Thèse

Présenté pour obtenir le grade de docteur de l'Ecole Centrale de Lyon

Aliaksandr V. PARADZINETS

Variable Resolution Transform-based Music Feature Extraction and their Applications for Music Information Retrieval

Soutenue le :

Devant le jury composé de :

Mme. Myriam Desainte-Catherine	(LABRI Bordeaux)	Examinatrice
M. Gaël Richard	(ENST Paris)	Rapporteur
M. François Pachet	(SONY CSL Paris)	Rapporteur
M. Liming Chen	(ECL Lyon)	Directeur de thèse
M. Evegeny I.Bovbel	(BSU Misnk)	Examinateur

Dans le cadre de l'Ecole Doctorale Informatique et Information pour la Société

Lyon 2007

Résumé

Dans le secteur de loisirs il y a un nombre considérable d'enregistrements numériques musicaux produits, diffusés et échangés qui favorise la demande croisante de services intelligents de recherche de musique. La navigation par contenu devient cruciale pour permettre aux professionnels et également aux amateurs d'accéder facilement aux quantités de données musicales disponibles. Ce travail présente les nouveaux descripteurs de contenu musical et mesures de similarité qui permettent l'organisation automatique de données musicales (recherche par similarité, génération automatique des playlistes) ainsi que l'étiquetage (classification automatique en genres). Ce travail s'intéresse au problème de la construction des descripteurs du point de vue musical en complément des caractéristiques spectrales de bas-niveau. Plusieurs aspects d'analyse musicale, telles que l'analyse du signal où une nouvelle technique de transformation fréquentielle à résolution variable est proposée et décrite. Le traitement de niveau plus haut touche aux aspects de l'extraction des connaissances musicales. Cette thèse présente les algorithmes de détection de coups (*beats)* et d'extraction de fréquences fondamentales multiples. Les deux algorithmes sont basés sur la transformation à résolution variable proposée. Les informations issues de ces algorithmes sont utilisées dans la construction des descripteurs musicaux, représentés sous forme d'histogrammes (nouvel histogramme rythmique 2D qui permet d'estimer directement le tempo, et les histogrammes de succession et profil de notes). Deux applications majeures qui utilisent les caractéristiques mentionnées sont décrits et évaluées dans cette thèse.

Summary

As a major product for entertainment, there is a huge amount of digital musical content produced, broadcasted, distributed and exchanged. There is a rising demand for content-based music search services. Similarity-based music navigation is becoming crucial for enabling easy access to the ever-growing amount of digital music available to professionals and amateurs alike. This work presents new musical content descriptors and similarity measures which allow automatic musical content organizing (search by similarity, automatic playlist generating) and labeling (automatic genre classification). The work considers the problem of content descriptor building from the musical point of view in complement of low-level spectral similarity measures. Several aspects of music analysis are considered such as music signal analysis where a novel variable resolution transform is presented and described. Higher level processing touches upon the musical knowledge extraction. The thesis presents algorithms of beat detection and multiple fundamental frequency estimation which are based on the variable resolution transform. The information issued from these algorithms is then used for building musical descriptors, represented in form of histograms (novel 2D beat histogram which enables a direct tempo estimation, note succession and note profile histograms etc.). Two major music information retrieval applications, namely music genre classification and music retrieval by similarity, which use aforementioned musical features are described and evaluated in this thesis.

Acknowledgments

There are many people who were accompanying me and made this dissertation possible, and whom I would like to express my gratitude.

This work has been done at the Department of Mathematics and Informatics, Ecole Centrale de Lyon, France, during the period of 2003-2007.

First of all I would like to thank my supervisor Prof. Liming Chen for giving me the opportunity to work in the research team of ECL and for supporting my work during the whole period.

I would like to thank my master thesis supervisor Prof. Evgeny Bovbel from the Department of Radio Physics, Belarusian State University, Belarus. Thanks to him and other people from the Department I could join the research team of ECL.

I would like to thank Gaël Richard from l'Ecole Nationale Supérieure des Télécommunications (ENST) and François Pachet from SONY CSL who were so kind as to agree to act as reviewers for this thesis. In the same context I would like to thank Myriam Desainte-Catherine from the Laboratoire Bordelais de Recherche en Informatique (LaBRI) for agreeing to take part in the examination jury.

It was a great pleasure to work in the research team at the Department of Mathematics and Informatics in Ecole Centrale de Lyon. I would like to personally thank Hadi Harb for many constructive discussions. Dzmitry Tsishkou and Viacheslau Parshyn were my colleagues, friends, and bureau companions during the period of thesis. Their company was always a pleasure for me. I thank all MathInfo department members – Christian and Colette Vial, Mohsen Ardebillian, Alexandre Saidi, Emmanuel Dellandrea and all the others. Special thanks goes to always helpful secretaries of the department: Franзoise Chatelin and Isabelle San-Jose.

I am very grateful to my friends for being who they are, especially Oleg Kotov for his valuable ideas in music signal processing, and Sergei Zhukovsky for helping with manuscript verifications. Finally I want to thank people who are the most important for me – my parents Valery and Galina Paradzinets and my sister Tatsiana for all their love and warm encouragements. Though they are far away, they are always with me. Last but certainly not least, I would like to express my warmest gratitude to my dear wife Katsiaryna for her love, support and understanding.

This page is intentionally left blank

Table of contents

RESUME ..... **III**

SUMMARY ..... **IV**

ACKNOWLEDGMENTS ..... **V**

1. INTRODUCTION **2**

1.1. RESEARCH TOPIC 2
1.2. PROBLEMS AND OBJECTIVES 2
1.3. OUR APPROACH AND CONTRIBUTIONS 3
1.4. ORGANIZATION OF THE MANUSCRIPT 5
1.5. LIST OF PUBLICATIONS 5

2. PROBLEM OF MUSIC SIMILARITY AND RELATED WORK **7**

2.1. STATE OF THE ART 7
2.1.1. Millions of audio features which are …similar *9*
2.1.2. Spectral similarity *11*
2.2. OUR APPROACH 14

3. MUSIC SIGNAL ANALYSIS **17**

3.1. ABOUT MUSIC SIGNAL 17
3.2. RELATED WORK 19
3.2.1. Fourier transform *20*
3.2.2. Wavelet transform *22*
3.2.2.1 Continuous wavelet transform 23
3.2.2.2 Discrete wavelet transform 25
3.2.3. Other transforms and filter banks *26*
3.2.3.1 Constant Q transform 26
3.2.3.2 Other filter banks 27
3.2.4. Discussion: FFT vs WT for music signal analysis? *28*
3.3. VARIABLE RESOLUTION TRANSFORM 31
3.3.1. Building Variable Resolution Transform *31*
3.3.1.1 The basis 31
3.3.1.2 Logarithmic frequency sampling 33
3.3.1.3 Varying the mother function 37
3.3.2. Properties of the VR transform *38*
3.3.3. Computation *44*
3.3.4. Discussion *45*
3.4. APPLICATION TO SPECTRAL SIMILARITY 47
3.5. CONCLUSION 48

4. RHYTHM-RELATED SIMILARITY FEATURES **51**

4.1. RELATED WORK 51
4.2. OUR VRT BASED APPROACH FOR BEAT CURVE EXTRACTION 54
4.2.1. An intuitive approach *54*
4.2.2. Procedure of beat curve extraction *56*
4.2.3. Discussion: VRT versus FFT based techniques *57*
4.3. RHYTHMIC FINGERPRINT 58
4.3.1. 2D beat histogram *58*
4.3.2. Rhythmic similarity measure *61*

4.4. A 2D BEAT HISTOGRAM BASED TEMPO ESTIMATION ALGORITHM AND ITS EVALUATION 65
4.4.1. A 2D beat histogram based tempo estimation algorithm 65
4.4.2. Experimental evaluations.....66
4.5. CONCLUSION.....71

5. MELODY-RELATED SIMILARITY FEATURES.....73

5.1. RELATED WORK 73
5.2. OUR VRT-BASED MULTIPLE F0 ESTIMATION ALGORITHM 74
5.2.1. Principle and procedure 74
5.2.2. Experimental evaluation 78
5.3. MELODY-RELATED SIMILARITY FEATURES.....81
5.3.1. Note profile histogram.....82
5.3.2. Note succession histogram.....84
5.3.3. Timbre histogram 85
5.4. CONCLUSION.....86

6. APPLICATIONS AND EVALUATION 88

6.1. AUTOMATIC GENRE CLASSIFICATION.....88
6.1.1. The problem 88
6.1.2. Related work.....89
6.1.3. Principle and architecture of our classification system 90
6.1.3.1 Single-classifier system 91
6.1.3.2 Multi-expert classification system..... 93
6.1.4. Experimental results 94
6.1.4.1 Reference database..... 95
6.1.4.2 Experimental results by single classifiers..... 96
6.1.4.3 Experimental results by Multi-expert system 100
6.1.4.4 Discussion..... 104
6.2. MUSIC SEARCH BY SIMILARITY.....105
6.2.1. The problem 105
6.2.2. Principle and architecture of our combination system of similarity measures.....106
6.2.3. Experimental results 107
6.2.3.1 Evaluation method..... 107
6.2.3.2 Listening test evaluation 109
6.2.3.3 Objective evaluation..... 113
6.2.3.4 MIREX2007 Audio Music Similarity and Retrieval..... 114
6.2.3.5 Discussion..... 118
6.3. CONCLUSIONS 119

7. CONCLUSIONS AND OUTLOOK 121

8. REFERENCES 124

LIST OF FIGURES.....132

LIST OF TABLES.....137

Introduction

Music is what feelings sound like

Unknown Author

1. Introduction

1.1. Research Topic

As a major product for entertainment, there is a huge amount of digital musical content produced, broadcasted, distributed and exchanged. There is a rising demand for content-based music search services. Similarity-based music navigation is becoming crucial for enabling easy access to the ever-growing amount of digital music available to professionals and amateurs alike. A professional user, such as a radio programmer, may want to search for a different interpretation of one song to include in a radio playlist. In addition, a radio programmer has the need to discover new songs and artists to help his listeners to discover new music. The music amateur on the other hand has different needs, ranging from active music discovery for the fans, to the simple seed song playlist generation of similar items. Such ways to organize musical collections as genre classification and title structuring are important as they facilitate music navigation and discovery.

Manual indexing of audio content is highly time-consuming and usually not compatible with the huge amount of audio data. In order to make high quality annotations musical experts are required. The need for experts combined with the huge amount of data to be annotated makes manual indexing hard to realize. As a consequence, systems that use human expert judgments exist but they are not numerous (e.g. All Music Guide and Music Genome Project).

The aim of this work is to develop new musical content descriptors and similarity measures which will allow automatic musical content organizing (search by similarity, automatic playlist generating) and labeling (automatic genre classification). We tried to consider the problem of content descriptor building from the musical point of view in addition to low-level spectral similarity measures.

1.2. Problems and Objectives

As compared to a vocal signal, a music signal is likely to be more stationary and posesses some very specific properties in terms of musical tones, intervals, chords, instruments, melodic lines and rhythms, etc. [TANG 93]. While many effective and high performance music information retrieval (MIR) algorithms have been proposed [CAS 05; LOG 01; MAN 05; MCK 03; MEN 05; SCAR 05; TZAN 02; WEST 04], most of these works unfortunately tend to consider a music signal as a vocal one and make use of MFCC-based

features which are primarily designed for speech signal processing. Mel Frequency Cepstrum Coefficients (MFCC) was introduced in the 60's and used since that time for speech signal processing. The MFCC computation averages spectrum in sub-bands and provides the average spectrum characteristics. Whereas they are inclined to capture the global timbre of a music signal and claimed to be of use in music information retrieval [FOOT 97; LOG 00], they cannot characterize the aforementioned music properties as needed for perceptual understanding by human beings and quickly find their limits [AUCO 04]. Recent works suggest combining spectral similarity descriptors with high-level analysis in order to overcome existing ceiling [PAMP 06].

The objective of this work is music retrieval by similarity and automatic labeling (music genre classification) by introducing an approach to high-level musical analysis.

1.3. Our Approach and Contributions

We propose an approach based on music features and corresponding similarity measures for music information retrieval. While popular spectrum-related techniques tend to characterize music timbre properties, we propose to complete them by musical features which can help to overcome the existing limits and, hence, to enhance the performance of music information retrieval algorithms. In this work, we suggest extracting music properties such as rhythmic, melodic, tonality and timbre fingerprints for automatic music search by similarity as well as automatic classification.

The Fast Fourier Transform and the Short-Time Fourier Transform have been the traditional techniques in audio signal processing. This classical approach is very powerful and widely used owing to its great advantage of rapidity. However, a special feature of musical signals is the exponential law of notes' frequencies. The frequency and time resolution of the FFT is linear and constant across the frequency scale while the human perception of a sound is logarithmic according to Weber-Fechner law (including loudness and pitch perception). Indeed, as it is well known, the frequencies of notes in equally-tempered tuning system in music follow an exponential law (with each semi-tone the frequency is increased by a factor of $2^{1/12}$). If we consider a frequency range for different octaves, this frequency range is growing as the number of octave increases. Thus, to cover a wide range of octaves with a good frequency grid large sized windows are necessary in the case of FFT; this affects the time resolution of the analysis. On the contrary, the use of small windows makes resolving frequencies of neighboring notes in low octaves almost impossible. The ability of catching all octaves in music with the same frequency resolution is essential for music signal analysis, in particular construction of melodic similarity features. Hence, as the basis of our work in music feature based MIR, we propose a new music signal

analysis technique by variable-resolution transform (VRT) particularly suitable to music signal.

Our VRT is inspired by Continuous Wavelet Transformation (CWT) introduced 20 years ago [KRON 87] and designed in order to overcome the limited time-frequency localization of the Fourier-Transform for non-stationary signals. Unlike classical FFT, our VRT depicts similar properties as CWT, i.e. having a variable time-frequency resolution grid with a high frequency resolution and a low time resolution in low-frequency area and a high temporal/low frequency resolution on the other frequency side, thus behaving as a human ear which exhibits similar time-frequency resolution characteristics [TZAN 01].

The Algorithms proposed in this work are all based on our VRT and can be divided into 3 groups according to their purposes: 1) spectral features extraction (used in segmentation task in [PAR 05] and music structuring task as well, which are not considered in this thesis), 2) beat detection and 3) automatic fundamental frequency (f_0) estimation. The beat detection algorithm issues a beat probability curve which is then transformed into rhythmic similarity features. The f_0 estimation algorithm delivers f_0 or note candidates as well as relative amplitudes of their partials. This information constructs melodic, tonality and timbre similarity features. Since these features are presented in the form of histograms, we also consider and apply various simple and efficient histogram comparison techniques.

Our similarity measures have been evaluated in several musical information retrieval applications such as automatic genre classification, listening test, reinterpreted compositions search and playlist relevance analysis. In the case of genre classification these similarity measures were combined with classical spectrum-based PGM-MLP [HARB 03] features and have shown a significant improvement of classification rates.

The listening test had an objective of evaluating the quality for the automatically generated lists of similar musical titles. The overall evaluation score as well as the score distribution have been obtained for various musical similarity measures and their combinations.

Playlist relevance analysis and reinterpreted compositions search are a kind of an objective evaluation of the similarity computation approaches. In playlist relevance analysis the number of musical titles in similarity playlists belonging to one artist or one genre were considered.

Our VRT has also been used in other related applications such as audio track segmentation in video content analysis and has shown promising results [PAR 05]. However, the main drawback of VRT is the burden of heavy computation which was partly overcome by using low-level programmed algorithms.

1.4. Organization of the manuscript

The thesis is organized into 5 major chapters starting from the definition of the problematic and description of related work in Chapter 2, then ascending different levels of musical signal analysis and finishing by its direct applications.

The Chapter 3 starts from a description of musical signal specificity. In the relation to musical signal properties various approaches to signal analysis are discussed. Finally, the chapter presents a technique of variable resolution transform which is then used across the whole thesis.

The next Chapter 4 passes from the signal analysis to extraction of rhythmical properties of a music signal. At this stage an algorithm of rhythm detection is presented and partly evaluated. The information delivered by the aforementioned rhythmic analysis algorithm is subsequently involved in the construction of rhythmical similarity features. A small evaluation of tempo estimation is carried out in this chapter.

Chapter 5 touches upon melodic-related similarity aspects. The chapter starts with a description of a multiple-f_0 extraction algorithm and its evaluation. The algorithm extracts pitch candidates together with their relative harmonic amplitude. The pitch candidates are then forming melodic-related fingerprints such note succession histogram.

Finaly, Chapter 6 is dedicated to applications of musical similarity measures. Two of them are considered – automatic music genre classification and music retrieval by similarity. Evolutions of both applications are given.

1.5. List of publications

Most of the work presented in this thesis is not published yet. The following conference publications only partially cover the various matters presented within this manuscript. Several journal papers are in preparation.

- A Paradzinets, O. Kotov, H Harb, L. Chen., Continuous Wavelet-like Transform Based Music Similarity Features for Intelligent Music Navigtion. *Proceedings of CBMI07, Bordeaux, France.* 2007.
- Kotov O., Paradzinets A., Bovbel E. Musical Genre Classification using Modified Wavelet-like Features and Support Vector Machines. *Proceedings of EuroIMSA, Chamonix, France.* 2007
- Paradzinets A., Harb H., Chen L., "Use of Continuous Wavelet-like Transform in Automated Music Transcription", *Proceedings of EUSIPCO* 2006
- Parshin V., Paradzinets A., Chen L. Multimodal Data Fusion for Video Scene Segmentation, *Proceedings of VIS* 2005
- Paradzinets A., Chen L., Bovbel E. (2004). Histogram-Based Algorithm for Speaker Segments Regroupment in Audio Databases Indexing Applications. *In proceedings of RIAO, pp 793-799, April 26-28, 2004, Avignon – France,* 2004

Chapter II

Problem of music similarity and related work

2. Problem of music similarity and related work

The ocean of music similarity search application is immense. One can imagine a music expert, or just an amateur, browsing a many-thousand files music collection and wishing to find music of a particular kind or spirit. The meta-data which are present nowadays bring very poor aid in this case. Navigation by an artist or pre-defined genre can hardly cover needs of intelligent navigation. For example, a fan may be looking for music similar to the one he likes but interpreted by a different orchestra or group.

One can imagine a radio-programmer, doing the same work but in different situation – programming playlists which will follow a defined format of music.

A music-selling online store is a direct application, where customers could be proposed to listen to/buy music which is ***similar*** to the music he or she has just purchased. There are means to use statistical information of a kind "those who had bought this also bought that". These methods, however, need a great number of transactions in order to obtain meaningful statistics and hence to be effective. Moreover, these methods tend to be biased towards famous titles.

Search by similarity can have its perfect place in portable mp3 players where it could be found behind an intelligent shuffle function.

In general, similarity of music is based on subjective judgments, making it difficult to define. We can attempt to define the musical similarity as the feature that lets a human subject create a "playlist" of music pieces based on his/her particular taste. Musical similarity can be expressed in musical terms, i.e. musical tones, intervals, chords, instruments, melodic lines and rhythms, etc. Additional music feature which can be mentioned is the timbre. The timbre is defined as the feature that permits humans to discriminate two sound objects having the same pitch.

Automatic music similarity computation can be defined as the problem of musical feature extraction, building models and comparing them.

2.1. State of the art

There exists a lot of works in the literature dedicated to automatic computation of music similarity. The most popular approach consists of (Gaussian Mixture Modeling) GMM modeling of several MFCC-like characteristics. *Pachet et al.* within the framework of the CUIDADO project [PACH 03], propose a combination of similarity measurements based on GMM modeling and co-occurrence analysis. [LOG 01] uses MFCC characteristics with GMM modeling and the distance "Earth Moving Distance" to estimate the acoustic similarity between two segments of music. Within a *peer to peer*

framework, [" AUM 03] uses basic characteristics: MFCC and GMM for the music search by similarity.

All the previous works thus build their models over MFCC characteristics which only capture somehow the global timbre property of a music signal while mostly omitting other major music properties such as melody, harmony and rhythm. They differ in the way of computing similarity distances.

In a similar context, [TZAN 03] proposes to describe a segment of music by a texture annotation and values of averages and variances of the traditional characteristics like MFCC, spectral flux, spectral centroid and some musical context characteristics such as beat histogram and histogram of pitch. This approach is an interesting one since it tries combining implicitly spectral and pro-musical features.

Another interesting approach [" ERE 03] proposes to make an anchor model for the music. A segment of music is characterized by a vector describing its membership to anchors. The analysis of vectors describing the segments of music makes it possible then to measure the acoustic similarity between two segments. The problem of the approach is that it is still MFCC-based.

Cheng Yang presents in [YANG 02] an algorithm for selection of spectrum locations which are most likely to contain relevant information. A spectral signature extracted from these locations is then used as the basis of a signature for the estimation of a similarity between two music segments. The author also shows that the problem of search for various interpretations of a work is a difficult problem which is unsolved by their algorithms.

A pioneer work in this field is MuscleFish [WOLD 96] which make use of averages, variances and correlations of features such as *loudness,* brightness bandwidth and others to characterize a segment of sound. The Euclidian distance is applied to estimate the similarity between two sound segments. The characteristics which are used are purely spectral and hence cannot bring any musical knowledge to similarity models.

In more recent works, [AUCO 05] propose an interesting approach of timbre related music similarity using Gaussian Mixture Models. In this work authors propose to model the global timbre of musical pieces and compute similarity between models. Further evolution of these methods is presented in [PAMP 05]. Authors introduce two new descriptors "Focus" and "Gravity". However, a detailed consideration of these works confirms that the basic technique behind is always Mel-Frequency Cepstrum Coefficients. The novelty of these works is somehow the way for further processing, i.e. model building and their comparison. Indeed, the first work is a very popular one (based on classical methods). At the same time, the notion of global timbre is rather not clear while the timbre itself is related to the way in which two

instruments sound differently for a human listener. From the point of view of global timbre, a melody played by the same instrument but transposed by some number of intervals will have different timbre while for a human listener it is still the same instrument and the same timbre. The question of perceptual timbre similarity is discussed in [MCAD 92]. In their work the authors develop the notion of timbre distances with the aim of testing whether musician and nonmusician listeners used the relations defined by the perceptual space to perform an analogies task of the sort: timbre A is to timbre ” as timbre C is to which of two possible timbres, D or D'.

Many other works propose various acoustic measurements to catch the different aspects of music similarity. ”ut the difficulty is always that the perceptive similarity is semantic and holds a good part of subjectivity. It was suggested by authors in [AUCO 04] the pure acoustic or spectral similarity quickly finds its limits. Authors of state-of-the-art works generally agree that by using the majority of MFCC-based similarity features it is not possible to accurately describe musical similarity aspects [PAMP 06].

Lots of other temporal, spectral and harmonic features are proposed in the context of sound description (see e.g. [PEET 04]).

During the last few years scarce works appeared which try to describe musical signal using musical context descriptors. One of such works is presented in [GOME 06]. The work considers aspects of high level musical similarity by musical tonality. Algorithms of tonality induction and their evaluation are described. Authors tried to apply methods of tonality comparison to intelligent music navigation problem and confirmed the validity of such application. It reinforces the evidence that classical MFCC-based and other related approaches must be complemented by musical-level descriptors in order to come closer to the human perception of music similarity.

2.1.1. Millions of audio features which are …similar

Here we list some of the basic spectral features based on the magnitude of the FFT (STFT) spectrum. These features are widely known and their definition can be found in many works (see e.g. [TZAN 02a])

- **Spectral Centroid**

The spectral centroid is defined as the center of gravity of the magnitude spectrum:

$$C_t = \frac{\sum_{n=1}^{N} S_t[n] * n}{\sum_{n=1}^{N} S_t[n]} \qquad \textbf{(2.1)}$$

here $S_t[n]$ is the magnitude of the Fourier transform of the time t at frequency bin n. The spectral centroid has been shown to have a relation with musical instrument timbre. It was used for example in [WOLD 96] for sound classification or music genre classification [TZAN 02].

- **Spectral Flux**

The spectral flux is defined as follows:

$$F_t = \sum_{n=1}^{N} \left(N_t[n] - N_{t-1}[n]\right)^2 \quad \textbf{(2.2)}$$

It is a squared difference of normalized spectrum $N_t[n]$ between successive frames t and $t-1$. As the spectral centroid, the spectral flux has been shown to have a relation to the instrument timbre. In [HAWL 93] it was used for harmonic music detection. It was used in music genre classification tasks [TZAN 02] as well.

- **Spectral Rollof**

The spectral rolloff is a measure of spectral shape which denotes a frequency R_t below which 85% of the spectrum distribution is concentrated.

$$\sum_{n=1}^{R_t} S_t[n] = 0.85 * \sum_{n=1}^{N} S_t[n] \quad \textbf{(2.3)}$$

- **Zero Crossing Rate**

Zero-crossing rate is the number of times when the signal changes its sign during one time period. It is used for example in such works on music/speech classification [SAUN 96] and in some works dedicated to music classification as well.

- **4Hz Modulation Energy**

This measure is mainly used to characterize speech. Speech has a tendency to have a 4Hz modulation [HOUT 73]. 4Hz modulation energy was found to be useful in speech detection [SCHE 97].

- **Mel-Frequency Cepstrum Coefficients (MFCC)**

We want to mention MFCC individually as nowadays a **vast** number of music information retrieval works are based on them.

Mel-Frequency Cepstrum Coefficients are the most known from the family of cepstral characteristics. These are cepstral coefficients obtained from a spectrum filtered by Mel scale [STEV 40]. The Mel scale is a scale that reflects characteristics of human perception. It affirms that high frequencies are caught by a human ear with less precision in comparison to low frequencies.

The MFCC are obtained as follows:

1. Divide signal into frames.
2. Compute the FFT to obtain the amplitude spectrum.
3. Take the logarithm.
4. Apply the Mel filter.
5. Take the Discrete Cosine Transform.

The MFCC are widely used in speech modeling and recognition. They also were claimed to be of use in music information retrieval [FOOT 97; LOG 00]. The work [LOG 00] which is always referred as a basic work proving the applicability of MFCC in music modeling stays in the context of speech / music discrimination. It makes a suggestion that the application of Mel scale is at least not harmful for this task.

MFCC was initially designed for speech signal analysis and describes the timbre aspect of a speech signal. As music signal is likely to be more stationary as compared to speech signals, its application for music analysis certainly also captures some timbre aspects of music signal. However, the other three major properties of a music signal such as rhythm, melody and harmony are not modeled by MFCC based features. As highlighted by several works ([HAR" 03; PAMP 06] etc), further advances in music analysis or retrieval clearly needs to go beyond MFCC based features and to consider some other music based features for describing rhythm, melodic line and harmonic properties.

A recent work [PACH 07] proposes a method of automatic construction of well-adapted solutions by using several elementary mathematical and logical operators to combine basic features like pitch, centroid, chroma etc., pointing out the fact that the number of *all possible* acoustic features is hardly calculable. Their proposed algorithms are able to search in a space of 10^{20} features and to construct effective analytical features using samples from training database. However, such search space is constructed from possible combination of a **very limited number** of initial features. "y an initial feature we mean a characteristic which is extracted directly from the signal or its spectrum (like MFCC). All further derivatives of the kind AVG(FFT(FFT(MFCC())) are considered to have one origin and hence to be strongly related to it.

2.1.2. Spectral similarity

In most of the cases the spectral similarity is calculated in the following ways. First, spectral vectors are extracted by means of the Short Term Fourier Transform. The original spectrum is further filtered by a filter bank containing 20 filters distributed based on the Mel scale.

In the work [HAR" 01] the Kull" ack-Leibler (KL) distance was used in audio segmentation, and it was found that this distance is suitable for the

problem of audio similarities. The reasons are that the segmentation is a similarity problem between consecutive windows and that KL distance enables the measure of dissimilarity between two spectral distributions.

The KL distance originates from the information theory [COVE 91]. It is a distance between two random variables. The original KL distance doesn't have the properties of a distance, but the symmetric KL is a distance. In the case of Gaussian distribution of the random variables the symmetric KL distance is computed by:

$$KL2(X,Y) = \frac{\sigma_X^2}{\sigma_Y^2} + \frac{\sigma_Y^2}{\sigma_X^2} + (\mu_X - \mu_Y)^2 (\frac{1}{\sigma_X^2} + \frac{1}{\sigma_Y^2}) \quad \textbf{(2.4)}$$

With $\sigma_X, \sigma_Y, \mu_X, and\, \mu_Y$ are respectively the standard deviation of X and Y and the mean of X and Y.

In the case of audio similarity, X and Y are sets of feature vectors obtained from window X, and window Y (several seconds for the two windows for instance).

The long term similarity or the global similarity can be defined as the similarity that a human subject would find when listening to excerpts of 10 to 20 seconds. Here are some long term similarities proposed by [HAR" 03].

Two musical pieces A and " can be segmented to twenty 1-second segments. Any segment from A can be similar to any segment from ", thus the Local Similarity (KL distance) can be calculated between all couples between " and A. However, the problem is how to obtain a global similarity measure between A and " based on their local similarities?

The KL distances between all couples of segments from A, and ", constitute a Local Similarity Matrix:

$$LSM_{A,B} = \begin{matrix} KL_{1,1}^{AB} & KL_{2,1}^{AB} & KL_{3,1}^{AB} & \bullet \\ KL_{1,2}^{AB} & KL_{2,2}^{AB} & KL_{3,2}^{AB} & \bullet \\ KL_{1,3}^{AB} & KL_{2,3}^{AB} & KL_{3,3}^{AB} & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{matrix} \quad \textbf{(2.5)}$$

Six features from the LSM matrix and other KL-based measures can be extracted. These features are: Homogeneity (H), Local Similarity (LS), Min Distance (MD), Sum Distance (SD), Local Similarity for Low Frequencies (LSLF), and Local Similarity for High Frequencies (LSHF). Each of the features is aimed at providing one aspect of acoustic similarity.

- **Homogeneity (H)**

The aim of the H feature is to measure how much A and " are similar in their homogeneity behavior. For example, theoretically a HipHop music piece will have different homogeneity behavior than a Metal music piece. These measures are computed as follows:

$$H = \left| H_A - H_B \right|$$
$$H_A = \frac{1}{M} \sum_{i=1}^{M} (KL_{i,i+1}^{AA})^2 \quad \textbf{(2.6)}$$
$$H_B = \frac{1}{M} \sum_{i=1}^{M} (KL_{i,i+1}^{BB})^2$$

- **Matching (M)**

The Matching (M) feature describes the distance between the minimal values and their diagonal counterparts in LSM. For example, if the component $KL_{i,j}^{AB}$ is the minimal value in row "i", we measure |j-i|, and M is the average of |j-i| for all rows and columns in LSM.

