



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

specjalność: Projektowanie systemów informatycznych

Praca dyplomowa - magisterska

Digital signal processing for music transcription

Michał Matuszczyk

słowa kluczowe:

Audio signal processing

Harmony extraction

Chord detection

krótkie streszczenie:

This thesis is concerned with automatic harmony extraction from audio recordings with a special emphasis on western popular music. Main attention is put on automatic chord detection and algorithms designed to perform this task. This thesis contains also description and evaluation of novel algorithm designed to extract harmonic data.

opiekun pracy	Dr inż. Marek Kopel
dyplomowej	<i>Tytuł/stopień naukowy/imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do:*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczętka wydziałowa

Wrocław 2016

Table of Contents

Abstract.....	4
Wstęp.....	4
1. Introduction.....	5
1.1. Motivation	5
1.2. Objectives and scope of the thesis	6
1.3. Thesis structure	7
2. Scientific background	8
2.1. Musical fundamentals	8
2.1.1. Frequency and pitch.....	8
2.1.2. Harmony	11
2.1.3. Chord inversions and extensions	12
2.2. State of the art.....	13
2.2.1. Template based approach using Pitch Class Profiles	13
2.2.2. Hidden Markov Models	14
2.2.3. Wavelet Transform	15
2.2.4. Neural networks.....	15
3. System overview	16
3.1. System overview	16
3.2. Feature extraction	18
3.2.1. Time to frequency transformation	18
3.2.2. Feature extraction	21
3.3. Chord recognition.....	22
3.4. Postprocessing.....	23
4. Improving system performance.....	25
4.1. Evaluation methods.....	25
4.1.1. Chord overlap	25
4.1.2. Chord vocabulary.....	26
4.2. Pitch range	27
4.3. Constant Q transform precision	32
4.4. History length	35
4.5. Fine tuning of neural network.....	38
4.6. Chord sequence post processing	40
5. Comparison with state of the art systems	44
5.1. MIREX 2015	44
5.2. Root only competition	45
5.3. Major, minor competition.....	46
5.4. Major, minor with inversions competition.....	47
5.5. Major, minor and sevenths competition.....	48
6. Summary and future work.....	49
6.1. Summary	49
6.2. Future work	50

6.2.1.	Additional music information.....	50
6.2.2.	Adding additional modules.....	50
6.2.3.	Extended data set	51
6.2.4.	Implementation enhancements	51
Bibliography		52
List of figures.....		54
List of tables.....		56
Appendix A - Example transcription		57

Abstract

This thesis is concerned with automatic harmony extraction from audio recordings with a special emphasis on western popular music. Main attention is put on automatic chord detection and algorithms designed to perform this task. Automatic chord detection is a part of a large problem generally described as automatic music transcription. In this thesis we present a novel approach for automatic harmony detection based on concepts from neo-Riemannian music theory. The approach is based on feature extraction from frequency spectrum of an audio signal. At the end of the thesis, presented algorithm is evaluated and compared to existing state of the art solutions participating in Music Information Retrieval Evaluation eXchange 2015 (MIREX 2015) contest. Results are presented for several different evaluation measures based on the “Audio Chord Estimation” competition.

Wstęp

Tematem niniejszej pracy magisterskiej jest wykorzystanie analizy sygnałów cyfrowych w procesie tworzenia tabulatur. Główna uwaga została poświęcona automatycznemu wykrywaniu akordów i algorytmach przeznaczonych do tego zadania. W niniejszej pracy przedstawiono nowatorskie podejście do tego zadania. Prezentowana metoda opiera się na ekstrakcji cech z widma częstotliwościowego sygnału audio. W ramach pracy, przedstawiony algorytm zostaje porównany z istniejącymi w literaturze rozwiązaniami biorącymi udział w konkursie Music Information Retrieval Evaluation eXchange 2015 (MIREX 2015). Wyniki prezentowane są w kilku kategoriach opartych na konkurencji "Audio Chord Estimation".

1. Introduction

This thesis we address automatic extraction of harmony information, specifically chords labels from audio signal. In this chapter, we explain the motivation and aim of our work in (Sections 1.1 and 1.2) and provide an overview of the material presented in the thesis (Section 1.3).

1.1. Motivation

In recent years automatic extraction of various musical properties from audio has become a popular research area in the computer science field called music information retrieval (MIR). Much work has been carried out for automatic obtaining key [1], harmony [2], beat and tempo [3] of an audio piece.

Main motivation behind research focused on automatic detection of harmony information is to help musicians in annotation of musical pieces. As an expression of the need of existence of such tool we can use websites such as Ultimate Guitar¹ or Songster² where thousands of musicians transcribe music so is it easy accessible to others. However even on such sites it may be hard to find some rare performances (for example covers of a song created by another artist). Also the quality of such home-made transcriptions may vary dependent on the musician who created it. High quality tool for automatic creation of such transcription is highly desired.

Chord progressions define the harmonic structure of a musical piece. As this they can be used as a base stage for more advanced processing. Possible practical applications include: automatic music transcription to staff notation [4], cover song detection [5] and genre classification [6].

As an additional motivation we can state that music transcription is interesting challenge from a perspective of a computer science, especially in the field of artificial intelligence. The goal of such research would be to create robust algorithm that understands human perception of music and applies this

¹ <https://www.ultimate-guitar.com>

² <http://www.songsterr.com>

knowledge to “listen” and transcribe specific song. It is also worth mentioning that even now it is not completely understood how individual mind understands sequences of various sounds as pleasant or unpleasant. Creating a computer model of such phenomenon may help understand processes that occur in our brain.

1.2.Objectives and scope of the thesis

The objective of this thesis is to design, implement and evaluate method for automatic music transcription from arbitrarily instrumented music. We want to combine music theoretical knowledge with current developments in the field of machine learning to properly classify various segments of audio data by assigning them musically valid chord label. This thesis will also look at the possible ways to improve said detection quality.

The stated objective is a practical one. As a result, this thesis describes a collection of techniques derived from different disciplines that aid automatic chord transcription. We use music theory, theory of music perception [7], digital signal processing and probability theory, with the aim of improving automatic chord recognition process.

The scope of the thesis consists of the following elements:

- Development of a method of automatic harmony extraction from audio recordings where part of an audio signal with specified characteristics will be classified as an exactly one chord value.
- Carrying out simulation tests in order to verify the correctness and performance of the proposed algorithm as compared to methods known in the literature.

1.3. Thesis structure

In this section we outline the structure of the thesis, chapter by chapter.

Chapter 2: Scientific background. This chapter introduces important background information on fundamental aspects of music theory from the perspective of chord recognition task. This section also contains description of current state of the art methods.

Chapter 3: System overview. In this chapter we propose system for automatic chord recognition that is central part of this thesis. The novelty of the approach is that it uses *Constant-Q* transform combined with relatively simple feed forward neural network with fine tuned parameters that can archive state of the art performance. Various methods of post processing will be introduced to further enhance system detection capabilities.

Chapter 4: Improving system performance. This chapter further enhances system presented in chapter 3. Main goal of this section is to improve detection performance. Most of the chapter is dedicated to finding best parameters for feature extraction from audio signal.

Chapter 5: Comparison with state of the art systems. This chapter contains empirical results. Described system is compared to state of the art entrants of the Music Information Retrieval Evaluation eXchange 2015 (MIREX 2015) contest.

Chapter 6: Summary and future work. This chapter provides a summary of the achievements of this thesis. We end by outlining planned work and, more generally, what we deem worthwhile future work in the area of chord recognition.

2. Scientific background

This chapter provides background information on fundamental concepts on music theory which are crucial in understanding of the thesis.

The final section of this chapter is devoted to current state of the art methods, where most common approaches to chord recognition task are described.

2.1. Musical fundamentals

In this section we will define musical terms of reference used throughout the rest of the thesis and introduce fundamental concepts required for discussions in later chapters.

2.1.1. Frequency and pitch

The fundamental signal feature related to tonality is **frequency**. The frequency f is the number of times that a cycle is repeated per second. For example, sinusoidal wave with a frequency equal to 440 Hz performs 440 cycles per second.

The subjective counterpart of frequency is called **pitch**. Pitch is perceived by humans in terms of its highness or lowness. For example, most women have higher voices than men.

Semitone	Enharmonics
C	
C#	D \flat
D	
D#	E \flat
E	
F	
F#	G \flat
G	
G#	A \flat
A	
A#	B \flat
B	

Table 2.1 Semitones with most common enharmonics

In this thesis we will use standard music notation in which a pitch is defined by a **pitch class** and an **octave** number.

A **pitch class** (sometimes called chroma) contains a natural name and an optional **accidental**. A sharp accidental (#) raises the pitch by one **semitone**, a flat accidental (b) lowers it by one semitone. The 12 semitones listed in ascending order beginning with C are: C, C#, D, D#, E, F, F#, G, G#, A, A#, B. The same pitch can be named with several names. This phenomenon is called **enharmonics**. Most common enharmonics are shown in Table 2.1.

Inter-val	Name	Exam-ple
0	perfect unison	c-c
1	minor second	c-c#
2	major second	c-d
3	minor third	c-d#
4	major third	c-e
5	perfect fourth	c-f
6	augmented fourth, diminished fifth	c-f#
7	perfect fifth	c-g
8	minor sixth	c-g#
9	major sixth	c-a
10	minor seventh	c-a#
11	major seventh	c-b
12	perfect octave	c1-c2

Table 2.2 First 12 intervals with examples for semitone C

The relationship (distance) between two pitches is called **interval**. As we can see in table 2.2 pitch raised 12 times by a single interval returns back to its original chroma (as there are only 12 semitones). Thus the second part of a pitch name is an **octave**. For example, pitch A4 means natural (as it does not have an optional accidental addition) semitone A in the 4 octave. Pitch that is exactly one octave higher is called A5 and the one octave lower is A3.

System in which octave is divided into twelve parts with equal frequency rates is called **equal temperament system**. Raising pitch by a single octave doubles its frequency. As the octave has a ratio of two, the ratio of frequencies

between two adjacent semitones is the twelfth root of two. This means that the pitch scale is logarithmic, i.e. adding a certain interval corresponds to multiplying a fundamental frequency by a given factor. In other words changing pitch by a single semitone equals changing frequency ratio of $\sqrt[12]{2}$, or approximately 1.059:

$$\frac{f_1}{f_2} = \sqrt[12]{2} \quad (2.1)$$

To find the frequency of any pitch the following formula can be used:

$$P_n = P_a \cdot 2^{\frac{n-a}{12}} \quad (2.2)$$

Where n is the distance of the desired pitch from reference frequency a .

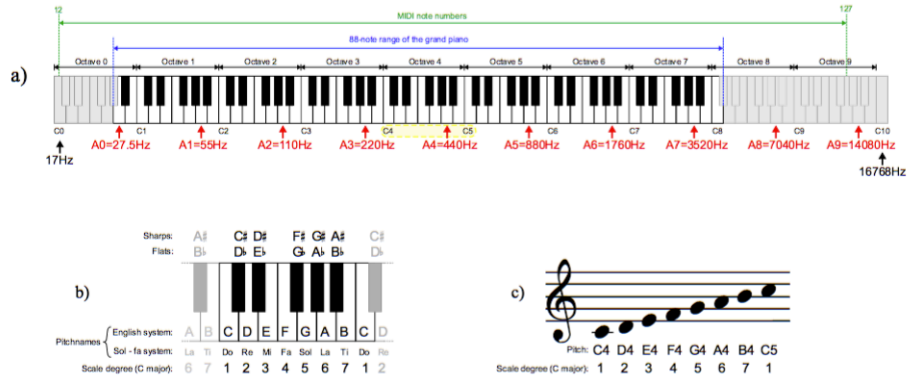


Figure 2.1 Pitches as found on grand piano keyboard (a). Pitch names from single octave on piano keyboard (b). Pitches from C4 to C5 on musical staff. [8]

The A4 - 440Hz is considered as the **standard reference frequency**, although we cannot assume that all of the musicians will always be tuned to this frequency. Sometimes players tune their instruments to each other to archive more interesting effects. The standard 88-key grand piano keyboard (as shown in figure 2.1a) has keys between pitch A0 at 27.5Hz and C8 at 4186Hz. White keys on the piano keyboard denotes natural semitones and black keys accidentals (in a way shown in figure 2.1b).

