the modules which were improving the algorithm in a clear way. During our work, we have proposed other approaches which are missing in that table: the frequency bin filtering and the bin contribution modification of the salient function. The main reason is that they proved to be inefficient when improving the bass extraction task.

Table 5.3 shows the evaluation results by adding the bin filtering module. This strategy filters the unstable salient bins between beats to remove percussive sounds. The version (1) filters all the frames between beats and the version (2) only filters the frames around the beat position.

Algorithm	Precision
EPMAC + bin filtering	71.65%
EPMAC + energy, note and bin filtering (1)	74.44%
EPMAC + energy, note and bin filtering (2)	72.63%

Table 5.3: Algorithm evaluation using bin filtering

As we can see, the precision is reduced with respect to our best approach. This could be explained by that fact that sometimes, more than one note can be present between two consecutive beats. If this is the case, the bins belonging to those notes are filtered out by this method. This approach should work fine with songs with bass notes at least longer than one beat.

Table 5.4 shows the evaluation results obtained by modifying the bin contribution of the peaks in the salient function. Essentia's default parameter is 200 cents, which means that every peak contributes to the bins equivalent to 100 cents above and under it, using a $\cos^2$ weighting function. In our error analysis, we spotted a high percentage of confusion mistakes with adjacent pitch classes and we decided to reduce the contribution to only half semitone (100 cents). Nevertheless, we found out that the initial contribution was much more beneficial for the overall algorithm.

Algorithm	100 cents	200 cents
EPMAC (optimized parameters)	69.12%	71.74%
EPMAC + energy and note filtering	72.29%	**74.64%**

Table 5.4: Algorithm evaluation modifying the bin contribution of the salient function

Finally, the following table shows the results of using harmonic percussive source separation (HPSS). This idea was sketched out by Salamon in [63]. Conceptually, it should be very beneficial for the bass extraction task since kick sounds share frequency range with the bass. However, results show that using HPSS reduces the precision of the algorithm.

Algorithm	Precision
EPMAC + HPSS audio	71.10%
EPMAC + energy and note filtering + HPSS audio	73.68%

Table 5.5: HPSS audio results

We can hypothesize that in the process of separating harmonics and percussive, important information for the bass analysis is discarded.

5.2.1.3 Error analysis

In this sub-section, we present varied information about the errors we have identified in our research about bass extraction. In general, we compare the type of errors done by the most basic approach (EPMAC) and the final version of the algorithm.

The table 5.6 shows the most common error types in the bass extraction task:

- Undetected silence frames: when the ground truth frame is silence but the algorithm is activated and detects a note.
- Undetected bass note frames: when the ground truth frame is a note but the algorithm estimates a silence.
- Pitch class confusion frames: the result of estimating a pitch class when the ground truth is another pitch class.

Error types	**EPMAC (default parameters)**	**EPMAC + energy and note filtering**
Undetected silence frames	11.68%	8.32%
Undetected bass note frames	2.97%	3.97%
Pitch class confusion frames	15.86%	13.19%

Table 5.6: Error types

The results show that we have improved considerably in better detecting the silences, which means that the control of the activation of the algorithm has been improved. Moreover, the pitch class confusions have been also reduced considerably, even if the percentage error is still high. The rise of undetected bass notes can be explained as a consequence of the filtering. When filtering percussive sounds, weak or short bass notes can be also filtered as a side effect.

For our work, it is very important to explain that the undetected silence error type is not totally realistic. During our research, we have spotted many ground truth errors in the RWC Popular dataset. It is difficult to estimate the impact of them because it would need a total revision of the collection but we think it is considerably high. We have categorized two main classes of error: pitch class confusion errors (not so common) and length note problems (very common). In the

ground truth, the length of the notes always tend to be shorter than they are. Figure 5.1 shows a typical example of a ground truth note which is shorter than the real sound. The upper part of the figure represents the low frequency range of the spectrogram and below, we can see the ground truth representation. Grey circles show the conflict zone. In many cases, our algorithm is activated at those zones but the evaluation reflects an error of undetected silence.