- **Local Similarity (LS)**

The LS measure seeks a dynamic comparison of A and ". It is based on similarities between one block from A (respectively "), with other blocks from " (respectively A), taking the local minimum for 3 neighbors.

$$LS = LS_{A,B} + LS_{B,A}$$

With
$$LS_{A,B} = \frac{1}{M} \sum_{i=1}^{M} \left| \min_{j=i-1}^{i+1} \left[KL_{i,j}^{AB} \right] \right| \quad \textbf{(2.7)}$$
$$LS_{B,A} = \frac{1}{M} \sum_{j=1}^{M} \left| \min_{i=j-1}^{j+1} \left[KL_{i,j}^{AB} \right] \right|$$

- **Min Distance (MD)**

The MD is the sum of the three minimum values in the LSM Matrix. Denote $MD^1{}_{A,B}$, $MD^2{}_{A,B}$ and $MD^3{}_{A,B}$ as the three minimal values in the LSM matrix. Thus the MD is simply the sum of these values.

$$MD_{A,B} = MD^1{}_{A,B} + MD^2{}_{A,B} + MD^3{}_{A,B} \quad \textbf{(2.8)}$$

- **Sum Distance (SD)**

The SD measures the similarity based on all the components of the LSM Matrix:

$$SD_{A,B} = \sum_i \sum_j KL_{i,j}^{AB} \quad \textbf{(2.9)}$$

- **Local Similarity for Low Frequencies (LSLF)**

LSLF is the LS feature but based on the KL distance calculation for frequencies below 1 KHz. It is targeted at concentrating the similarity towards low frequencies.

- **Local Similarity for High Frequencies (LSHF)**

LSHF is the LS feature but based on the KL distance calculation for frequencies between 1 and 4 KHz. The goal of this measure is to target the similarity towards high frequencies.

2.2. Our approach

The number of ways to compute the similarity features of musical compositions is hardly countable. The main goal of constructing similarity features is to come as closer as possible to humans perception. Instead of turning in all senses the MFCC based features which capture in a certain way the timbre property of a music signal, we propose to complement them by the features characterizing music properties such as rhythm, melody and tonality.

Extraction of several groups of features requires lower-level algorithm of musical content analysis which touches upon all stage of processing. The idea can be visualized by the following diagram (Figure 2.1).

	Music-based similarity		**Classical audio-based similarity**	
5	Applications	⇒ playlists, genres	Applications	⇒ playlists, genres
4	Similarity modeling	⇒ distances	Similarity	⇒ distances
3	Music features extraction	⇒ chords, tonality, melody lines, tempo, etc	Modeling	⇒ GMM, fluctuation patterns, etc…
2	Music info computation	⇒ notes, beats	Spectral features extraction	⇒ MFCC, ZCR
1	Signal processing	⇒ spectrum	Signal processing	⇒ spectrum

Figure 2.1. 5-level music-based similarity analysis dataflow model compared to audio-based similarity dataflow.

In the Figure 2.1 we show our vision of music signal analysis in a music information retrieval process which is represented by 5 levels of treatment. It is compared to an approximate model of popular music analysis known in literature. In this thesis we are interested in all stages of our model starting from signal processing. The classical way of doing it is to use the FFT in order to obtain the spectrum of a signal. The advantages and disadvantages of these two approaches as well as the VRT technique we propose are discussed in Chapter 3.

The next stage is the stage of musical information extraction. In the case of classical way of musical signal analysis the phase of musical information extraction does not have any direct analogue; instead the spectral features are extracted directly. The role of higher-level features in that case is played

by descriptive models such as GMM. We focus at the second stage on beat detection and multiple f_0 estimation algorithms we have developed. These lower level algorithms do not provide musical similarity features, yet they provide the underlying information necessary for the construction of such features. The process of their computation and comparison is covered by the stages 3 and 4 of the model, which chapters 4 and 5 of this thesis are dedicated to. The chapter coverage is illustrated in Figure 2.2.

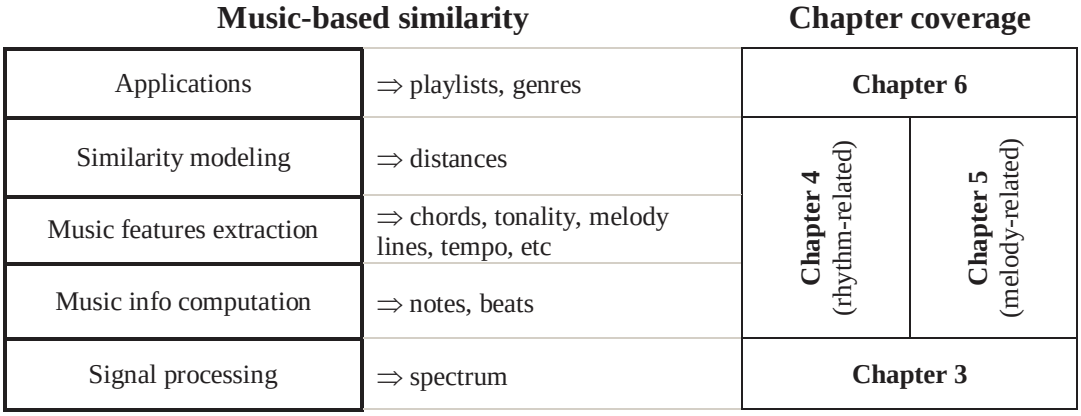


Figure 2.2. 5-level music similarity analysis dataflow and chapter coverage.

Chapter III

Music signal analysis

3. Music signal analysis

The primary stage in every kind of audio based music information retrieval is signal data analysis. Some algorithms perform analysis in the time domain as for example several beat detection algorithms. ”ut the majority of music information retrieval algorithms perform their computation in the frequency domain, or a time-frequency representation, to be exact. So, the performance of all further steps of processing is strictly dependent on the initial data representation.

This chapter gives an introduction to signal treatment. It gives a brief coverage of the classical FFT-based approach and its drawbacks. As opposed to the FFT, the chapter describes the wavelet transform as a novel and promising instrument in musical signal processing. The chapter explains the main principle of classic wavelet transform. A novel variable resolution transform, which is introduced in this work, is also presented in this chapter.

3.1. About music signal

Music is an art form consisting of sound and silence expressed through time. Elements of sound as used in music are pitch (including melody and harmony), rhythm (including tempo and meter), structure, and sonic qualities of timbre, articulation, dynamics, and texture.[1]

Melodies are usually sequences of pitches that are created in Western music with respect to scales and modes and having a certain rhythm. Scales and modes are notions of music theory which describe a set of notes involved in the play. In Western music there are 12 notes. An interval between neighbor notes is called *semitone*.

For better understanding of particularity of music signal, let’s consider the following Figure 3.1 containing a musical excerpt expressed in a form of pattern.

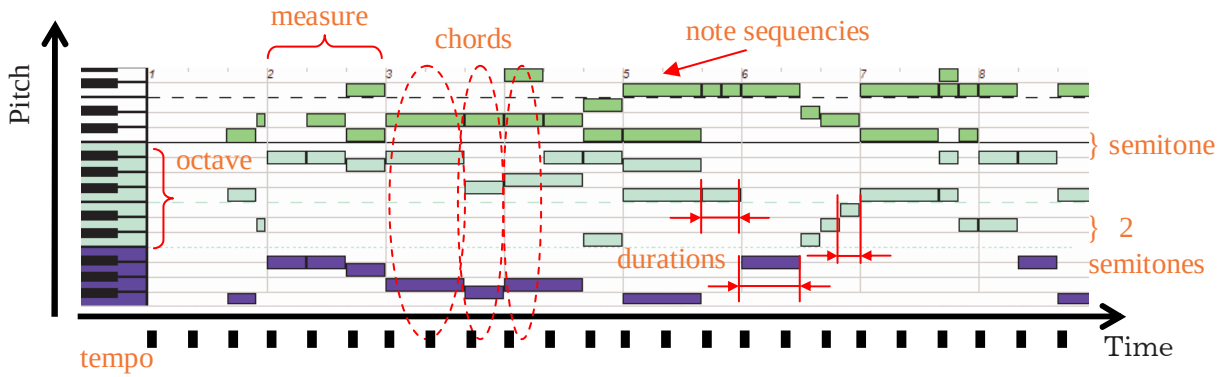


Figure 3.1. Typical music pattern.

[1] http://en.wikipedia.org/wiki/Music

The term *pitch* is directly related to frequency, the higher is the pitch – the higher is the frequency. The relation between pitch and frequency is logarithmic.

$$p = 69 + 12 \cdot \log_2\left(\frac{f}{440Hz}\right) \quad \textbf{(3.1)}$$

In music, pitch is the perceived *fundamental frequency*. Sounds of real instruments do not have single-frequency spectra, but rather comprise a packet of frequencies – the fundamental one and its harmonics. The Figure 3.1 would represent a spectrogram of a musical signal if the instruments in it had a single frequency in their spectrum.

As we have noticed, musical signal have two main properties – temporal and frequency granulation. In reality, we don't speak about frequency granulation of musical signal since there could be a singing voice or percussion instruments in it. Speeding up or slowing down the signal also leads to shifts of note frequencies from their theoretical values corresponding to integer pitches.

In comparison to speech signals music signals can be assumed to be more stationary (during one note or one chord the spectrum of the signal does not change much). Duration of notes can be considered to be around 120-250 ms for 1/8 - 1/16 (quaver - semiquaver) at the most popular tempo of 120 ” PM (the minimal duration of a note which could be found in music at 120 ” PM is then 65 ms for 1/32). Nevertheless, music signals may contain percussion instruments rapidly changing in time or of a very short duration. Singing voice of one or many persons may be also present. The second characteristic feature of a musical signal is *multipitch*. Music is usually represented by multiple simultaneous note events, such as chords. A chord may contain from 2 to many (>10) pitches at the same time. Some instruments may already contain multiple pitches in one note. These facts make musical spectrum much more complicated in the meaning of frequency contents in comparison to speech signals. Such complexity can be referred as *timbre* of the musical sound. The timbre characterizes each voicing instrument defined by its spectrum (presence of harmonic and inharmonic components) and envelope. The timbre allows us to distinguish different instruments.

Unlike voicing instruments, percussion instruments in music may not have their fundamental frequency. Thus, they are undetermined-frequency instruments (no pitch can be perceived). As an example we can mention snare drums, hi-hats and cymbals. These instruments have mainly noisy components in their sound which are then superposed to voiced instruments and form more complex spectrum.

We have briefly described some specificities of a musical signal. In order to be able to extract its musical properties, a well adapted analysis tool is required which will resemble somehow the human auditory system. Human perception of pitch is linear, thus, the tool should have a logarithmic frequency scale. It is also known, that human ear is more sensitive to rapid changes in high frequencies and less sensitive to rapid changes in low frequencies. A tool with similar characteristics might be better suited for analysis of musical signals.

In this work we consider by different level of analysis, starting from obtaining a musical spectrum to extraction of musical features and finally to building similarity models for different musical features and applying them to high-level music analysis such as automatic classification or similarity search. In this context, we are interested to have a tool with both frequency sampling suitable for analyzing notes in equal tempered note system and good frequency resolution in high frequency area since harmonic resolution is important as well. The tool must also provide acceptable time resolution, suitable for beat/onset detection.

3.2. Related work

There are plenty of works in the literature dedicated to musical signal analysis. The common approach is the use of FFT (Fast Fourier Transform) which has become a de-facto standard in music information retrieval community. The use of FFT seems straightforward in this field and relevance of its application for music signal analysis is almost never motivated.

There are some works in music information retrieval attempting to make use of wavelet transform as a novel and powerful tool in musical signal analysis. However, this new direction is not very well explored. [TZAN 02] proposes to rely on discrete wavelet transform for beat detection. Discrete packet wavelet transform is studied in [GRIM 02] to build time and frequency features in music genre classification. In [KADA 92], wavelets are also used for automatic pitch detection.

Spectrum analysis tools with geometric frequency spacing are well known. One of classic examples is Constant Q transform, considered further. Many other custom filterbank techniques have been proposed in literature. These are bounded versions of constant Q transform [KASH 85], fast filter banks [LIM 92] having the goal of minimizing frequency bins dispersion, bounded constant Q filter banks [DINI 07]. In fact, there is no universal analysis tool with ideal frequency grid. It is ether fast transform with linear or quasi-linear (linear within octaves) frequency grid or it is logarithmic frequency heavy computational solutions. Depending on target application, different solutions may perform better.

Another class of approaches to signal analysis is an analysis by modeling. The aim of these approaches is to approximate a multi-sinusoidal model to the signal issuing frequencies and phases of separate harmonic components. An example of such approach is described in [" ADE 02]. Although these methods represent a great interest in musical signal analysis, we are concentrated on spectrum based approaches in our work.

3.2.1. Fourier transform

As it is well known, Fourier transform enables a spectral representation of a periodic signal as a possibly sum of a series of sines and cosines. While Fourier transform gives an insight into the spectral properties of a signal, its major disadvantage is that a decomposition of a signal by Fourier transform has infinite frequency resolution and no time resolution. It means that we are able to determine all frequencies in the signal, but without any knowledge about when they are present. This drawback makes Fourier transform to be perfect for analyzing stationary signals but unsuitable for irregular signals whose characteristics change in time. To overcome this problem several solutions have been proposed in order to represent more or less the signal in time and frequency domains.

One of these techniques is windowed Fourier transform or short-time Fourier transform. The idea behind is to bring time localization into classic Fourier transform by multiplying the signal with an analyzing window:

$$X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-j\omega t}\,dt \qquad \textbf{(3.2)}$$

where $w(t)$ – is a windowing function (usually Gauss window, Hamming or Hann function and many others. The same principle is applied in the **discrete** short-time Fourier transform.

$$X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \qquad \textbf{(3.3)}$$

The problem here is that the short-time discrete Fourier transform has a fixed resolution. The width of the windowing function is a tradeoff between a good frequency resolution transform and a good time resolution transform. Shorter window leads to smaller frequency resolution but higher time resolution while larger window leads to greater frequency resolution but lower time resolution. This phenomenon is related to Heisenberg's uncertainty principle which says that

$$\Delta t \sim \frac{1}{\Delta f} \qquad \textbf{(3.4)}$$

where ψt is a time resolution step and ψf is a frequency resolution step.

A classical example illustrating weakness of Fourier transform is two signals with two sine waves in it (Figure 3.2). The signal on figure a) is a sum of two sine waves while figure b) represents a signal where two sine waves follow each other. It can be noticed that their Fourier transforms (figures c and d correspondingly) have very few differences enabling discrimination of these two signals.

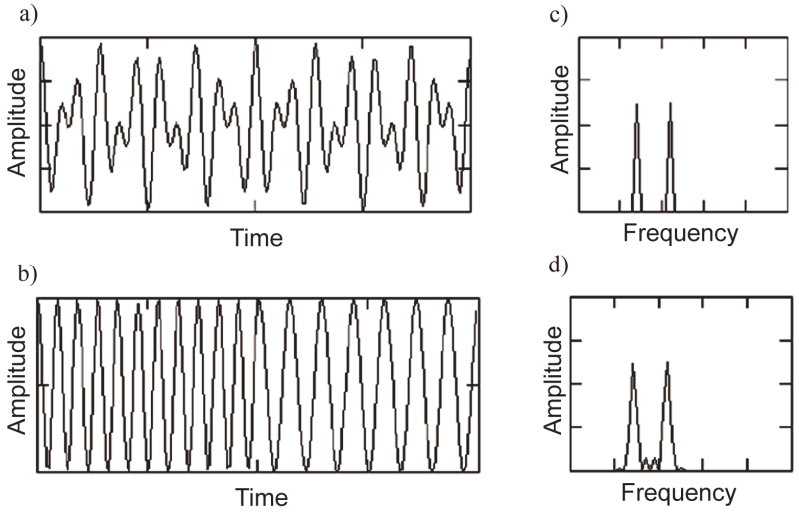


Figure 3.2. Fourier transform of two test signals. The signal a) is composed with two superposed waves with different frequencies, b) the same waves, but concatenated one after another, c) d) their Fourier spectrum

Of course, this example cannot be used to illustrate the weakness of the Fast Fourier Transform in music analysis applications (Fast Fourier Transform is a fast version of the discrete Fourier transform). For example, a 512-sample window in FFT applied on 16kHz-sampled signal gives about 30ms of time resolution which is sufficient for most of music applications where duration of notes can be considered to be around 50-200 ms. However, the frequency resolution step of 512-sample FFT is about a semi-tone in the 3rd octave, what is obviously not enough. Usually short notes are notes which are involved in melodic lines (average-to-high frequencies) while long notes stand more often for bass lines (low frequencies).

Remember that in our work the main goal is music analysis. In this respect, we consider a rather music-related example which illustrates specificities of musical signals. As it is known, the frequencies of notes in equally-tempered tuning system in western music follow a logarithmic law, i.e. adding a certain interval (in semitones) corresponds to multiplying a frequency by a given factor. For an equally-tempered tuning system a semitone is defined by a frequency ratio of $2^{1/12}$. So, the interval between two frequencies is

$$n = 12 \cdot \log_2\left(\frac{f_2}{f_1}\right) \tag{3.5}$$

If we consider a frequency range for different octaves, it is growing as the number of octave is higher. Thus, applying the Fast Fourier Transform we either lose resolution of notes in low octaves (Figure 3.3) or we are not able to distinguish high-frequency events which are closer in time and have shorter duration.

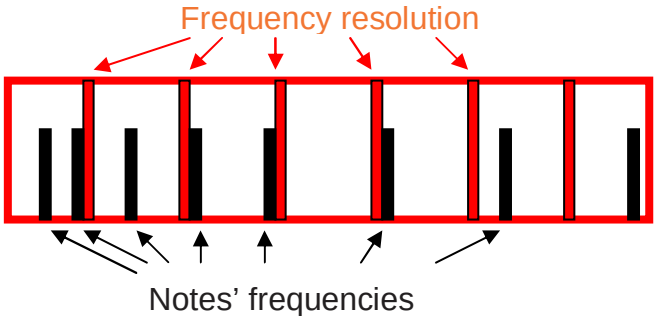


Figure 3.3. Mismatch of note frequencies and frequency resolution of the FFT.

Time-frequency representation, which can overcome resolution issues of the Fourier transform is ***Wavelet transform***. Wavelets (literally "small waves") are a relatively recent instrument in modern mathematics. Introduced about 20 years ago, wavelets have made a revolution in theory and practice of non-stationary signal analysis [KRON 87; MALL 99]. Wavelets have been first found in the literature in works of Grossmann and Morlet [GROS 84]. Some ideas of wavelets partly existed long time ago. In 1910 Haar published a work about a system of locally-defined basis functions. Now these functions are called Haar wavelets. Nowadays wavelets are widely used in various signal analysis, ranging from image processing, analysis and synthesis of speech, medical data and music [KADA 92; LANG 98].

3.2.2. Wavelet transform

Wavelet transform is a form of time-frequency representation. It is divided into continuous and discrete subclasses. Wavelets are functions satisfying certain mathematical rules which are used to represent data or other functions as it was done by Joseph Fourier, who has proposed to represent all periodic functions from 0 to 2π with an infinite sum of sines and cosines.

For many decades researchers wanted more appropriate functions than sines and cosines – the basis of the Fourier analysis – to approximate choppy signals and functions. Wavelets are localized functions and therefore well-suited for approximating data with sharp changes.

Reference fundamental works on wavelets are [COHE 92; DAU" 92; GROS 84; MALL 89] and many others. A work explaining the use of the continuous wavelet transform in signal analysis is, for example, [LANG 98].

3.2.2.1 Continuous wavelet transform

Continuous wavelets transform of a function $f(t) \in L^2(R)$ is defined as follows:

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt \qquad \textbf{(3.6)}$$

where $a, b \in R, a \neq 0$.

In the equation (3.6) $\psi(t)$ is called basic wavelet or mother wavelet function (* stands for complex conjugate). Parameter a is called wavelet scale. It can be considered as analogous to frequency in the Fourier transform. Parameter b is localization or shift. It has no correspondence in the Fourier transform.

The wavelet transform, primarily, is a correlation of a signal being analyzed with mother wavelet function, which is shifted and zoomed on its time axis. Mother wavelet function usually, but not necessarily, looks like a wave, faded to zero on its sides. Therefore, the window of short-time Fourier or Gabor transform is somehow included.

Inverse integral wavelet transform is defined as

$$f(t) = C_\psi^{-1} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W(a,b) \psi \left(\frac{t-b}{a} \right) \frac{1}{\sqrt{a}} \frac{dadb}{a^2} \qquad \textbf{(3.7)}$$

where C_ψ - is a normalizing factor defined as follows:

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\Psi|^2}{|\omega|} d\omega < \infty \qquad \textbf{(3.8)}$$

Inequality (3.8) is a condition for the existence of the inverse transform. Here $\Psi(\omega)$ stands for the Fourier transform of $\psi(t)$. To satisfy the aforementioned expression, the Fourier transform of $\psi(t)$ at zero frequency must be equal to zero. Hence, we can rewrite it as follows:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \qquad \textbf{(3.9)}$$

In other words, $\psi(t)$ must be a wave, i.e. it must satisfy to ***zero mean condition***.

It is sometimes necessary that wavelet function satisfy an equality to zero of higher moments:

$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0 \tag{3.10}$$

Center $\langle t \rangle$ and radius Δ_t of a function can be used in order to estimate its localization.

$$\langle t \rangle = \frac{1}{\|\psi\|^2} \int_{-\infty}^{\infty} t |\psi(t)|^2 dt \tag{3.11}$$

$$\Delta_t^2 = \frac{1}{\|\psi\|^2} \int_{-\infty}^{\infty} [t - \langle t \rangle]^2 |\psi(t)|^2 dt \tag{3.12}$$

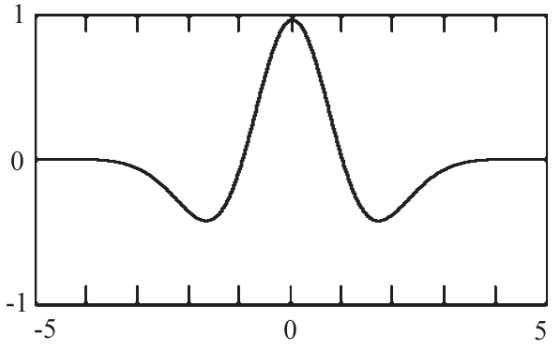
Effective width of a wavelet is usually taken as $2\Delta_t$. The same expressions are also true for the frequency axis:

$$\langle \omega \rangle = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega |\Psi(\omega)|^2 d\omega \tag{3.13}$$

$$\Delta_\omega^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} [\omega - \langle \omega \rangle]^2 |\Psi(\omega)|^2 d\omega \tag{3.14}$$

Let's consider now some examples of wavelets functions.

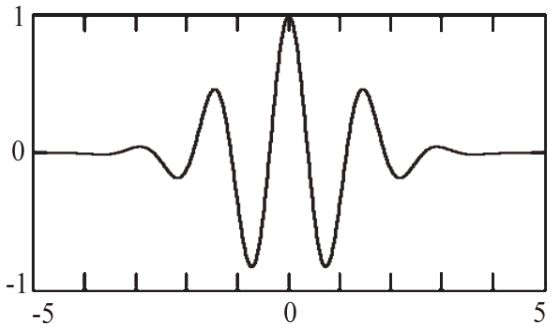
- Mexican Hat wavelet



$$\psi(t) = \left(1 - t^2\right) e^{-t^2/2} \tag{3.15}$$

The name of this wavelet comes from its shape. This kind of wavelets is obtained by differentiation of Gaussian exponents. MHAT wavelets are well localized in time and frequency domain.

- Morlet wavelet



$$\psi(t) = e^{-t^2/\alpha^2}\left(e^{ik_0 t} - e^{-k_0^2\alpha^2/4}\right) \tag{3.16}$$

Morlet wavelet is a Gaussian-modulated flat wave. Parameter α defines Gaussian's width and k_0 stands for frequency (usually taken $k_0=2\pi$). Parameter α adjusts time-frequency localization and hence time-frequency resolution scale of the transform.

3.2.2.2 Discrete wavelet transform

Discrete wavelet transform is a sampled version of the wavelet transform where sampling points are selected following the system $(a^m, na^m b)$ with integers $m,n \notin Z$.

The discrete wavelet transform is often calculated by a *pyramidal algorithm* by passing a signal $x(t)$ through a series of low-pass (g) and high-pass (h) filters related to each other. The filter outputs are then downsampled by factor 2.

$$\begin{aligned} y_{low}[n] &= \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \\ y_{high}[n] &= \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \end{aligned} \tag{3.17}$$

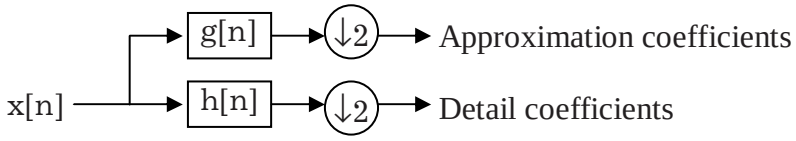


Figure 3.4. Block diagram of DWT filter cascade.

The decomposition is repeated by cascading filter pairs in a form of binary tree for further increase of frequency resolution. This tree is also called a filter bank.

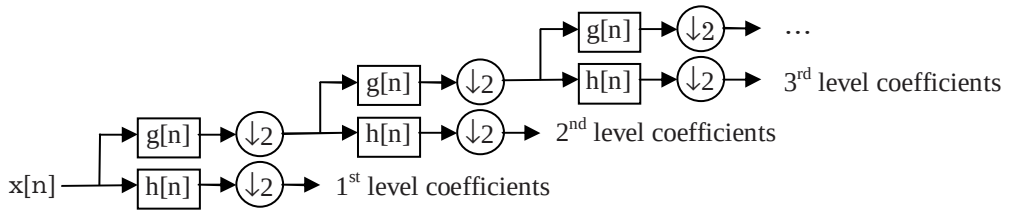


Figure 3.5. Example of DWT filterbank.

Discrete wavelet transform is mostly used in signal coding and data compression.

A variety of discrete wavelet transform is *wavelet packet transform* (see e.g. [CODY 94; MALL 99]). Wavelet packet transform can tile the frequency space in a discrete number of intervals. It can be represented as a binary tree, defining subspaces of details W_L, as illustrated in Figure 3.6.

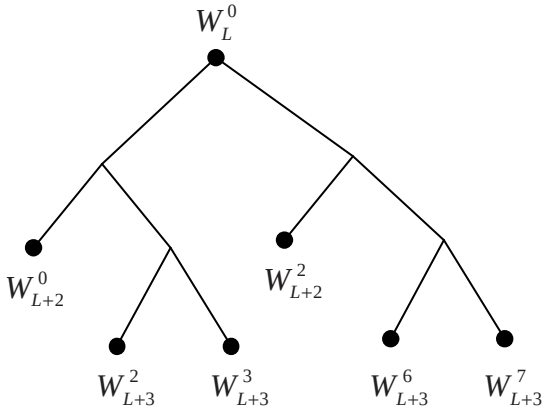


Figure 3.6. Example of a valid wavelet packet tree.

Wavelet packets can be adapted for music analysis for example by defining Heisenberg boxes matching musical octaves and musical notes [GRIM 02].

3.2.3. Other transforms and filter banks

3.2.3.1 Constant Q transform

The idea to adapt the time/frequency scale of a Fourier-related transform to musical applications is not completely novel. A technique called **Constant Q Transform** [" ROW 91] is related to the Fourier transform and it is used to transform a data series to the frequency domain. Like the Fourier transform a constant Q transform is a bank of filters, but contrary to the Fourier transform it has geometrically spaced center frequencies $f_k = f_0 \cdot 2^{\frac{k}{b}}$ (k = 0; ...), where b is the number of filters per octave. In addition it has a

constant frequency resolutions ratio $R_{f/\Delta} = \left(2^{\frac{1}{b}} - 1\right)^{-1}$. Choosing appropriately k and f_0 makes central frequencies to correspond to the frequencies of notes.

In general, the transform is well suited to musical data (see e.g. [NAWA 01], in [ESSI 05] it was successfully used for recognizing instruments), and this can be seen in some of its advantages compared to the Fast Fourier Transform. As the output of the transform is effectively amplitude/phase against log frequency, fewer spectral bins are required to cover a given range effectively, and this proves useful when frequencies span several octaves. The downside of this is a reduction in frequency resolution with higher frequency bins.

The transform mirrors the human auditory system, whereby at lower frequencies spectral resolution is better, whereas temporal resolution improves at higher frequencies, and so this kind of analysis makes sense for musical signal.

In addition, the harmonics of musical notes form a pattern characteristic to the timbre of the instrument in this transform. Assuming the same relative strengths of each harmonic, as the fundamental frequency changes, the relative position of these harmonics remains constant. This makes identification of instruments much easier. In this extent it is similar to log-sampled continuous wavelet transform with sinusoidal mother function. Its advantages and disadvantages we described in section 3.3.

3.2.3.2 Other filter banks

”esides constant Q transform there are bounded version of it (”QT) which use quasi-linear frequency sampling when frequency sampling remains linear within separate octaves. This kind of modification allows construction of medium complexity computation schemes in comparison to standard CQT. However, making the frequency sampling quasi-linear (within separate octaves) renders the finding of harmonic structure much more complex task.

Fast Filter ”anks are designed to deliver higher frequency selectivity maintaining low computational complexity. This kind of filter banks inherit all disadvantages of FFT in music analysis applications (discussed in §3.2.4).

More advanced techniques, described for example in [DINI 07] are medium-complexity methods which aim to overcome disadvantages of FFT and try to follow note system frequency sampling. However, octave-linear frequency sampling keeps the same disadvantage as in the case of bounded Q transforms.

3.2.4. Discussion: FFT vs WT for music signal analysis?

Here we would like to focus on the windowed Fourier transform and the wavelet transform. One way to see the time-frequency resolution differences between the Fourier transform and the wavelet transform is to look at the basis function coverage (tiling) of time-frequency plane. Figure 3.7 shows a windowed Fourier transform. The window is square and it truncates the sine and cosine function of the transform by a particular width. ” ecause the STFT uses the window of the same size for all frequencies, the resolutions is the same at all locations in the time-frequency plane.

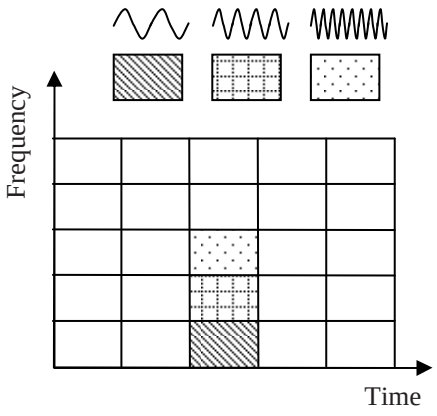


Figure 3.7. Fourier basis functions, time-frequency tiles, and coverage of the time-frequency plane.

The main advantage of the wavelet transform is that the window size is not constant. It changes across the frequency axis. Figure 3.8 shows the coverage in the time-frequency plane with a wavelet mother function, namely Morlet wavelet.

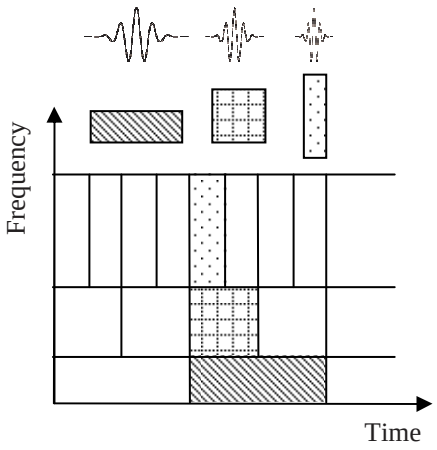


Figure 3.8. Morlet wavelet basis functions and time-frequency coverage.