Pitch with assigned duration is called a **note**. The higher the note is on the staff the higher is its pitch (as shown in figure 2.1c).

2.1.2. Harmony

Harmony is a term that denotes the simultaneous sustain of multiple notes. Such sound is called **chord**. Sequence of chords over time is called **chord progression**. In this work we will consider only the aspects of the harmonic content related to the combination of notes into chords.

Chord	Root	Third (III)	Fifth (V)
C Major	C	E	G
C Minor	C	D#	G
C Augmented	C	E	G#
C Diminished	C	D#	F#

Table 2.3 Triad chords for based on C root

Chords are characterized by the intervals between notes they contain. To properly identify chord we need a **root** pitch and chord **family** (chord type). For example, the chord formed with tones C, E and G simultaneously would be given the chord named “C Major”, where “C” is the root (note that it does not contain the octave) and “Major” is the chord family. Other examples of chords sharing the same family would be “D Major”, “A Major” etc.

The most fundamental chords are triads, consisting of only three notes. Except for the root note they contain also a third (III) and a fifth (V). A triad containing, root note, major third (as an interval between root and III) and minor third (III-V) is called a **major chord**. **Minor chords** consist of the root note, a minor third and a major third. Other common triad families are augmented, diminished chords. An **augmented chord** is built up of two major thirds. By contrast, the **diminished chord** is built up of minor thirds. Examples of different chord types of C triads are illustrated in table 2.3.

The degree to which a combination of notes is perceived to be acceptable or pleasing in a given musical context is called **consonance**. The opposite term i.e. how unpleasant it is, is called **dissonance**. All of the notes used in a chord have to create consonance so that the played sound is pleasant to the human ear. Discussion why we use these exact intervals for chord creation is out of scope of this thesis.

The process of finding chords to an existing melody is called **harmonization** (or transcription). Though there are several rules for harmonizing melodies,

there is rarely one "true" progression. Result of transcription depends on musician who created it as such piece would be adjusted to his musical preferences and skills.

Usually for a single piece of music there is a single major or minor chord, that represents the **harmonic (tonic) center**. This special chord is called the **key** of the piece of music. Tonality is the system of composing music around such a tonal center. The word tonality is frequently used as a synonym for key. In classical music the key is often named in the title (e.g. Beethovens 5th Symphony in C-Minor). A piece of music does not need to hold one key over the whole song duration. Changing the tonal center of a piece of music is called **modulation**.

2.1.3.Chord inversions and extensions

The lowest (in terms of frequency) note played in chord is called bass note. Usually the bass note is the same as root. Although, sometimes to archive desired musical effect (on the chord sound or the way the chord functions in progression) other tone may be used as the bass note. Which degree of the chord becomes the bass note defines the chord's **inversion**. The inversion of a chord is often written by specifying the chord name and the bass note. For example, the first inversion of a C major chord would be written C major/E and is referred to as "C over E". Despite the fact that now the lowest frequency will be the frequency associated with the E tone, it still will be an C major chord as the overall chord tonality has not been changed (it will still contain: C, E, G tones).

Although, all of the chords introduced in chapter "2.1.2 Harmony" were triads (major, minor, augmented, diminished) more chords are defined in music theory. By addition of certain pitches new chords with a different structure can be generated from the basic triads. This is called **extension**. The most important of these are the seventh chords which originate by addition of an additional third called seventh (VII). Adding a major 7th to a major triad produces the **Major 7th** chord. Adding the minor 7th to the major triad produces a 7th often referred to as a **dominant 7th** chord. Adding the minor 7th degree to a minor triad produces the **minor 7th** chord.

Further extensions are also possible although they exceed the original octave. By adding yet another interval of a third onto a 7th chord a 9th chord is

formed. Such chords are rarely used in western popular music and will not be taken into scope of this thesis.

It is also worth mentioning that the frequencies of occurrence of specific chord families heavily depend on the musical genre of an audio piece. For example, 7th chords are more popular in blues than in rock.

2.2.State of the art

Automatic chord transcription is a classification problem. For a signal with given characteristics adequate chord label is assigned. In most cases the process includes two successive steps:

- **feature extraction** which captures harmonic information contained in the musical piece
- **recognition process** which task is to match extracted features with predefined chord labels

2.2.1.Template based approach using Pitch Class Profiles

The first step of chord recognition consists in the extraction of relevant audio features from musical content. This process is based on either on the Short Time Fourier Transform [8] or on the Constant Q Transform [9]. The exact features used in chord transcription differ from a method to method but in most of the cases they are variants of the Pitch Class Profiles [2]. Those features (called chroma vectors) are 12 dimensional vectors. Every component represents the spectral energy of a semitone on the chromatic scale regardless of the octave. Sequence of these chroma vectors over time are called chromagrams.

The recognition process relies on chord templates. Each template contains information about single chord in a form of binary Chord Type Template (CTT).

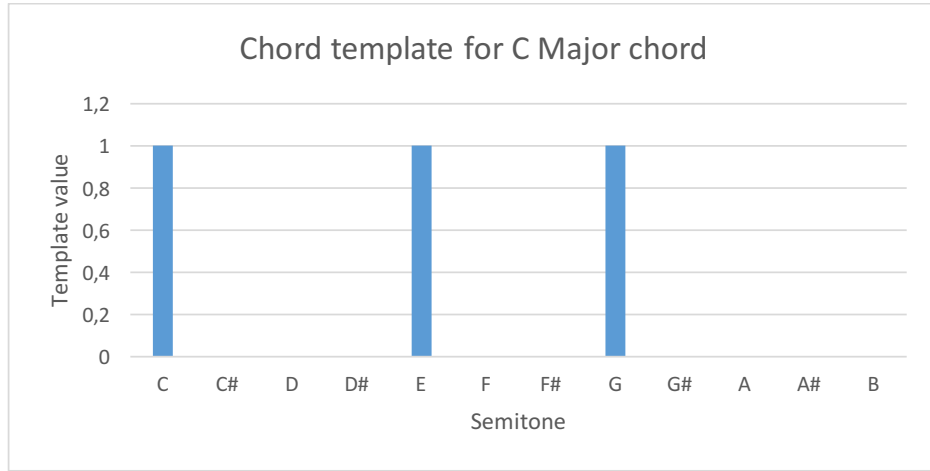


Figure 2.2 Chord type template for C Major chord (semitones C, E and G).

In order to perform chord recognition inner product is taken between the PCP obtained from audio signal and all of the predefined templates. The chord template that returns the highest value is chosen.

2.2.2. Hidden Markov Models

In another approach to chord recognition hidden Markov Models are used [9]. A hidden Markov model is a statistical model of a finite state machine in which the state transitions obey the Markovian property, such that the future state is dependent only on the present state and not on the past. This feature is very useful in chord recognition as the current chord value is highly dependent on the previous one. It can be very easy derived from music theory that although we don't know how likely is to C Major to be played before D Major, it is even less likely for this chord to be before F Minor. This observation is the foundation of using HMM for chord recognition.

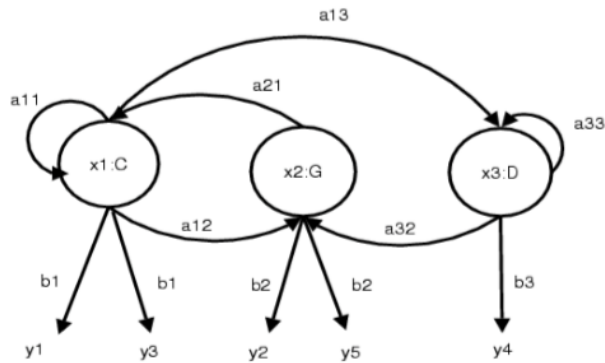


Figure 2.3 Hidden Markov model for chord sequences

As shown in figure 2.3 each state (x) generates an observable output (y) with a certain probability (b). Transition probabilities (a) are assigned to each pair of states. While the sequence of states is not visible ("hidden"), the outputs can be observed and used to draw conclusions on the current state.

In terms of chord recognition each state is responsible for a single chord. The output is defined as the Pitch Class Profile (PCP) that represents the spectrum produced by playing given chord. Transition and output probabilities are obtained using expectation maximization algorithm (EM) on hand-labelled chord sequences. Finally, Viterbi algorithm is used to find the most likely sequence of chords that could have generated the observed PCP sequence.

2.2.3.Wavelet Transform

Although different variations of PCP feature vectors are by far the most common representation of audio content other approaches are proposed [12]. The most notable one uses wavelet transform calculated on the full frequency spectrum. The result of the wavelet transform is a direct input for neural-network used as an chord-classification unit. The neural network consists of a self-organized map (SOM) with one node (neuron) for every chord that shall be recognizable. Initial synaptic weights of each neuron on the SOM are set according to music theory. The SOM then learns from a set of training data using supplied information.

The authors report an accuracy rate of 100%. However, as their test set consists only of 8 music pieces from Beethoven Symphony, their result cannot be regarded as comparative to other methods.

2.2.4.Neural networks

Neural networks are also popular among Music Information Retrieval (MIR) researchers. As described in section devoted to hidden Markov models the exact value of chord in sequence is highly dependent on its predecessors. Because of this property recurrent neural networks (RNN) are popular for chord recognition as they use time-dependent context information.

Boulanger-Lewandowski, Yoshua Bengio and Pascal Vincent [13] used mix of deep belief networks (DBN) and RNN. As input to their system they used whitened Short Time Fourier Transform spectrum. DBN was used to automatically construct complex abstractions based on this data, for example for estimating active pitches. Information extracted by DBN were used as an input for RNN. Final chord sequence was obtained by using dynamic programming to find most probable arrangement.

3. System overview

In the previous chapter we gave an overview of an existing approaches to chord transcription. In this chapter we will introduce a new chord detection algorithm that has been designed for this thesis. First section gives an overview over the developed system. Following sections describe each module in more details: Section 3.2 deals with obtaining frequency information from audio signal, Section 3.3 describes feature extraction, Section 3.4 focuses on neural network used for chord recognition. Finally, Section 3.5 deals with post processing method used to maximize system performance.

3.1. System overview

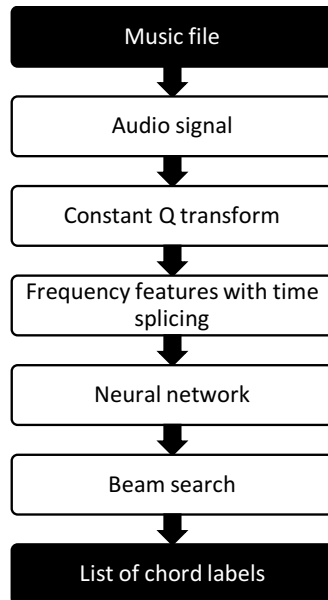


Figure 3.1 Overview of the described chord recognition system.

Figure 3.1 depicts the flow chart diagram of proposed chord detection system. The first and the last blocks (highlighted with black color) represent system input and output. The algorithm takes audio file as an initial parameter. As a result, system will output a list of chord labels with information about their starting and finishing times. Example of a system output is presented in Appendix A.

First step in presented algorithm is to retrieve digital signal encoded in audio file. Support for following file formats was implemented: MPEG Layer 3 (.mp3), MPEG 4 Audio (.aac), WAVE (.wav). Subsequently extracted audio data is stored using Pulse Code Modulation (PCM) method. If needed, at this stage audio signal is also converted to its single channel (monophonic) form. Result of such transformations is called waveform.

To obtain frequency information *Constant-Q transform* [10] is used. This transformation was selected as it produces logarithmically spaced frequency bins which resembles the layout of musical frequencies in equal tempered system. Data in such form serves as a base for extracting features. For this task direct frequency information from *Constant-Q transform* are used in a range of multiple octaves. This allows us to recognize chord extensions and inversions. To include temporal context to system *time splicing* method [11] was implemented. It is worth noticing that presented feature vector values directly correspond to amount of energy concentrated around frequency for a given pitch.