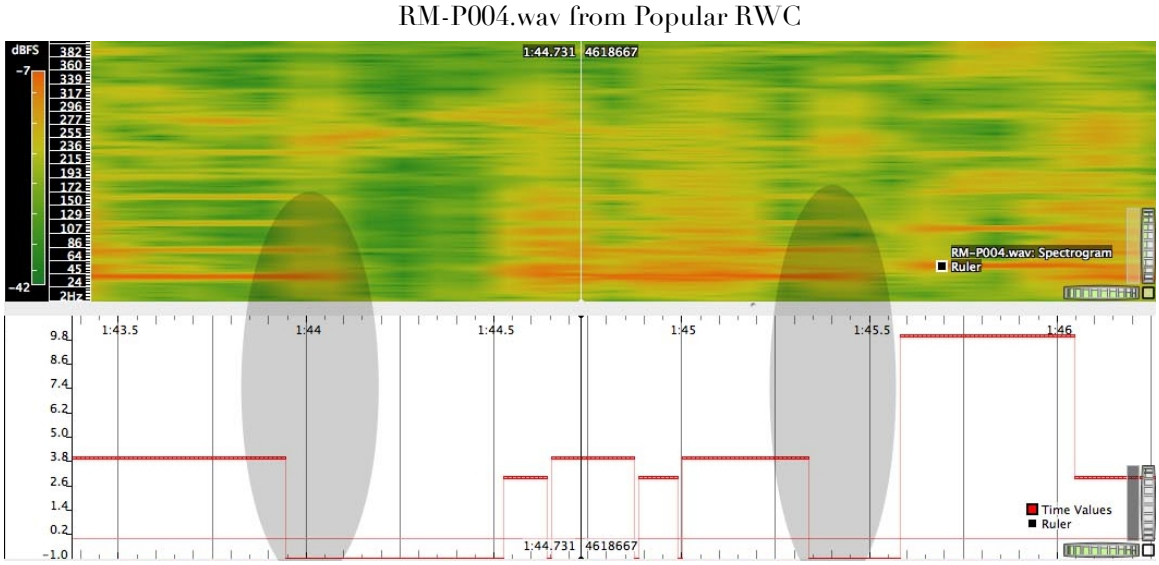


Figure 5.1: Spectrogram and ground truth conflict

Only as a fast experiment, without proper study, we decided to extend bass notes (or connect them) by about 150ms. The total performance of our algorithm was raised by 3 points.

As we have seen in the previous table, pitch class confusion is a major problem in bass line estimation and has a large improvement field. We are interesting in determining which are the most common mistakes of this type. Table 5.7 shows the

confusion vector of the pitch classes. Since the concept of pitch class is circular, the confusion distance of 11 semitones is considered in this case as an adjacent pitch class error.

Confusion distance	**EPMAC (default parameters)**	**EPMAC + energy and note filtering**
1 semitone	**15.07%**	**11.43%**
2 semitones	8.97%	12.11%
3 semitones	6.15%	7.32%
4 semitones	3.92%	3.67%
5 semitones	**11.38%**	**16.60%**
6 semitones	2.81%	2.08%
7 semitones	**17.69%**	**11.86%**
8 semitones	3.67%	4.08%
9 semitones	5.27%	5.90%
10 semitones	8.66%	11.73%
11 semitones	**16.34%**	**13.19%**

Table 5.7: Pitch class confusion vector

We can observe two interesting facts. On one hand, the main problems are found with adjacent pitch classes and also with the ones located at a 5 and 7 semitone distances. On the other hand, there is an evolution between the type of errors when comparing the most basic algorithm and our final approach.

The errors with adjacent notes are probably due to the analysis difficulty of very low frequencies: to resolve the low partials we need very big windows but at the same time, note changes can be blurred because of the length (we were using 280ms window size). The errors with the 5 and 7 semitone distance are potentially

the same error. Indeed, as we can see in figure 5.2, if we consider the 5th of a chord (musically speaking), it can be at a 7 semitone distance if it's above the fundamental but also at a 5 semitone distance if its below the fundamental. Still, it's the same note.

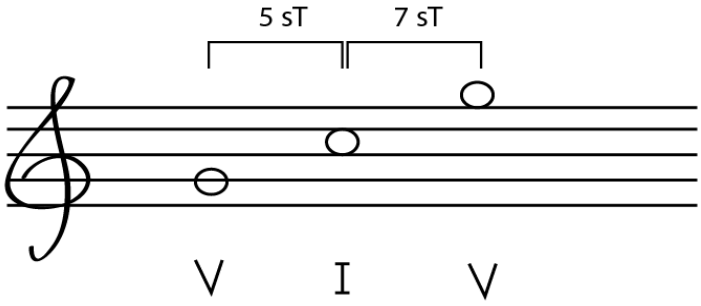


Figure 5.2: Pitch class equivalence but different semitone distance

These errors can occur due to the salient function, which uses harmonics contribution (very similarly to the HPCP algorithm).