One important thing is that the wavelet transform does not have a single set of basis functions like the Fourier transform. Instead, the wavelet transform utilizes an infinite set of possible basis functions. Thus, it has an access to a wide range of information including the information which can be obtained by other time-frequency methods such as Fourier transform.

As explained in brief introduction on music signal, a music excerpt can be considered as a sequence of note (pitches) events lasting certain time (durations). ”eside beat events, singing voice and vibrating or sweeping instruments, the signal between two note events can be assumed to be quasi-stationary. The duration of a note varies according to the main tempo of the play, type of music and type of melodic component the note is representing. Fast or short notes usually found in melodic lines in high frequency area while slow or long notes are usually found in bass lines with rare exceptions. Let's consider the following example in order to see the difference between the Fourier transform and wavelet one. We construct a test signal as containing two notes E1 and A1 playing simultaneously during the whole period of time (1 second). These two notes can represent a bass line, which, as it is well known, does not change quickly in time. At the same time, we add 4 successive notes ”5 with small intervals between them (around 1/16 sec). These notes can theoretically be notes of the main melody line. Let's see now the Fourier spectrogram of the test signal with a small analyzing window.

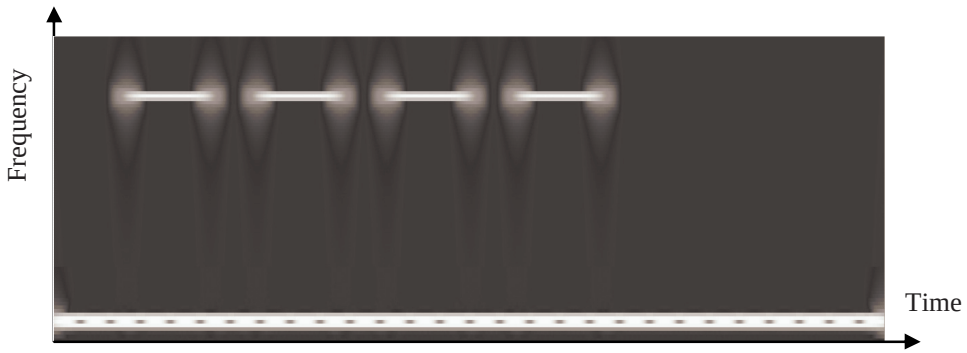


Figure 3.9. Small-windowed Fourier transform (512 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top.

As we can see from Figure 3.9, while high-octave notes can be resolved in time, two bass notes are irresolvable in frequency domain. Now we increase the size of the window in the Fourier transform. Figure 3.10 illustrates the resulting spectrogram.

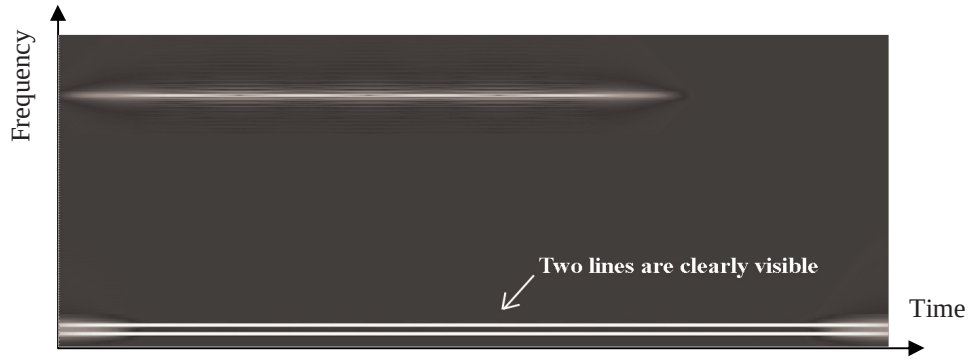


Figure 3.10. Large-windowed Fourier transform (≥1024 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top.

As we can see, two lines at the bottom of the spectrogram are now clearly distinguishable while the time resolution of high-octave notes has been lost.

Finally we apply wavelet transform to the test signal. Figure 3.11 shows such Morlet-based wavelet spectrogram of our test signal.

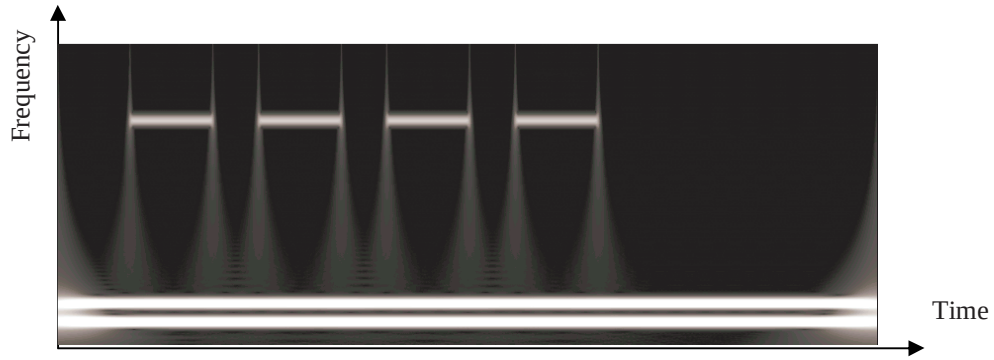


Figure 3.11. Wavelet transform (Morlet) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top**.**

Of course, the given example is quite artificial; however it explains well our motivation for a wavelet like time-frequency representation of a signal. It is also known, that human ear exhibits time-frequency characteristic closer to that from wavelet transform [TZAN 01].

A crucial disadvantage of the wavelet transform is its heavy computational demands. There are fast algorithms for wavelet transform computation (E.g. [RIOU 91]). They are based on self-similarity of wavelets at different scales or on the fact of particular sampling of the scale axis of the wavelet. We overcome this problem by manually programming wavelet-like transform algorithms (including our implementation of VRT on which our work is based on) using fast vector arithmetic of modern CPUs (MMX/SSE instruction sets), which is usually never done automatically by high-level language compilers.

3.3. Variable Resolution Transform

In this section, we introduce our Variable Resolution Transform (VRT) specifically designed for music signal analysis. Inspired by the wavelet and constant Q transforms, our VRT is a set of specially crafted filters with central frequencies following a logarithmic scale which only "resembles" a wavelet transform and in reality does not appear to be a true wavelet transform (generally, all kinds of transforms as for example FFT, CQT can be considered as filter sets [SMIT 07]). Our VRT is a fundamental tool for music signal analysis in our work as it is used for deriving music features such as beat detection, multiple f_0 estimation etc.. which, in turn, enable the construction of melodic, tonality and timbre similarity features. There are no "universal" and 100% efficient filterbanks suitable for all kinds of applications. Our tool is one of numerous possible solutions. We position it to be better adapted for harmonic structure extraction owing to its higher frequency resolution in upper frequency area.

3.3.1. Building Variable Resolution Transform

Our Variable Resolutions Transform (VRT) is first derived from the classic definition of Continuous Wavelet Transform (CWT) given in §3.2.2.1 in order to enable a variable time-frequency coverage which should fit to music signal analysis better. The consideration of specific properties of music signal finally leads us to change the mother function as well and thus our VRT is not a true CWT but a filter bank.

3.3.1.1 The basis

We start the construction of our VR Transform from Continuous Wavelet Transform defined by (3.6). Thus, we define our mother function as follows

$$\psi(t) = H(t,l)e^{j\cdot 2\pi\cdot t} \tag{3.18}$$

where *H(t,l)* is the Hann window function of a length l with $l \in Z$ as defined by (3.19). In our case l will lie in a range between 30-300 ms. Notice that using different different length values l amounts to change the mother wavelet function Ψ.

$$H(t,l) = \frac{1}{2} + \frac{1}{2}\cos\frac{2\pi t}{l} \tag{3.19}$$

Once the length l is fixed, function (3.18) becomes much more similar to a Morlet wavelet. It is an oscillating function, a flat wave modulated by a Hann window. The parameter l defines the number of periods to be present in the wave. Figure 3.12 illustrates such a function with *l=20 waves*.

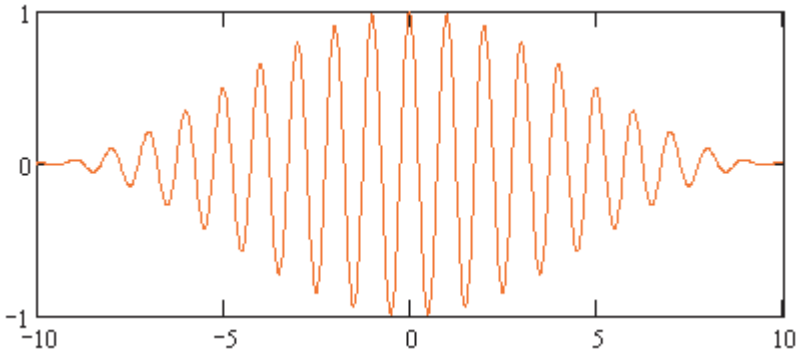


Figure 3.12. Our mother wavelet function. A flat wave modulated by a Hann window with *l=20*.

We can write according to the definition of the function (since $l < \infty$):

$$\int_{-\infty}^{\infty} |\psi(t)| dt < \infty \text{ and } \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{3.20}$$

The function is oscillating symmetrically around its 0 value, hence

$$\int_{-\infty}^{\infty} \psi(t) dt \to 0 \tag{3.21}$$

Using (3.6) we write a discrete version of the transform for a sampled signal between the instants of time form $t-l/2$ to $t+l/2$. Applying the wavelet transform to the signal, we are interested in spectrum magnitude

$$W(a,b) = \frac{1}{\sqrt{a}} \sqrt{\left(\sum_{t=-l/2}^{l/2} s[t+b] \cdot H\left[\frac{t}{a}, l\right] \cdot \cos\left(2\pi \frac{t}{a}\right)\right)^2 + \left(\sum_{t=-l/2}^{l/2} s[t+b] \cdot H\left[\frac{t}{a}, l\right] \cdot \sin\left(2\pi \frac{t}{a}\right)\right)^2} \tag{3.22}$$

Here $W(a,b)$ is the magnitude of the spectral component for the signal $s[t]$ at time instant b and wavelet scale a.

The value of $W(a,b)$ can be obtained for any a and b provided that b does not exceed the length of the signal. The equation (3.22) thus defines a Continuous Wavelet Transform for a discrete signal (time sampling).

The scale of wavelet a can be expressed in terms of central frequency corresponding to it since our mother function is a unit oscillation:

$$a = \frac{f_S}{f} \tag{3.23}$$

where f_S is the sampling frequency of the signal.

A higher value of a stands for a lower central frequency.

3.3.1.2 Logarithmic frequency sampling

First of all, the sampling of the scale axis is chosen to be logarithmic in the meaning of frequency. It means that each musical octave or each note will have an equal number of spectral samples. Such a choice is explained by the properties of a music signal, which is known to have frequencies of notes to follow a logarithmic law (following the human perception). Logarithmic frequency sampling also simplifies harmonic structure analysis and economizes the amount of data necessary to cover the musical tuning system effectively.

A voiced signal with single pitch is in the general case represented by its *fundamental frequency* and the fundamental frequency's *partials (harmonics)* with the frequencies equal to the fundamental frequency multiplied by the number of a partial. Hence the distances between partials (harmonic components) and f_0 (basic frequency) in logarithmic frequency scale are constant independently from f_0. Such harmonic structure looks like a "fence", depicted on Figure 3.13.

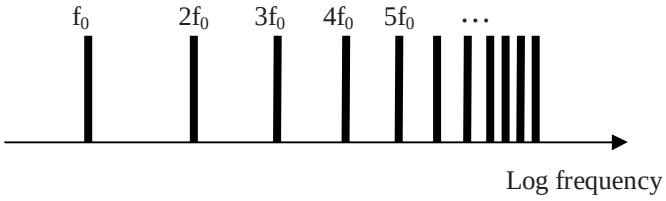


Figure 3.13. Harmonic structure in logarithmic frequency scale.

In order to cover the frequency axis form f_{min} to f_{max} with N frequency samples with a logarithmic law we define a discrete function $a(n)$, which denotes the scale of wavelet and where n stands for a wavelet bin number ranging in the interval 0..N-1.

$$a(n) = \frac{f_S}{f_{\min} e^{\frac{n}{N} \ln\left(\frac{f_{\max}}{f_{\min}}\right)}} \tag{3.24}$$

Now the transform (3.22) sampled in both directions gives

$$W(n,b) = \frac{1}{\sqrt{\frac{f_S}{f_{\min} e^{n \cdot C}}}} \left| \sum_{-l/2}^{l/2} s[t+b] \cdot H\left[\frac{t \cdot f_{\min} \cdot e^{n \cdot C}}{f_S}, l \right] \cdot e^{-i \frac{t f_{\min} \cdot e^{n \cdot C}}{f_S}} \right| \tag{3.25}$$

where the constant $C = \frac{1}{N} \ln\left(\frac{f_{\max}}{f_{\min}}\right)$.

Expression (3.25) is the basic expression to obtain an N-bin spectrogram of the signal at time instant b. Thus, for a discrete signal of length S, expression (3.25) provides $S \times N$ values for each instant of time, N being the number of frequency samples. The expression (3.25) is still a sampled version of the Continuous Wavelet Transform where the sampling of the scale axis has been chosen logarithmic for N samples.

Frequency dependency on the bin number has the following form (with $f_{min}=50$, $f_{max}=8000$, $N=1000$).

$$f(n) = f_{\min} e^{\frac{n}{N}\ln\left(\frac{f_{\max}}{f_{\min}}\right)} = f_{\min} e^{nC} \qquad \textbf{(3.26)}$$

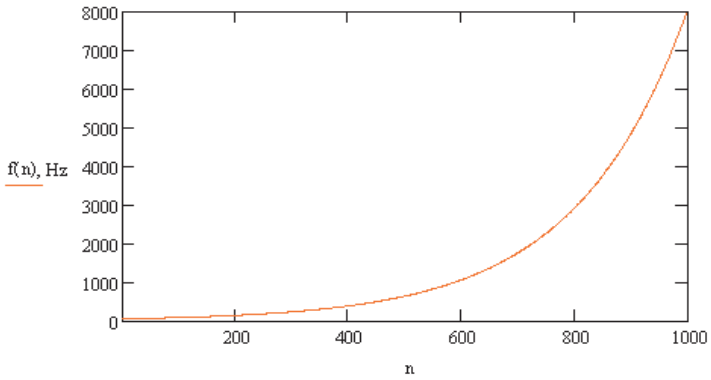


Figure 3.14. Equivalent central frequency of the wavelet according to its bin number. $f_{min}=50$, $f_{max}=8000$.

In order to depict the time/frequency properties of music signals by this discretized wavelet transform with a fixed length value (l=20), let's consider wavelet spectrograms of several test signals. Figure 3.15 shows the wavelet spectrogram $W(n,b)$ of a piano recording. One can observe single notes on the left and chords on the right. Fundamental frequency (f_0) and its harmonics can be observed in the spectrum of each note. As we can see from the Figure 3.15, up to 5 harmonics are resolvable. Higher harmonics after the 5^{th} one become indistinguishable especially in the case of chords where the number of simultaneously present frequency components is higher.

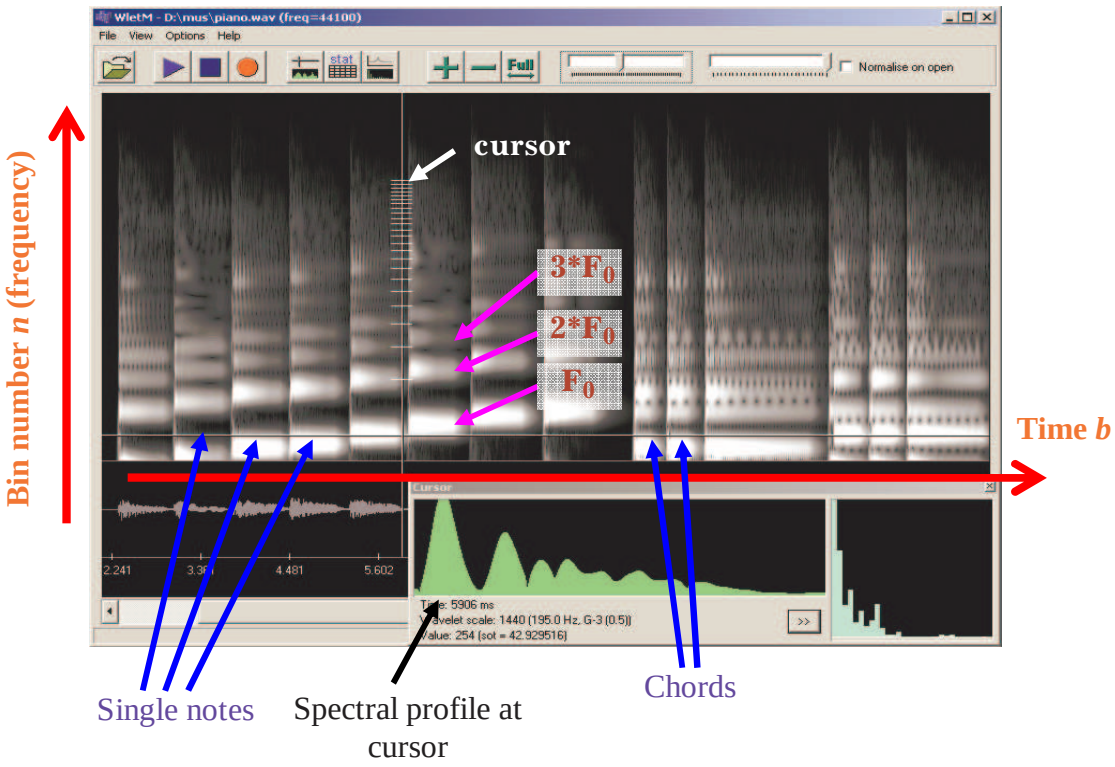


Figure 3.15. Wavelet spectrogram of a piano recording (wavelet (3.18)). Single notes on the left and chords on the right. Up to 5 harmonics are resolvable. Higher harmonics after the 5th one become indistinguishable especially in the case of chords where the number of simultaneous frequency components is higher. The main window illustrates our wavelet analysis tool developed within this thesis work.

Good time resolution is important in such tasks as beat or onset detection for music signal analysis as we will see in the next chapter. The next example serves to illustrate the time resolution properties of the Variable Resolution Transform we are developing. In this example we examine a signal with a series of delta-pulses (Dirac) as illustrated in Figure 3.16 which is a wavelet spectrogram of 5 delta-pulses (1 on the left, 2 in the middle and 2 on the right). As we can see from this figure, Delta-pulses on the picture are still distinguishable even if the distance between them is only 8 ms (right case). In the case of FFT one need 64-sample window size in order to obtain such time resolution.

Figure 3.16. Wavelet transform of a signal containing 5 delta-pulses. The distance between two pulses on the right is only 8 ms.

A quite straightforward listening experiment that we have carried out reveals that the human auditory system is capable to distinguish delta-pulses when a distance between them is around 10 ms. On the other hand, the human auditory system is also able to distinguish very close frequencies - 4Hz in average[1], and down to 0.1Hz (see the following figure).

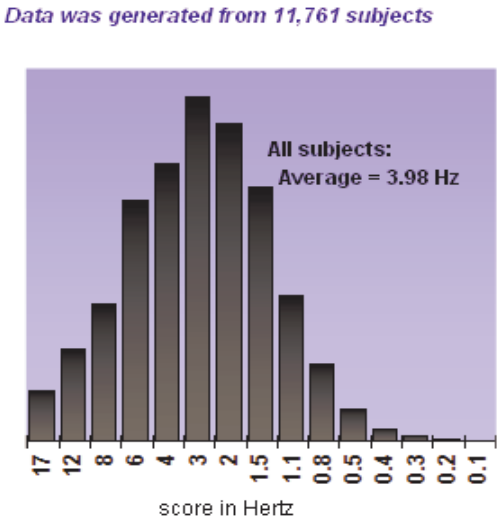


Figure 3.17. Human auditory system frequency resolution histogram for frequencies around 500Hz.

To maintain a time resolution around 5-10 ms, the FFT requires a window size around 64-128 samples. Figure 3.18 illustrates a Fast Fourier spectrogram with 64-sample window of the same piano recording which clearly contrasts to the spectrogram in Figure 3.15. Neither fundamental frequencies nor partials can be extracted in this case.

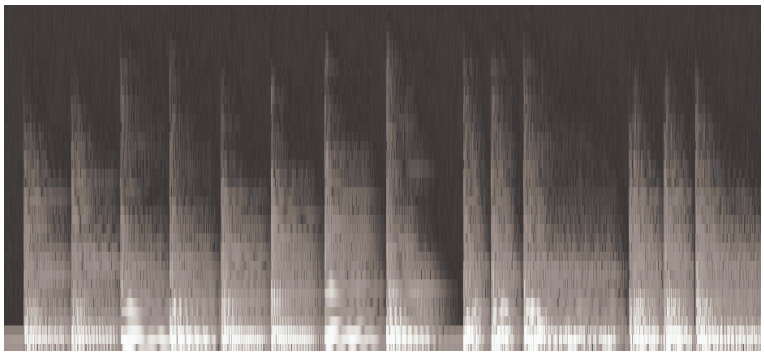


Figure 3.18. Fourier transform of the signal with notes played by a piano (the same signal with was used in previous wavelet experiment on Figure 3.15). Neither fundamental frequencies nor partials can be extracted.

[1] http://tonometric.com/adaptivepitch/

3.3.1.3 Varying the mother function

However, music analysis requires good frequency resolution as well. As we can see from the spectrogram in Figure 3.15, neither high-order partials nor close notes are resolvable, because the spectral localization of the used wavelet is too wide. Increasing the length parameter l in (3.18) or (3.25) of the Hann window would render our wavelet transform unusable in low-frequency area since the time resolution in low-frequency area would rise exponentially. Thus, we propose in this work to make dynamic parameter l **with** a possibility to adjust its behavior across the scale axis**.** For such a purpose we propose to use the following law for parameter l in (3.25) instead of applying scale $a(n)$ to parameter t in $H(t,l)$:

$$l(n) = L \cdot \left(1 - k_1 \frac{n}{N}\right) \cdot e^{-k_2 \frac{n}{N}} \tag{3.27}$$

where L is the initial window size, k_1 and k_2 – adjustable parameters

The transform (3.25) becomes:

$$W(n,b) = \frac{1}{\sqrt{\frac{f_S}{f_{\min} e^{n \cdot C}}}} \left| \sum_{-l/2}^{l/2} s[t+b] \cdot H\left[t, L \cdot \left(1 - k_1 \frac{n}{N}\right) \cdot e^{-k_2 \frac{n}{N}}\right] \cdot e^{-i \frac{t f_{\min} \cdot e^{n \cdot C}}{f_S}} \right| \tag{3.28}$$

The expression (3.27) allows the effective"wavelet" width to vary in different ways: from linear to completely exponential to follow the original transform definition. When $L = \frac{f_S}{f_{\min}}$, k_1=0 and k_2=$C \cdot N$, (3.28) is equivalent to (3.25).

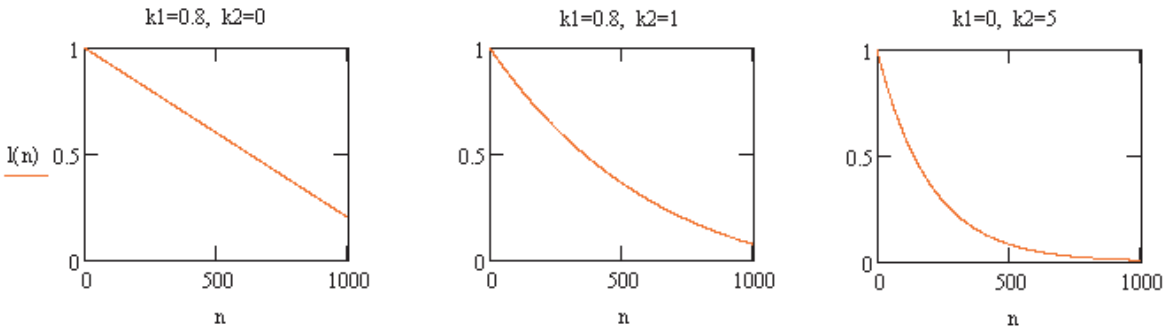


Figure 3.19. Various $l(n)$, depending on parameters. From linear (left) to exponential (right).

Doing so, we are now able to control the time resolution behavior of our transform. In fact, such transform is **not anymore a wavelet transform** since the mother-function changes across the scale axis. For this reason we call the resulted transform as ***variable resolution transform*** (**VRT**). It can be also referred as a custom filter bank.

As the effective mother-function width (number of wave periods) grows in high-frequency relatively to the original mother-function, the spectral line width becomes more narrow, and hence the transform allows to resolve harmonic components (partials) of the signal. An example of the spectrogram with new variable resolution transform is depicted in Figure 3.20.

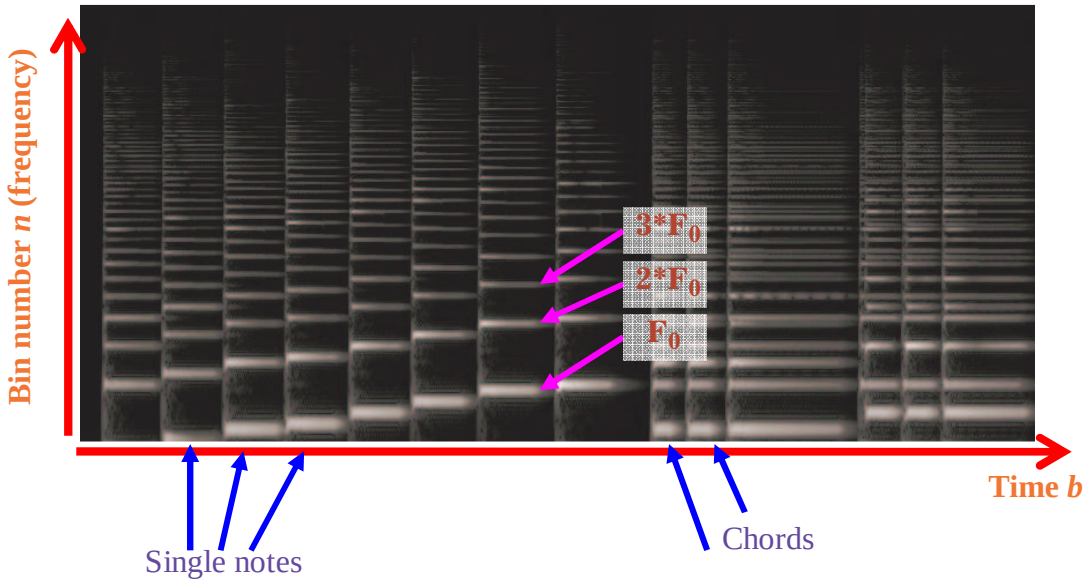


Figure 3.20. VRT spectrogram of the piano recording used in the previous experiment. Fundamental frequencies and partials are distinguishable (k1=0.8, k2=2.1).

3.3.2. Properties of the VR transform

Here we proceed to study the properties of our VR transform within the scope of the present work, i.e., with regard to music signals.

A music signal between 50 and 8000 Hz contains approximately 8 octaves. Each octave consists of 12 notes, leading to a total number of notes around 100. A filterbank with 100 filters would be enough to cover such octave range. In reality, frequencies of notes may differ from the theoretical note frequencies of equal-tempered tune because of recording and other conditions. Therefore for music signal analysis considered here, we are working with spectrogram size of 1024 bins – 10 times the amount necessary which covers the note scale by 10 bins per note. It is important to know the size of the spectrogram because all spectral graphs and their derivates are represented in terms of *spectral bin number n*, which has a logarithmic relation with frequency behind it. The other reason of increasing the number of bins comes from the necessity to distinguish high-order partials. However, there is a tradeoff to be found as further increase of the number of bins renders the computation much too heavy.

Timbre is a one of major properties of music signal along with melody and rhythm. Let's consider now a structure of partials of a harmonic signal (harmonic structure). In Figure 3.13 we have depicted an approximate view of such structure in logarithmic frequency scale. According to the definition of the function *f(n)* (3.26), the distance between partial *i* and partial *j* in terms of number of bins is **independent** of the absolute fundamental frequency value.

Indeed, according to (3.26) $n(f)=\frac{1}{C}\ln\frac{f}{f_{\min}}$ and taking into account $f_i=i*f_0$ and $f_j=j*f_0$ we obtain:

$$n(f_j)-n(f_i)=\frac{1}{C}(\ln(f_0\cdot j)-\ln f_{\min})-\frac{1}{C}(\ln(f_0\cdot i)-\ln f_{\min})=\frac{1}{C}(\ln(f_0\cdot j)-\ln(f_0\cdot i))=\frac{1}{C}\ln\frac{j}{i}$$

Figure 3.21 illustrates these decreasing distances between neighboring partials *i* and *i-1* when harmonic component number *i* increases, covering the first 20 harmonics. As we can see, the number of bins between the two first partials f_0 and f_1 are relatively high (128 bins). However, when the partial number of the harmonic increases, the number of bins decreases quickly but stays around 10 bins for high-order harmonics.

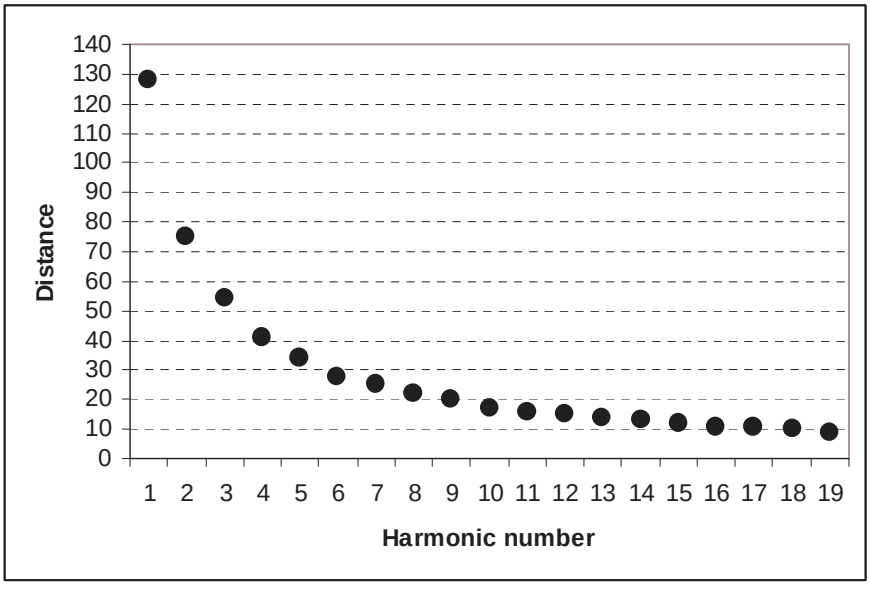


Figure 3.21. Dependency of the distance between partials form partial number.