Such extracted features are used as an input to neural network. Role of network is to find likelihood of an event such that the specific chord was played. Network was trained to recognize 25 events (classes): 12 major chords, 12 minor chords and an event that no chord was played. Although in chapter 5 we adapted and tested our design on bigger chord dictionary in the goal of comparing it with state of the art systems that realize such task. Output of the neural network (as a list of chord probabilities for each of the processed time-frames) is then post processed to find most probable progression using beam search algorithm.

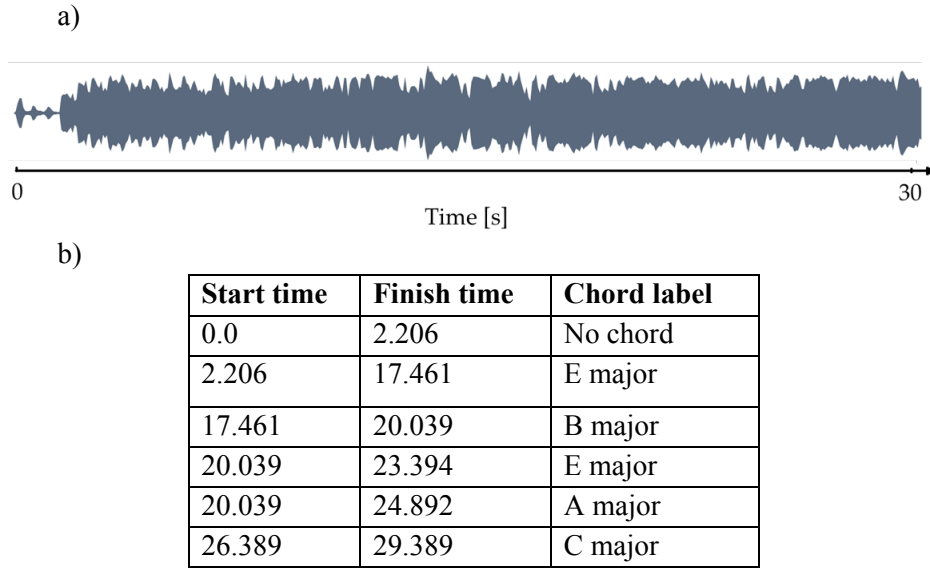


Figure 3.2 Waveform (a) and example of system output (b) for first 30 seconds of “I Saw Her Standing There” by The Beatles.

3.2.Feature extraction

The first stage of our system is feature extraction. It consists of 2 sub-stages: time to frequency transformation using *Constant-Q transform* and calculation of a *time spliced* features described in sections 3.2.1 and 3.2.2.

3.2.1.Time to frequency transformation

The heart of feature extraction process is a *Constant-Q transform* [10] applied to PCM audio signal. *Constant-Q transform* converts data series to the frequency domain just like Short Time Fourier Transform [8]. The difference between them is that *Constant-Q transform* produces logarithmically spaced frequency bins, instead of using the same bin width independently of analyzed frequency as in STFT. This feature makes it perfect for musical applications as this form of audio spectrum resembles the layout of musical frequencies in equal tempered system. This guarantees that identical number of bins per single musical pitch is used across the whole audible spectrum, thus maintaining a constant level of accuracy for each pitch.

Constant-Q transform can be thought as a set of filters (where each filter contains low and high pass filter to separate that of the spectrum) in a frequency domain. The ratio of a filter's bandwidth δf to its center frequency f is called its quality or Q factor.

$$Q = \frac{f}{\delta f} \quad (3.1)$$

For musical analysis, in the goal of using frequency components corresponding to semitone spacing of the equal tempered scale, the k^{th} bin frequency can be defined as:

$$f_k = (\sqrt[\beta]{2})^k f_{min} \quad (3.2)$$

Where f_{min} is a minimal frequency for which we perform analysis. β parameter is responsible for setting number of bins per octave. In equal tempered tuning $\beta = 12$, therefore the centers of frequency bins are separated by the ratio of $\sqrt[12]{2}$. The upper pitch is not defined, although generally it does not make sense to calculate values for bins above the Nyquist frequency (22050 Hz for most common sampling rate for musical data - 44100 samples/sec).

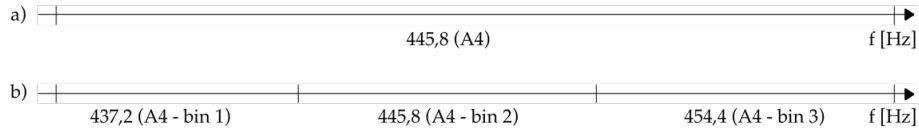


Figure 3.3 Division of audio spectrum using constant Q transform near $A4=440$ Hz reference frequency using 12 (a) and 36 (b) bins per octave.

For frequency analysis in music where tuning may not conform to standard tuning $A4=440$ Hz, β parameter may be set to different values (although always multiplicity of 12) to distinguish between adjacent semitones. As an example resolution of 36 bins per octave is equivalent to three bins per semitone (situation illustrated in Figure 3.3).

The *Constant-Q* spectrum Q_k of the time sequence x_n for a single frequency bin is given by the transform (please note that similarity to discrete STFT is not an accident):

$$Q_k = \frac{1}{N_k} \sum_{n=0}^{N_k} w_{k,n} x_n e^{-j \frac{2\pi Q n}{N_k}} \quad (3.3)$$

where N_k is the analyzed frame length for frequency bin k , w is a windowing function [12] and the frequency is $\frac{2\pi Q n}{N_k}$.

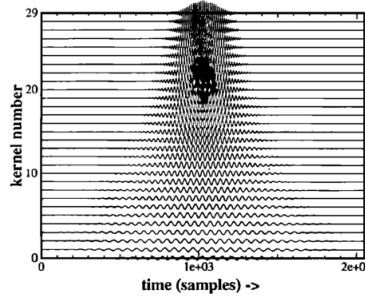


Figure 3.4 Real part of the spectral kernel as presented in (Brown i Puckette 1992).

Although *Constant-Q transform* can be calculated as a series of Discrete Fourier Transforms [10] in our implementation more efficient algorithm is used [16]. The heart of this method is to apply spectral kernel (Figure 3.4) to result of Fast Fourier Transform of an audio data. Using window with appropriate size (lower frequencies needs longer windows) we are able to obtain enough bin density to convert data to log frequency scale. This operation can be performed by multiplication by sparse matrix containing raw windows transformed with FFT. In our implementation we used Hamming windows as a base of spectral kernel.

3.2.2.Feature extraction

Spectrum representation obtained by using *Constant-Q transform* is used in process of creating *feature vectors*. Although raw output of the transformation can be used for this task following optimizations were implemented: *frequency filtering* and *time splicing*.

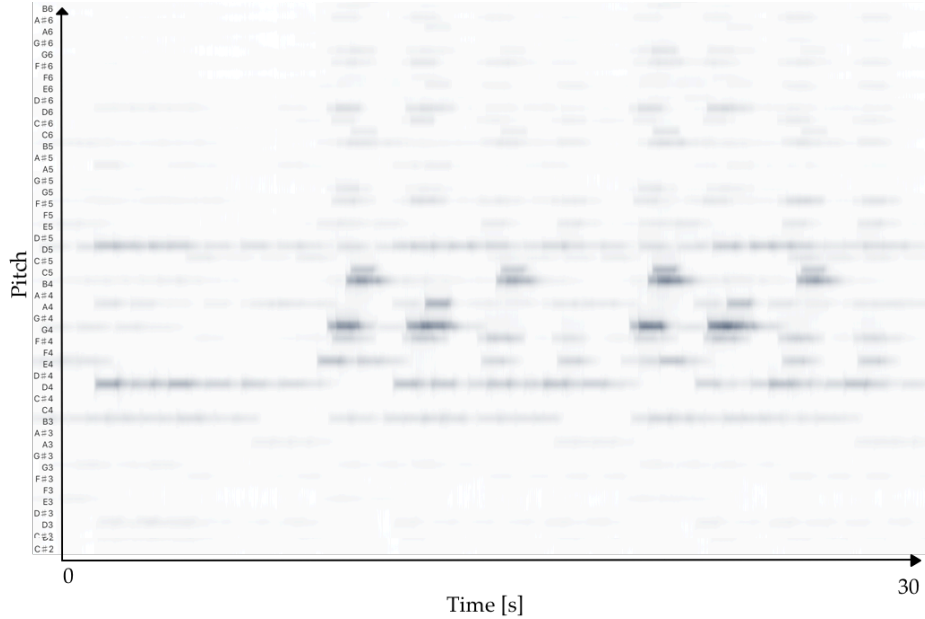


Figure 3.5 Result of *Constant-Q transform* of first 30 seconds of “*Comptine d'un autre été : L'Après-Mid*” by Yann Tiersen.

Frequency filtering is based on the observation that most of frequency information are concentrated in a small portion of audio spectra. For example, in spectrogram shown in figure 3.5 most of the information is stored between B3 and F5 pitches. It is very rare for composers to use notes with pitch greater than C8 (4186.0 Hz). We use this information to separate part of the spectrum that contains most of the harmonic data.

Time splicing is a process of extending the current frame with the data of neighboring frames so together they create one larger superframe. Its role is to provide the recognition system contextual information about the current event (currently played chord) by adding historical values. Since in presented system length of a single time-frame is vastly greater than distance between frames (known as *hop size*) an overlap exists between neighboring superframes.

Such feature values directly correspond to amount of energy concentrated around frequency for a given pitch in a given time-frame. The length of the feature vector can be obtained from equation:

$$L = 12 \cdot n \cdot h \quad (3.4)$$

Where n is number of octaves considered during calculation of the Constant-Q transform and h is the number of frames used in history.

3.3.Chord recognition

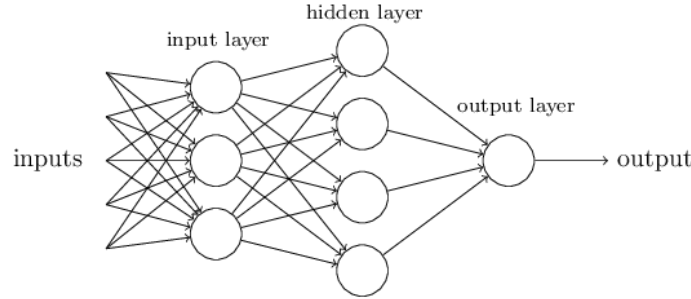


Figure 3.6 Representation of used neural network with single hidden layer.

The heart of an recognition process is neural network [13]. For this task feed-forward neural network was implemented. This type of network is one of the most popular forms of neural networks and as this it will not described in this thesis. General architecture of this structure it is widely available in literature [14]. Ability to calculate (or more of an approximate) almost any function has made neural networks very popular in music applications in recent years.

Our network is trained using supervised learning with annotated audio data. For training purposes, “The Beatles” discography provided as a part of Christopher Harte PhD thesis [15] was used. Network is fine-tuned with respect to a cross-entropy cost:

$$L(v^t, z^t) = - \sum_{j=1}^N z_j^t \log y_j^t + (1 - z_j^t) \log(1 - y_j^t) \quad (3.5)$$

where $y(t)$ is the prediction obtained at the top-most layer and $z \in \{0, 1\}^N$ (N is the number of possible chord values) is a binary vector serving as a training

target at time step t . Note that training target $z(t)$ must have only one active element (one-hot) at a given time step as only one chord can be played.

Errors on a previous layers are obtained by backpropagation. That combined with stochastic gradient descent to minimize cost allows to gradually improve system performance during training period. For better performance L^2 normalization is also applied.

3.4. Postprocessing

To determine most probable progression of chords beam search algorithm is used. Beam search is a breadth-first tree search algorithm where only the w most promising paths (or nodes) at depth t are kept for future examination. In this case depth t means that the current chord sequence (history) contains exactly t chords (one chord label for each traversed tree level). It is assumed that all of the descendants of such node share the same history. In consequence all of the siblings of the current node may differ only by the value of the last chord. By storing cumulative likelihood for each node we can identify most promising paths.

Algorithm 1 Beam search

Find the most likely sequence $\{z^t \in C | 1 \leq t \leq T\}$ given x with beam width $w \leq N^T$.