Finally, we have studied the behavior of the algorithm in two different zones: around the beat position and out of it. For this purpose, we have calculated the precision of the starting point of the algorithm and the precision of the final approach at those segments and we have also quantified the pitch class confusions.

Table 5.8 shows how in general, the precision of the algorithms is lower around beat positions (between 30ms before and 120ms after the detected beat). Salient percussive bins are likely to be responsible for the errors done by the algorithm. The table also shows the improvements we have achieved with our final approach.

Algorithms	Around beat	Off beat
EPMAC (default parameters)	63%	71.72%
EPMAC + energy and note filtering	**68.87%**	**76.42%**

Table 5.8: Algorithm precision in beat zone and non-beat zones

The next table shows also how percussive bins affect pitch class confusions in error percentage. Higher confusion is shown in beat position areas and there is also an important improvement with our final approach.

Algorithms	Around beat	Off beat
EPMAC (default parameters)	20.22%	14.38%
EPMAC + energy and note filtering	**16.12%**	**12.12%**

Table 5.9: Pitch class confusion errors in beat zone and non-beat zones

The higher confusion around beat positions could be explained by the behavior of the salient function. The percussive bins could contribute to the salience of wrong pitch classes, overcoming the salience of the real notes.

5.2.2 Note-wise evaluation

In this sub-section, we present the evaluation results of the optimized version of our algorithm in the music transcription task. We will not show the results of intermediate steps but only the version which got the best score in the extraction task.

The following table shows the results of the EPMAC + energy and short note filtering and also the results of the approach by Ryynänen and Klapuri in [62]. We selected that article over this one [61] because in the former, the whole song duration was used for the analysis and the latter only used short song fragments. We tested the algorithm with the exact song list (84 songs from the RWC Popular dataset).

Algorithm	**Precision**	**Recall**	**F-measure**
Ryynänen & Klapuri	57.5%	57.7%	56.3%
EPMAC + energy and note filtering	67.09%	70.06%	67.08%

Table 5.10: Bass note transcription evaluation results

These results are very promising but unfortunately they are not comparable. The transcription by Ryynänen and Klapuri is in midi notes. They are taking into account two octaves and a half for the bass range. On the contrary, we are only considering one octave because our output is pitch class based. The only way to compare the results in a fair way would be to ignore the octave errors by their algorithm. We could also not fold the frequency range into one octave but that is not the point of our research. Indeed, we are only interested in pitch class because the octave information is not very useful when using the bass line in the chord estimation problem.

5.3 Chord segmentation algorithm performance

In this section, we describe the results we have obtained in the chord estimation task. First we present the algorithm performance comparing the different segmentation approaches. Then we do a comparative evaluation with the *Chordino* algorithm and finally we show the results of two different experiments to show the potential of our segmentation strategy.

5.3.1 Algorithm performance

Before comparing our chord detection algorithm with the state-of-the-art, we are presenting the results we have achieved by the different segmentation procedures.

Table 5.11 shows the evaluation results in automatic chord estimation achieved by our approach using different segmentation strategies. The values represent the chord symbol recall of the different combinations of the approaches. For instance, the result in bold (60.08) corresponds to the combination of segmenting the audio using the merging bass notes strategy (without key information) and the limiting by number of beats strategy (with number of beats equal to 1).

Beat information approach	**Bass note to bass note**	**Merging bass notes with harmonic relationship**	**Merging bass notes with harmonic relationship and key**
No information	58.59%	52.82%	56.39%
Limiting by nº beats (n=1)	-	**60.08%**	59.88%
Limiting by nº beats (n=2)	-	59.19%	59.14%
Limiting by nº beats (n=4)	-	56.85%	57.87%
Limiting by nº beats (n=8)	-	54.24%	56.90%
Downbeats	-	56.45%	57.99%

Table 5.11: Chord symbol recall results in chord estimation task

The table presents numerous results which look similar and complex to understand. Nevertheless, we can observe several patterns and interesting facts which deserve our attention.

Range of recall values

The range of precision values is narrow. With the exception of the merging strategy without any type of information, all the values are between 56% and 60%. Moreover, in many cases, the precision of the algorithms is worse than the most simple approach: segmenting from bass note to another one.