An accurate harmonic analysis of music signal implies that frequency resolution in terms of spectrogram bin number, as expressed by the spectral dispersion (3.14), should be always below the distance between neighboring components under consideration. Figure 3.22 illustrates an example of spectral dispersion for a wavelet transform (e.g. Morlet or another one defined in (3.18). It is computed by sweeping a sine wave by all frequencies across the wavelet scale axis and calculating the dispersion.

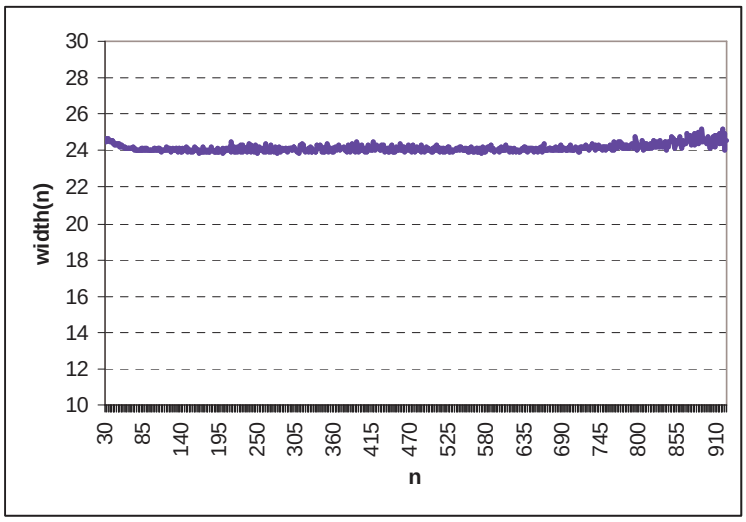


Figure 3.22. Spectral dispersion for a **wavelet** transform. It is taken in terms of wavelet number n on x axis.

This flat dispersion comes from the fact that the frequency axis is in a logarithmic scale. As we saw in Figure 3.15, such a constant frequency dispersion over the frequency axis leads the situation where only 5 partials of a harmonic signal can be distinguished without peak intersection by such wavelet transform which is obviously not enough for majority of applications (higher order harmonic might be still resolvable while the step of the frequency resolution is still lower than the dispersion). Further improving frequency localization of such wavelet transform (lowering the dispersion) decreases its time resolution dramatically, causing the maximum time localization to grow exponentially. In the Figure 3.23 we can observe the result of such manipulation with Morlet wavelet which was used to transform a delta-pulse.

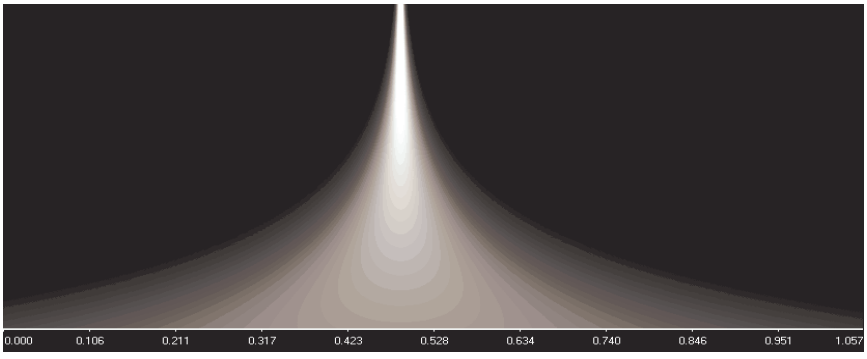


Figure 3.23. Morlet wavelet transform with $\alpha^2=200$ of a delta-impulse. The maximum time localization **exceeds one second bounds**.

Despite an unacceptable time localization at low frequencies, classical wavelet transform in logarithmic scale remains an attractive tool in analyzing rapidly changing signals. Together with impressive time resolution in high-frequency area it has constant frequency localization because of the frequency axis sampling we have chosen is logarithmic.

Let's return to our variable resolution transform. As it was described in §3.3.1.3, we can vary the effective window size using different laws independently from the main scale of the mother-function. With $k_2=2.1$ and $k_1=0.8$ we obtain the following spectral dispersion distribution as illustrated in Figure 3.24.

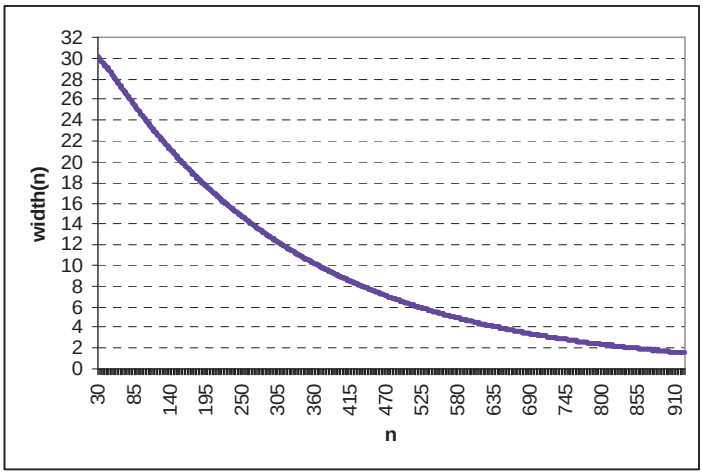


Figure 3.24. Spectral dispersion graph of VR transform with $k_2=2.1$ *and* $k1=0.8$ for bins numbers from 30 to 900.

Having the total width of 20-partial harmonic structure to be a constant around 600 points in terms of number of bins ($n(f_{20})$ - $n(f_0)$), we can establish that the frequency resolution of the obtained transform is large enough to resolve high-order partials we are interested in at all positions of the VRT spectrogram, especially for low octave notes. It means that a 20-partial harmonic structure starting from the beginning of the spectrogram will always lie *above* the dispersion curve. If we consider now the time resolution of the transform, we must recall Figure 3.19, where various dependencies on the effective width of filter were given. If we define the maximum effective window size to be 180ms (recall our musical signal properties) we obtain the following time resolution grid as illustrated in Figure 3.25.

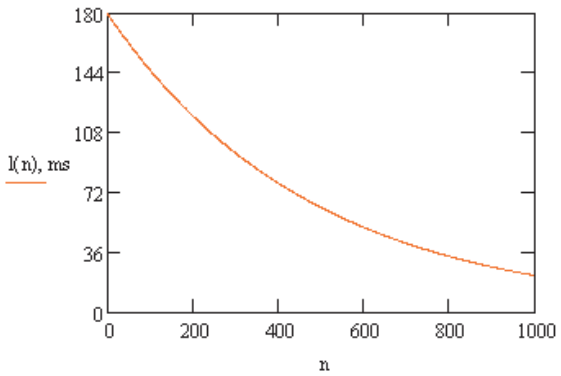


Figure 3.25. Time resolution dependency of VR transform with $k_2=0.8$, $k_2=2.1$.

Final resolution plots of our variable resolution transform is given in Figure 3.26 and Figure 3.27 in terms of frequency in Hz and time in milliseconds (given for frequencies between 50 and 5000Hz where the most of notes frequencies are found).

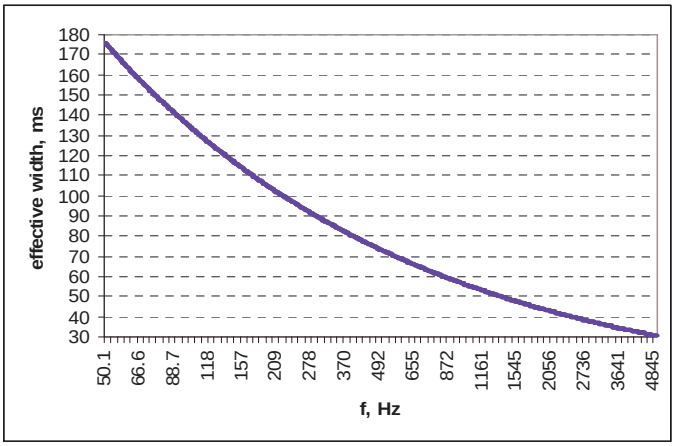


Figure 3.26. Time resolution grid of our VR transform ms/Hz, estimated form the transform of a test signal.

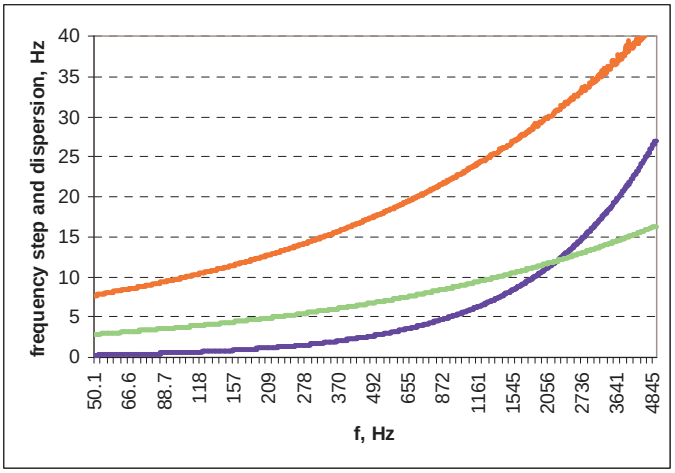


Figure 3.27. Frequency resolution grid of our VR transform. The red curve stands for dispersion graph according to frequency; blue curve signifies a frequency step (Hz/bin) of the transform. Green curve on the picture stands for equivalent FFT frequency resolution (Hz/sample) as if it had a window size equal to that one from VRT (from Figure 3.26).

To understand the previous figures better, let's consider an example. The blue curve in Figure 3.27 corresponds to a spectral step size – the difference of equivalent frequencies of n^{th} and $n\text{-}1^{th}$ filter. Taking as example 500Hz, we obtain a frequency step size around 3Hz. The same equivalent characteristic can be calculated for an FFT. What does it mean? If we take 500Hz and look at Figure 5.5, we get an effective window size of VR transform to be approximately 70ms. We then imagine an FFT with window size of 70ms (with 16kHz sampling rate it is going to be close to 1024 samples). The size of frequency step in 1024-sampled FFT is 8Hz, as can be observed by the green curve in Figure 3.27. It means that for an FFT with 1024-sample

window, frequencies of 500Hz and 508Hz cannot be resolved unlike the case of the VR transform, where the size of frequency step at 500Hz was equal to 3Hz. However, the VR transform has considerable frequency dispersion – around 18Hz at 500Hz point. Therefore, on the spectrogram, two frequencies of 500 and 508Hz will have an intersection as illustrated in Figure 3.28.

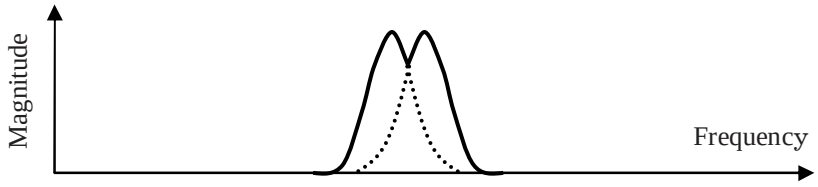


Figure 3.28. Intersection of two close frequencies on VR spectrogram.

The main point of the VRT is nonlinearity of time-frequency tiling which makes it possible to resolve rapidly changing high-frequency events and at the same time to have a constant number of spectral steps per semitone in all octaves in order to be well adapted to music analysis.

Now let's consider some wavelet spectrogram examples of real music excerpts.

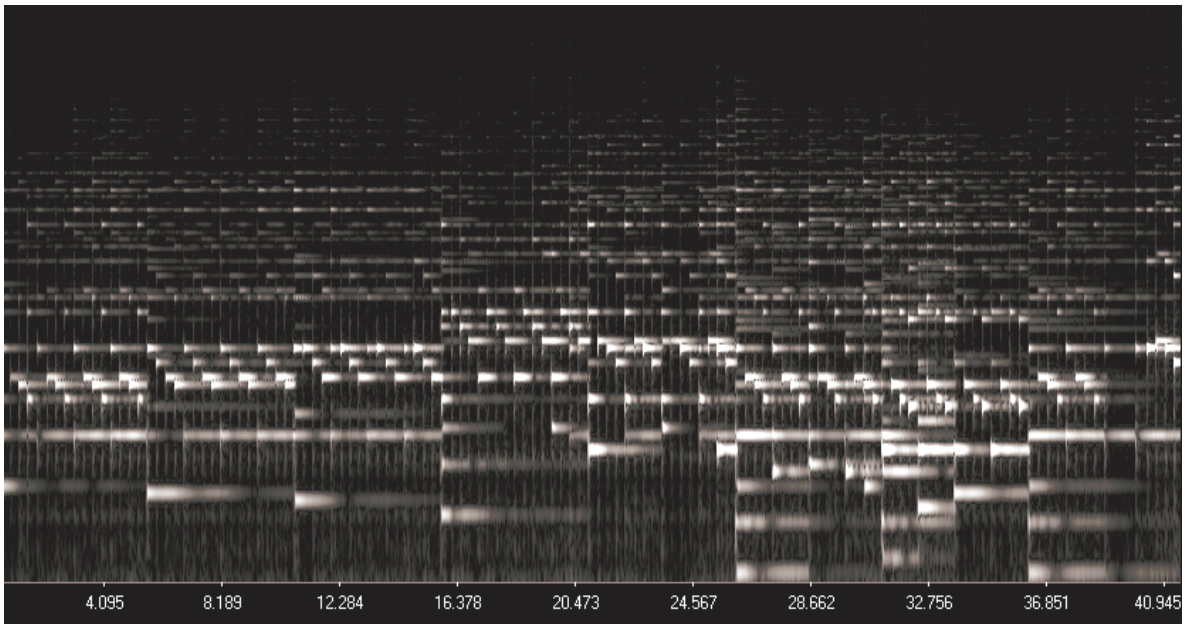


Figure 3.29. VRT spectrogram of an excerpt from *Era – Flowers of the Sea*, piano.

Figure 3.29 is a VRT spectrogram for an excerpt from "Era – Flowers of the Sea". In this excerpt only one instrument is present – a piano. "ars, measures and note events are distinguishable. As we can see, positions of notes on the frequency axis looks like (and actually is!) linear. This example contains about 800 frequency planes (calculation points), with 50ms step. Computing time on a *PentiumM-1.4GHz* was around 3 seconds.

The same excerpt was studied in a known sound edit software. The result is given in Figure 3.30. In this example the window size of the FFT was 2048 points. Calculation time is also around 3 seconds.

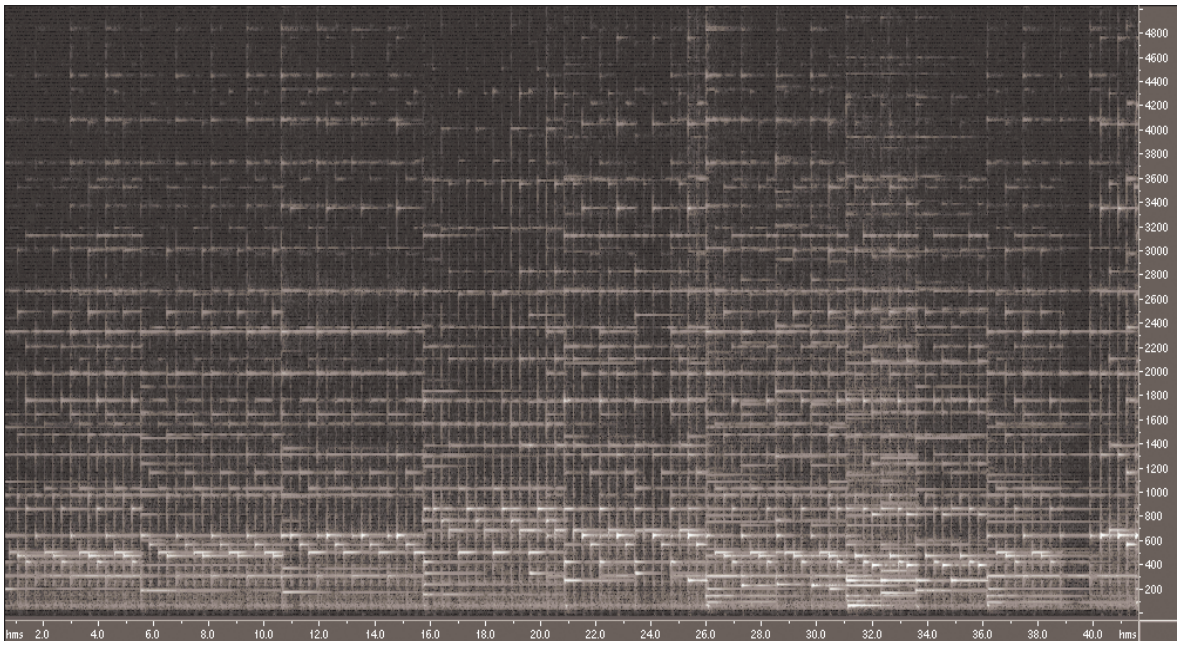


Figure 3.30. 2048-point FFT spectrogram of the excerpt from *Era – Flowers of the Sea*.

As we can see from the figure, the main melody notes are crowded at the bottom of the spectrogram, but the "forest" of partials is very distinct.

Another interesting fact is that frequencies of partials are not necessarily linear as n^*f_0. (See e.g. [KLAP 99]). Regarding to this aspect, our VR transform might be more advantageous as the logarithmic frequency scale will most probably "eat" the non-linearity of partials' frequencies, and hence, non-linear partials will be situated not far away from their linear theoretical positions in terms of spectral bin number.

3.3.3. Computation

Regardless of the existence of relatively fast algorithms of continuous wavelet transform computation as for instance octave algorithm, we implement a direct scheme of computation as the most precise one. The choice of direct computation is imposed also by an absence of self-similarities in wavelet-like functions introduced in this work, and therefore, an absence of fast algorithms of computation.

However, several efforts have been made in order to accelerate the computation procedure. First of all, we store previously crafted values of wavelet functions for all frequencies in memory arrays. Avoiding the sine and cosine calculation every time during wavelet analysis computation significantly increases the overall speed. Finally the procedure of computation is done via vector arithmetic of modern processors where a SIMD instruction

can operate many data at the same time (*SIMD – Single Instruction Many Data)* (see [Intel 03]). Each integer 64-bit MMx register may contain 8 bytes (char), 4 words (short), 2 double words (int) or 1 quad word (int64 or longlong). SIMD instructions make arithmetic operations on all operands stored in MMx register simultaneously. More recent instruction sets manipulate 128-bit registers with packet integers or floating points. Some examples of packed arithmetic instructions are given on figure.

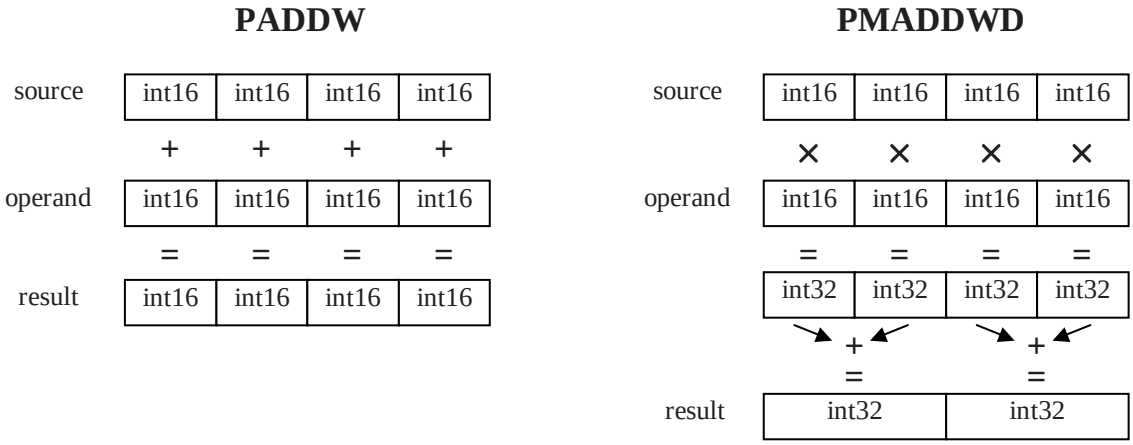


Figure 3.31. Example of integer SIMD instruction from Intel's MMX instruction set.

Using Intel MMX technology it looks like

```
movq MM2,qword ptr [esi]          // take 4 samples of the signal
movq MM3,MM2
pmaddwd MM2, qword ptr [edi]      // multiply and add pairs, real part
paddd MM0,MM2                     // accumulate
pmaddwd MM3, qword ptr [ebx]      // compute complex part
paddd MM1,MM3                     // accumulate
```

The example of the computation code provides a gain of performance by about 5-8 times in comparison to a standard C++ implementation.

3.3.4. Discussion

Our Variable Resolution Transform is derived from the classic definition of Continuous Wavelet Transform given in 3.2.2. In our previous work, we referred to our VRT as "Wavelet-Like" or "Pseudo-Wavelet" transform [PARA 06; PARA 07]. Actually, our VRT is not a CWT even though they have many similarities. The main difference between VRT and CWT consists in the frequency axis sampling, as well as in the mother wavelet function which is changing its form across the scale (or frequency) axis in the case of VRT in order to have enough resolution details for high order frequency partials. This last property is not a wavelet transform, because in the true wavelet transform the mother function is only scaled and shifted making a

discrete tiling of the time-frequency space in the case of DWT or infinite coverage in the case of CWT. Our VRT can be also referred to as a specially crafted filter bank. Major differences between our VRT and a wavelet transform are:

- no 100% space tiling
- no 100% signal reconstruction (depending on parameters)
- mother function changes

Major similarities between our VRT and a wavelet transform are the following:

- They are based on specially sampled version of CWT
- with certain parameters they can provide 100% signal reconstruction
- low time resolution and high frequency resolution in low frequency area and high time with low frequency resolution in high frequency area

Comparison of our VRT to other techniques can be summarized by the following table.

Table 3.1. Comparison of VRT to other approaches according to their properties

Technique	Frequency sampling	Frequency bin dispersion	Computational complexity	Note resolution	Harmonic resolution	Temporal resolution
FFT (small window)	Linear	High	Low	Very low	Low	High
FFT (large window)	Linear	High	Low	Medium	High	Low
FFB	Linear	Low	Low	Same as FFT	Same as FFT	Same as FFT
CQT	Log	High	High	High	Low	High
CWT (general)	Variable	Variable	High	High	Low	High
BQT	Linear in octaves	High	Medium	High	Medium	High
BQFFB	Linear in octaves	Low	Medium	High	Medium	High
VRT	Log	High	High	High	High	Medium

In another comparison table we provide to summarize various approaches according to their applicability for different problems.

Table 3.2. Comparison of VRT to other approaches according their applicability

Technique	Beat detection	Note detection	Spectral features
FFT (small window)	++	- -	+ -
FFT (large window)	- -	+ -	+ +
FFB	Same as FFT	+ -	Same as FFT
CQT	+ + / - -	+ - / + +	+ +
CWT (general)	+ + / - -	+ - / + +	+ +
BQT	+ +	+ -	+ +
BQFFB	+ +	+ ±	+ +
VRT	+ +	+ +	+ +

Generally, "QFF" approach could be used for our music similarity retrieval application, but due to its specific frequency sampling, it complicates spectral modeling (distances between harmonic are not constant).

3.4. Application to spectral similarity

The following spectral similarity measure was proposed in one of our previous works [PAR 05] – spectral ratio coefficients matrix. Coefficients are computed in the following way. First, the signal spectrum is obtained from the VR Transform. Then the filterbank of MEL frequencies is applied to absolute values of the spectrum (it is directly mapped to the VRT spectrogram). The spectral descriptor is then represented in a form of a matrix $N \times N$, which is constructed as follows:

$$M_{i,j} = \begin{cases} \log\left(\frac{B_i}{B_j}\right), \text{when } B_i \neq 0 \text{ and } B_j \neq 0 \\ -CONST, \text{when } B_i = 0 \\ +CONST, \text{when } B_j = 0 \\ 0, \text{when } B_i = B_j = 0 \end{cases} \quad \textbf{(3.29)}$$

where N is the total number of spectral bands, B_i is the energy in the i^{th} band.

The matrix M is used as an initial spectral feature. A sequence of them can then be modeled by a set of Gaussian distributions (each element

independently). The distance between two sound excerpts is calculated as a sum of KL distances for distributions of each $M_{i,j}$.

The feature we have presented does not have the disadvantage of volume dependency because only energy ratios are taken so it is more robust in acoustic similarity and it showed better results in audio segmentation issue of video segmentation context [PAR 05].

A second spectral feature we have proposed is based on cutting the VRT spectrum into time-frequency rectangles (with the following dimensions: 25ms × 1/8 of total frequency range) and computing histograms of values in each obtained rectangular area. The main principle is shown on Figure 3.32.

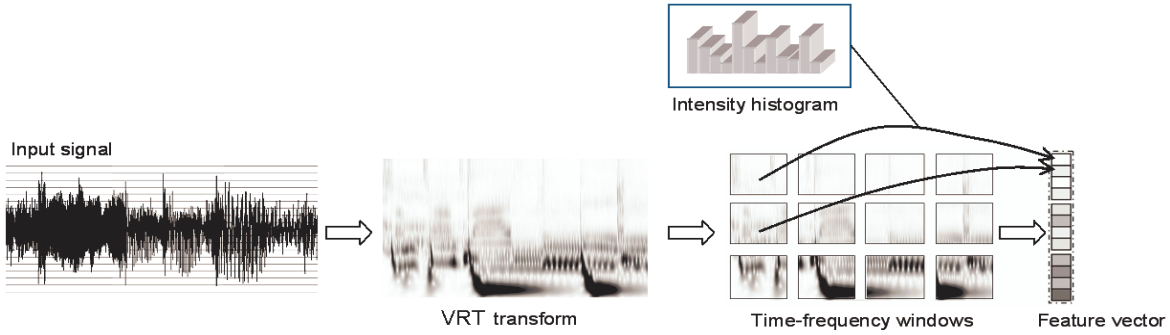


Figure 3.32. Spectral feature extraction procedure. Wavelet or VRT spectrum is divided into time-frequency tiles. Histograms of values are computed in each tile. Histograms are then serialized in form feature vectors.

” ins of obtained histograms are then serialized and concatenated into one feature vector. The usage of this spectral similarity characteristic was found in genre classification problem [KOTO 07] where histogram is summarized by the mean values.

3.5. Conclusion

In this chapter we have introduced our Variable Resolution Transform as a novel signal processing technique specifically designed for music signal analysis. A music signal is characterized by four major properties: melody, harmony, rhythm and timbre. The classic Fast Fourier transform, a de-facto standard in music signal analysis in the current literature, has its main drawback of having a uniform time-frequency scale which makes it impossible to perform efficient spectrum analysis together with good time resolution. The wavelet transform overcomes this limit by varying the scale of mother-wavelet function and, hence, the effective window size. This kind of transform keeps frequency details in low-frequency area of the spectrum as well as time localization information about quickly changing high-frequency components. However, the dramatic decrease of frequency resolution of the basic wavelet transform in high-frequency area leads to

confusion in high order harmonic components where a sufficient resolution is necessary for the analysis of harmonic properties of a music signal. We have thus introduced our Variable Resolution Transform in varying mother-function. The law of variation is controlled by two parameters, linearity and "exponentiality", which can be carefully chosen in order to adjust the frequency-time resolution grid of the VRT. Hence, our VRT takes advantage of the classic continuous wavelet transform and the windowed or short-time Fourier transform by diminishing the frequency dispersion in high-frequency area and therefore keeping the frequency resolution in high-frequency area good enough to resolve harmonic components of musical instruments. As compared to time-frequency characteristic of the Fourier and the wavelet transform, the major advantage of the VRT is the dynamic variability of time-frequency tiling in contrast to constant tiling in the case of the STFT. This ability to change the tiling allows a better description of a music signal while keeping a fixed number of frequency samples per octave and a good time resolution where it is necessary, i.e. in high-frequency area, where all rapid changes usually take place.

As we will see in the next chapters, the families of music analysis algorithms introduced in this thesis are exclusively based on the aforementioned VR transform and their influence on the performance of MIR algorithms are also discussed.

Chapter IV

Rhythm-related similarity features

4. Rhythm-related similarity features

Rhythm is one of four major music properties along with melody, harmony and timbre. Automatic extraction of rhythmic pulse from musical excerpts has been a topic of active research in recent years. Also called beat-tracking and foot-tapping, the goal is to construct a computational algorithm capable of producing behaviors that correspond to the experience of beat or pulse in a human listener. Rhythm as a musical concept is intuitive to understand, but somewhat difficult to define.

Let's define beats as a sequence of pulses and accents over a musical composition issued either by percussion instruments or by note onsets. In general, beats are forming musical tempo which is measured in *beats per minute (BPM)*. While characterizing the rhythmic feature of a music excerpt, the tempo in " PM may not be enough in music classification and similarity search. For example a classical composition and at the same time a rock composition may have 140 beats per minute while these two compositions are from very different genres. On the other hnad, the perception of the tempo (foot-tapping) is ambiguous [MOEL 04] it may usually have an error of 2 or other multiples.

4.1. Related work

" eat detection-related or tempo estimation-related algorithms have found many implementations. They can be divided into two groups treating different problems: beat/onset detection algorithms and periodicity search algorithms.

The first group of algorithms is designed to detect rhythmic activity such as percussion instrument beats, note onsets, accents, etc. Sounds of percussion instruments are generally non-voiced, strong attack, and noisy (although they can contain harmonic components), leaving considerable noisy traces in spectrum. Note onsets are less remarkable events and, according to the instrument, may be rather smooth. Figure 4.1 depicts an energy envelope example of a note onset.

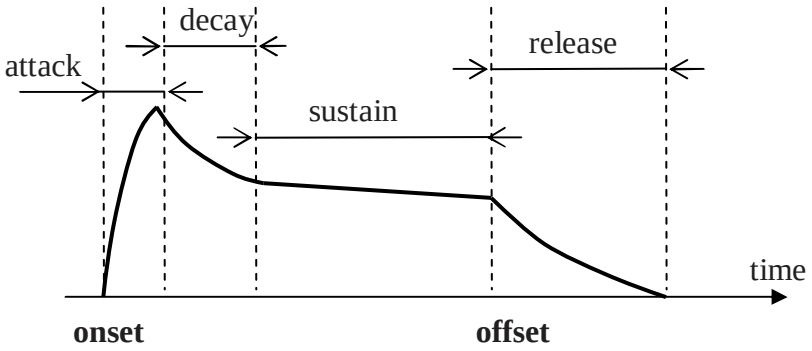


Figure 4.1. Typical energy envelop of a note (piano-like instrument)

The easiest way to extract beats events is to use energy envelope directly from the raw signal with automatically adjusted threshold (Figure 4.2). This method is very fast and can be used as the first approximation in beat detection or for example visualizing applications. Unfortunately this method is very sensitive to the musical content – there must be really strong beats represented by percussion instruments with light music on background. Moreover it is not suitable for onset detection, which has an equal importance in rhythmical analysis of a musical composition.