1. $q \leftarrow$ priority queue
2. $q.insert(0, \{\})$
3. for $t = 1 \dots T$ do
4. $q' \leftarrow$ priority queue of capacity w^*
5. for z in C do
6. for l, s in q do
7. $q'.insert(l + \log P(z(t) = z | x, s), \{s, z\})$
8. $q \leftarrow q'$
9. return $q.max()$

**A priority queue of fixed capacity w maintains (at most) the w highest values at all times.*

Algorithm 1 provides quick overview of steps used in beam search method. Following notation was used: C is chord dictionary and T is the number of feature vectors extracted from audio data. $P(x|z)$ is the probability of chord given chord history z . As this it can be formulated using Bayes rule:

$$P(x^t|z^t) \propto \frac{P(z^t|x^t)}{P(z^t)} \quad (3.6)$$

where $P(z^t|x^t)$ is the output of the classifier and constant terms given x have been removed. That gives the resulting distribution:

$$P(z^t, x^t | \{z^r, r < t\}) \propto \frac{P(z^t|x^t)}{P(z^t)} P(z^t|z^{(t-1)}) \quad (3.7)$$

which depends only on $z^{(t-1)}$ - previous value in sequence. From now is trivial to derive a recurrence relation (as previous value depends on its previous value) to optimize using dynamic programming. This was already done in literature giving rise to the Viterbi algorithm. To estimate probability of chord transitions we used annotated The Beatles discography [15].

It is worth mentioning that $w = I$ reduces algorithm to a greedy search where only single most promising path is traversed. Similarly using $w = NT$ (where N is the number of possible chord values) corresponds to an exhaustive breadth-first search. However, beam search is only a heuristic and unless we traverse all of the paths in tree we can't be sure that discovered result is really the best one (although many real-life applications shown that the approximation is very close and the profit from smaller number of calculations is significant).

4. Improving system performance

In this chapter we will describe methods we used for fine tuning presented chord recognition system. In chapter 4.1 we will provide an overview of the evaluation metrics used in this thesis. Following sections will be devoted to each of the examined parameters: in chapter 4.2 we will look at the influence of the pitch range used during feature extraction step. In chapter 4.3 we show techniques used to improve performance for out of tune musical pieces. Chapter 4.4 will show the importance of introducing temporal data into chord recognition. Finally, chapters 4.5 and 4.6 contain optimization techniques for our neural network and beam search respectively.

4.1. Evaluation methods

4.1.1. Chord overlap

The quality of a chord transcription can be evaluated only in connection to the piece of music it transcribes. Automatic chord transcription tries to imitate the way an experienced human annotator would annotate given audio segment based on its harmonic properties. Because of this it is common practice to compare automatic transcription to some reference (“ground truth”) transcription provided by expert.

Start time	Finish time	Chord label
0.0	2.206	No chord
2.206	17.461	E major
17.461	20.039	B major
20.039	23.394	E major
20.039	24.892	A major

Table 4.1 Example of system output for first 25 seconds of “I Saw Her Standing There” by The Beatles.

Evaluation methods used in this thesis will be based on those used during Music Information Retrieval Evaluation eXchange (MIREX) contest for an “Audio Chord Estimation” competition. For this the output of an algorithm is defined as a list of values with following properties: start time, finish time and chord label (example in Table 4.1). Chord label follows syntax proposed by C. Harte [16].

Typical measure for quality of an automatic chord transcription is Chord Symbol Recall (CSR):

$$CSR = \frac{\text{Total time of properly annotated chords}}{\text{Song duration}} \quad (4.1)$$

When results are calculated for more than one song weighted average will be used. Weight of a single input will be based on its duration according to:

$$WCSR = \frac{1}{N} \sum_{n=1}^N CSR_n \cdot w_n \quad (4.2)$$

Where N is the testing corpus. This measure is commonly known as weighted chord symbol recall (WCSR).

4.1.2. Chord vocabulary

Transcriptions made by human annotators often feature a great amount of detail. For instance the biggest available collection for testing of automatic chord transcription [15] contain hundreds of different chord labels, constructed from around 160 chord types.

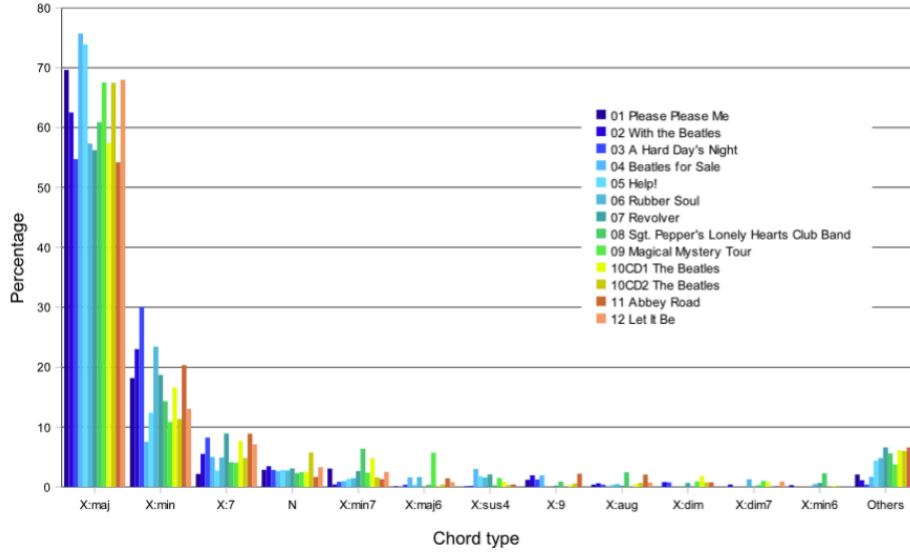


Figure 4.1 Chord structure of *The Beatles* discography [15]

However as shown in Figure 4.1 most of the chords used in modern music are part of a small number of chord families. Therefore, algorithms usually use a limited set of chord classes. Often, only the 12 major and 12 minor chord labels are used with addition of special “no chord” class.

Based on number of recognized chord classes MIREX distinguishes following categories in “Audio Chord Estimation” contest:

- Chord root note only
- Major and minor: {N, maj, min}
- Seventh chords: {N, maj, min, maj7, min7, 7}
- Major and minor with inversions: {N, maj, min, maj/3, min/b3, maj/5, min/5}

Where N always represents “no chord” class. In this thesis we will concentrate on “Major and minor” category.

4.2. Pitch range

As shown in equation 3.4 number of octaves considered during *Constant-Q* transform has proportional relationship to a length of the presented feature vectors. Using larger frequency range gives more features. Knowing that each feature contains the same information (amount of energy concentrated around single pitch) using more features seems to be logical step forward better chord recognition.

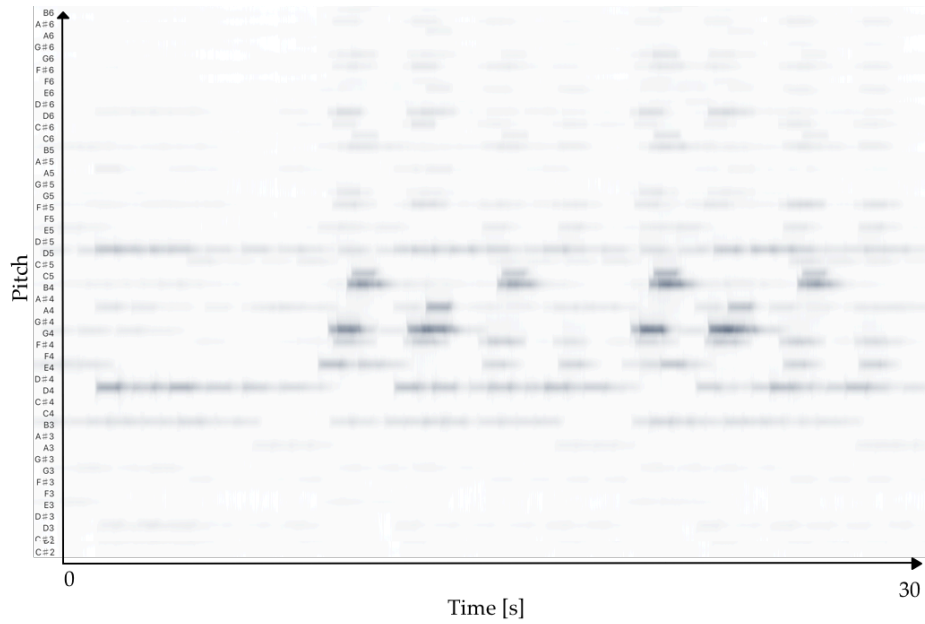


Figure 4.2 Result of Constant- Q transform of first 30 seconds of “Comptine d'un autre été : L'Après-Mid” by Yann Tiersen in a pitch range from C2 to B5.

However, as shown in figure 4.2 most of frequency information are concentrated in a small portion of audio spectra (in this specific case between C3 and F5). This thesis is true for most of the modern musical pieces. It is very rare for composers to use notes with pitch greater than C8 (4186.0 Hz). We could use this information and separate this part of the audio spectrum that contains most of the harmonic data.

Min pitch	Max pitch	Window length	FFT window length
C1	C8	22680,0	32768
C2	C8	11340,0	16384
C3	C8	5670,0	8192
C4	C8	2835,0	4096
C5	C8	1417,5	2048
C6	C8	708,7	1024
C7	C8	354,3	512

Table 4.2 Relation between used pitch range and length of a used time window.

It is also worth noticing that minimal pitch has a significant influence on length of the used time-window according to equation:

$$N_k = \frac{S \cdot Q}{f_k} \quad (4.3)$$

Where N_k is the length of a k window in kernel and S symbolizes audio sample rate. As we can see window length is inversely proportional to frequency represented by k bin. This confirms our intuition that smaller frequencies need bigger window to obtain desired amount of spectral data. Examples of this relationship is shown in Table 4.2. In presented system *Constant-Q* transform was calculated using efficient method presented by Brown and Puckette [16] which requires that length of each window have to be power of 2 (as it uses Fast Fourier Transform with such radix). In order to satisfy this condition each window length must be rounded to the next power of 2 (also shown in table 4.2 in “FFT window length” column) to obtain sufficient amount of spectral data.

We could assume that using smaller number of octaves (thus reducing frequency range of transformation) will speed up calculations by introducing lower number of features in consecutive steps. However, using too limiting range will harm detection ability as it would remove frequencies containing important data from presented feature vectors.

To establish most optimal pitch range, the reference value (meaning situation when all of the spectral data was present) was set to pitches between C1 to C8. This range represented full audio spectrum. Subsequently best minimal and maximal pitch were determined. Together those two values provided enough information to specify optimal pitch range of the *Constant-Q* transform.

To examine influence of narrowing the spectrum Harmonic Pitch Class Profile (HPCP) [3] was used. This type of feature vector is frequently used in multiple chord recognition systems [8]. HPCP vector H may be calculated from a *Constant-Q* spectrum in the following way:

$$H_b = \sum_{g=0}^{G-1} |Q_{b+\beta g}| \quad (4.4)$$

Where g is the octave number, G is total number of octaves produced by *Constant-Q* transform with β bins per octave (for purpose of this test it is always

set to 12) and b is index in HPCP value. To determine actual loss of the spectral data inner product between HPCP from narrowed spectrum and full spectrum will be taken. This operation will be performed on the whole The Beatles discography in goal to obtain indisputable values.

Min pitch	Frequency information when compared to full audio spectrum
C1	100%
C2	100%
C3	98%
C4	58%
C5	17%
C6	3%
C7	1%

Table 4.3 Amount of remaining frequency information after applying hi-pass filter at “min pitch”.

As we can see in table 4.3 setting minimal pitch to values greater than C2 result in losing audio data. However, with only 2% loss spectrum starting at C3 this may provide to be an acceptable trade-off as the time frame shortened from 32768 to only 8192 values.

Max pitch	Frequency information when compared to full audio spectrum
C3	0%
C4	17%
C5	56%
C6	74%
C7	96%
C8	100%

Table 4.4 Amount of remaining frequency information after applying low-pass filter at “max pitch”.

Just as in the previous case in table 4.4 we can clearly identify range of pitches that contain most of the harmony data in audio signal. Setting maximal frequency lower than C8 will filter out some of the frequency data. Although with a 96% information preserved C7 pitch based feature vectors may prove to be equally efficient.