Time limitation

It is very obvious from the data that the more permissive the algorithm is with large segments, the worse is the precision. When limiting by maximum number of beats, we are changing the maximum length allowed for a segment creation. If the segment is very large, several chords can be gathered together inside a chunk. This would lead to a very confusing chroma calculation and to a very bad result in the chord estimation process. Let's consider the following example:

Bass notes:	C	G	G	D
Chord ground truth:	Cmajor		Gmajor	

We can see two bars: the first one represents a Cmajor chord and the second one a Gmajor chord. Using the merging strategy, the grouping of the bass notes would be:

C-G-G // D

1st segment 2nd segment

This is a very bad segmentation since by grouping the notes like this, the first segment cuts the second chord in half and therefore a lot of noise is introduced in the first segment (since we are mixing two chords). This could be the main reason why the larger the segment, the lower the precision. This example brings us to the next point which is also related to this phenomenon.

Downbeat strategy performance

The performance of the algorithm which takes into account downbeat information is much lower than expected. This strategy should deal with the problems presented in the previous section. One possible explanation for this deficient result is that the approach is very dependent on the downbeat position. If it is miscalculated, generally all the downbeats of the song will be wrong since the algorithm considers a 4/4 measure and assigns the downbeat every four beats. From this point of view, the approach is not very robust because it fundamentally depends on the correct estimation of the downbeat.

Key information impact

From the table 5.11, we can also conclude that the inclusion of the key information in the segmentation algorithm is very positive. In general, there is always an improvement when taking it into account, especially when the segments can be larger. This is the case of the approaches which allow four or more beats, the downbeat approach and the simplest one, without any limitations.

Bass notes and beat alignment

We have also evaluated the impact of the bass note onsets alignment with the beat positions in the automatic chord estimation task. Due to the various filtering we did in the bass estimation, it was possible that some of the bass note onsets were moved backwards or forwards. Table 5.12 shows the impact of the time correction of the onsets by using beat positions.

Algorithm	No alignment	Beat alignment
Bass note to bass note	59.40%	59.84%
Merging bass notes with harmonic relationship	60.08%	60.07%
Merging bass notes with harmonic relationship and key	59.88%	59.90%

Table 5.12: Impact of beat alignment

As we can see, the impact of aligning bass onsets to beats is minimal. There is a very short improvement in general but we consider that is not significant.

Comparative evaluation

Using the Beatles songs from the Isophonics dataset, we have tested our chord estimation approach against a well-known almost state-of-the-art algorithm: the *Chordino* plugin. The following table shows the labeling precision of both programs:

Algorithm	Labelling precision
Note to note with EPMAC (default parameters)	57.46%
Our approach	60.08%
Chordino	73.16%

Table 5.13: Comparative evaluation with *Chordino*

As we can see, we have improved the chord estimation results by introducing our segmentation strategies with respect to our starting point. Still, it is is obvious that *Chordino* outperforms our approach very significantly: by about 13 points. Nevertheless, it is fair to say that our algorithm is missing one of the most important parts of an automatic chord estimator: the mid-level model. Its main contribution is the segmentation process for the chromagram smoothing. In the next section we present the results of the segmentation experiment, which shows the importance of a good segmentation.

5.3.2 Experiment results

In this sub-section we present the outcome of the experiments to show the potential of segmentation based on bass information.

5.3.2.1 Segmentation experiment

With this experiment, what we try to show is that segmenting in a more intelligent way could lead to a general improvement of the chord estimation algorithms. The following table shows the results in chord estimation by smoothing the chromagram using different segmentation strategies.

Segmentation approach	Labelling precision
Frame level	43.64%
Beat level	**52.82%**
Note to note with EPMAC (default parameters)	57.46%
Our approach	**60.08%**

Table 5.14: Segmentation results

The results of this experiment are very interesting. We can observe a great difference between the frame level segmentation (no smoothing at all) and our segmentation approach. When the smoothing is done at the beat level, the template matching algorithm scores a 52% in chord labeling. This type of smoothing is used by almost all the state-of-the-art algorithms. When the smoothing is done using our segmentation algorithm, the template matching algorithm scores 60%. It represents an 8 points difference with respect to the beat segmentation. We think it is a very promising result because our segmentation technique could also be included in more sophisticated chord detection algorithms. In fact our approach only uses a binary template matching for the chord estimation.

5.3.2.2 "Best case" estimation experiment

With this experiment we want to prove that, theoretically, by having the right information (about the bass notes and the beat position), it would be possible to reach good results in chord estimation by only using a template matching approach and a good segmentation technique, even without having a mid-level transition model. Therefore, we have annotated three songs by the Beatles from the Isophonics dataset and used the annotation for the estimation. Table 5.14 shows a comparative evaluation with *Chordino*.