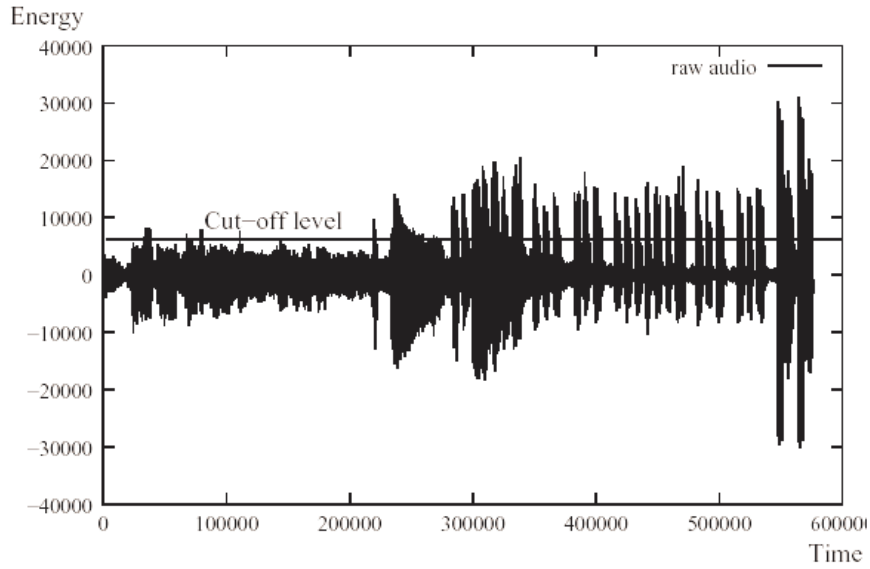


Figure 4.2. Beat detection from the raw WAV. The signal's waveform is cut at certain energy level. The obtained peaks are considered as detected beat or high-energy events.

Other deterministic methods work with time-frequency representation of the signal such as STFT spectrogram (e.g. [COLL 05]) and filter banks (e.g. [KLAP 99a]). An approach of beat detection using image treatment techniques applied to a spectrogram is presented in [NAVA 04]. An interesting work is presented in [ALON 03]. In their work the authors decomposes the signal into 12 subbands, each subband is processed separately by representing it in bi-component form containing sinusoidal signal plus noise (using Exponentially Damped Sinusoidal, described in [" ADE 02]). Attacks extraction and periodicity detection is done separately in these subbands.

Another kind of methods are statistical methods are based on assumptions that the music signal can be described by some probabilistic model trying to guess locations in the signal where a potential accentuation event may occur. Obviously, the detection performance of these methods is strongly dependent on musical content and on training data.

The approach we present in our work is a deterministic one. It analyses the musical signal directly using VR Transform and processes the resulting

spectrogram as a grayscale image. No assumption on music nature or theoretical prediction of beat positions is made.

The second group of algorithms concerns higher level processing in order to find periodicities. These algorithms can be divided into some sort of general classes:

- Autocorrelation-based. This class of algorithms is based on computing of one or multiple autocorrelation functions from the signal, its spectrum or on sequence of feature vectors in order to find the most salient period and mainly used for tempo estimation. Such works can be mentioned as [" ROW 93; ECK 05; GOUY 03; PAUL 02]. In [PEET 05] authors proposed a method for the automatic estimation of the tempo based on the reassigned spectral energy flux, a combination of DFT and Frequency Mapped autocorrelation function and a Viterbi decoding algorithm.

- " ased on a mathematical model of resonator set (e.g. [KLAP 06; SCHE 97a]). The idea behind these methods is to find the most probable periodicity using multiple resonators or oscillators. Each oscillator has its own frequency and produces a probability output (resonance). Non-linear resonators are also known, e.g. [LARG 94].

- Histogram building. One of rhythmical information representation is the beat histogram as used for example by *George Tzanetakis* [TZAN 02]. In his work the beat histogram accumulates peaks of autocorrelation function over the whole sound. Each bin of the beat histogram corresponds to a peak lag (i.e. the beat period in " PM – beat per minute). This kind of representation is quite useful for example for genre classification as it carries information about the number of beats with different periods or tempo determination by finding the maximal value. There are other works known in the literature which are based on histograming. The work [AREN 01] proposes a simple time-domain method in order to extract the tempo in " PM.

Other methods of periodicity detection are known. Authors of the work [ALON 06] use frequency domain methods, such as spectral sum and spectral product for periodicity analysis.

Autocorrelation and resonator-based approaches are not suitable for music similarity analysis since they operate with periodicity detection and issue the main tempo of the play. We are interested in representing of rhythmic structure, thus we need a mechanism of periodic and non-periodic beat and also onset detection together with rhythmic fingerprint computation.

Non-periodic beat tracking and rhythmic-structure representation is described in the work [TZAN 02a]. In this work beat periods are organized in form of beat histogram allowing their rhythmic comparison. A disadvantage of the proposed method is the absence of beat strength information included into rhythmic-structure representation.

Rhythm-related similarity characteristics are among a few numbers of semantic features which are quite well explored in literature. For instance, in the work of *Eric Scheirer* [SCHE 00], the question of beat related characteristics holds a dominant part. There are many other works on rhythm analysis in the context of automatic music retrieval. The work in [ALGH 99] proposes to achieve beat tracking and their hierarchical organization in order to obtain periodicity patterns. The beat tracking is done in low frequency area, therefore, not all beat events are theoretically detected since instruments like hi-hats or cymbals do not have low-frequency components. Another beat tracking system is presented in [GOTO 01] which describes a real-time system able to detect different rhythm-forming events such as beats, onsets or chord changes and to organize them hierarchically. Sound signal in this work does not require drum instruments in it.

More recent works discuss questions of rhythmic similarity. The work in [FOOT 02] is based on "beat spectrum" extraction and comparison. The goal of the work is automatic retrieval of musical pieces with similar rhythm. Some theoretical evaluation is also provided on a very small corpus. A theoretical aspect of rhythmic similarity is given in [HOFM 02].

In our work we focus on construction of rhythmic similarity feature which consists of two parts:

a) beat detection

b) fingerprint extraction

Additionally, the fingerprint we obtain allows doing tempo estimation which is then evaluated and compared with other works.

4.2. Our VRT based approach for beat curve extraction

In our work, the beat detection relies on our previously defined VRT giving a time-frequency (spectral) representation of the signal.

4.2.1. An intuitive approach

Consider a music excerpt as represented by our VRT. Figure 4.3 is a spectrogram of a musical segment obtained using the VR transform as described in paragraph 3.3. The musical segment comes from metal music with noisy instruments, distortion guitars etc. *a priori*, contrast between beats and the rest of the music is rather low.

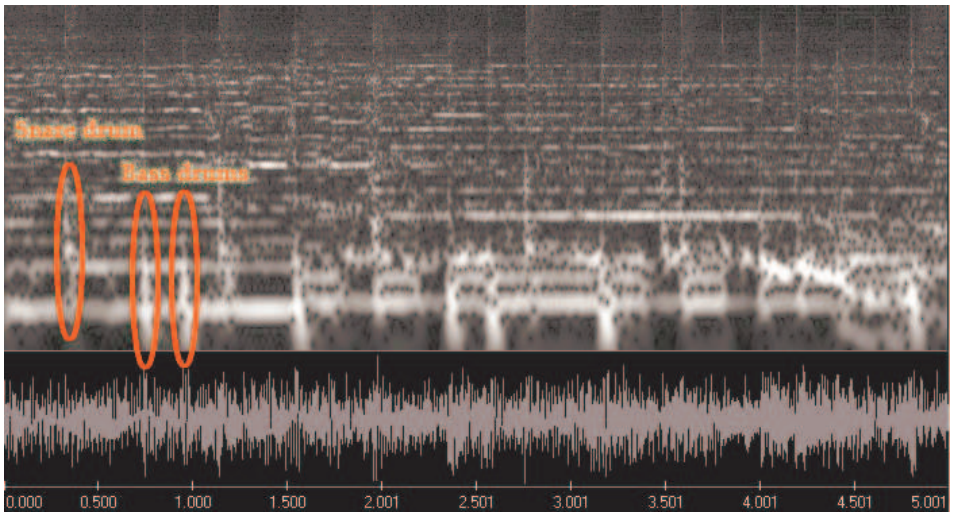


Figure 4.3. VR transform representation of a musical excerpt (of metal genre).

However, as we can discover from Figure 4.3, the VRT based spectrogram has remarkable elements in a form of vertical spots. These are drum events (bass drums and snare drums). In high frequency area they stand out against the background in a form of thin vertical lines.

Since the information about beats and onsets is concentrated on vertical constituent, it is possible to make use of image processing technique, such as gradient filters, to mark out all fragments of this "spectral image" more or less interesting in the meaning of beats and onsets. We thus propose to apply a Sobel operator in horizontal plane [SO" E 90], with its mask depicted on the next image.

-1	0	1
-5	0	5
-1	0	1

Figure 4.4. Sobel gradient operator in horizontal plane.

The central horizontal line in the mask is directly responsible for emphasizing vertical lines and may be adjusted in order to give more importance to horizontal gradient. The values -5 0 5 are found during experiments to be more appropriate for further beat detection.

The result of such treatment is shown in Figure 4.5.

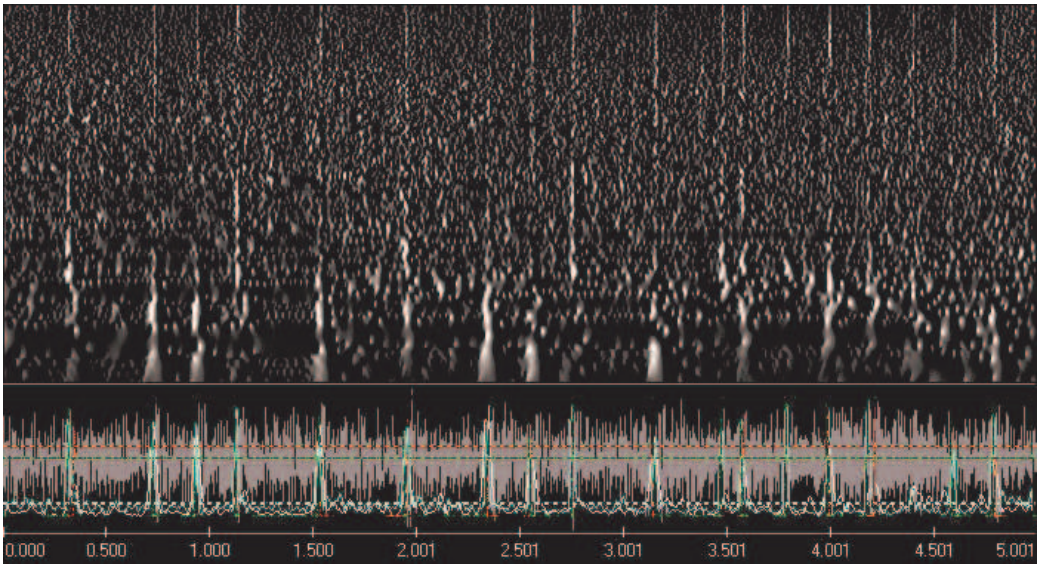


Figure 4.5. Sobel-filtered VRT spectrogram for a muscal excerpt with detected beat strength curve at the bottom.

After image enhancement all events we are interested in are now much clearer. Subsequently, the enhanced (Sobel-filtered) spectrogram $W^*(b, a)$ is treated by calculating a beat curve in the following way. A small window together with a large one is moved across the enhanced spectrogram. The value of the beat curve in each time moment is the number of points in the small window with values higher than a threshold which is obtained from the average value of points in the large window. Numerous beat curves may be computed separately by dividing the spectrum into bands. For the general question of beat detection only one beat curve is used.

4.2.2. Procedure of beat curve extraction

Figure 4.7 and Figure 4.6 summarize our procedure of beat curve extraction.

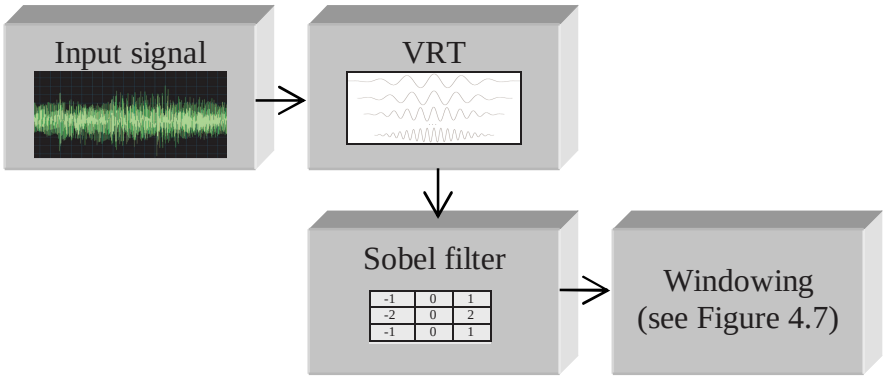


Figure 4.6. Beat curve extraction procedure diagram.

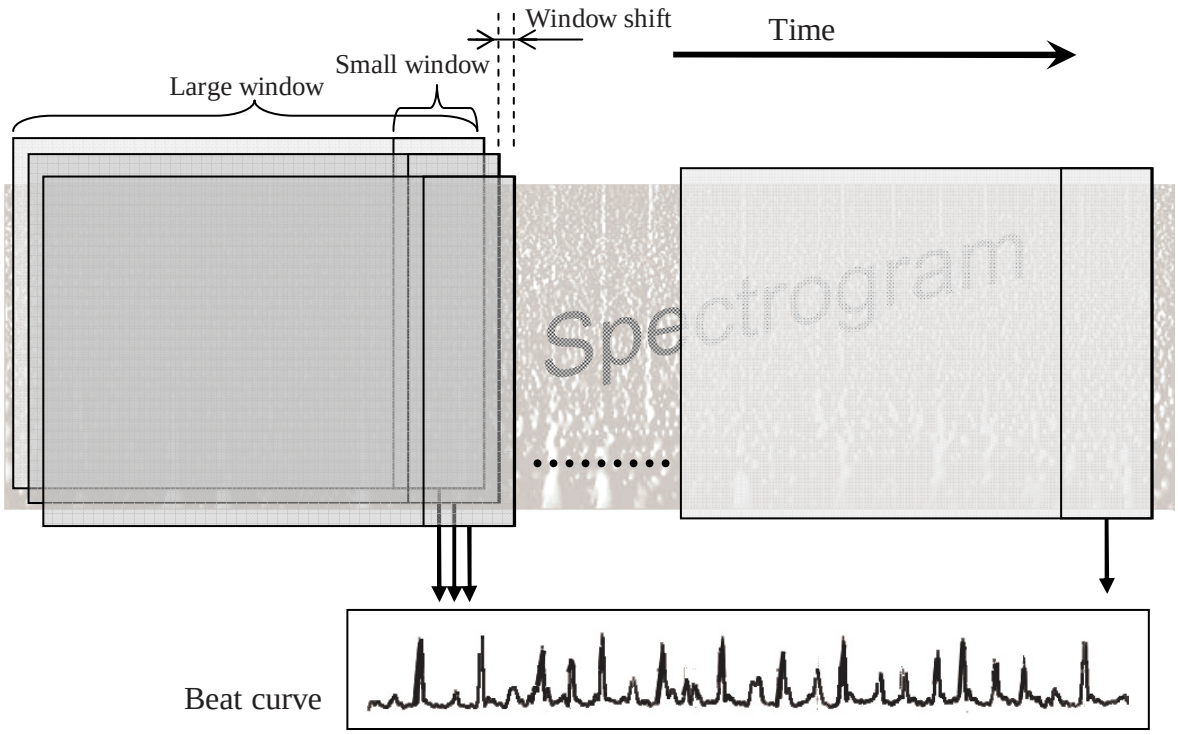


Figure 4.7. Beat detection windowing procedure. Beat probability or beat strength curve (at the bottom) is obtained by moving a sliding window and calculating number of spectrum points whose values are higher than the average value for bigger one.

Sizes of windows as mentioned in our procedure are chosen in the following way. The small window should well cover a snare drum beat or tom. Its size was selected to be 50-100ms (depending on analysis step size). The length of the large window was chosen to be 20 times the size of the small window, i.e. 1-2 seconds. As the beat curve extraction procedure is aimed to be volume-independent, we thus assume that a music excerpt has a relatively constant volume during each interval of 1-2 seconds which appears quite reasonable. Moreover, these two windows are shifted every 10 to 30 ms, producing thus beat curve values accordingly.

4.2.3. Discussion: VRT versus FFT based techniques

Instead of VRT based spectrogram, we may also apply our beat curve extraction technique on a Sobel-filtered image from the FFT transform. Figure 4.8 illustrates the result obtained from a FFT spectrogram that can be compared with the one processed by our VRT on the same musical excerpt (Figure 4.9).

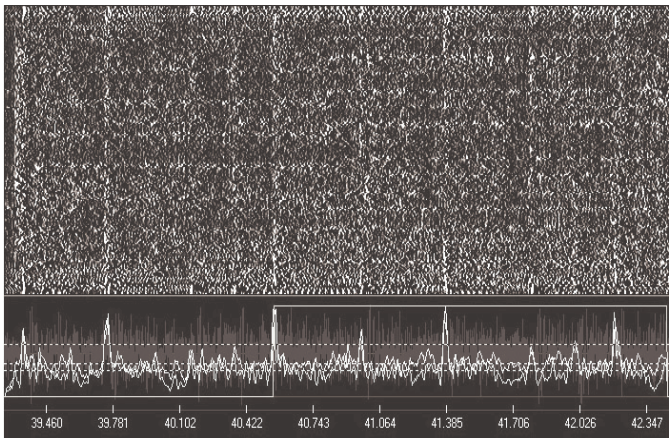


Figure 4.8. FFT spectral image (excerpt from "Nightwish – Come Cover Me")

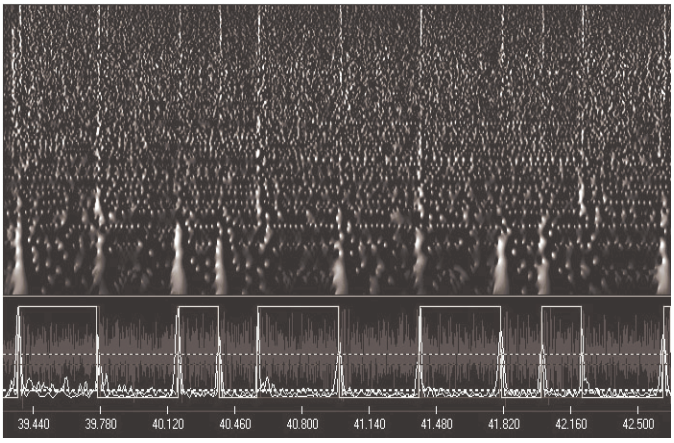


Figure 4.9. VRT spectral image (excerpt from "Nightwish – Come Cover Me")

As we can see, VRT based beat curve extraction shows 100% detection of percussion instrument beats in the test excerpt. This example suggests that VRT tend to be better suited than FFT for beat detection.

4.3. Rhythmic fingerprint

4.3.1. 2D beat histogram

In our work, we propose to improve beat histogram representation for a better characterization of rhythmic property of a music signal. Indeed, classical one-dimensional beat histogram (Figure 4.10) only provides some knowledge about the different beat periods while the distribution of beats in the meaning of their strength is not clear. At the same time beat detection algorithm and its parameters affect the form of the histogram.

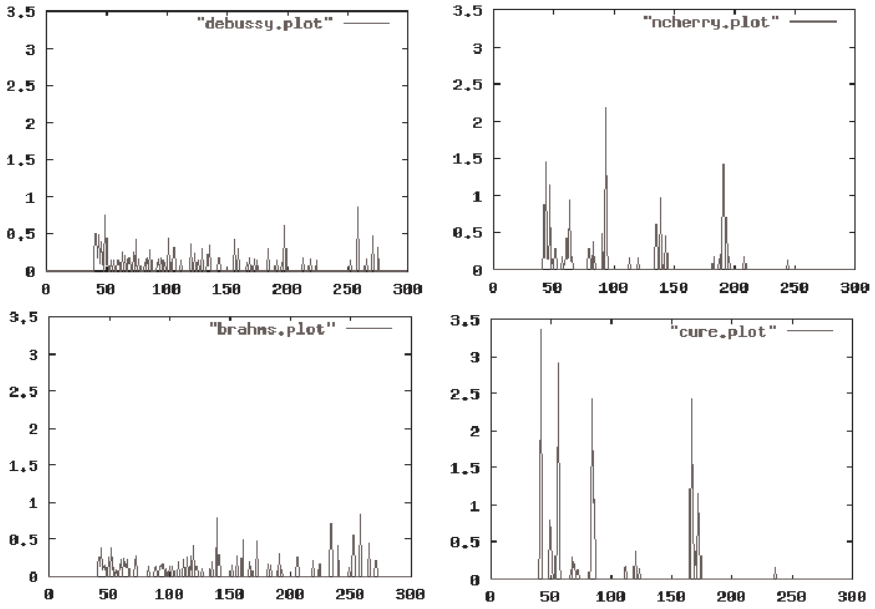


Figure 4.10. Classical one-dimensional beat histogram from the work of *G. Tzanetakis.*

A more accurate rhythmic property representation of a music signal should thus bring some knowledge about the strength of periodic beats into the histogram and avoid the dependency on the beat detection algorithm parameters. In this work, we propose a 2D form histogram which can be built with a beats period on the X axis and with amplitude (strength) of a beat on the Y axis (Figure 4.11). The information about beat strength in our histogram is included since the histogram is computed upon the threshold varying in Y axis. It is thus possible to avoid the disadvantage of recording conditions dependency (e.g. volume) and peak detection method. The range of threshold variation is taken from 1 to the found maximum-1. Thus, the beat strength is taken relatively and the volume or recording level dependency is avoided

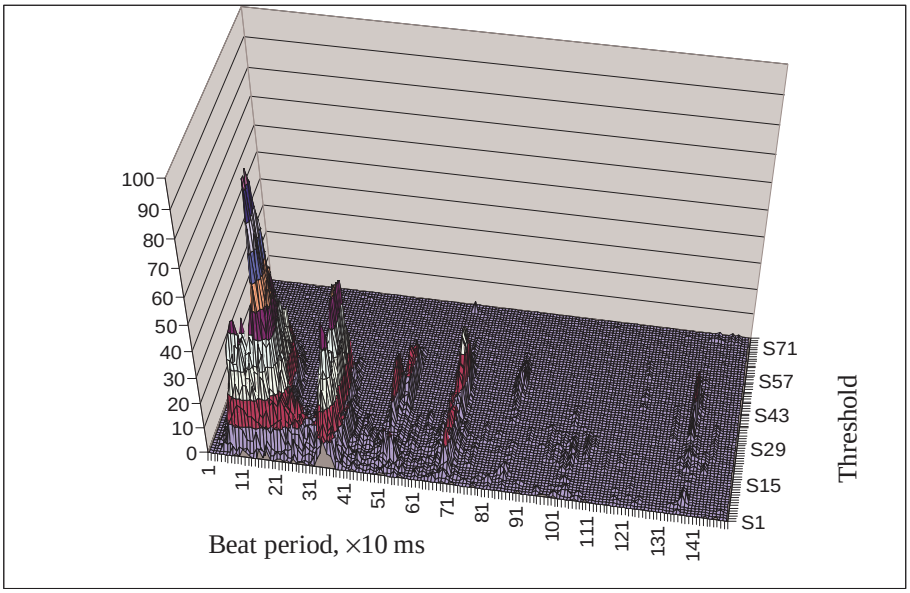


Figure 4.11. Example of two-dimensional beat histogram. Here the x axis represents beat period, y axis stands for threshold value used in peak detection on the beat curve.

As we can see, the histogram in Figure 4.11 contains several peaks. One of these peaks can be considered as main tempo peak, and the strongest peak is likely to represent *tatum* – the pulse of the lowest metrical level.

Such histogram characterizes rhythmic property of a music signal. It can thus be used as feature vector in music analysis applications such as genre classification or music matching. Figure 4.12 illustrates two 2D beat histograms for a dance composition (left) and a classical composition (right).

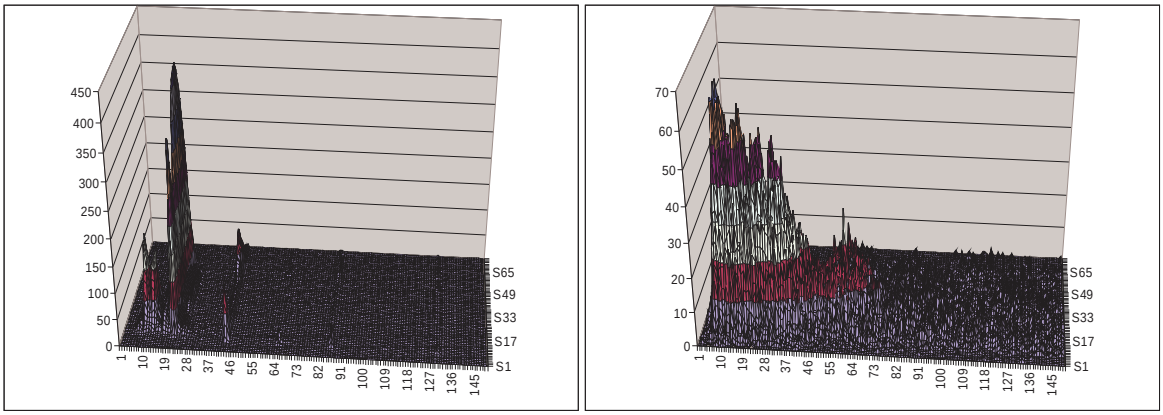


Figure 4.12. Two-dimensional beat histograms for dance (left) and classical (right) musical pieces.

As we can see in Figure 4.12, the left histogram has sharp peaks, indicating that the rhythmic structure of the play contains strong rhythmic beats. The right histogram, on the contrary, is an evidence of absence of strong periodic events as expected since classical composition does not contain either percussion instruments or strong attack instruments. The difference in 2D beat histogram shapes expresses different rhythmic properties of music signals and can then be used for music analysis applications such as music genre classification.

The following experiment tends to prove the independence of our 2D beat histogram from recording conditions. A musical composition is filtered with treble and bass cut filters (telephone filter 300-3000Hz). As we can see in Figure 4.13, the resulting histograms of beats still have the same shape and peaks. Moreover, the absolute difference between these two 2D histograms is under 10%.

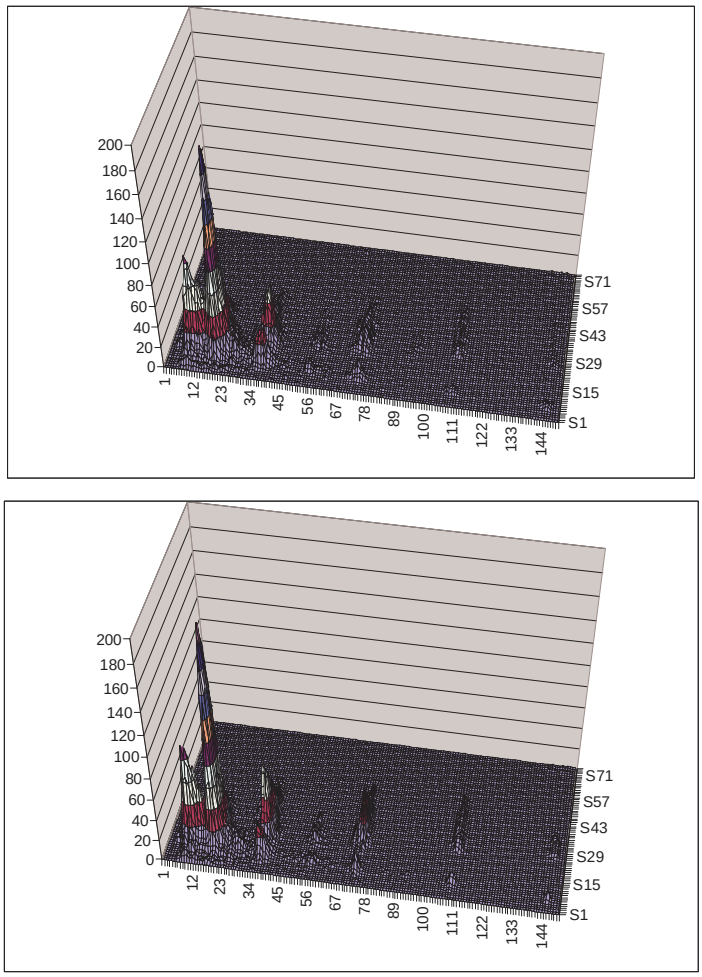


Figure 4.13. Beat histogram for *"Rammstein – Mein Herz Brennt"* without filtering (top) and after bass/treble cut filter (bottom).

Our 2D beat histogram is used in our work as a rhythmic feature vector. The size of the 2D histogram in our case is 70×150 bins where 150 bins are devoted to beat period axis in order to maintain high temporal accuracy whereas 15 bins has been observed enough for the details of strength axis.

4.3.2. Rhythmic similarity measure

Rhythmic similarity of two music pieces is then based on comparison of their beat histograms. Several measures can be used for similarity comparison of two histograms $H=\{h_i\}$ and $K=\{k_i\}$. They are classified into two categories: bin-by-bin similarity and cross-bin similarity measures. The bin-by-bin similarity measure only compares contents of corresponding histogram bins while cross-bin similarity measure includes a comparison of non-corresponding bins. Naturally, bin-by-bin similarity measures are more sensitive to slight variations of neighboring bins.

In the family of bin-by-bin similarity measures, several distances can be mentioned:

- **Minkovsky-Form Distance**

$$d(H,K)=\left(\sum_{i=1}^{N}|h_i-k_i|^r\right)^{1/r} \quad \textbf{(4.1)}$$

- **Histogram Intersection**

$$d(H,K)=1-\frac{\sum_{i=1}^{N}\min(h_i,k_i)}{\sum_{i=1}^{N}k_i} \quad \textbf{(4.2)}$$

- **Jeffrey Divergence**

Jeffrey divergence is a symmetric form of Kullback-Leibler divergence.

$$d(H,K)=\sum_{i=0}^{N}\left(h_i\log\frac{2h_i}{h_i+k_i}+k_i\log\frac{2k_i}{h_i+k_i}\right) \quad \textbf{(4.3)}$$

The main disadvantage of bin-by-bin similarities is their sensitivity to slight variations in neighboring bins. In the case of beat histograms, small changes in rhythm may result in vast modifications in the distance between two histograms as we can see in Figure 4.14.

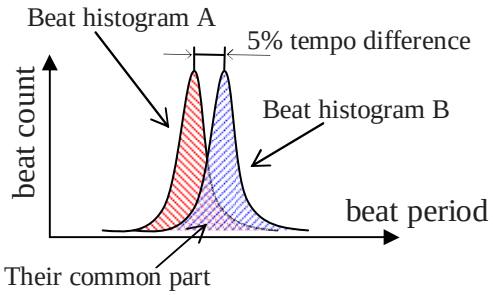


Figure 4.14. Small rhythmic changes lead to large bin-by-bin beat histogram distance (at strength plane).