Based on information obtained from table 4.3 and 4.4 we narrowed used pitch range to contain only those between C3 and C7.

To provide final comparison between full audio spectrum and presented narrowed one, their performance was compared during chord recognition using WCSR metrics. Of course, as the number of features changed all of the neural network hyper-parameters had to be adjusted to adapt to new situation. This procedure is described in chapter 4.5. In this chapter we will not compare training times of both systems as it does not matter for the output of the algorithm. Although, system with smaller neural network input learned considerably faster than the one that used full audio spectrum during extraction process.

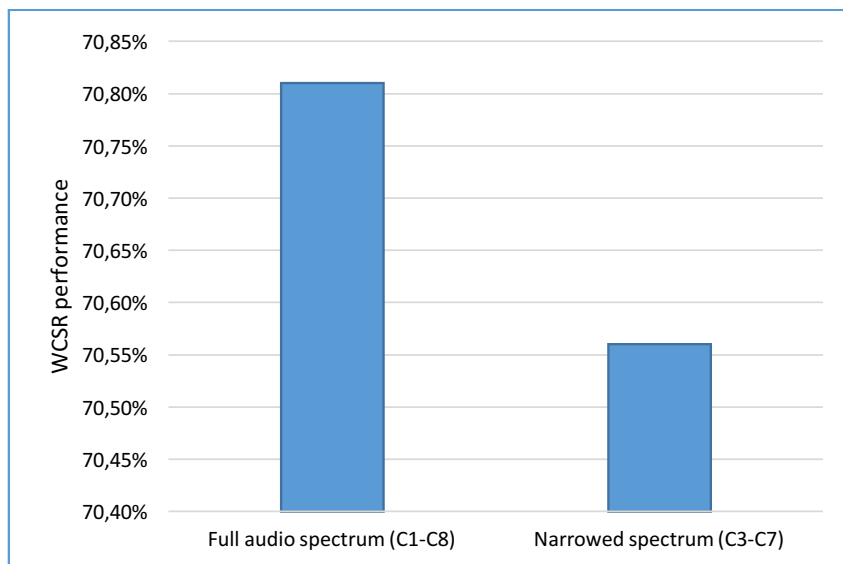


Figure 4.3 WCSR performance as a function of a pitch range used during feature extraction process.

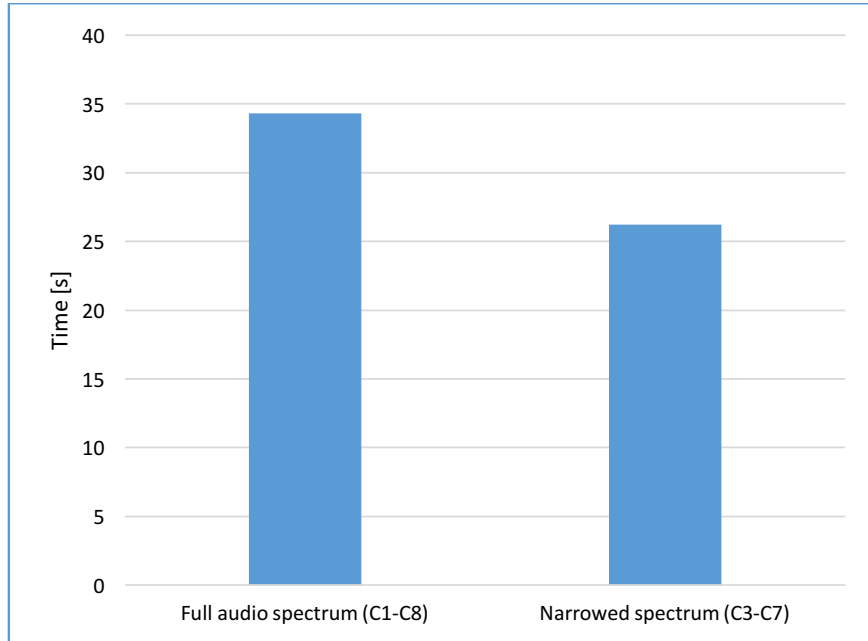


Figure 4.4 Recognition time as a function of pitch range used during feature extraction process.

Figures 4.3 and 4.4 show performance and computational time comparison between using full audio spectrum and our limited one. As we can see general system performance are almost the same but smaller computational time greatly favors usage of narrowed frequency range. For the rest of this thesis we will use C3 as a minimal pitch as it is a maximal value that does not negatively impact system performance. For similar reasons we will use C7 as a maximal pitch.

4.3.Constant Q transform precision

For musical audio that is guaranteed to be perfectly tuned to concert pitch (i.e. reference pitch $A4 = 440\text{Hz}$), a 12 bins per octave in *Constant-Q* transform would be ideal as 1 bin would directly map into 1 semitone. In general, however, we cannot guarantee that audio we analyze from real recordings will be at concert pitch. This is statement certainly true for the Beatles albums which we use as our test set in this work. Every song after recording goes through a final step called mastering during which additional musical effects are introduced in post-production, with the goal making sound more appealing to audience.



Figure 4.5 Result of Constant- Q transform of first 30 seconds of "Comptine d'un autre été : L'Après-Mid" by Yann Tiersen using 36 bins per octave.



Figure 4.6 Result of Constant- Q transform of first 30 seconds of "Comptine d'un autre été : L'Après-Mid" by Yann Tiersen using 84 bins per octave.

To compensate for this effect systems which incorporate *Constant-Q transform* [3] [8] manipulate β parameter (equation 3.2) which is responsible for the number of bins per octave in resulting spectrum. Result of this change is clearly visible in Figure 4.5 and 4.6. As we can see for perfectly tuned piano playing “*Comptine d'un autre été : L'Après-Midi*” by Yann Tiersen in ideal acoustics conditions the distances between bins which contain most energy increased. This fact is caused by introduction of additional bins between the “pure” semi-tones. Using β parameter set to 36 would result in adding 1 bin before “pure” tone and 1 bin after.

Majority of artist during recording process use additional effects which distort original sounds produced by instruments. In addition, most musical pieces contain more than 1 instrument (human voice is also counted as an additional one) and if even a single one of them is out of tune it will be visible in audio spectrum. Very often sounds of multiple instruments blend into one which effects in temporary peak that span across multiple pitches.

Bins per octave	Window length	FFT window length
12	5670,0	8192
36	17342,8	32768
60	29016,5	32768

Table 4.5 Relation between number of bins per octave used during *Constant-Q transform* and time frame length with $\min pith = C3$.

On the other hand, using multiple more bins per octave in results in using bigger time-window as shown in table 4.5 (calculations based on equation 4.3). If we want to use fast algorithm for calculation of *Constant-Q transform* [16] we have to use windows with length as a power of 2. This is connected to usage of Fast Fourier Transform as the base of this method. Using more bins per octave changes the ratio of frequency to its bandwidth (equation 3.1) which is the Q parameter in equation 4.3 and that raises the window length.

Using time-frames with bigger length increases chance that in analyzed audio segment there is a chord change which introduces disturbances in spectrum. This may have unfavorable consequences for chord detection algorithm as it won't be able to correctly classify this part of the signal.

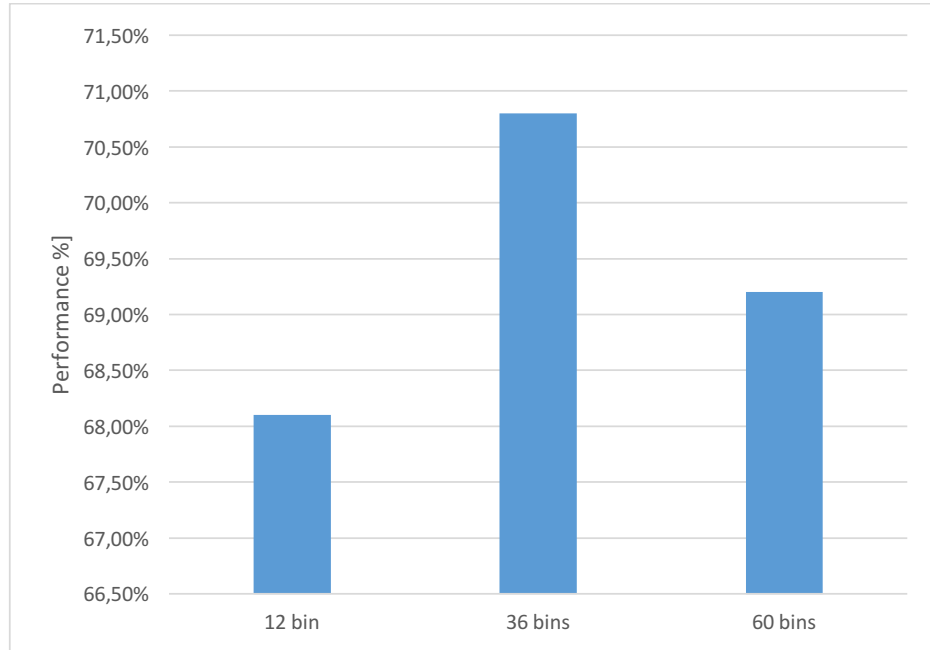


Figure 4.7 Chord recognition performance (WCSR) according to number of bins used during *Constant-Q* transform.

As we can see in Figure 4.7 hypothesis about the negative impact of big number of bins per octave in *Constant-Q* transform is confirmed. Using 60 bins performs worse than using only 36 bins (both contains 32768 frames in single window). It is worth noticing that using 36 bins per octave gives better result than using only one. We can say that this value is a middle ground between the ability to recognize out-of-tune pitches and length of analyzed window.

4.4. History length

As shown in equation 3.4 amount of time-frames considered during feature extraction has a major influence on length of presented feature vectors. The relationship between those 2 values is a proportional one, meaning that including additional time-frame in our calculation increases dimension of the feature vector by additional 48 values (optimal number of features extracted from a single time-frame was determined in chapter 4.2). By extending the current frame with the data of neighboring frames we create one larger superframe. This process is known as *time splicing*. Its role is to provide the recognition system contextual data about the current event (currently played chord) by introducing historical values.

To examine impact of this parameter on our system performance we tested different values according to WCSR criteria. As for each tested value length of used feature vector was different it was necessary to relearn neural network used in recognition step for each test. For time-saving purposes only a subset of the available training data was used for this task. We have chosen following albums from “The Beatles” discography:

- training data: Please Please Me, Rubber Soul, Help
- validation data: Abbey Road
- test data: With The Beatles

Please note that this selection was completely arbitrary and by no means it represents whole spectrum of all of the available popular music. However, we think that using the same dataset for every evaluation makes the result comparable. The exact methods used for training the neural network are described in chapter 4.5.

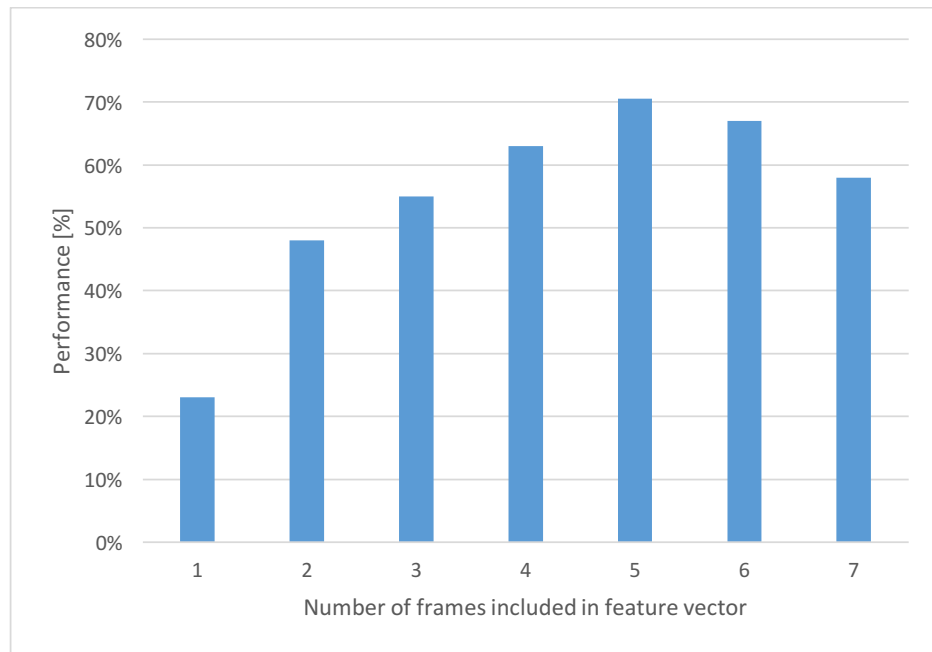


Figure 4.8 Chord recognition performance according to number of time frames used in feature vector.