Algorithm	Labeling precision
Our approach	**75.68%**
Chordino	74.27%

Table 5.15: "Best case" evaluation results

As the table shows, if the bass and beat estimations used by the segmentation algorithm are correct, a simple template-matching algorithm without transition model could achieve better results than *Chordino*.

5.4 Conclusions

In this chapter we have presented and discussed the results obtained in our research in the fields of bass estimation and chord estimation. We started by showing the changes and improvements that we did to the Essentia's predominant melody algorithm in the bass extraction task, outperforming the initial stage by more than 5 points. Then, we analyzed the most typical errors and identified the aspects of the algorithm that could be improvable, especially the pitch class confusion errors. We also showed that the current results could be even better if the ground truth was revised. We also compared the extraction task to the salient function by Salamon [63] (outperforming by more than 5 points) and the transcription task with Ryynänen and Klapuri's approach [62], showing promising results.

Regarding the chord estimation task, we presented the evaluation results of our algorithm using different segmentation techniques. The best score was obtained by using beat information to limit the length of the segments created by grouping bass

notes with harmonic relationship. Nevertheless, comparing our approach with a state of the art chord estimator like *Chordino* showed that our overall strategy has room for improvement. In fact, it lacks a mid-level transition model, which is used by all the chord estimation algorithms. However, the experiments showed that as a segmentation tool for chromagram smoothing, the algorithm could have a high potential. In fact, it could be part of a larger and more sophisticated system.

CHAPTER 6

Conclusion

6.1 Contributions

We want to start this section by offering a reminder about the goals we proposed at the beginning of this thesis (section 1.2). We note that all of them have been fulfilled:

- Study the role of bass line and beats in the automatic chord estimation problem.
- Develop a new method for audio segmentation to enhance audio chord estimation based on bass notes and beat positions.
- Modify Essentia's predominant melody algorithm to improve its performance in the bass transcription task.
- Provide comparative evaluation of our approaches with respect to other algorithms

From the methodological process and the evaluation results, we can make several final conclusions related to the bass estimation task and the chord estimation task.

Regarding the former, we have obtained good results and improvement in the algorithm by tuning it and adopting the filtering strategies: we have increased its performance by more than 5 points. Even so, we think that there is still room for improvement, especially regarding the pitch class errors, but this would mean working on a lower level. Indeed, experiment with the spectrogram and also with the salience function would be necessary.

With respect to the chord estimation task, we have also improved our initial results by adopting new segmentation strategies. Still, our approach is far from the state-of-the-art algorithms. However, it is worth mentioning the results of the two proposed experiments, especially the first one which is related with the different segmentations. With a simple chord detection approach, we compared our segmentation strategy with the most common segmentation approach for chromagram smoothing in the chord estimation literature and we outperformed it by 8 points. This fact leads us to believe that our segmentation algorithm based on bass notes and beat positions could be used as a segmentation tool for more complex chord estimation systems.

6.2 Future Work

The creation process of our current work has happened very fast and some of the ideas or strategies proposed during the thesis could have been developed more extensively. We want to list some of the future work that could be done to extend or complement this document:

- Bass estimation. As we have commented in the previous section, important improvements could be done in the bass extraction algorithm at the spectral level by filtering noisy sounds. Moreover, the highest number of errors in bass estimation are found in the pitch class confusions with adjacent notes and also with the fifths.

This issue deserves special attention since there is a lot of room for improvement and the algorithm performance would increase significantly.

- Segmentation tool test. The evaluation of the segmentation tool proposed in our work could be improved if we could include it in a state-of-the-art algorithm, replacing the typical beat segmentation. A comparative evaluation with the original algorithm would be a very informative experiment.

- Mid-level chord transition model. The strategy that we used in our thesis is a very simple approach based on template matching. It doesn't have any transition model. It would be very interesting to design one which could work together with our segmentation tool and evaluate it with state-of-the-art algorithms.

6.3 Final words

Writing this document has not been an easy task. As my first serious research work, I've found it painful and rewarding at the same time. I have discovered that researcher's life is not an easy one, but it can give great satisfaction and it is worth giving it a chance. Finally, I just want to thank all the people who helped me in any way during my short path in the sound and music computing world.

Urbez Capablo Riazuelo

September 2014

BIBLIOGRAPHY

[1] Anglade, A., Ramirez, R., & Dixon, S. (2009). Genre classification using harmony rules induced from automatic chord transcriptions. *ISMIR*, 669–674.

[2] Bello, J. P. (2007). Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. *ISMIR*.

[3] Bello, J., & Pickens, J. (2005). A Robust Mid-Level Representation for Harmonic Content in Music Signals. *ISMIR*.