A well known cross-bin similarity measure is the Earth Mover's Distance (e.g. [RU" N 00]). The Earth Mover's Distance (EMD) is based on a solution of transportation problem. It is a problem of suppliers and consumers where goods are to be transported from suppliers to consumers according to logistic costs. Stocks of suppliers are limited as well as consumers' appetites. Thus, the EMD can be used to measure the necessary effort or cost to transform one thing (histogram, image, signature etc.) into another one. Figure 4.15 illustrates its basic principle.

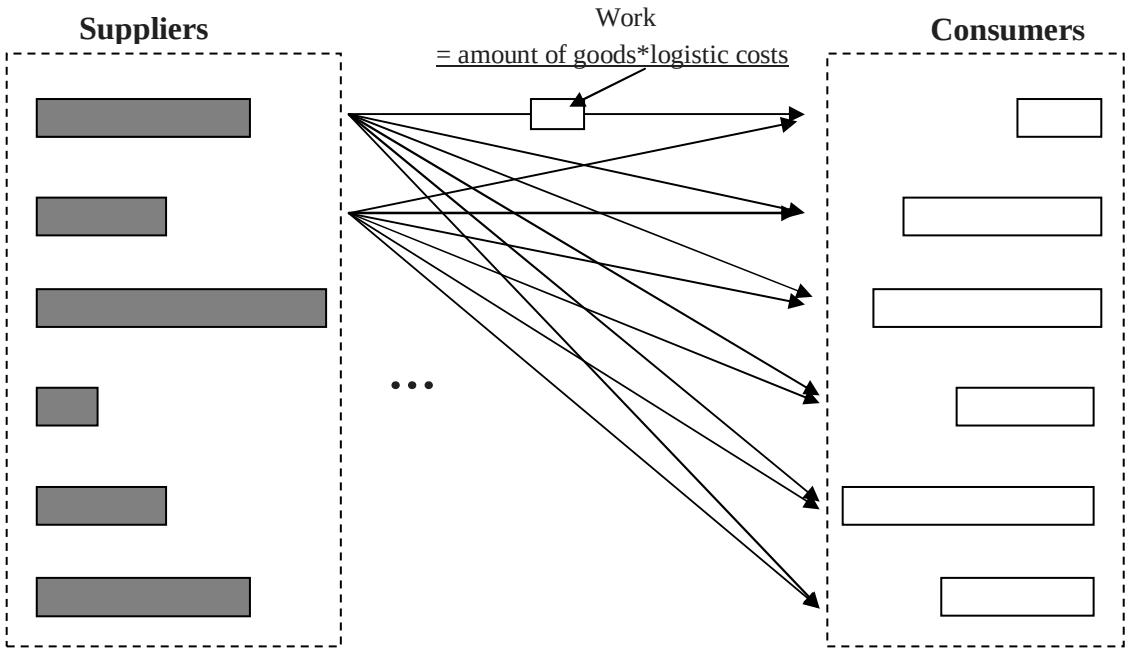


Figure 4.15. Illustration of EMD between suppliers and consumers.

Transfer of goods from a supplier to a consumer has a transportation cost. The work done on the transportation is amount of goods transferred from supplier X to consumer Y multiplied by the cost of transportation. The goal of EMD computing is to minimize the total cost. It can be formalized as the following linear programming problem: Let $A=\{(a_1,w_{a1}),\ldots,(a_m, w_{am})\}$ be the supplier grid with m suppliers (or histogram with m bins). Where a_i is a bin representative (coordinate or bin number etc.) and w_{am} is the weight of the bin. $B=\{(b_1,w_{b1}),\ldots,(b_n, w_{bn})\}$ is the second signature with n elements. $D=[d_{ij}]$ is the ground distance matrix where d_{ij} is distance between elements a_i and b_j (transportation cost).

The question is to find a flow $F=[f_{ij}]$ where f_{ij} is the flow between a_i and b_j that minimizes the total cost

$$Work(A,B,F)=\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij}\, f_{ij} \qquad \textbf{(4.4)}$$

with the following constraints:

$$f_{ij} \geq 0 \qquad 1 \leq i \leq m, \;\; 1 \leq j \leq n \qquad \textbf{(4.5)}$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{a_i} \qquad 1 \leq i \leq m \qquad \textbf{(4.6)}$$

$$\sum_{j=1}^{n} f_{ij} \leq w_{b_j} \qquad 1 \leq j \leq n \qquad \textbf{(4.7)}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min\left(\sum_{i=1}^{m} w_{a_i}, \sum_{j=1}^{n} w_{a_j}\right) \quad \textbf{(4.8)}$$

Once the minimization problem is solved, the Earth Mover's Distance is defined as the normalized total cost:

$$EMD(A,B) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \quad \textbf{(4.9)}$$

The advantage of the EMD to allow all possible variations of rhythms in our case is "compensated" by the cost of EMD computation. Complexity of EMD computation is polynomial ($\sim n^3$), it makes it hardly applicable for comparing large histograms that are in our case 70×150 bins and building complete similarity matrix of many-thousand musical files collection.

While all the previous similarity measures can be used for comparing our 2D beat histogram and outputting a rhythmic similarity, we propose in our work a tradeoff between computation efficiency and robustness of similarity measure to slight tempo variation. Indeed, we assume that two songs having similar rhythmic structure is likely to have similar beat histograms except small tempo variations. On the other hand, a music retrieval system by similarity needs to give the impression of being real time: once fingerprints are calculated for the musical collection, the algorithm of similarity search should run quickly to issue the complete similarity matrix or to give a list of similar songs to a query song. We thus propose a modified version of bin-by-bin similarity measure so that it is insensitive to slight histogram changes while keeping its advantage of low computation cost.

Our rhythmic similarity measure base on 2D beat histogram is defined as follows:

$$Dist_{H1,H2} = \sum_{x=1,y=1}^{N,M} \frac{1}{2}\left(\min_R\left(\left|H1_{x,y} - H2_{(x,y)+R}\right|\right) + \min_R\left(\left|H1_{(x,y)+R} - H2_{x,y}\right|\right)\right) \quad \textbf{(4.10)}$$

where

$H1$, $H2$ − beat histograms to compare

N, M − beat histogram size

R − tolerance area as for instance of the following form (Figure 4.16)

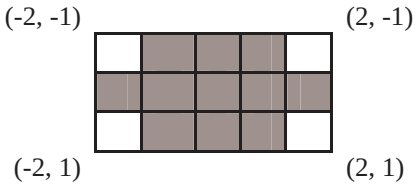


Figure 4.16. Tolerance area used in the similarity computation of two beat histograms.

As we can see from Figure 4.16, the tolerance area R thus defines thus slight histogram shifts in beat period axis X and beat strength axis Y.

4.4. A 2D beat histogram based tempo estimation algorithm and its evaluation

While general evaluation of our 2D beat histogram and the associated rhythmic similarity measure will be made through their applications. These applications are automatic genre classification and music similarity search, which are described in chapter 6. We describe in this section a sideway product of our 2D beat histogram, namely a tempo estimation algorithm and also propose an evaluation of our algorithm as a first validation of our VRT-based beat curve extraction procedure.

4.4.1. A 2D beat histogram based tempo estimation algorithm

As we can see from Figure 4.17, peaks on our 2D beat histogram correspond to the tempo of the piece or the tempo multiples like 2x, 3x, 4x, or 1/2x, 1/3x, 1/4x. We thus derive a tempo estimation algorithm from this 2D beat histogram.

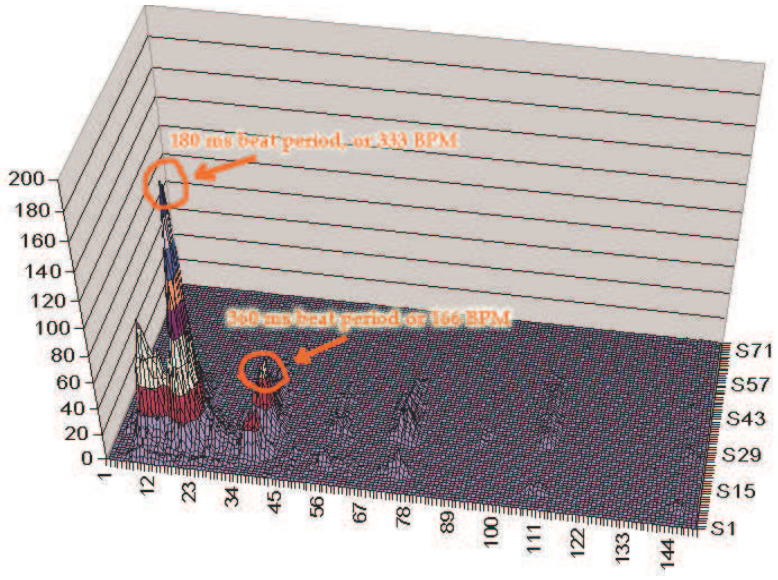


Figure 4.17. A 2D beat histogram. Main tempo and its 2x alias are marked on the image.

The basic assumption is that the true tempo can have its ” PM value ranging from 65 to 200 (rarely past 200 and fewer than 60, see e.g. [ALON 03]). Our algorithm proceeds first by seeking the maximal peak on the 2D beat histogram. If the value of its period hit the range of possible tempos, then the peak is considered as the found tempo. Otherwise, its multiples are examined in the following manner:

1) find the first multiple with significant (non-zero) amplitude
2) check if its period lies in the acceptable tempo range
3) if in acceptable range return the obtained tempo, otherwise find the next multiple

4.4.2. Experimental evaluations

Two evaluations of this simple tempo estimation algorithm have been carried out using different databases.

A detailed evaluation of our tempo estimation (and hence, beat detection) was conducted by Miguel Alonso form the research team of ENST Paris in the context of **MusicDiscover** project. Evaluation methods they used were pretty much the same as in known tempo induction contests (e.g. MIREX) with 8% of confidence window. The dataset which was used is the dataset of 474 30-sec. pieces annotated by Anssi Klapuri from Tampere University (Finland).

Here are the results (algorithm from this work is referred as EC-Lyon in the tests).

Acc1 – exact tempo estimation within acceptance window
Acc2 – tempo with 2x and 3x multiples
Acc3 – tempo with 1/2x and 1/3x multiples.

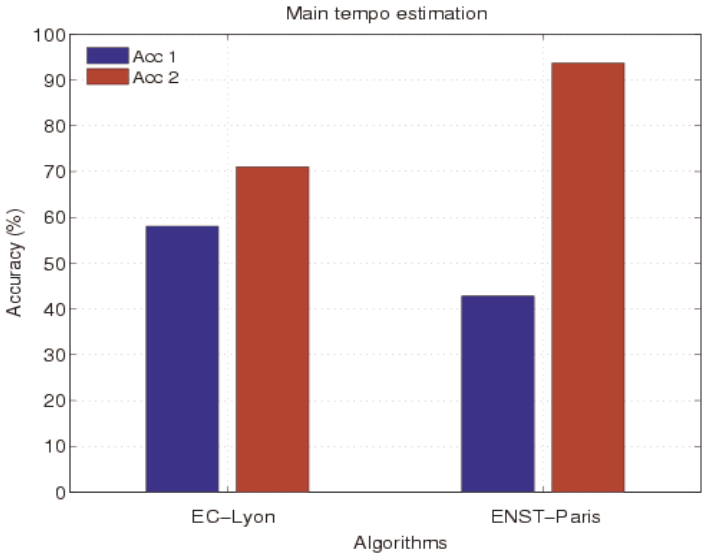


Figure 4.18. Performance comparison of two tempo estimation algorithms – EC-Lyon and ENST-Paris.

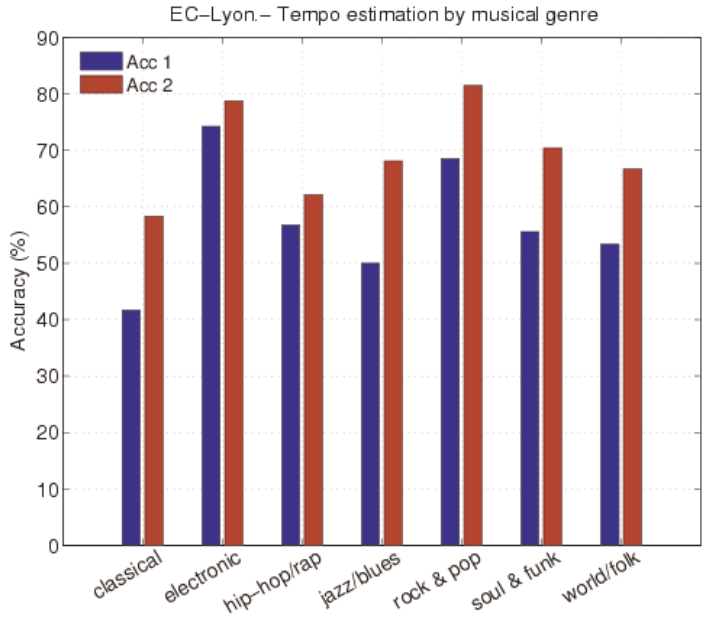


Figure 4.19. Distribution of the tempo estimation accuracy as function of genre for ECL algorithm.

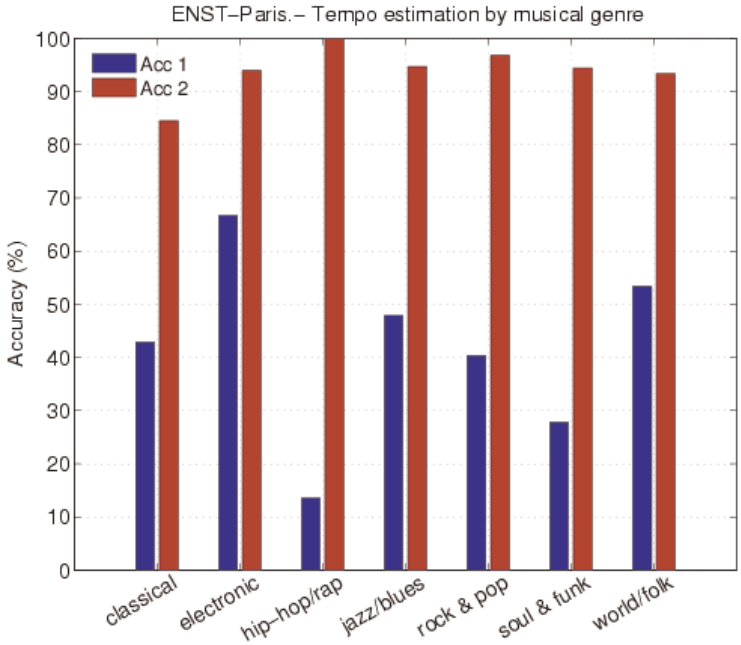


Figure 4.20. Distribution of the tempo estimation accuracy as function of genre for ENST algorithm.

The following figures concern tatum estimation. The tatum may be established by the smallest time interval between two successive notes, but is in general best described by the pulse series that most highly coincides with all note onsets.

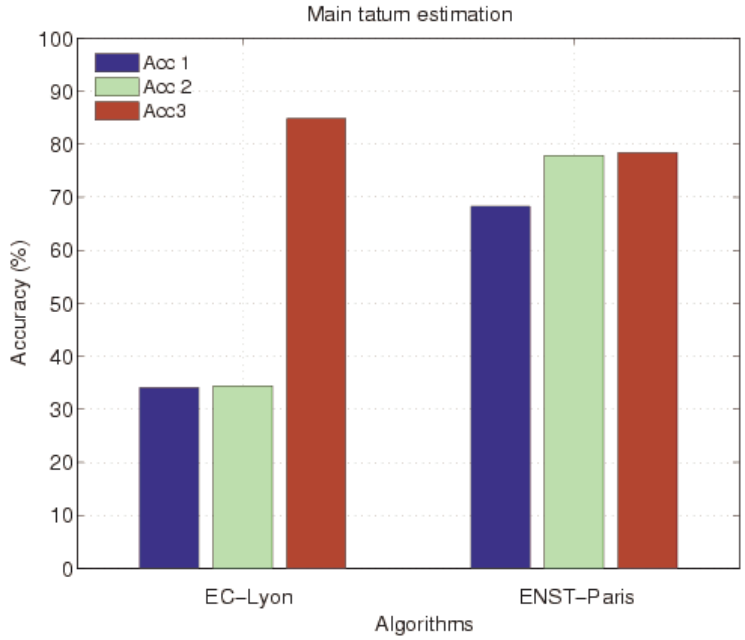


Figure 4.21. Tatum estimation accuracy of two algorithms.

The picture above shows an overestimation of tatum by our algorithm. It was extracted by our algorithm as a period of maximal peak on the beat histogram. The true tatum appeared to be two or three times the period obtained by our algorithm, according to the image since the measure Acc 3 takes into account these multiples.

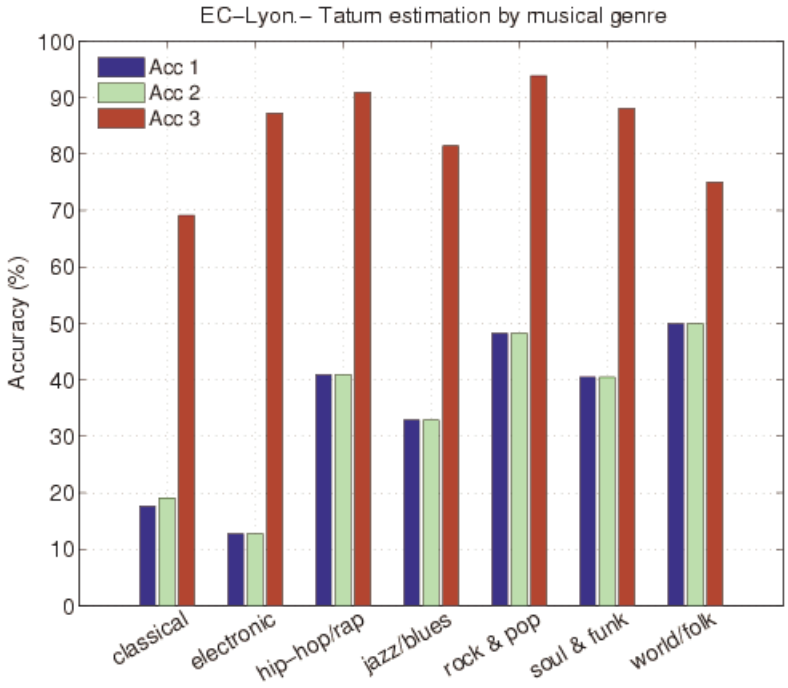


Figure 4.22. Tatum estimation performance by musical genre. EC-Lyon algorithm.

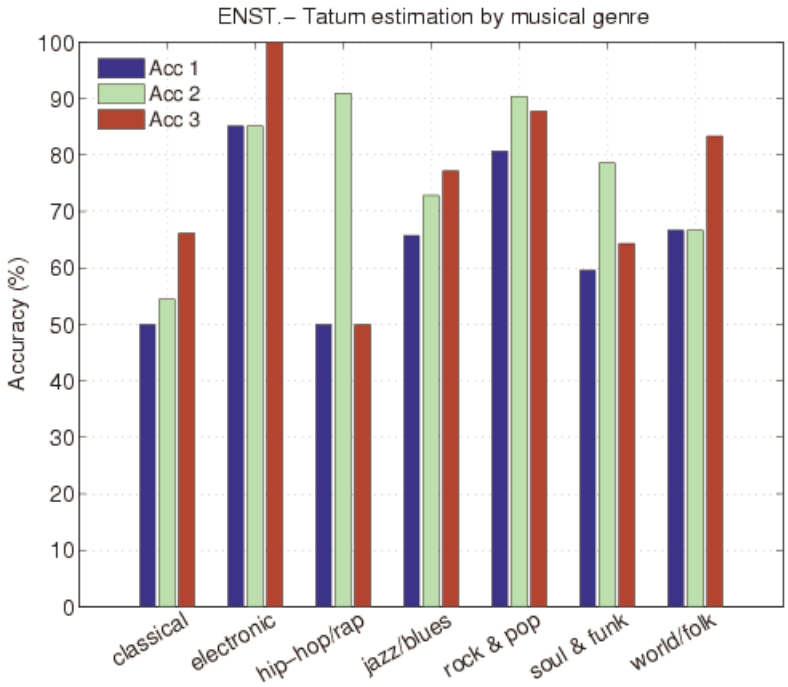


Figure 4.23. Estimation performance by musical genre. ENST-Paris algorithm.

As we can see from the previous figures, our tempo estimation algorithm has outperformed the algorithm of ENST-Paris [ALON 07] more than 10 points in exact tempo estimation accuracy. However, in the case of tatum estimation, our algorithm underestimates the exact tatum frequency and delivers values two or three times less than the actual tatum. This problem comes from the confusion between tatum and the declared tatum by our algorithm, i.e. the detected maximum peak in 2D beat histogram which is not necessarily tatum. For a better discrimination between tempo and tatum, a learning scheme on 2D beat histogram might be used to improve the previous result.

The second experiment was based on database which consisted of a subset of musical files used in *ISMIR2004* tempo estimation contest. Musical segments were selected from *BallroomDancers* collection. There were about 100 30-second files in 6 genres: Cha-Cha, Quickstep, Salsa, Samba, Tango and Waltz. All files are sampled at 16kHz 16 bit.

Precision metric A was taken as a percentage of tempos correctly estimated within 4% of absolute error. Precision metric ” was a percentage of tempos correctly estimated within 4% of absolute error and have its close multiples, i.e. 2x, 3x, 4x, 1/2x, 1/3x and 1/4x to be a correct estimation as well as it is done at common tempo estimation contests.

The following results were obtained for 3 analysis window shifts - 10, 15, 30ms (Table 4.1 - Table 4.4). The last table is shown here to indicate the fact that increasing the confidence area strongly influences the results.

Table 4.1. Tempo estimation results for 10ms analysis window shift.

	ChaCha	Quickstep	Salsa	Samba	Tango	Waltz	Average
A	100	0	66.7	0	33.3	0	33.3
B	100	60	66.7	60	33.3	21.4	56.9

Table 4.2. Tempo estimation results for 15ms analysis window shift.

	ChaCha	Quickstep	Salsa	Samba	Tango	Waltz	Average
A	100	0	55.6	0	60	35.7	41.9
B	100	86.7	55.6	66.7	60	57.1	**71.0**

Table 4.3. Tempo estimation results for 30ms analysis window shift.

	ChaCha	Quickstep	Salsa	Samba	Tango	Waltz	Average
A	92.9	0	22.2	66.7	66.7	28.6	**46.2**
B	92.9	66.7	33.3	86.7	66.7	42.9	64.8

Table 4.4. Tempo estimation results for 30ms analysis window shift, 10% confidence area.

	ChaCha	Quickstep	Salsa	Samba	Tango	Waltz	Average
A	100	0	78.8	66.7	93.3	78.6	69.4
B	100	93.3	88.9	86.7	93.3	92.9	92.5

The best result known in the literature for the complete *BallroomDancers* database (~3200 musical excerpts) is 67% (measure A) and 84% (4% accuracy and 2x, 1/2x etc. tempos taken into account, measure ”). All results of ISMIR 2004 tempo induction contest is given in the Table 4.5.

Table 4.5. Complete table of reference results from ISMIR2004 tempo estimation contest.

4% accuracy	4% accuracy, w/ alias tempos
1. Klapuri 66.91%	1. Klapuri 84.36%
2. Uhle 46.26%	2. Dixon_auco 81.74%
3. McKinney 44.89%	3. McKinney 80.68%
4. Dixon_auco 38.66%	4. Dixon_trac 74.14%
5. Scheirer 37.70%	5. Dixon_indu 72.98%
6. Alonso_sppr 35.98%	6. Alonso_sppr 69.30%
7. Dixon_indi 31.60%	7. Uhle 68.02%
8. Tzan_medsumbands 30.94	8. Scheirer 67.73%
9. Tzan_medmultibands 30.72%	9. Alonso_auco 56.92%
10. Alonso_auco 27.38%	10. Tzan_medsumbands 55.17%
11. Dixon_trac 26.53%	11. Tzan_medmultibands 50.48%
12. Tzan_histsumbands 22.42%	12. Tzan_histsumbands 49.87%

The experiment shows that our tempo estimation algorithm achieves comparable results on a partial randomly chosen subset of ISMIR 2004 database.

These two experiments, while displaying acceptable performance of our simple 2D beat histogram tempo estimation algorithm, thus tend to show the

soundness of our VRT-based beat extraction curves as they characterize rhythmic events in a music signal. The relevance of the derived rhythmic similarity measure for music genre classification or music retrieval by similarity will be further studied in chapter 6 dedicated to applications.

The results of the second experiment we provide here are given as an indication only, since dataset we used is a small subset of the common database.

4.5. Conclusion

In this chapter, we have proposed some VRT-based solutions for characterizing rhythmic property of music signal and its similarity. Rhythmical features are first extracted from a VRT spectrogram as a beat curve using some basic image enhancing algorithms. The resulting beat strength curve can be assumed to be a beat probability-related curve. This beat curve is further summarized by a beat histogram in two-dimensional space. There are two advantages of such representation. First, the histogram is built with the second axis covering a range of thresholds in the beat detection algorithm, so the necessity of threshold choosing is omitted. The second advantage is that this histogram is therefore volume-independent as the range of thresholds is taken from 0 to maximum.

The rhythmical similarity has been defined as the distance between two 2D beat histograms. The main question is how to compare two beat histograms? Several most popular approaches to histogram comparison have been studied in this chapter – from bin-by-bin to complete cross-bin comparison algorithms. We have proposed an intermediate solution which keeps the rapidity of bin-by-bin methods and add a tolerance to slight neighbor bin variations.

” ased on our 2D beat histogram, we have also proposed a simple tempo estimation algorithm which has displayed, when evaluated by two experiments, comparable performance to other tempo estimation algorithms known in the literature.

Our 2D beat histogram and its associated rhythmic distance measure will be further used as rhythmic features in music genre classification and music retrieval by similarity as we will see in chapter 6.

Chapter V

Melody-related similarity features

5. Melody-related similarity features

We call *melody-related similarity features* a series of similarity measures based on melodic and harmonic properties of a music signal. The true melodic similarity metrics are used to compare melodies upon their *score* representation like MIDI (Musical Instrument Digital Interface). The question of score-based melodic similarity is well known and well explored in the literature (see e.g. [TOIV 02; TYPK 03]). A work providing a comparison of different approaches is for example [GRAC 02].

Unfortunately, working with raw audio signal makes symbolic similarity algorithms useless since a general 100% effective signal-to-score translation is not yet available. Therefore, we do not focus here on score-based melodic similarity but rather on signal-based one. We propose in this chapter a VRT-based approach for approximate extraction of multiple fundamental frequencies with relative amplitudes of their harmonics from the signal. Furthermore, based on these estimated multiple f_0 and their harmonics we also derive several melody related similarity measures within the framework of music information retrieval.

5.1. Related work

A melody-related characteristic currently used in the MIR community is *pitch*. The notion of pitch is often related to monophonic pitch obtained using autocorrelation-based algorithms (see e.g. [MCKI 03]). However, this definition of pitch is not equivalent to converting a polyphonic music signal into note score. Pitch in musical signal is defined as fundamental frequency (f_0) of a sounding voiced instrument.

Early works on automatic pitch detection were developed for speech signal. (see e.g. [A"E 96; HU 01]). Much literature nowadays treats the ***monophonic*** case (only one f_0 present and detected) of fundamental frequency estimation. In the monophonic case several approaches are well known. Autocorrelation based algorithms are among the most frequently used approaches for f_0 estimation (e.g. [CHEV 02; TALK 95]). Operating directly in the time-domain, zero-crossing rate based methods are the simplest techniques for fundamental frequency estimation. Other techniques operate in frequency domain. For instance, harmonic matching methods propose to extracting peaks in the spectrum which are further compared to predicted harmonics of each f_0 candidate (see e.g. [DOVA 91; PISZ 79]).

The previous works are mostly suitable for to speech signal analysis. Unfortunately, it is admitted that algorithms designed for monophonic or speech case are rarely usable for multi-pitch detection.

There are also works studying the polyphonic case of music signal in the literature. However, in most of these works the polyphonic music signal is usually considered with a number of restrictions such as the number of notes played simultaneously or some hypothesis about the instruments involved.

The work [KLAP 99] presents a pitch detection technique using separate time-frequency windows. "oth monophonic and two-voice polyphonic cases are studied. Multiple-pitch estimation in the polyphonic single-instrument case is described in [LAO 04] where authors propose to apply a comb-filter mapping linear frequency scale of FFT into logarithmic scale of notes frequencies. As the method is FFT-based, the technique inherits drawbacks of FFT for music signal analysis as we highlighted in Chapter 3, namely requiring large FFT analysis windows thus leading to low time resolution.

An advanced f_0 detection algorithm is presented in [GOTO 01a] which is based on finding frequencies which maximize a f_0 probability density function. The algorithm is claimed to work in the general case and have been tested on CD recordings.

We can also mention many other recent works on multiple fundamental frequency estimation, for instance the ones in [LI 07; YEH 05]. "oth these works are probabilistic methods. The first one uses a probabilistic HMM-based approach taking into account some a priori musical knowledge such as tonality. Variable results from 50% to 92% of recognition rates for different instruments in MIDI synthesized sequences are reported. The second algorithm is evaluated on synthetic samples where each file contains only one combination of notes (1 note or 1 chord).

It is not evident how to compare these different multiple f_0 estimation algorithms as assumptions or models on the polyphonic music signal are often not explicitly stated. On the other hand, there is no single evident way of multiple f_0 detection. Some algorithms are strong in noisy environment; some algorithms require a priori training; others are able to detect inharmonic tones etc. The most popular approach to f_0 estimation is harmonic pattern matching in frequency domain. Our multiple-f_0 estimation algorithm makes use of this basic idea and relies on our VRT specifically designed for music signal analysis.

5.2. Our VRT-based multiple f0 estimation algorithm

5.2.1. Principle and procedure

The basic principle of the f_0 estimation algorithm consists of modeling of our VRT spectrum with harmonic models. Real musical instruments are known to have inharmonic components in their spectrum [KLAP 04]. It means that the frequency of the n^{th} partial can be not strictly equal to $f_0{}^*n$.

The algorithm we describe does not take such inharmonic components into account, but it tolerates some displacement of partials in a natural way.

A typical "flat" harmonic structure used to model the spectrum is depicted in the Figure 5.1.

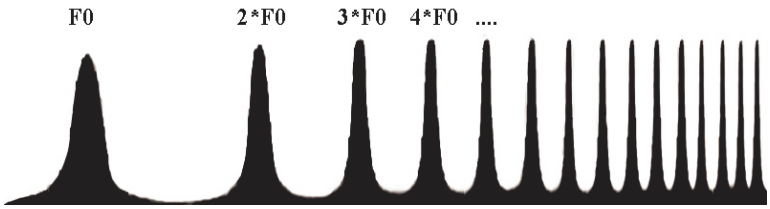


Figure 5.1. Harmonic structure.

This fence is a vertical cut of VRT spectrogram calculated from a synthetic signal representing an ideal harmonic instrument. The width of peaks and space between them is variable because the VR transform has a logarithmic frequency scale.