As shown on Fig. 4.8 performance of chord recognition system greatly depends on number of time-frames considered during feature extraction step. It is

clearly visible that using history length of 1 gives the lowest results. This may be caused by the fact that in this case only single time-frame was considered without any contextual information. As we know music is a sequence of sounds in which order is very important. Using history parameter with value greater than 1 introduces information about previous sounds in musical piece. As we can see this data is very valuable as the performance of analyzed system doubles after adding just a single history frame. However, after reaching peak of performance at value 5 our recognition capability starts to gradually decrease. It may be connected to that, that in our system every extracted feature has the same weight. Neural network used in next step does not recognise that certain values come from the “past” and it treats them as one big input. Thus, using bigger history length values may lead to unclear results for frames where spectrum represents moments during chord change. In this situation part of the extracted features will describe new chord but some of them will still represent mixed spectrum that occurs during such change.

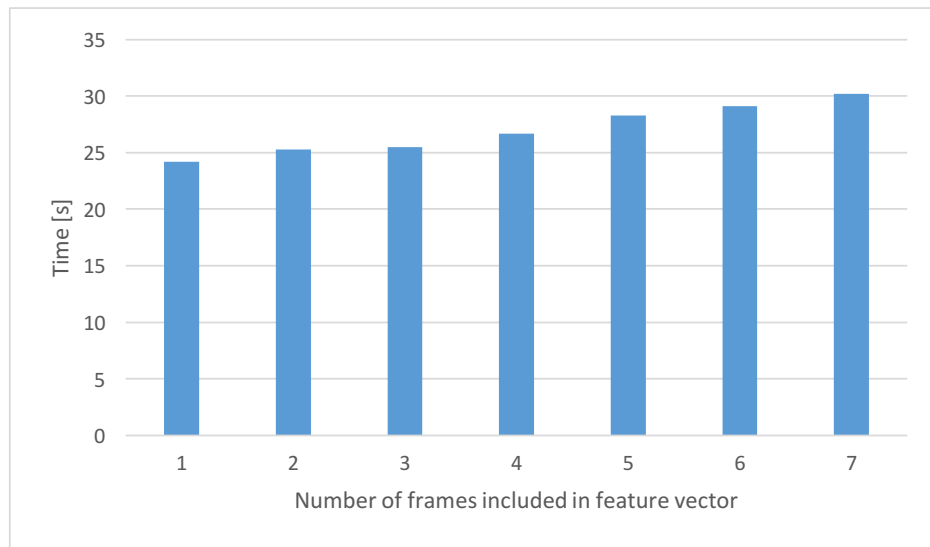


Figure 4.9 Chord recognition time performance according to number of time frames used in feature vector.

In figure 4.9 we compared different values of history length parameter according to time needed to annotate whole “With The Beatles” album. As we can see there is a direct dependence between those 2 values. Using more historical values increases time needed to reach goal (during our experiments we found out that most of the time during calculation is consumed by matrix multiplication in our neural network, although we were not able to further optimize performance of this element). However, we could argue that all of the presented

values are suitable for real-life applications (on average there are about 60 chords in a single song).

We decided that for this parameter we will neglect research connected to CPU or memory usage. Although those parameters vary dependent on the number of history time-frames used, they do not impact system performance in terms of number of recognized chords. At no point in time for any of the tested parameters the memory usage exceeded 70MB.

We decided to use history length parameter set to 5 in further research.

4.5. Fine tuning of neural network

As stated in chapter 2.2 most techniques proposed in the literature do not incorporate machine learning methods. Notable exceptions are recent works by Osmalskyj, Embrechts, Piérard and Van Droogenbroeck (2012) and Boulanger-Lewandowski, Bengio and Vincent (2013). As shown other computer science fields (most notably in the image recognition) in many cases designs based on machine learning vastly outperform usage of other methods [14].

In our system we decided to use feed-forward neural network. Our network consists of single hidden layer. In chapters 4.2 and 4.4 we determined optimal length of the feature vector. Using pitch range from C3 to C7 with history length of 5 gives us 240 values used as an input to our network (calculations based on equation 3.4). Output of our network for “major, minor and no chord” MIREX concurrency consists of 25 classes, it takes form of a vector where each value contains probability of a single chord.

During learning of our neural network we focused on following hyper-parameters:

- number of neurons in hidden layer
- learning rate
- number of iterations
- weight decay

We used holdout-based early stopping to prevent overfitting. This ensures us that algorithm stops at the most optimal iteration. Initially weight decay hyper-parameter was set to 0 which resulted in completely turning off this hyper-

parameter. This leaved only 2 parameters left to optimize: number hidden neurons and learning rate.

For time-saving purposes we decided to employ the same subset of training data as in chapter 4.4. Again, as a test data we used “With The Beatles” album by “The Beatles”. First test when we saw that presented network was learning occurred when we used 20 hidden neutrons and learning step of 0.0001. We decided to use those values as a base for grid search method for obtaining optimal values of those hyper-parameters.

		Neurons in hidden layer			
		20	30	45	100
Learning rate	0,1	24,24%	48,54%	65,38%	63,67%
	0,01	26,73%	48,23%	66,24%	68,13%
	0,001	27,24%	52,13%	69,83%	67,24%
	0,0001	29,56%	52,56%	69,78%	66,63%

Table 4.6 Relationship between number of neurons in hidden layer and the learning rate as WCSR recognition

		Neurons in hidden layer			
		20	30	45	100
Learning rate	0,1	8,75	15,83	20,42	23,33
	0,01	31,67	40,83	36,25	42,92
	0,001	60,83	63,33	47,08	64,58
	0,0001	84,58	76,25	101,25	98,75

Table 4.7 Relationship between number of neurons in hidden layer and the network learning time (in minutes)

As presented in tables 4.6 and 4.7 system performance vary depending on values of both parameters. Using small learning step drastically increases length of the learning process. We can also see that big learning rate very quickly finds optimal parameter value. We can attribute this behavior to the fact that in those cases number of iterations was very small as an additional epochs did not provide any substantial improvements (the algorithm was “stepping over” determined local minimum). Based on our research we think that using learning step with dynamic length (proportional to error) value would be optimal approach. Although due to time constraints we were not able to test this hypothesis. As a result, we decided to use neural network with 45 hidden neurons and learning rate of 0.001.

	Weight decay						
	0	0,00001	0,0001	0,001	0,01	0,1	1
Performance [WCSR %]	69,83%	70,03%	70,48%	48,32%	24,65%	12,21%	7,24%

Table 4.8 Relationship between value of weight decay parameter and system performance.

The last parameter of presented neural network is weight decay. Again we used grid search approach but this time for only one parameter. As we can see in table 4.8 recognition improvements when compared to results without decay (value 0) are relatively minor. During research we observed that for large values of this parameter system performs only a small number of iterations. After the early stopping was turned off we saw that the results of those configurations were not improving at all as if network was struggling to maintain current performance level. The peak of the performance can be seen at weight-decay set to 0.0001.

4.6. Chord sequence post processing

Result of our neural network is a sequence of vectors. Each value of in such vector is responsible for probability of a single chord (or in the special case: no chord at all).

Chord root	Probability	
	Major	Minor
A	5%	3%
A#	12%	15%
B	8%	12%
C	34%	21%
C#	3%	79%
D	57%	43%
D#	23%	17%
E	84%	67%
F	43%	34%
F#	13%	15%
G	16%	12%
G#	25%	34%
No chord	52%	

Table 4.9 Chord probabilities obtained from neural network around 5 second of "I saw her standing there" by The Beatles.

Ideally all of those vectors would be one-hot, meaning that one of the recognized patterns is definitely more likely to appear at this specific time. However, in real life scenarios this situation almost never happens. Most of the time output of the neural network resembles situation shown in Table 4.9. According to annotations in our testing set [15] at this exact moment E major chord should be recognized. Even though this value has the maximum likelihood out of all of the recognized classes, it would be easy to confuse it with C# minor. It is worth mentioning that those 2 chords share 2 of the same notes (E major is the sound E, G# and B played together and C# minor is C#, E and G#) and that may be the reason of such similar probability.

To deal with such situation we use special post-processing method called beam search. The main role of this algorithm is to establish most probable sequence of chords out of all of the labels recognized by neural network.

Beam search is a breadth-first tree search algorithm where only the w most promising paths (or nodes) at depth t are kept for future examination. In our case depth t means that the current chord sequence (history) contains exactly t chords (one chord label for each traversed tree level). It is assumed that all of the descendants of such node share the same history. In consequence all of the siblings of the current node may differ only by the value of the last chord. By storing cumulative likelihood for each node we can identify most promising paths. We will focus on finding most optimal value of w parameter both in term of recognition accuracy and computation time.

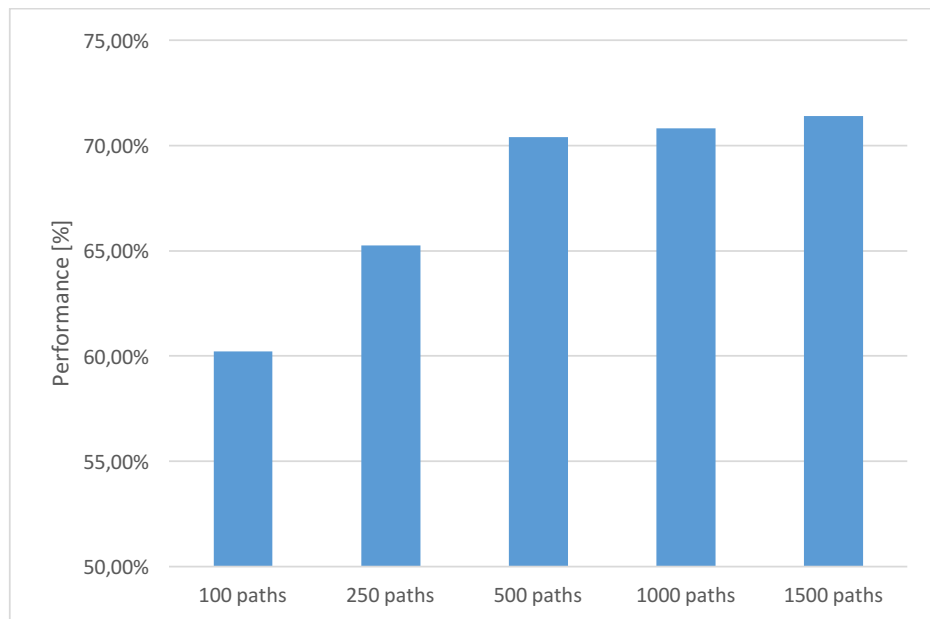


Figure 4.10 Recognition performance as a function of beam width.

As shown in figure 4.10 using bigger beam widths generally yields better results. However, as we can see the momentum of growth of this value rapidly slows down after exceeding 500 best paths. To definitely designate best value of this parameter we have to also look at the computation time needed to find the best path.