[4] Burgoyne Laurent, John Ashley, Pugin Corey, K. I. F. (2007). A cross-validated study of modelling strategies for automatic chord recognition. *ISMIR*, 251–254.

[5] Chen, R., Shen, W., Srinivasamurthy, A., & Chordia, P. (2012). Chord recognition using duration-explicit hidden Markov models. *ISMIR*.

[6] Chew. (2000). Towards a mathematical model of tonality. Massachusetts Institute of Technology.

[7] Cho, T., & Bello, J. (2011). A feature smoothing method for chord recognition using recurrence plots. *ISMIR*, 651–656.

[8] Cho, T., & Bello, J. (2013). Large vocabulary chord recognition system using multi-band features and a multi-stream HMM. *Music-Ir.org*.

[9] Cho, T., Weiss, R., & Bello, J. (2010). Exploring common variations in state of the art chord recognition systems. *Proceedings of the Sound and Music Computing Conference*.

[10] Fujishima, T. (1999). Realtime chord recognition of musical sound: a system using Common Lisp Music. *International Computer Music Conference*.

[11] Gómez, E. (2006). Tonal description of music audio signals. Unpublished doctoral dissertation, Universitat Pompeu Fabra.

[12] Goto, M. (2004). A real-time music-scene-description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, *43*(4), 311 329.

[13] Goto, M., & Muraoka, Y. (1999). Real-time beat tracking for drumless audio signals: chord change detection for musical decisions. *Speech Communication*, *27*(3-4), 311 335.

[14] Haas, B. de, Magalhães, J., & Wiering, F. (2012). Improving audio chord transcription by exploiting harmonic and metric knowledge. *ISMIR*.

[15] Haas, W. B. de, Veltkamp, R. C., & Wiering, F. (2008). Tonal pitch step distance: a similarity measure for chord progressions.

[16] Harte, C. (2010). *Towards automatic extraction of harmony information from music signals*. Doctoral dissertation. Queen Mary, University of London.

[17] Harte, C., & Sandler, M. (2005). Automatic chord identification using quantised chromagram. *Audio Engineering Society Convention*, (2), 2 4.

[18] Harte, C., Sandler, M., & Gasser, M. (2006). Detecting harmonic change in musical audio. *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, 21.

[19] Khadkevich, M., & Omologo, M. (2011). Time-frequency reassigned features for automatic chord recognition. *Acoustics, Speech and Signal Processing*, 181 184.

[20] Lerdahl, F. (2001). *Tonal Pitch Space*.

[21] Mauch, M. (2010). *Automatic chord transcription from audio using computational models of musical context*.

[22] Mauch, M., & Dixon, S. (2010). Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech and Language Processing*, *18*(6), 1280 1289.

[23] McVicar, M., Santos-Rodríguez, R., Ni, Y., & Bie, T. De. (2014). Automatic chord estimation from audio: a review of the state of the art. *IEEE Transactions on Audio, Speech and Language Processing*, *22*(2), 1 20.

[24] Müller, M., & Ewert, S. (2011). Matlab implementations for extracting variants of chroma-based audio features. *ISMIR*, 1 6.

[25] Murphy, K. P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. Doctoral dissertation. UC Berkeley.

[26] Ni, Y., & McVicar, M. (2012). An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech & Language Processing*.

[27] Oudre, L., Grenier, Y., & Févotte, C. (2009). Chord recognition using measures of fit, chord templates and filtering methods. *Applications of Signal Processing to Audio and Acoustics*, 9 12.

[28] Papadopoulos, H., & Peeters, G. (2007). Large-scale study of chord estimation algorithms based on chroma representation and HMM. *Content-Based Multimedia Indexing*.

[29] Papadopoulos, H., & Peeters, G. (2011). Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(1), 138 152.

[30] Pardo, B., & Birmingham, W. P. (2002). Algorithms for chordal analysis. *Computer Music Journal*, *26*(2), 27 49.

[31] Pauwels, J., & Peeters, G. (2013). Evaluating automatically estimated chord sequences. *Acoustics, Speech and Signal Processing*.

[32] Pauws, S. (2004). Musical key extraction from audio. *ISMIR*.

[33] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257 286.

[34] Rhodes, C., Lewis, D., & Müllensiefen, D. (2009). Bayesian model selection for harmonic labelling. *Mathematics and Computation in Music*, 107-116.

[35] Rocher, T., Robine, M., Hanna, P., & Oudre, L. (2010). Concurrent estimation of chords and keys from audio. *ISMIR*, *1*(ISMIR), 141 146.