In the next step, these models are used to approximate the spectrum of the signal being analyzed in order to obtain a list of f_0 candidates.

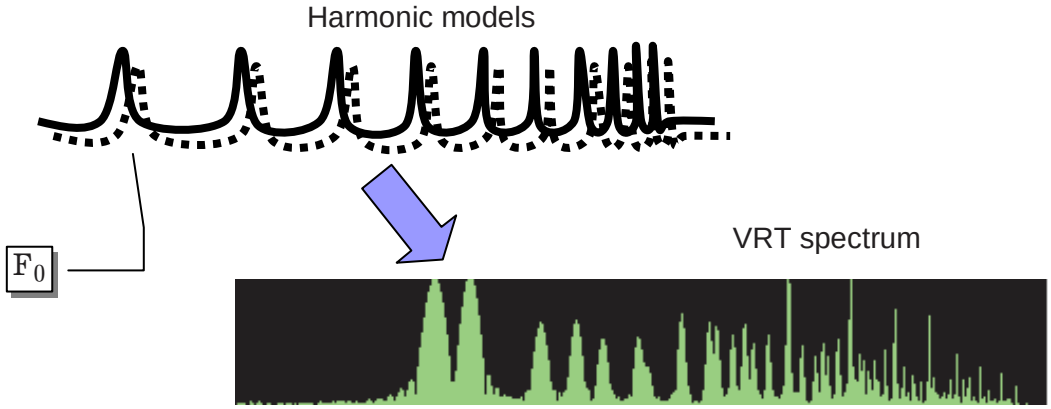


Figure 5.2. Matching of harmonic models to spectrum.

During every iteration of the algorithm, such harmonic fence is shifted along the frequency axis of the spectrogram and matched with it at each starting point.

The matching of the harmonic model is done as follows. At every harmonic their amplitudes a_i are taken from the values of the spectrogram for the frequencies of i^{th} harmonics. As frequencies of harmonics do not necessarily have integer ratios to the fundamental frequency, we take the maximum amplitude in a close neighborhood, as it is explained in Figure 5.3.

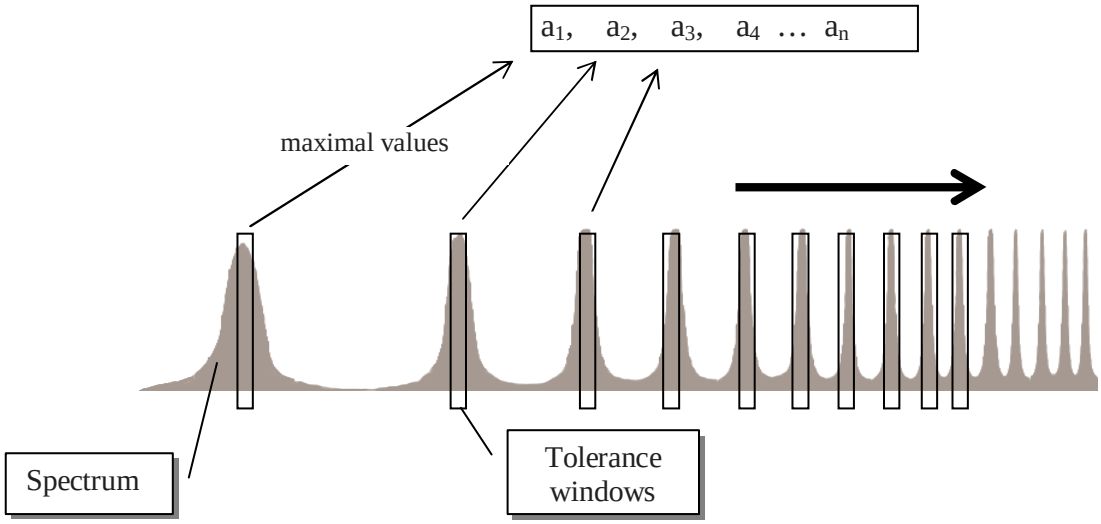


Figure 5.3. Procedure of extraction of harmonic amplitude vector.

This procedure forms a function $A(f)$ which is a norm of the vector $\boldsymbol{a}$ for the frequency f. The value of frequency for which the function A takes its maximum value is considered as an f_0 candidate.

Further, the obtained f_0 candidate and the corresponding vector $\boldsymbol{a}$ of harmonics amplitudes is transformed into a spectrum slice like in Figure 5.1. The shape of peaks is taken from the shape of VRT spectrum of a signal with a sine wave with corresponding frequency. This slice is then subtracted from the spectrum under study. The iterative process is repeated either until the current value of harmonic structure $A(f)$ becomes inferior compared to a certain threshold or until the maximum number iterations has been reached. We limit the maximum number of iterations to 4, and therefore the maximum number of notes that can be simultaneously detected is 4. As it was observed in preliminary experiments, increasing the number of simultaneously detected notes doesn't improve the f_0 detection performance significantly for high-polyphonic music, because after 3rd or 4th iteration the residue of spectrum is already quite noisy as almost all harmonic components have been already subtracted from it due to harmonic overlaps. The procedure of multiple f_0 estimation can be depicted by the following algorithm:

```
max_match = 0
model_width = number_of_bins taken by N-harmonic model
for starting_point = 0 to spectrum_size – model_width {
      A[starting_point] = match model with the spectrum at starting_point;
      if max A[starting_point] then max_match = starting_point;
}
f0_candidate = convert to pitch (max_match);
subtract the model from spectrum at max_match;
```

```
if spectrum is not empty and max iterations is not reached then continue iterations;
```

The matching algorithm can be described by the following function:

```
Function match (starting_point)
{
    for i = 0 to number_of_harmonics_in_model {
    y = starring_point + disp[i]; // disp – an array of relative harmonic positions, disp[0]=0
    a[i] = max(W[y-1], W[y], W[y+1]); // W[0...spectrum_size] – current spectrum slice
                                      // a tolerance window size of 3 bins is applied here
    }
    match = norm (a[…]);
}
```

The procedure of note extraction is applied each 25 ms to the input signal sampled at 16 kHz 16 bits. Hence, for the shortest notes with duration around 50-70 ms we obtain note candidates at least twice in order to be able to apply filtering techniques. Every slice produces a certain number of f_0 candidates; then, f_0 candidates are filtered in time in order to remove noise and unreliable notes. The time filtering method used is the nearest neighbor interframe filtering. 3 successive frames are taken and f_0 candidates in the middle frame are changed according to the f_0 candidates in the side neighbors. This filter removes noisy (false detected) f_0 candidates as well as holes in notes issued by misdetection. It can be explained by the following algorithm:

```
for all f0 candidates at time t-1 {
    if candidate absent at t-2 or candidate absent at t the remove candidate at t-1
}
for all f0 candidates at time t-2 {
    if candidate present at t then add candidate at t-1
}
```

” lock diagram of the note detection algorithm is shown in Figure 5.4.

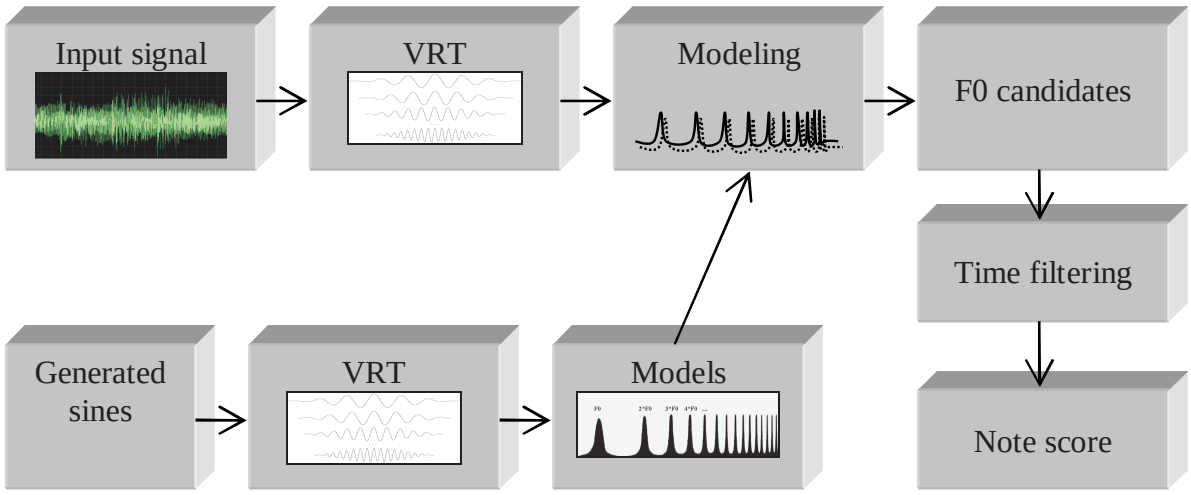


Figure 5.4. Block diagram of note detection procedure.

5.2.2. Experimental evaluation

The easiest way to make basic evaluation experiments in automated music transcription is to use MIDI files (plenty of them can be freely found on the Internet) rendered into waves as input data. The MIDI events themselves serve as the ground truth. However, the real life results must be obtained from recorded music with true instruments and then transcribed by educated music specialists.

In our work we used wave files synthesized from MIDI using hardware wavetable synthesis of Creative S" Audigy2 soundcard with a high quality 140Mb SoundFont bank "Fluid_R3" freely available on the Internet. In such wavetable synthesis banks all instruments are sampled with good sampling rates from real ones: the majority of pitches producible by an instrument are recorded as sampled (wave) block and stored in the soundfont. In the soundfont we used, acoustic grand piano, for example, is sampled every four notes from a real acoustic grand piano. Waves for notes which are in between these reference notes are taken as resampled waves of closest reference notes. Therefore, signal generated using such wavetable synthesis can be considered as a real instrument signal recorded under ideal conditions. And a polyphonic piece is an ideal linear mixture of true instruments. To make the recording conditions closer to reality in some tests we passed the signal over speakers and record it with a microphone.

Figure 5.5 is a screen-shot of our note detection program which is used for note detection. With a capability of real-time input signal processing, our note detection program achieves candidate f_0 detection and time filtering with 25 ms interframe time. It is capable to parse and play MIDI files and compare the result of note detection with the original music score.

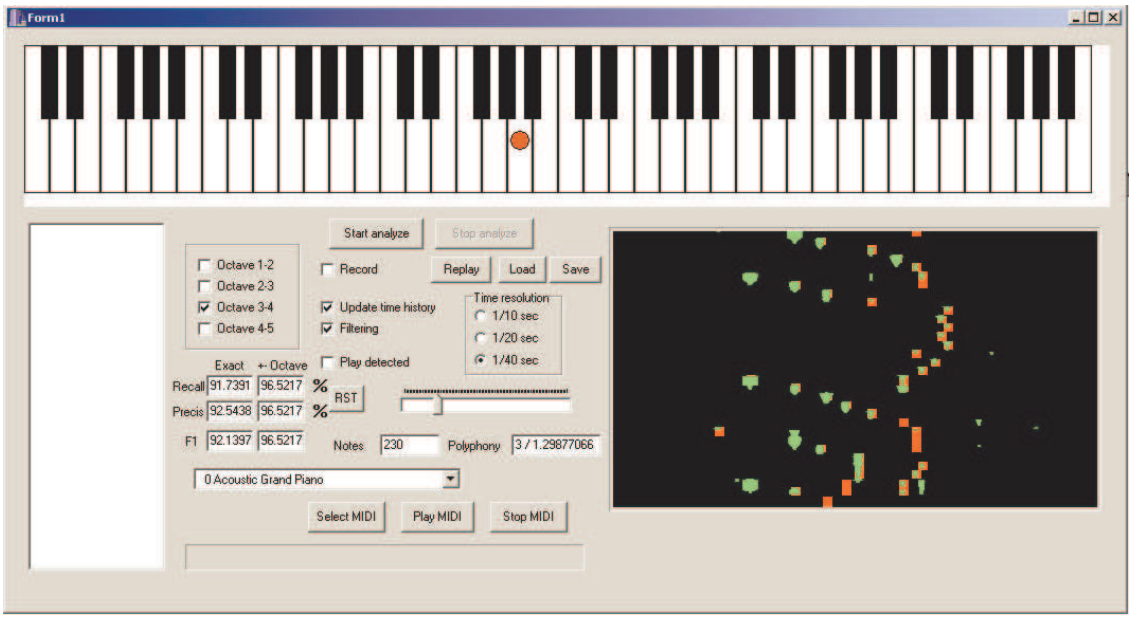


Figure 5.5. Note recognition and evaluation program.

Recall and Precision measures are used to measure the performance of the note detection. Recall measure is defined as:

$$\text{Recall} = \frac{\textit{the number correct notes detected}}{\textit{the actual number of notes}} \tag{5.1}$$

Precision is defined as follows:

$$\text{Precision} = \frac{\textit{the number correct notes detected}}{\textit{the number of all notes deteced}} \tag{5.2}$$

For the overall measure of the transcription performance, the following *F1* measure is used

$$F1 = \frac{2 \cdot \textit{Recall} \cdot \textit{Precision}}{\textit{Recall} + \textit{Precision}} \tag{5.3}$$

All falsely detected notes also include those with octave errors. For some tasks of music indexing as for instance tonality determination, what is important is the note basis and not the octave number. For this reason, the performance of note detection without taking into account octave errors is estimated as well.

Our test dataset consists of 10 MIDI files of classical and pop compositions containing 200 to 3000 notes. Some other test sequences were directly played using the keyboard. The following tables (Table 5.1 - Table 5.4) display precision results of our multiple pitch detection. Perf.Oct column stands for performance of note detection not taking into account notes' octaves (just the basic note is important). The *polyphony* column indicates the

maximum and the average number of simultaneously sounding notes found in the play.

Table 5.1. Note detection performance in monophonic case. Sequences are played manually using the keyboard.

Name	№ of notes	Polyphony	Performance			Perf. Oct
		max / avg	Recall	Prec	F1	F1
Piano Manual	*150*	*1 / 1*	*100*	*100*	*100*	*100*
Violin Manual	*160*	*1 / 1*	*100*	*97*	*98.5*	*100*

Table 5.2. Note detection performance in polyphonic case. Sequences of chords are played manually using the keyboard.

Name	№ of notes	Polyphony	Performance			Perf. Oct
		max / avg	Recall	Prec	F1	F1
Piano Manual	*330*	*2 / 1.8*	*98.5*	*100*	*99.5*	*99.7*
Piano Manual	*214*	*5 / 2.2*	*95.8*	*100*	*97.8*	*99.1*
Flute Manual	*174*	*4 / 2*	*97.7*	*97.7*	*97.7*	*99.7*

Table 5.3. Note detection performance in polyphonic case. Classical music titles (single- and multi-instrument, no percussion).

Name	№	Polyphony	Performance			Perf. Oct
	of notes	max / avg	Recall	Prec	F1	F1
Fur_Elize	*924*	*6 / 1.6*	*91.1*	*88.7*	*88.9*	*95.6*
Fur_Elize w/ microphone	*924*	*6 / 1.6*	*88.1*	*86.9*	*87.5*	*95.4*
Tchaikovski 01	*177*	*4 / 3.5*	*84.7*	*95.5*	*89.8*	*95.4*
Tchaikovski 16	*186*	*4 / 2.6*	*86.5*	*100*	*92.8*	*97.2*
Bach 01	*687*	*5 / 1.7*	*91.1*	*88.7*	*89.9*	*98.2*
Bach 03	*549*	*5 / 2.1*	*98.9*	*91.9*	*95.2*	*96.8*
Bach Fugue	*252*	*5 / 2.4*	*83.7*	*76.1*	*79.8*	*93.2*
Vivaldi Mandolin Concerto	*1415*	*6 / 2.9*	*70.1*	*74.8*	*72.4*	*91.5*

Table 5.4. Note detection performance in polyphonic case. Popular and other music (multi-instrument with percussion).

Name	№	Polyphony	Performance			Perf. Oct
	of notes	max / avg	Recall	Prec	F1	F1
K. Minogue	*2545*	*10 / 4.7*	*40.6*	*37.1*	*38.8*	*64.3*
Madonna	*2862*	*8 / 3.9*	*43.9*	*56.9*	*49.5*	*66.4*
Soundtrack f/ Godfather	*513*	*9 / 4.1*	*88.7*	*67.2*	*76.5*	*90.4*

As we can see from these tables, our algorithm performs quite well in the monophonic case. Good results are also obtained in polyphonic case with classical music having a low average level of polyphony (number of notes simultaneously played). More complex musical compositions which include percussion instrument and have high polyphony rate have produced lower recognition rates. In our note detection algorithm, we have limited the maximal detectable polyphony to 4 while the maximal and average polyphony in the case of popular and other music is 10 and 4.7

correspondingly. The octave precision, however, stays high (perf. Oct F1 field).

For comparison purpose, we also implemented our note detection algorithm based on FFT with different window size instead of our VRT. We carried out an experiment with a set of polyphonic classical compositions (~1000 notes) using this FFT-based note detection algorithm. Table 5.5 and Figure 5.6 summarize the experimental results.

Table 5.5. Comparison of transcription performance based on different time-frequency transforms (the FFT with various window sizes versus VRT).

Transform	FFT	FFT	FFT	VRT
FFT size or number of VRT frequency samples	*1024*	*2048*	*4096*	*1024*
Result (F1)	*66.2*	*77.6*	*80.5*	***91.3***

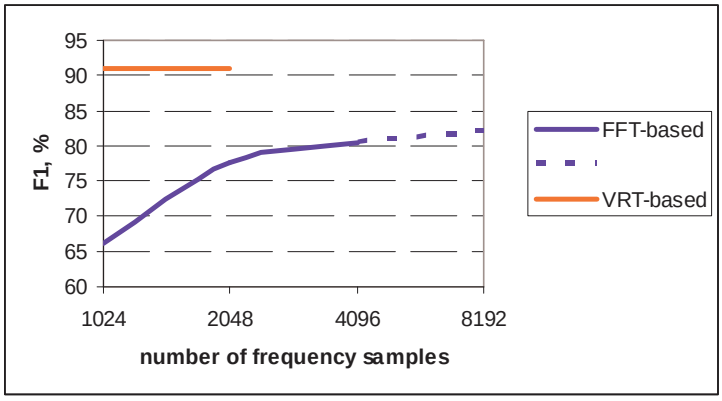


Figure 5.6. Note detection algorithm performance according to underlying spectral analysis approach.

Further increase of the FFT window size lowers the time resolution down to 0.5-1 seconds so that note changes quicker that 0.5 seconds cannot be resolved anymore.

These experimental results show the advantage of our VRT and its simple use performs multiple note detection quite well in the case of low average polyphony rate.

5.3. Melody-related similarity features

Let us point out that the main goal of this work is music retrieval by similarity which is quite subjective. As such, a 100% accurate note detection algorithm may not be necessary. In the following, we thus assume that our VRT-based note detection algorithm delivers ***partial transcription*** of a music signal on which we can build melody-related similarities. The basic idea is that for each window our multiple f_0 detection algorithm issues a list of detected f_0`s together with relative amplitudes of their partials. This information is then used for building several kinds of statistical characteristics (histograms).

5.3.1. Note profile histogram

The simplest way to calculate a similarity distance is to calculate a distance between note histograms. Note histograms, or *chroma profiles* and *chromagrams* [PAUW 04] are well known and they are computed across the whole musical file or its part and serve for estimation of musical similarity by tonality as well as tonality estimation (musical key) itself.

Tonality is a musical term denoting a system of relationships between a series of pitches (notes) that form melodies and chords. In simple words, tonality can be referred as describing a collection of pitches, used to build a musical piece. A tonality has a *tonic* – a central note as its most stable element. Besides the tonic, one of the most important notes of a tonality is the dominant (the 5^{th} note) and subdominant (the 4^{th} note).

Besides the tonic tonalities are distinguished by *modes* – major and minor (natural, harmonic, melodic). Each of them has different musical characteristics regarding the position of tones and semitones. Figure 5.7 gives an example of two tonalities – Do-major (C-major[1] or C-dur) and Do-minor.

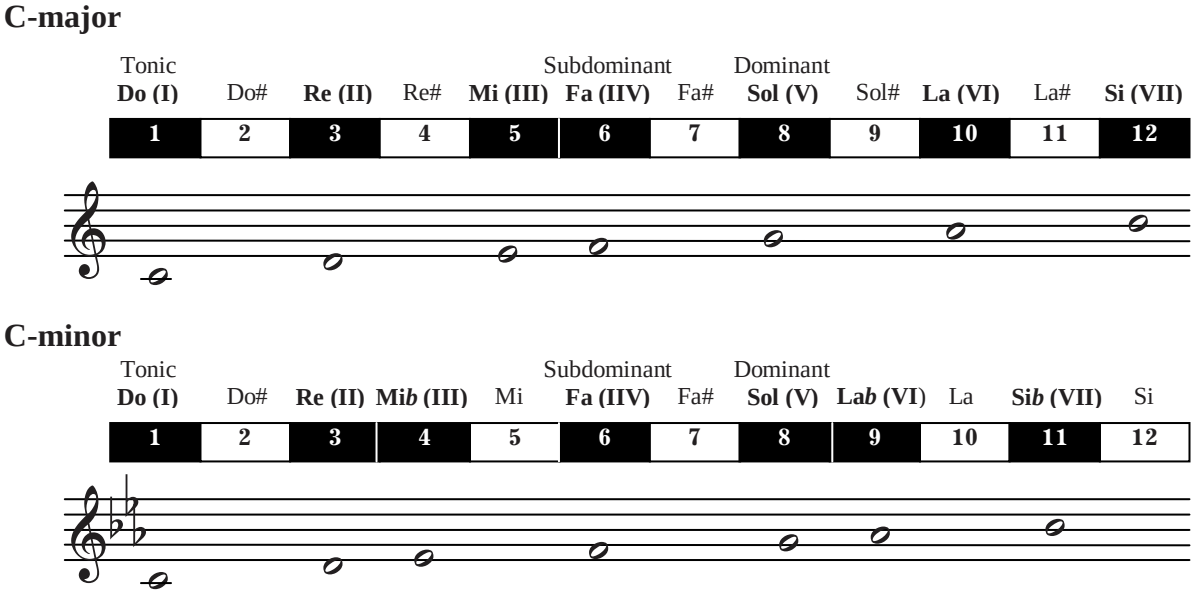


Figure 5.7. Tonality. Do-Major and Do-Minor (natural) tonalities.

These are parallel tonalities in respect of the tonic. There are also parallel tonalities in the meaning of notes used. For do-major it is la-minor.

The note histogram or tonal profile has a close relationship with the tonality as tonality and its mode (minor/major) are related to the probabilities of notes that can be observed in the play. For example, for a song in Do-major it is not very probable to find Do# in it. An example of two

[1] Notes are counted Latin alphabet letters A, B, C, D, E, F, G where C corresponds to Do

such tonal profiles (for C-major and C-minor) is given in Figure 5.8. In music analysis applications the tonality can be "guessed" using note histogram of the musical piece [CHUA 05; PEET 06].

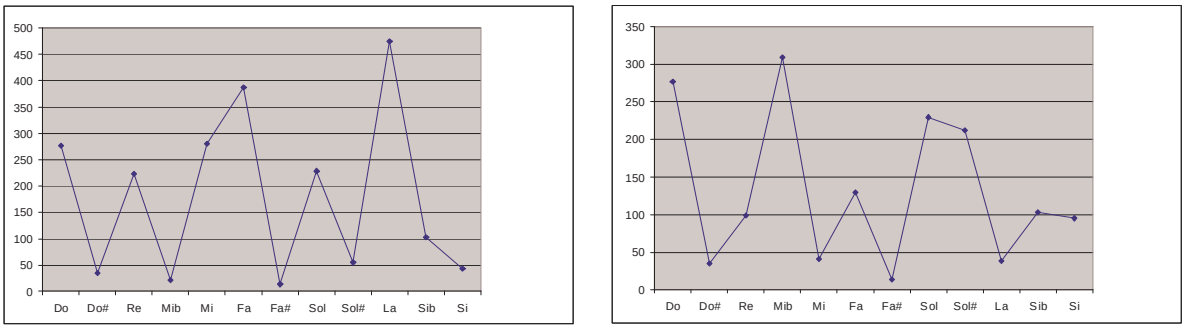


Figure 5.8. Note profiles for major (C-major, left) and minor (C-minor, right) tonalities (approximate).

In our work, we propose to utilize note profile histogram as a measure in order to characterize a tonal similarity between two musical compositions using Minkovski-form distance (4.1). Since a transposition in music does not change the perception of melody, the distance must be computed taking into account all possible transpositions. An illustration of the process is given in Figure 5.9. The algorithm makes a rotation of one of the two histograms. As the number of notes in histograms is 12, the rotation produces 12 variants. These variants of the first histogram are compared to the second histogram by computing a distance between histograms. The minimum value of obtained distances is used as the measure of tonal distance.

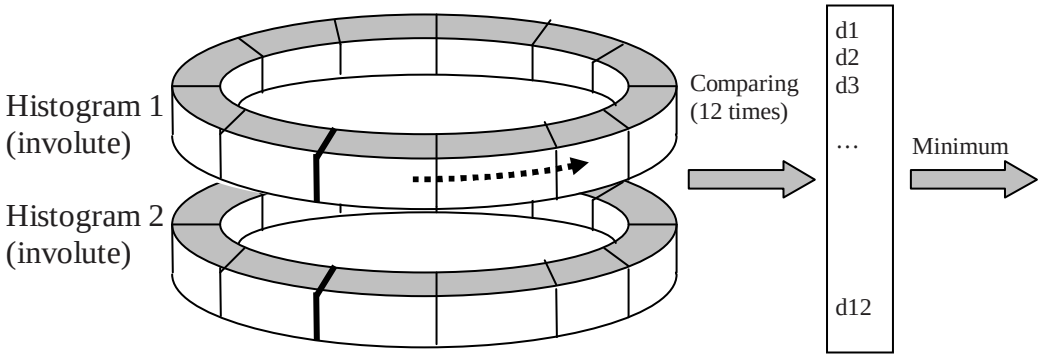


Figure 5.9. Comparison of note histograms taking into account all possible transpositions.

This tonal distance is assumed to be useful in finding the musical pieces which are similar in tonality, and therefore, probably similar in their mood – music in a minor key is often claimed to sound sad whereas music in a major key is typically viewed as sounding cheerful[1].

[1] http://en.wikipedia.org/wiki/Major-Minor

The variant of note histogram that we use is the multi-octave note histogram. In this type of tonal profile, octaves are not rolled up into one to produce a histogram, but histogram is produced from the absolute height of the notes. Comparison of such multi-octave histograms can be done by shifting them by up to ± ½ octaves with one semi-tone shift (melodies can be transposed and we want to accept such transpositions within one octave). Multi-octave tonal profile is mostly indicating the total distribution of notes in the spectrum, for example a balance between bass and melody lines can be extracted, etc.

5.3.2. Note succession histogram

Another musical similarity measure studied in this work is a similarity based on note successions histogram. Here the probability of 3-note chains is collected and its histogram is then used as a "fingerprint" of musical title. A musical basis of such similarity measure is that if passages are frequent in two musical compositions, it gives a chance that these two compositions have similarities in melody or harmony.

The note successions histogram extracted from a music signal is computed as follows. First, note extraction over the whole piece is carried out. Then the detected notes are grouped in local note histograms in order to find a dominating note in each grouping window. The size of the grouping window may vary from 100ms to 1 sec. Finally, all dominant notes are extracted from local histograms and their chains are collected in the note successions histogram. The process is depicted in Figure 5.10.

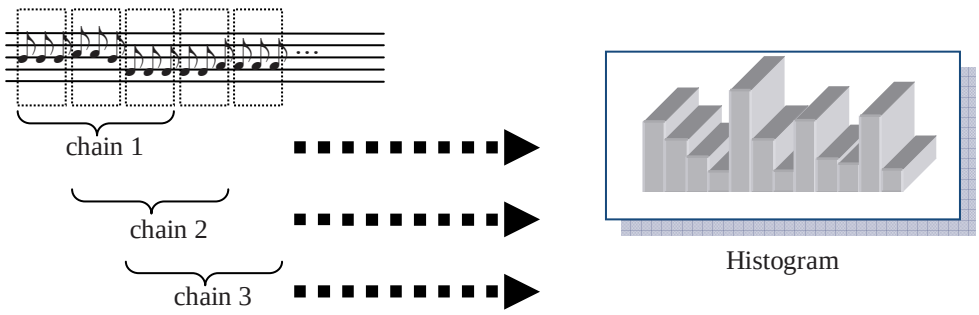


Figure 5.10. Procedure of note succession histogram calculation.

The resulting histogram is a 3-dimensional histogram where each axis is one note of 3-note chain found in the musical piece under study.

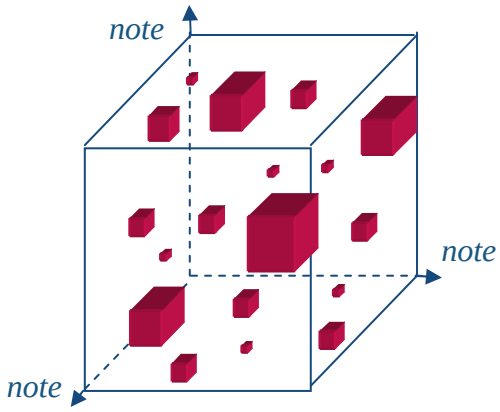


Figure 5.11. 3D note successions histogram example.

Figure 5.11 illustrates such a histogram where red cubes represent histogram bins. The bigger is the cube – the higher is the value of the corresponding histogram bin. ” ins are depicted in three-dimensional space since the histogram is three-dimensional. As each note axis has 12 positions (corresponding to notes), the whole histogram therefore comprises 12^3 bins. In the case of 4-note succession histogram, it contains 12^4 bins which would be too large and too sparse for good comparison as few notes in one play would fill the histogram.

To compare the note succession histograms we apply the same principle as for one-dimensional note histograms. All rotations around 3 axes of the histogram are considered simultaneously and the minimum distance value is taken.

5.3.3. Timbre histogram

The third characteristic we extract from a musical piece is the timbre histogram. In general, “voiced” instruments differ from each other also by their timbre – profile of their partials. In our work we collect all detected notes with relative amplitude of their harmonics. Further, relative amplitudes of harmonics are reduced to 3-4 bits and attached together in order to form an integer number. Histogram of these numbers is then computed.