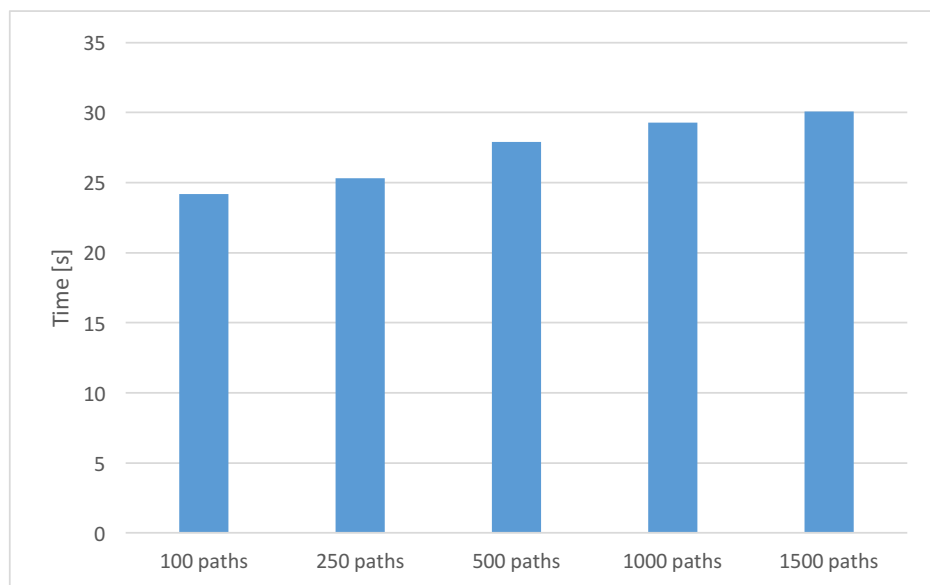


Figure 4.11 Calculation time as a function of beam width.

Comparing figure 4.10 and 4.11 clearly shows a trade-off between the recognition precision and the computational time. In general, the more time we spend on calculations the better result we will get. Although all of the values seems to be relatively close. For the rest of this thesis we will remember 1000 best paths during our post processing stage.

5. Comparison witch state of the art systems

In this chapter we will concentrate on evaluating system performance in comparison with methods known in literature. In section 5.1 we will introduce methods used during 2015 Music Information Retrieval Evaluation eXchange (MIREX) contest. Consequently, we will show evaluation results using Billboard 2012 and 2013 data sets in following competitions: “Root only” (section 5.2), “Major minor” (section 5.3), “Major, minor with inversions” (section 5.4) and “Major minor and sevenths” (section 5.5).

5.1.MIREX 2015

In this chapter we will compare performance of our chord recognizer with methods known in literature. For this we will confront our algorithm with systems designed by participants of most recent MIREX edition (2015). Contest results are freely available online at the competition webpage.

We will compare our design with following state of the art methods:

- CM3 - Chordino [19]
- DK [16] - Although multiple algorithms were submitted by Deng and Kwok in this chapter we will focus only on “DK9” as its performance was significantly better than the others.
- KO - shineChords [17]

Our own algorithm will be denoted as “MM”. As the chord dictionary varies between categories neural network was retrained for each test. Each time we used whole The Beatles discography as an training data set.

5.2. Root only competition

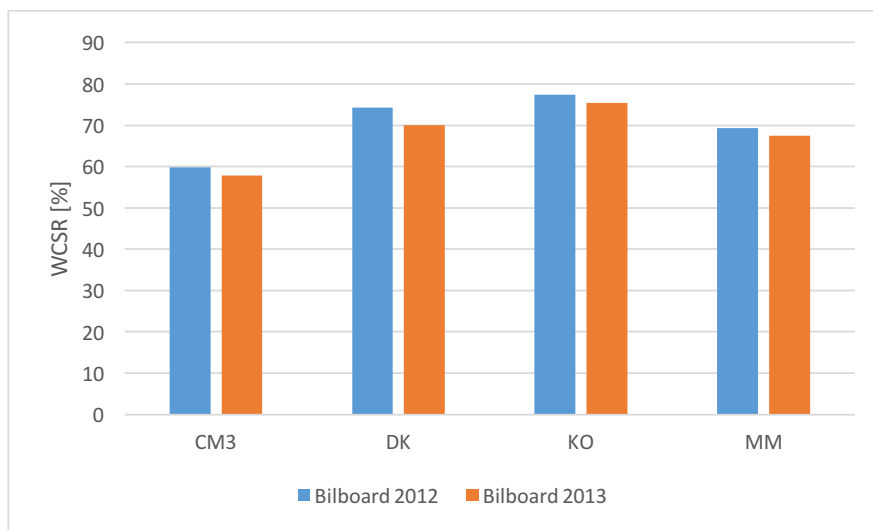


Figure 5.1 Algorithm performance comparison in "Root only" category

In this concurrency no limits are hold for the size of the chord dictionary. However, during the evaluation process only chord root is compared with the hand annotated transcription. To compete in this category, we decided to use our system trained originally for recognition of major and minor triads.

As we can see in the figure 5.1 our design represents state of the art performance. Although algorithms DK and KO performed better than our implementation, presented system recognized chords more efficiently than Chordino, obtaining ~68,5% according to WCSR measure.

5.3. Major, minor competition

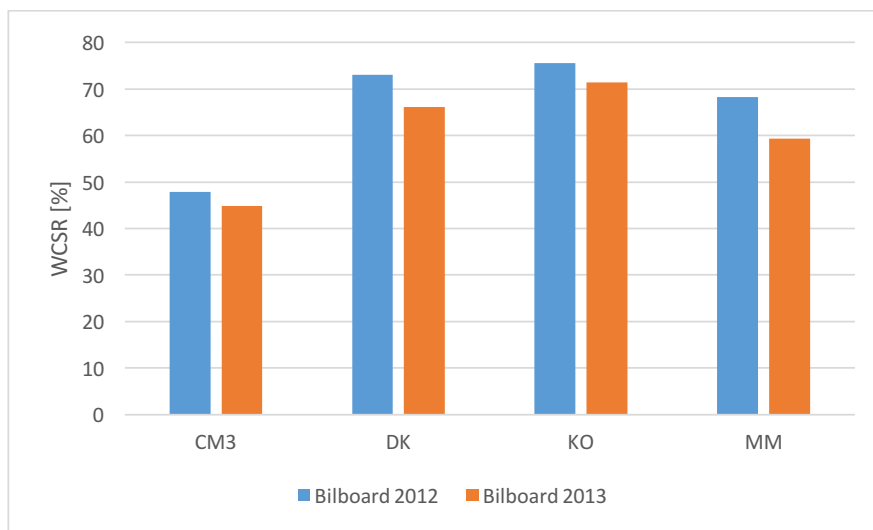


Figure 5.2 Algorithm performance comparison in “Major, minor” category

In this concurrency all of the systems have to recognize 25 classes: 12 major, 12 minor chords and an additional no chord label.

Results shown in figure 5.2 further confirm quality of our design. Although DK and KO are still performing better our algorithm with the overall score of 68,3% our implementation is close third.

It is worth noticing that results in figure 5.1 closely resemble those presented in figure 5.1. From this we can state that for concurrency “Major, minor” estimating the right root note will most likely result in correctly determining chord label (as there are only 2 chord families available).

5.4. Major, minor with inversions competition

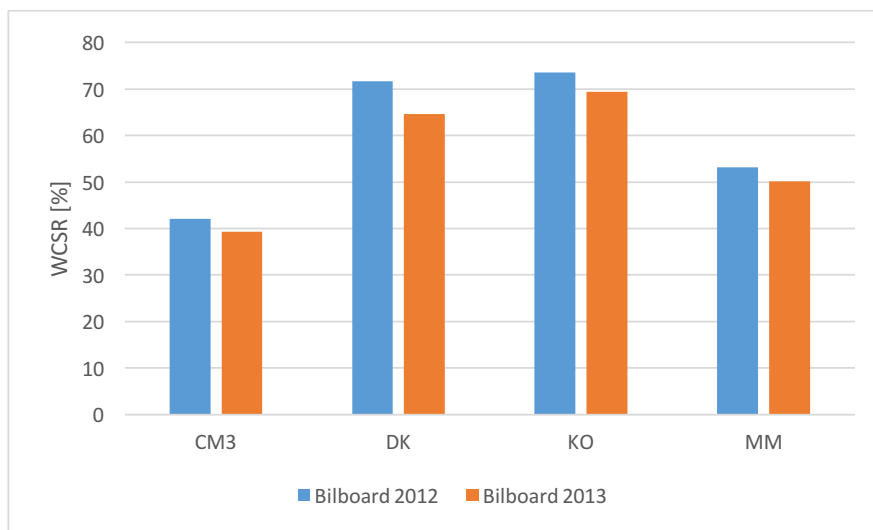


Figure 5.3 Algorithm performance comparison in "Major, minor with inversions" category

Adding inversions greatly increases size of our chord dictionary. Each chord (from base set of major and minor) will have 2 additional versions each with different bass note, giving 73 possible labels.

As we can see in figure 5.3 our algorithm does not perform well in this category. It is still better than chordino but the distance from 2 leading designs increased. We assume that this low performance can be a result of low number of inverted chords in our training set.

5.5. Major, minor and sevenths competition

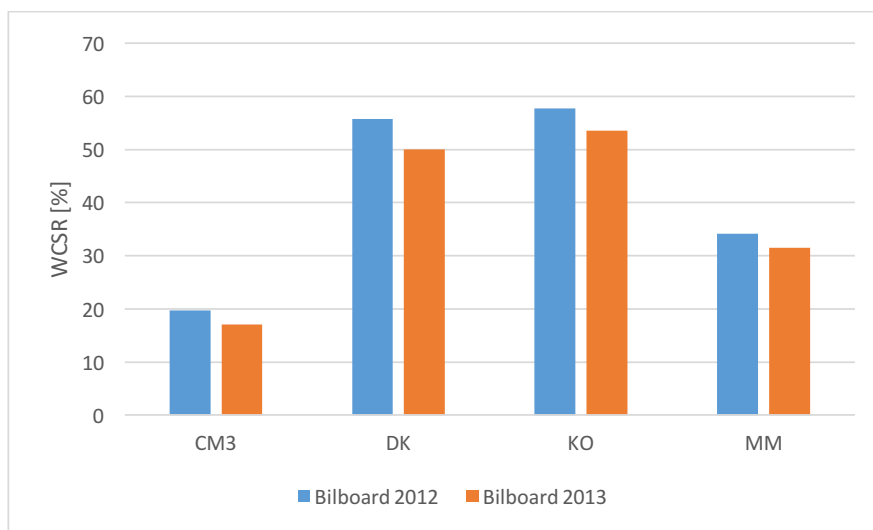


Figure 5.4 algorithm performance comparison in "Major, minor and sevenths" category

Last category consists of major, minor and seventh chords giving 121 possible labels.

As we can see in figure 5.4 the gap between our implementation and 2 leading algorithms increased just like in "Major, minor with inversions" case. Although, using part of a Constant-Q spectrum as a feature vector should provide enough information to properly classify such signal the results seem to contradict this thesis. Our system was mostly designed for recognition of major and minor chords as those families are most common.

6. Summary and future work

In this thesis we have presented a harmony detection system from audio signal capable of state-of-the-art chord transcription. We will summarise the main outcomes of our research in chapter 6.1.

However, in the process of working on this thesis, many possible enhancements came to our mind, that would have been too time-consuming to be included in this version. In chapter 6.2 we present overview of such ideas.

6.1. Summary

Our approach to chord detection was to integrate as much music theory information as possible into the design. In chapter 3 we presented such system, which we proven to be capable of state of the art transcription.

Our design is based on feature extraction using *Constant-Q* transform and *time splicing* method. Both of them were chosen due to their musical relevance. *Constant-Q* transform was selected as it produces logarithmically spaced frequency bins which resembles the layout of musical frequencies in equal tempered system. This feature assures us that every pitch is analyzed with the same precision. In addition, *Constant-Q* transform allows us to easily detect different tuning of the instruments used during recording and correctly adapt to this situation. By using *time splicing* we introduced time-dependent context into our system. As shown in chapter 4.4 this information is crucial for the chord recognition performance. The design of this feature is based on theory that in vast majority of cases single chord spans across multiple time-frames. This allows us detect correct chord even if the superframe is partially in transition to a new chord.

The final step of our system is chord recognizer. This stage consists of 2 modules: neural network and beam search. Neural network was used for its versatility. In our design this component is responsible for finding likelihood of event such as the specific chord was played. The output of the network as a list of chord probabilities for all of the processed time-frames is used as an input to beam search algorithm. Its role is to find most probable progression using chord transition matrices derived from audio data. Example of a system output is presented in Appendix A.

Chord progressions define the harmonic structure of a musical piece. As this they are a key step for an automatic music transcription - process in which an audio file is analyzed and its musical properties are extracted. Presented method is a step towards this goal.