[36] Ryynänen, M., & Klapuri, A. (2008). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, *32*(3), 72 86.

[37] Sheh, A., & Ellis, D. (2003). Chord segmentation and recognition using EM-trained hidden Markov models. *ISMIR*.

[38] Shepard, R. N. (1964). Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, 36, 2346.

[39] Su, B., & Jeng, S. (2001). Multi-timbre chord classification using wavelet transform and self-organized map neural networks. *Acoustics, Speech, and Signal Processing*, 3377 3380.

[40] Sumi, K., Itoyama, K., Yoshii, K., & Komatani, K. (2008). Automatic chord recognition based on probabilistic integration o chord transition and bass pitch estimation. *ISMIR*, 39 44.

[41] Takuya Yoshioka, Tetsuro Kitahara, Kazunori Komatani, Tetsuya Ogata, H. G. O. (2004). Automatic chord transcription with concurrent recognition of chord symbols and boundaries. *ISMIR*.

[42] Temperley, D., & Clercq, T. (2013). Statistical analysis of harmony and melody in rock music. *Journal of New Music Research*.

[43] Varewyck, M., Pauwels, J., & Martens, J.-P. (2008). A novel chroma representation of polyphonic music based on multiple pitch tracking techniques. In *Proceeding of the 16th ACM international conference on Multimedia* (p. 667). New York, New York, USA: ACM Press.

[44] Zenz, V., & Rauber, A. (2007). Automatic chord detection incorporating beat and key detection. *Signal Processing and Communications*, 2 5.

[45] Klapuri, A. (2006). Introduction to Music Transcription. Springer Science + Business Media.

[46] Patel, A.D. (2003). Language, music, syntax and the brain. *Nature Neuroscience*, 6(7), 674 681.

[47] Thompson, W.F. (1993). Modelling perceived relationships between melody, harmony, and key. *Perception & Psychophysics*, 53(1), 13 24.

[48] Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*.

[49] Westwood, P. (1997). Bass Bible: a world history of styles and techniques. AMA Verlag.

[50] Sailer, C., & Rosenbauer, K. (2006). A bottom-up approach to chord detection. *International Computer Music Conference Proceedings*, 612–615.

[51] Shenoy, A., & Wang, Y. (2005). Key, chord and rhythm tracking of popular music recordings. *Computer Music Journal*, *29*(3), 75–86.

[52] Çataltepe, Z., & Altinel, B. (2007). Hybrid music recommendation based on different dimensions of audio content and an entropy measure. *Eusipco 2007 Conference, Poznan, Poland*.

[53] Cho, T., & Bello, J. (2011). A feature smoothing method for chord recognition using recurrence plots. *ISMIR*, 651–656.

[54] Cohen, L. (1989). Time-frequency distributions-A review. *Proceedings of the IEEE, (July), 941-981.*

[55] Goto, M. (2004). Development of the RWC music database. *Proceedings of the 18th International Congress on Acoustics*, (April), 553–556.

[56] Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2002). RWC Music Database: Popular, Classical and Jazz Music Databases. *ISMIR*, (October), 287–288.

[57] Goto, M., & Hayamizu, S. (1999). A real-time music scene description system: Detecting melody and bass lines in audio signals. *IJCAI Workshop on Computational Auditory Scene*, (August), 31–40.

[58] Harte, C., Sandler, M., Abdallah, S., & Gómez, E. (2005). Symbolic representation of musical chords: a proposed syntax for text annotations. *ISMIR*, 66–71.

[59] Mbikayi, H. (2013). Toward Evolution Strategies Application in Automatic Polyphonic Music Transcription using Electronic Synthesis. International *Journal of Advanced Computer Science and Applications*, *4*(3), 244–249.

[60] Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *Advances in Neural Information Processing Systems*, 2643-2651.

[61] Ryynanen, M., & Klapuri, A. (2007). Automatic bass line transcription from streaming polyphonic audio. International Conference on Acoustics, Speech, and Signal Processing, (April)

[62] Ryynänen, M., & Klapuri, A. (2008). Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, *32*(3), 72 86.

[63] Salamon, J. (2008). *Chroma-based predominant melody and bass line extraction from music audio signals*. Unpublished master thesis, Universitat Pompeu Fabra.

[64] Salamon, J., & Gómez, E. (2009). A chroma-based salience function for melody and bass line estimation from music audio signals. *6th Sound and Music Computing Conference*, (July), 23 25.