6.2.Future work

In this final section we want to state possible optimizations of our algorithm.

6.2.1.Additional music information

We believe that additional music information such as key or genre would be greatly helpful in improving system performance. Knowing the musical piece key would give us information which chord progressions are most likely to happen. We could incorporate this knowledge in our post-processing step for example by improving chord transitions matrices.

Additional information about musical genre would also be very helpful. Progressions used in blues are vastly different from the ones used in rock music. We expect that there is not one best chord transcription algorithm for all of the music, even within one genre. However, using genre classification system at the beginning of the transcription would greatly improve performance of presented system.

6.2.2.Adding additional modules

We could further improve performance of presented system by adding new modules. For example, simple algorithm created for detecting silence (for example by measuring total amount of energy present at given time in spectrum) would immediately have a positive impact on performance. Such places would be automatically classified as “no chord” and would be treated as a always 100% correctly guessed places during recognition.

Additionally, during our development we saw vastly different classifications of the same audio fragment using separately trained neural networks. This situation occurred mostly in places where chord was in the middle of transaction to the next one, as such moments the spectrum often contain a lot of additional sounds. One of the possible ways resolve this problem is to use multiple algorithms working in parallel on the same piece of audio data to generate separate

chord estimates. Average of such estimates would be used as an input to our post processing step.

6.2.3.Extended data set

Our focus throughout this thesis was put on chord transcription of western popular music which was mainly reflected by using “The Beatles” discography as our training data set. However, one could argue that using songs from just single performer would prevent our algorithm from properly generalizing chord spectral representations. Results presented in chapter 5 seems to contradict this thesis although we were not able to test our algorithm on vastly different audio set (e.g. korean K-POP). We expect that without relearning of the neural network our algorithm would not perform very well in such situation.

6.2.4.Implementation enhancements

One of the intentions of this thesis was to create mobile application capable of chord recognition. Unfortunately, this goal was fulfilled only partially. Although we created chord detection system with state of the art performance, the mobile application is still a distant dream. Even though we used Swift as our main coding language moving into iOS would cost much effort. And even then, in its current form it would be nowhere near desired production quality in terms of user experience.

It is also possible to improve application performance in terms of resource utilization as current code focused more on modularity. This approach allowed us to quickly swap components to get the best possible design but in some cases (mostly in post-processing stage) the performance suffered. We hope that after this changes application would be able to recognize chords in real time.

Bibliography

- [1] Geoffroy Peeters, "Chroma-based Estimation of Musical Key from Audio-Signal Analysis," in *7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.
- [2] Takuya Fujishima, "Real time chord recognition of musical sound: a system using Common Lisp Music," in *International Computer Music Conference (ICMC)*, 1999, pp. 464–467.
- [3] Emilia Gómez, *Tonal Description of Audio Music Signals*. Barcelona, 2006, PhD thesis, Universitat Pompeu Fabra.
- [4] J. L. Durrieu, and Gael Richard Jan Weil, "Automatic generation of lead sheets from polyphonic music signals," 2009.
- [5] Juan Bello, "Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats," in *8th International Conference on Music Information Retrieval (ISMIR 2007)*, Vienna, 2007.
- [6] Amelie Anglade, Raphael Ramirez, and Simon Dixon, "Genre classification using harmony rules induced from automatic chord transcriptions.," in *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, 2009, pp. 669-674.
- [7] Carol Krumhansl, "Cognitive Foundations of Musical Pitch," 1990.
- [8] John Allen, "Short Time Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform," *Transactions on Acoustics, Speech, and Signal Processing.*, pp. 235-238, 1977.
- [9] Lee Kyogu and Malcolm Slaney, "Automatic Chord Recognition from Audio Using a HMM with Supervised Learning," in *6th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.
- [10] Borching Su and Jeng Shyh-kang, *Multi-timbre chord classification using wavelet transform and self-organized map neural networks.*, 2001.
- [11] Xinquan Zhou and Alexander Lerch, "Chord detection using deep learning," 2015.
- [12] Alan Oppenheim, Ronald Schafer, and John Buck, *Discrete-time signal processing*. New Jersey: Prentice Hall, 1999.
- [13] Teuvo Kohonen, "An introduction to neural computing," 1988.
- [14] Michael Nielsen, "Neural Networks and Deep Learning," 2015.
- [15] Christopher Harte, "Towards automatic extraction of harmony information from music signals," 2010.
- [16] Junqi Deng and Yu-Kwong Kwok, "Automatic chord estimation with chord correction using deep neural network," , 2015.
- [17] Maksim Khadkevich and Maurizio Omologo, "Time-frequency reassigned features for automatic chord recognition," , Trento, 2015.
- [18] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent, "Audio Chord Recognition with Recurrent Neural Networks," in *14th*

- International Conference on Music Information Retrieval (ISMIR 2013)*, 2013.
- [19] Judith Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Am.*, pp. 425–434, 1991.
- [20] Julien Osmalskyj, Jean-Jacques Embrechts, Sébastien Piérard, and Marc Van Droogenbroeck, "Neural networks for musical chord recognition," 2012.
- [21] Judith Brown and Miller Puckette, "An Efficient Algorithm for the Calculation of a Constant Q Transform," *Acoustical Society of America*, pp. 92-97, 1992.
- [22] Chris Cannam et al., "Vamp plugins from the centre for digital music," , 2015.

List of figures

Figure 2.1 Pitches as found on grand piano keyboard (a). Pitch names from single octave on piano keyboard (b). Pitches from C4 to C5 on musical stave. [8].....	10
Figure 2.2 Chord type template for C Major chord (semitones C, E and G)...	14
Figure 2.3 Hidden Markov model for chord sequences.....	14
Figure 3.1 Overview of the described chord recognition system.	16
Figure 3.2 Waveform (a) and example of system output (b) for first 30 seconds of “I Saw Her Standing There” by The Beatles.	18
Figure 3.3 Division of audio spectrum using constant Q transform near A4=440 Hz reference frequency using 12 (a) and 36 (b) bins per octave.	19
Figure 3.4 Real part of the spectral kernel as presented in (Brown i Puckette 1992).	20
Figure 3.5 Result of Constant-Q transform of first 30 seconds of “Comptine d'un autre été : L'Après-Mid” by Yann Tiersen.	21
Figure 3.6 Representation of used neural network with single hidden layer...	22
Figure 4.1 Chord structure of The Beatles discography [15].....	26
Figure 4.2 Result of Constant-Q transform of first 30 seconds of “Comptine d'un autre été : L'Après-Mid” by Yann Tiersen in a pitch range from C2 to B5.....	28
Figure 4.3 WCSR performance as a function of a pitch range used during feature extraction process.	31
Figure 4.4 Recognition time as a function of pitch range used during feature extraction process.....	32
Figure 4.5 Result of Constant-Q transform of first 30 seconds of “Comptine d'un autre été : L'Après-Mid” by Yann Tiersen using 36 bins per octave.	33
Figure 4.6 Result of Constant-Q transform of first 30 seconds of “Comptine d'un autre été : L'Après-Mid” by Yann Tiersen using 84 bins per octave.	33
Figure 4.7 Chord recognition performance (WCSR) according to number of bins used during Constant-Q transform.	35
Figure 4.8 Chord recognition performance according to number of time frames used in feature vector.	36
Figure 4.9 Chord recognition time performance according to number of time frames used in feature vector.	37
Figure 4.10 Recognition performance as a function of beam width.....	42
Figure 4.11 Calculation time as a function of beam width.	42
Figure 5.1 Algorithm performance comparison in “Root only” category	45
Figure 5.2 Algorithm performance comparison in “Major, minor” category..	46
Figure 5.3 Algorithm performance comparison in "Major, minor with inversions" category.....	47

Figure 5.4 algorithm performance comparison in "Major, minor and sevenths" category	48
---	----

List of tables

Table 2.1	Semitones with most common enharmonics.....	8
Table 2.2	First 12 intervals with examples for semitone C	9
Table 2.3	Triad chords for based on C root	11
Table 4.1	Example of system output for first 25 seconds of “I Saw Her Standing There” by The Beatles.	25
Table 4.2	Relation between used pitch range and length of a used time window.....	28
Table 4.3	Amount of remaining frequency information after applying hi-pass filter at “min pitch”.	30
Table 4.4	Amount of remaining frequency information after applying low-pass filter at “max pitch”.	30
Table 4.5	Relation between number of bins per octave used during Constant-Q transform and time frame length with min pith = C3.	34
Table 4.6	Relationship between number of neurons in hidden layer and the learning rate as WCSR recognition.....	39
Table 4.7	Relationship between number of neurons in hidden layer and the network learning time (in minutes).....	39
Table 4.8	Relationship between value of weight decay parameter and system performance.	40
Table 4.9	Chord probabilities obtained from neural network around 5 second of “I saw her standing there” by The Beatles.	40
Table 0.1	Example annotation of “I Saw Her Standing There” by The Beatles scoring 70,4%according to CRS measure. Used algorithm was trained for “Major and minor” task.....	58

Appendix A - Example transcription

Presented algorithm	Reference transcription [8]
0.000 2.097 No chord	0.0 0.755 N
2.097 17.597 E	0.755 2.206 N
17.597 18.344 B	2.206 17.461 E
18.344 24.722 E	17.461 20.039 B
24.722 25.821 C	20.039 23.394 E
25.821 27.879 E	23.394 24.892 A
27.879 29.169 B	24.892 26.389 C
29.169 48.583 E	26.389 27.504 E
48.583 49.163 C	27.504 27.876 D/5
49.163 51.682 E	27.876 29.373 B
51.682 52.697 B	29.373 41.297 E
52.697 56.076 E	41.297 43.874 B
56.076 56.843 A	43.874 47.195 E
56.843 59.187 E	47.195 48.681 A
59.187 59.873 A	48.681 50.167 C
59.873 65.476 E	50.167 51.293 E
65.476 68.839 B	51.293 51.664 D/5
68.839 70.118 A	51.664 53.139 B
70.118 87.537 E	53.139 56.134 E
87.537 88.588 C	56.134 65.167 A
88.588 90.877 E	65.167 68.220 B
90.877 91.684 B	68.220 71.192 A
91.684 92.968 E	71.192 80.260 E
92.968 93.312 G	80.260 83.232 B
93.312 106.266 E	83.232 86.216 E
106.266 106.498 B	86.216 87.713 A
106.498 110.791 E	87.713 89.234 C
110.791 111.321 A	89.234 89.989 E
111.321 119.008 E	89.989 92.230 B
119.008 119.670 A	92.230 93.716 E:7
119.670 122.770 E	93.716 94.436 G/3
122.770 122.999 A	94.436 95.515 N
122.999 125.268 E	95.515 104.141 E
125.268 126.011 A	104.141 107.125 B:7
126.011 128.459 E	107.125 107.892 E:min7
128.459 130.880 B	107.892 110.155 E
130.880 132.481 A	110.155 112.779 A
132.481 143.397 E	112.779 114.637 E
143.397 143.929 B	114.637 115.008 B:min
143.929 150.778 E	115.008 116.135 D#:dim
150.778 151.343 C	116.135 117.586 E
151.343 153.773 E	117.586 119.014 N

153.773 154.575 B	119.014 128.058 A
154.575 159.550 E	128.058 130.705 B
159.550 160.520 B	130.705 134.072 A
160.520 165.587 E	134.072 136.707 E
165.587 166.644 B	136.707 138.577 A
166.644 170.841 E	138.577 143.093 E
170.841 175.613 No chord	143.093 146.077 B
	146.077 147.574 E:maj7
	147.574 149.049 E/3
	149.049 150.558 A
	150.558 152.056 C
	152.056 153.205 E
	153.205 153.588 D/5
	153.588 154.703 B
	154.703 159.208 E
	159.208 159.591 D/5
	159.591 161.088 B
	161.088 165.593 E
	165.593 167.126 B
	167.126 168.658 A
	168.658 171.828 E
	171.828 175.804 N

Table 0.1 Example annotation of “I Saw Her Standing There” by The Beatles scoring 70,4% according to CRS measure. Used algorithm was trained for “Major and minor” task.