[65] Salamon, J., & Gómez, E. (2012). Melody extraction from polyphonic music signals using pitch contour characteristics. IEEE Transactions on *Audio, Speech, and Language Processing*, 20(6), 1759 1770.

[66] Salamon, J., Serra, J., & Gómez, E. (2013). Tonal representations for music retrieval: from version identification to query-by-humming. *International Journal of Multimedia Information Retrieval, 2(1), 45-58.*

[67] Lindsay, B.G. (1995). Mixture Models: Theory, Geometry, and Applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 5, Institute of Mathematical Statistics, Hayward

[68] Babbitt, M. (1955). Some aspects of twelve-tone composition. *The score and IMA Magazine*, 12, 53-61.

[69] Bogdanov, D. et. al. (2013). ESSENTIA: an Audio Analysis Library for Music Information Retrieval. *International Society for Music Information Retrieval Conference (ISMIR)*, 493-498.

[70] Davies, M.E.P., & Plumbley, M.D. (2007). *Context-dependent beat tracking of musical audio. Audio, Speech and Language Processing, IEEE Transactions*, 15(3), 1009-1020.

[71] Noland, K., Sandler, M. (2007). Signal Processing Parameters for Tonality Estimation. *Proceedings of Audio Engineering Society 122nd Convention*, 7155

[72] Klapuri, A. (2006). Multiple fundamental frequency estimation by summing harmonic amplitudes. *International Society for Music Information Retrieval Conference (ISMIR)*, 216-221

[73] Dempster, A.P., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B*, 39(1), 1-38.

[74] Billier, S. (2000). *Le déchiffrage ou l'art de la première interprétation*. Paris, France: Alphonse Leduc.

APPENDIX A

Music Database File List

The following list contains the 66 song IDs of the RWC Popular Music Database songs used in our evaluation of the **bass extraction task:**

RM-P001, RM-P002, RM-P004, RM-P006, RM-P007, RM-P008, RM-P011, RM-P012 RM-P014, RM-P016, RM-P017, RM-P018, RM-P019, RM-P020, RM-P021, RM-P022, RM-P023, RM-P024, RM-P025, RM-P026, RM-P027, RM-P028, RM-P032, RM-P034, RM-P035, RM-P036, RM-P037, RM-P039, RM-P040, RM-P041, RM-P042, RM-P044, RM-P046, RM-P047, RM-P048, RM-P049, RM-P050, RM-P051, RM-P052, RM-P054, RM-P055, RM-P058, RM-P059, RM-P061, RM-P063, RM-P064, RM-P065, RM-P067, RM-P068, RM-P069, RM-P070, RM-P081, RM-P083, RM-P084, RM-P085, RM-P086, RM-P087, RM-P088, RM-P089, RM-P091, RM-P092, RM-P093, RM-P094, RM-P096, RM-P097, RM-P100.

The following list contains the 84 song IDs of the RWC Popular Music Database songs used in our evaluation of the **bass transcription task:**

RM-P001, RM-P002, RM-P003, RM-P004, RM-P005, RM-P006, RM-P007,

RM-P008, RM-P009, RM-P011, RM-P012 RM-P013, RM-P014, RM-P015,

RM-P016, RM-P017, RM-P018, RM-P019, RM-P020, RM-P021, RM-P022,

RM-P023, RM-P024, RM-P025, RM-P026, RM-P027, RM-P028, RM-P029,

RM-P030, RM-P031, RM-P032, RM-P035, RM-P036, RM-P037, RM-P039,

RM-P040, RM-P041, RM-P042, RM-P043, RM-P044, RM-P045, RM-P046,

RM-P047, RM-P048, RM-P049, RM-P050, RM-P051, RM-P052, RM-P053,

RM-P054, RM-P055, RM-P057, RM-P058, RM-P059, RM-P060, RM-P061,

RM-P062, RM-P063, RM-P064, RM-P065, RM-P066, RM-P067, RM-P068,

RM-P069, RM-P070, RM-P081, RM-P082, RM-P083, RM-P084, RM-P085,

RM-P086, RM-P087, RM-P088, RM-P089, RM-P090, RM-P091, RM-P092,

RM-P093, RM-P094, RM-P095, RM-P096, RM-P097, RM-P098, RM-P100.

The following list contains the album names by the Beatles of the Isophonics Database that have been used in our evaluation of the **chord estimation task:**

Please Please Me, With the Beatles, Help!, Rubber Soul, Revolver, Sgt. Pepper's Lonely Hearts Club Band, Magical Mystery, The Beatles (the white album), Let It Be