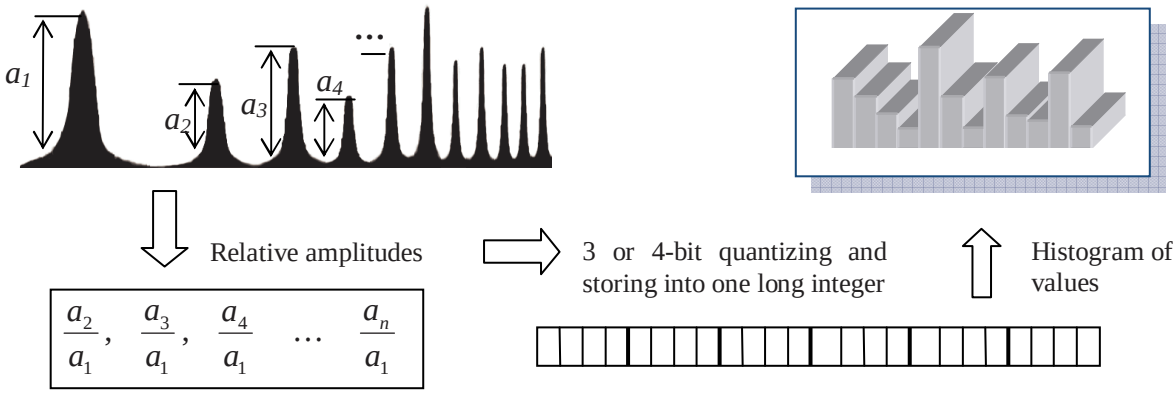


Figure 5.12. Computing of timbre histogram.

Unlike the classical approach which consists of describing the timbre globally, our timbre features are first defined based on separate notes isolated by the multiple-f_0 estimation algorithm and then they are summarized in a histogram. Comparing such histograms gives a similarity measurement, which is supposed to be somehow instrument-related.

5.4. Conclusion

In this chapter we have presented a VRT-based multiple-f_0 estimation algorithm characterized by its simplicity, rapidity and high temporal resolution as opposed to the FFT-based methods. It performs pretty well in the detection of multiple pitches with non-integer rates. However, as other similar algorithms, our VRT-based multiple f_0 estimation algorithm does not solve the following problem: two notes with a distance of an octave can hardly be separated, because the second note does not bring any new harmonics into the spectrum, but rather changes the amplitude of existing harmonics of the lower note, so some knowledge of the instruments involved in the play or instrument recognition techniques and multi-channel source separation is necessary to resolve the problem.

Our note detection mechanism was evaluated in its direct application – musical transcription from the signal. In this evaluation ground truth data was taken as note score files – MIDI. These files from various genres (mostly classical) were rendered into waves using high-quality wavetable synthesis. The resulting wave files were passed as input for the transcriptions algorithm. The results of the transcription and the ground-truth data were compared and a performance measure was calculated, producing results that are reliable enough to be suitable for use of this partial transcription in similarity metrics.

Several musical features and corresponding similarity measures have been proposed in this chapter. Some of them were already known in literature (the pitch class or note profile), some of them are newly presented in this thesis (the note succession histogram, the timbre histogram). All these music features are closely related to musical content. As we will see in the next chapter on music genre classification and music retrieval by similarity, they are precious as complementary features for previous purely spectral or timbre similarity features developed so far in the literature.

Chapter VI

Applications and evaluation

6. Applications and evaluation

From the start, our basic assumption was that music signal analysis within MIR framework needs to go beyond pure acoustic or spectral features, as was also suggested by many works in the literature [AUCO 04]. Consideration of some music-specific features in addition to popular acoustic and spectral ones can improve the performance of music information retrieval algorithms, in particular music genre classification and music retrieval by similarity.

In our work we consider the task of automatic genre classification *as a tool of an objective evaluation* of our previously defined music features and the associated similarity measures.

While evaluation of the first application is rather clear, the evaluation of the second application presents a certain difficulty. The reason is the absence of objective ground truth data since the question of musical similarity is inherently a subjective one. Of course, the ground truth data can be found in some extent in the meaning of genre classification or artist identification. In this chapter we will discuss about it in details.

6.1. Automatic genre classification

Music genres are categories used by music editors and distributors. These categories facilitate the navigation in a large music collection. Genres are naturally used to categorize digital music collections. To date, genres are manually assigned by music editors, distributors and aggregators. Assigning genres to music titles automatically is becoming important for several reasons. The main reason is that, in the digital era the music unit is the title in contrary to the album in the physical era, meaning a clear increase in the number of units to be categorized.

6.1.1. The problem

The main practical difficulty in the problem of automatic genre classification is the absence of reliable ground-truth data. Definitions of genres are often ambiguous and unclear. Moreover, even if genres are well defined, their borders are still a question to discuss. There exist many musical pieces with transitional genres sharing multiple classes.

It is known that even a human-based genre classification does not give 100% of classification rates is known. The work [PERR 99] has provided 72% of classification that agrees with genres among 10 attributed by record companies. Another study [LIPP 04] has found inter-participant genre agreement rates of only 76%. With these facts one can assume the existence of an unavoidable ceiling on automatic genre classification performance.

Another fact which makes ground-truth data for automatic genre classification less reliable is that genres are often attributed to artists or albums while in general artists have songs belonging to different genre even within the same album. An example is easy to find. Quite a large number of *Rock* (*Metal*) albums can contain calm and romantic compositions which can be closely related to *Ballad* or *Power Ballad* to be exact. From listener's point of view these songs are different from the original artist's genre and the absence of detailed per-song genre attribution is sometimes confusing. From the point of view of automatic genre classification algorithms such exception songs may produce big disturbances in classification models since these songs will generate descriptors significantly different from those of the principal attributed genre.

The aforementioned facts along with little observed progress during last years prevent the task of automatic genre classification from being given a precise definition. Hence the utility of the automatic genre classification techniques can be disputed. However, there are works in literature which provide arguments in favor of using the genre concept (see e.g. [MCKA 06]).

The automatic genre classification may be useful with re-defining the genres as sub-clusters of some low-level meta-genres which are closely related to *styles* of music instead of genres. In this case the genres would be presumably better defined form the viewpoint of acoustics or perception. At the same time musical files from the same genre would produce more homogeneous audio descriptors. So, this will increase classification rates as well. " ut this would also require detailed manual labeling of every song with the new genre taxonomy, which can be a time consuming procedure, especially in the case of large music collections.

As we stated at the beginning of this chapter, we consider the task of automatic genre classification *as a tool of an objective* (to some extent) *evaluation* of our previously defined music features and the associated similarity measures. The goal here is thus to show that our previously defined music features and the associated similarity measures bring about a performance gain when they are used in addition to purely acoustic or spectral based features, even in the case of simple classifiers.

6.1.2. Related work

Several systems for automatic genre classification have been proposed. In their majority, these systems are an adaptation of a general audio classifier to the task of music genre classification. They use a signal-only analysis.

[TZAN 02] uses frequency centroid, spectral flux, zero crossing rate, cepstral characteristics as well as characteristics of musical rhythm and other aspects. Proposed features are classified by GMM classifier. For six musical

genres: Classic, Country, Disco, Hip Hop, Jazz and Rock the results in terms of average classification precision were up to 62% for 30-second segments.

[DAN 02] proposes to use peak valleys of spectral characteristics coupled with GMM for genre classification. For five genres of music (baroque, romantic, pop songs, jazz, and rock) the precision of classification up to 82% was reported for 10-second segments of their proprietary database constructed from 1500 pieces.

A classical approach of genre classification, based on cepstral features (MFCC and delta MFCC) and GMM classifier was used in [PYE 00]. The author reports a precision of 92% in the classification of entire songs (musical titles) of the following genres: ”lues, Easy Listening, Classic, Opera, Techno and Indy Rock.

An original approach of temporal structure modeling of musical signals using neural networks is introduced in [SOLT 98]. This approach was evaluated in genre classification of four genres: Rock, Pop, Techno and Classic. The authors provide an average precision of 70% for 4-second segments.

Recent approaches presented in literature use spectral features such as MFCC, ZCR etc. together with support vector machines [MAN 05] and Ada”oost methods [”ERG 06]. They reported nearly 64% and 69.5% of normalized raw classification accuracy respectively on *Magnatune* database at MIREX2005 Genre Classification Contest.

Direct comparison of algorithms found in the literature is quite a complicated question since they were often evaluated on **different databases with different genres**. For example, similar approaches used respectively in [TZAN 02] and [PYE 00] have produced precision rates as different 64% versus 92%. In addition, the length of segments used for classification also has an influence on the results. It is very probable, for example, that one song could be correctly classified by major vote over all segments of the song when only 30% of its duration is classified correctly.

6.1.3. Principle and architecture of our classification system

The idea we present here is to apply musical similarity measures described in the previous chapters to the automatic genre classification problem. It is supposed that songs from the same genre should induce resemblances in musical and acoustic properties. Specifically, songs from the same genre must sound similarly to some extent (timbre, rhythm, melody, etc.). This assumption forms the basis of our genre classification system.

In our work, two kinds of classifiers are considered. They are described in the following two sections. Each classifier (expert) is first investigated using some specific music feature and correlated similarity measure

previously defined. This respectively leads to four different experts, namely acoustic expert, rhythmic expert, timbre expert and tonality expert. Starting from the assumption that each expert gets some insight into the genre according to its specific feature, we also investigate a multi-expert classification system which further synthesizes these individual expert decisions into a global classification decision, thus making use of all acoustic and music features developed, so far.

6.1.3.1 Single-classifier system

The single classifier version of our genre classification system uses classical architecture where a universal supervised classifier which is trained on learning partition of the dataset produces n outputs equivalent to the probabilities of classes.

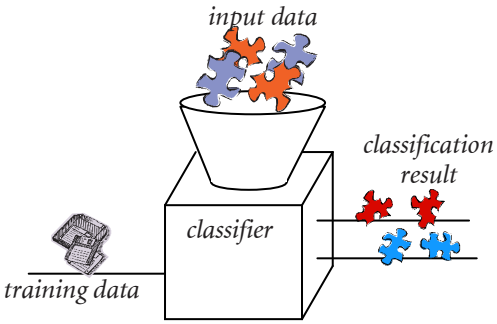


Figure 6.1. General principle of classification

For our system we have chosen the k-NN (k-Nearest Neighbors) classifier [AHA 91]. K-Nearest Neighbor classification is a very simple, yet powerful classification method. The main idea behind the k-NN is that similar observations belong to similar classes. Thus, one simply has to look for the class designators of a certain number (k) of the nearest neighbors for an unknown sample and weigh the values their classes' appearance in order to assign a class to the unknown sample (Figure 6.2).

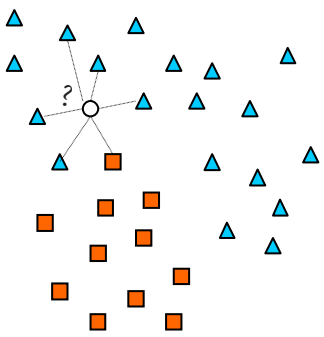


Figure 6.2. The k-NN classification principle.

The choice of naïve k-NN classifier is motivated by its simplicity. ”esides, our subsequent multi-expert system requires a class probability made by each individual expert that each music excerpt belongs to one of six genres, according to its specific expertise. Alternatively, popular classifiers such as neural networks with an appropriate architecture or SVM may also be used.

Of course, we are aware that a major drawback of k-NN classifiers (as a downside to its ability to operate without the need for a separate learning procedure) is its heavy computational cost during classification because it requires all the available learning dataset. Alternative classification strategies may trade off learning complexity against computational demands during classification.

Since the k-NN classifier does not need any special training, for each song to be classified it looks through the training data set, counts a distribution of classes of k closest samples and attributes the resulting class according to the most representative class of the closest samples.

Here is the list of single classifiers:

• Rhythmic classifier or *rhythmic expert* is a k-NN classifier which uses 2D beat histograms of musical pieces as feature vector and (4.10) as similarity measure.

• *Timbre expert* is a k-NN classifier fed by timbre histograms (§5.3.3) as feature vectors. Minkosvki-form distance (4.1) is used in place of a distance measure.

• *Tonality expert* is much like the other experts, taking note profile histograms (§5.3.1) as feature vectors and wrapped-around Minkovski-form distance using the principle described by Figure 5.9.

• The *acoustic classifier* we used in our work is PGM-MLP (Piecewise Gaussian Model – Multi Layer Perceptron) audio classifier [HAR” 05]. PGM models the context effect by an auditory memory, supposing that the classification of a stimulus at the time instant t is based on the status of the memory at the time instant t. The auditory memory is supposed to be a Gaussian distribution of the spectrum in the past time window, called the Integration Time Window ITW. The auditory memory is therefore modeled by one Gaussian distribution for each frequency band. The frequency bands are chosen according to the MEL psychoacoustic scale in order to model the frequency resolution of the human ear. The auditory memory model is updated continuously by a new acoustic observation and hence by new spectral features. For the sake of simplicity it was supposed that the duration of the memory model is constant, which means that the ITW is constant. Also, the Gaussian distributions are described by their mean and diagonal covariance. The PGM features are then coupled to a Multi Layer Perceptron

(MLP) trained using the error back-propagation algorithm. The MLP has 40 input neurons corresponding to the 20 mean values and the 20 variance values obtained from each ITW window. The MLP has 1 hidden layer with 100 hidden nodes, 6 output neurons according to number of genres. The MLP estimates, after a training phase using the gradient descent algorithm, the probability of the audio classes given the PGM features of the ITW window.

6.1.3.2 Multi-expert classification system

As our previous single classifier makes use of different music features for music genre classification, we further consider the conjunction of these different music features by fusing the outputs of these single classifiers into a final classifier, thus leading to a *multi-expert* system. The intuition behind such architecture is that each single classifier uses its specific music feature and is more suitable for characterizing some music genres according to the feature chosen. A final classification result is needed to combine these very complementary single expert decisions.

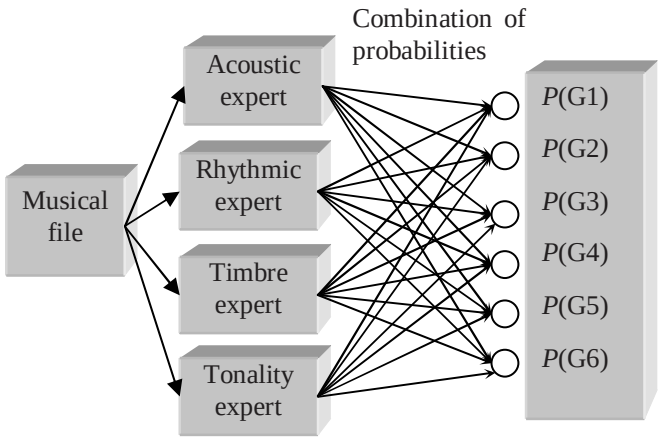


Figure 6.3. Multi-expert architecture of the classification system. Individual experts issue genre probabilities which are then mixed using the final combining expert.

The usage of a mixture of experts is not entirely new. The properties of expert mixtures have been investigated in [JORD 94]. They have been successfully applied in a general audio classification system in Hadi Harb's thesis work [HAR" 03; HAR" 05]. Multi-expert systems have shown good results in music genre classification (see e.g. [SCAR 05a]). They are known to divide the problem into sub-problems with reduced complexity, which improves the global accuracy.

One of the experts constituting our multi-expert classification system is the expert of classification by acoustic analysis (the first expert on Figure 6.3). This expert (a classifier presented in [HAR" 05]) is one or several Multi Layer Perceptrons (MLPs) having the perceptually motivated PGM (Piecewise Gaussian Model) features as input characteristics. Each MLP of

this expert is trained independently on the entire learning dataset or on a part thereof.

All the other individual experts are naïve k-NN classifiers used in single-expert version of the system where they were applied on similarity measures like rhythmic, melodic (note profile and note succession histograms) and similarity by timbre. Since the k-NN classifier produces one class designator as a classification result, is was adapted to produce multiple probabilistic outputs standing for probabilities of a song being analyzed to belong to the corresponding class.

$$P_i = \frac{C_i}{\sum_{j=1}^{N} C_j} \tag{6.1}$$

In previous works by our team a weighted sum of individual classifiers was used for producing resulting probabilities [HAR" 04]. However, a contribution of each classifier is not necessarily linear. Hence in this work we used a method of combining based on MLP as a generic non-linear classifier with the following architecture. Normalized outputs of individual classifiers form the input for the MLP. Six outputs of the MLP are probabilities that a song belongs to six genres. The MLP is trained on the testing set where target vectors for training are formed with one member value equal to 1 for the genre in question and 0-values for all the rest. The MLP which was used has one hidden layer with 24 neurons. Further increase of the number of neurons in the hidden layer leads to augmenting of the network complexity and requires more training data. It can therefore result in overtraining with bad generalization.

The training algorithm used for the neuron network is back error propagation [RUME 86]. The function of activation was the sigmoid function.

6.1.4. Experimental results

One important difficulty to overcome in the development of an automatic music genre classification system is the constitution of a reference database. A reference database should be sufficiently representative of the real world situation in order to draw reliable conclusions on system architecture, features, classifiers etc. The database must also reflect the real needs in real world applications, especially in the definition of genres and the variability of music excerpts for each genre. In the following, we first introduce and discuss ECL-Music genre dataset which is used as reference database in addition to Magnatune dataset. Then, performances by single classifiers and multi-expert classifier are presented and discussed.

6.1.4.1 Reference database

We have chosen six music genres for the reference database. The genres were chosen to be those generally found on several online music stores. The selected list of genres includes: Rock (Pop), Rap (HipHop, R&"), Jazz (" lues), Classic, Dance (Disco, Electro, House), Metal (Hard Rock, Heavy Metal)

Each of these "general" genres consists of several sub-genres which have a more precise definition. For example, the Rap genre consists of such sub-genres as Rap, HipHop, R&" , Soul etc... Each sub-genre corresponds to a specificity which means that two songs of the given sub-genre are closer to each other (at least from the musical edition's point of view) than two songs from different sub-genres. Unfortunately, a detailed genre taxonomy can be defined in multiple ways [PACH 00], which is a limit for the definition of universal musical genres taxonomy. Hence, we propose to well defined representative sub-genre form each "general" genre. The choice of the most representative sub-genre is made according to the number of songs associated to it by a musical distributor, for instance *fnacmusic*. E.g. Rock → Pop rock, Metal → Hard rock.

For each representative sub-genre we have selected the list of artists associated to it on the music distributor store. This list was then used to capture music from webradios[1]. The musical segments were captured as 20-seconds records starting from the 50th second of the play and saved as PCM 8KHz 16bit Mono files. In total the reference database consists of 1873 titles from 822 artists which make 37480 seconds in total.

It is crucial to note an important variability of musical titles in this reference database owing to a significant number of artists (see Table 6.1). As far as we know, this is the first reference database where the attribution of genres to each title is not made in subjective manner by one person but takes into account the musical distribution's attribution. Also, in comparison with other freely available databases like *magnatune* (see Table 6.2) used in ISMIR 2004 genre classification contest[2], the current reference database is better balanced in the meaning of representation of classes (e.g., there are 640 classic songs vs. 52 jazz songs in the case of *magnatune*).

[1] www.shoutcast.com
[2] http://ismir2004.ismir.net/genre_contest/index.htm

Table 6.1. Database details – number of songs per class, number of various artists per class etc.

Genre	Titles	Artists	Duration (seconds)
Classic	214	113	4280
Dance	335	226	6700
Jazz	305	104	6100
Metal	324	105	6480
Rap	311	152	6220
Rock	384	122	7680
Total	1873	822	37480

For the reason of comparison we also provide some results conducted on the *magnatune* (ISMIR 2004) database which is a known reference database. The *magnatune* database contains 1458 files of 6 genres and 138 artists. The database is heavily unbalanced – there are 12 times less songs in Jazz genre than in Classic genre (see Table 6.2 for details). Better balanced datasets allow better training of classification models. An important issue with the *mangatune* database that needs to be addressed is the genre "World" which is a collection of genres not very well defined in a musicological way and doesn't mention the acoustic definition. *""World" is a common "catch-all" for ethnic/folk music that is not easily classified into another group and can contain such diverse music as Indian tabla and Celtish rock"*.[1]

Table 6.2. *Magnatune* database details.

Genre	Titles	Artists
Classic	640	40
Electronic	230	30
Jazz_Blues	52	5
Metal_Punk	90	8
Pop_Rock	202	36
World	244	19
Total	1458	138

6.1.4.2 Experimental results by single classifiers

In this series of experiments we separately apply single music feature-based classifiers to make genre classification. The aim here is to show the relative performance achieved by an individual music feature and the associated similarity measure with regard to the music genre classification.

[1] http://www.music-ir.org/mirex2005/index.php/Audio_Genre_Classification

In the case of rhythmic and timbre classifiers every musical title in the testing set is compared to all titles in the learning set by the distance between their histograms. The learning set is taken as 50% of the entire database. The classifier used is the k-NN classifier with $k=10$.

First we consider the *Magnatune* reference database for which classification results can be taken form the literature. We got the classification accuracies as follows. In the single classifier methods the two musical descriptors most important in genre classification were rhythmic similarity and timbre similarity. Note profile descriptors applied as the only ones do not distinguish well between genres since tonality cannot be related to a specific genre (we do not provide its classification confusion matrix - its mean classification accuracy was around 28%). For the classification by rhythmic similarity the result obtained was 68.1% of raw classification accuracy, which means that 496 songs from 729 were classified correctly. The result normalized with respect to the probability of classes is 54.6%. The confusion matrix of the classification result is presented in Table 6.3.

Table 6.3. Rhythm expert classification result on *Magnatune* (normalized mean accuracy **54.6%**, raw accuracy **68.1%**)

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**89.7**	0.6	0.6	0.3	3.1	5.6
Electronic	6.1	**56.8**	1.7	3.1	13.1	19.2
Jazz-Blues	11.5	1.9	**30.8**	0	3.8	51.9
Metal	4.4	3.3	0	**46.7**	30.0	15.6
Rock-Pop	10.8	7.4	1.0	13.3	**52.7**	14.8
World	23.0	9.0	0	0.4	16.8	**50.8**

It can be noticed that the recognition rate is relatively high for classical songs. This result is expectable since classical music naturally does not have strong rhythmic structure and has high rhythmic dissimilarity with all the other classes except World. Regarding the class World one can conclude a significant confusion with other classes, especially with Classic and Pop. The results show that this class actually contains different types of music. Among the other classes, Metal and Rock-Pop have high cross-confusion rates as well.

While displaying comparable ISMIR 2004 genre classification contest results, these experiments show that rhythm based music features influence music genre classification, and appear particularly efficient for discriminating classics from the rest of music genres.

The second experiment concerns another musical similarity metric – the timbre distance. A k-NN classifier based on this distance has performed with lower but still correct results. The raw classification accuracy obtained was 52.2% which makes 380 songs from 729 to be classified exactly. The

normalized mean accuracy was 39.6%. The whole confusion matrix is given in Table 6.4.

Table 6.4. Timbre expert result (normalized mean accuracy **39.6%,** raw accuracy **52.2%)**

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**75.6**	1.9	0	2.5	5.0	15.0
Electronic	16.6	**17.0**	0.4	14.0	15.3	36.7
Jazz-Blues	19.2	0	**5.8**	3.8	13.5	57.7
Metal	10.0	2.2	0	**58.9**	18.9	10.0
Rock-Pop	23.2	2.0	0.5	24.1	**30.5**	19.7
World	27.5	9.0	0	5.3	7.4	**49.6**

As we can see from this table, the best distinguished classes are Classic and Metal owing to the characteristic nature of instruments involved. The percentage of recognition of Metal genre is even higher than in the case of classification by rhythmic distance. However, classes where songs are characterized mostly by their rhythmical structure such as Jazz-" lues and Electronic have weak classification rates. These results tend thus to suggest that our timbre feature and associated similarity measure are useful for distinguishing the kinds of music genres clearly characterized by the involved instrumentations such as classic and metal.

Our baseline music genre classification system is the acoustic PGM-MLP-based system [HAR" 05]. When applied to Magnatune dataset, the acoustic PGM-MLP-based system has produced the following classification performance (Table 6.5). The raw classification accuracy was 53.6% with 390 correctly classified songs out of 729. The normalized mean accuracy obtained was 49.6%, which is higher than in the case of the timbre distance-based classification but lower than in the case of the rhythmic similarity classifier. However, the acoustic classifier has outperformed both rhythm and timbre classifiers in the Metal genre.

Table 6.5. Acoustic expert on *Magnatune* (normalized mean accuracy **49.6%,** raw accuracy **53.6%)**

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**69.6**	3.4	6.9	0	3.8	16.3
Electronic	6.6	**33.6**	5.7	11.4	28.4	14.4
Jazz-Blues	11.5	3.8	**53.8**	0	15.4	15.4
Metal	3.3	7.8	6.7	**61.1**	20	1.1
Rock-Pop	9.9	14.3	10.3	11.3	**44.3**	9.9
World	26.6	11.1	7.8	4.1	15.2	**35.2**

It can be observed from these experimental results that each classifier-expert is strong in classifying certain genres and rather weak in classifying the others which can be better classified by other experts. That reinforces

our hypothesis of necessity of multi-expert system and fits the assumption that non-linear combination of classifiers can improve global classification accuracy.

For comparison we also provide here the best and the worst result of the ISMIR2004 genre classification contest[1].

The system of Dan Ellis & ”rian Whitman has produced the following classification accuracy: raw accuracy of 64% and normalized 51.48% (Table 6.6).

Table 6.6. Dan Ellis & Brian Whitman's system classification result (51.48% mean and 64% raw).

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**97.8**	0.9	0.3	0	0	0.9
Electronic	36.2	**33.**8	0.8	4	15.3	9.6
Jazz-Blues	18	1.6	**21.3**	8.1	31.1	19.6
Metal	2.2	11.1	0	**51.1**	35.5	0
Rock-Pop	19.8	18.8	1.9	7.9	**51.4**	0
World	62.6	9.7	0	0.8	4.8	**21.9**

Another classification system from the ISMIR2004 we reference here is the system of Elias Pampalk ([PAMP 06]). The system has shown very high classification accuracy (84.07% of correctly classified songs and 78.78% of normalized rate). The confusion matrix is given in (Table 6.7).

Table 6.7. Classification accuracy by a reference system (E. Pampalk) (78.78% normalized accuracy and 84.07% raw).

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**97.8**	0.3	0	0	0.6	1.2
Electronic	0.7	**62.2**	0	16.2	14.8	5.9
Jazz-Blues	7.6	3.8	**80.7**	0	3.8	3.8
Metal	0	4.4	0	**75.5**	20	0
Rock-Pop	2.9	9.9	0	4.9	**77.2**	4.9
World	14.6	6.5	0.8	0	9.7	**68.2**

The ECL database described in §6.1.4.1 was also examined. For this database the obtained classification rates were higher than in the case of the Magnatune database. It can be explained by the better quality of the ECL database where all classes are nearly equal in terms of representation and better defined (no such classes as World).

[1] http://ismir2004.ismir.net/genre_contest/results.htm

Both rhythmic and timbre k-NN classifiers were applied and the following results were observed. Table 6.8 gives the class recognition confusion matrix for the rhythmic classifier.

Table 6.8. *ECL_genres* database classification results with rhythmic classifier (normalized mean accuracy 71.4%)

	Classic	Dance	Jazz	Metal	Rap	Rock
Classic	**82.8**	0.5	4.4	5.4	1.0	5.9
Dance	0.3	**76.8**	2.1	4.6	7.9	8.2
Jazz	5.5	3.8	**68.9**	1.4	10.0	10.4
Metal	8.3	3.5	8.0	**69.7**	2.2	8.3
Rap	0.0	6.0	10.7	3.3	**75.7**	4.3
Rock	7.4	3.6	15.3	8.2	11.2	**54.4**

As it was noticed before, the rhythmic classifier is stronger in classifying rhythmically bright genres of music such as Dance, Rap. Classic genre also belongs to this category since it doesn't have strong rhythmic patterns. An absence of strong rhythm is a rhythmic descriptor as well.

Table 6.9. *ECL_genres* database classification results by timbre (normalized mean accuracy 42.4%)

	Classic	Dance	Jazz	Metal	Rap	Rock
Classic	**49.3**	3	13.8	17.7	4.4	11.8
Dance	7.9	**19.8**	5.8	30.8	7.6	28.0
Jazz	8.0	11.4	**31.5**	14.9	9.0	25.3
Metal	5.1	3.2	1.6	**85.7**	0.3	4.1
Rap	5.3	12.0	15.0	23.0	**28.3**	16.3
Rock	12.3	7.9	6.0	30.6	3.6	**39.6**

Table 6.10. *ECL_genres* database classification results by acoustic expert (normalized mean accuracy 49.3%)

	Classic	Dance	Jazz	Metal	Rap	Rock
Classic	**53**	5	12	2	5	10
Dance	3	**40**	7	7	11	8
Jazz	23	4	**38**	2	6	21
Metal	7	24	15	**75**	16	19
Rap	2	16	12	7	**55**	7
Rock	12	11	16	7	7	**35**

The classification by timbre and by acoustic classifier again shows a superiority of Metal recognition rate owing to a specific instrumentation used.

6.1.4.3 Experimental results by Multi-expert system

As explained in the section on architecture of classifiers, our multi-expert system aims at fusing single music feature based-experts into a global

one for music genre classification. As the goal of this thesis work is the use of music features in a complementary way to purely acoustic features, we have also included a pure acoustic feature based classifier as one of our single classifiers as a baseline music genre classifier. Recall also that the fusion strategy is a Multi-Layer Perceptron having one hidden layer which synthesizes a global classification result from the outputs of single classifiers. The following Table 6.11 shows the result obtained for combinations of all experts on the Magnatune dataset.

Table 6.11. All experts combined by MLP, (normalized mean accuracy 66.9%, raw accuracy 74.2%)

	Classic	Electronic	Jazz-Blues	Metal	Rock-Pop	World
Classic	**88.7**	0.6	0	0.6	1.2	8.9
Electronic	3.5	**58.8**	9.6	3.5	7.9	16.7
Jazz-Blues	7.7	3.8	**57.7**	0	11.5	19.2
Metal	0	8.9	0	**66.7**	22.2	2.2
Rock-Pop	1	11.8	2	10.9	**64.7**	9.8
World	13.9	8.2	2.5	0.8	9.83	**64.6**

As it can be seen from the table, normalized mean accuracy rate is up to 66,9%. Thus the combination of experts that uses music features in addition to purely acoustic feature based PGM-MLP expert, brings a significant improvement of classification precision as compared to the normalized mean accuracy rate of 49,6% achieved by the single PGM-MLP expert. Our multi-expert system outperforms the single acoustic feature based PGM-MLP expert in each of considered six genres. The best improvements were achieved on Electronic and World genres, changing from an accuracy rate of 33,6% and 35,2% for single PGM-MLP to 58,8% and 64,6% for multi-expert system, respectively. These two music genres presumably need more music features for a better discrimination due to their varieties.

With the ECL Music genres dataset, which has a higher artist variability and generally better defined genres, our multi-expert system achieves even better results. Table 6.12 gives the classification results by our Multi-expert system. Indeed, our multi-expert system displays a normalized mean classification accuracy rate up to 80.9% as compared to 49.3% achieved by the single PGM-MLP expert.

Table 6.12. *ECL_genres* database classification results (normalized mean and raw accuracy 80.9%)

	Classic	Dance	Jazz	Metal	Rap	Rock
Classic	**91.6**	0	2	0	0	6.2
Dance	0	**69.2**	1	3.2	6.5	19.7
Jazz	1.1	8	**71.2**	0	3.4	16
Metal	0	1	0	**89.2**	2.1	7.5
Rap	0	4.6	2.3	4.6	**80.2**	8.1
Rock	1.7	4.4	2.6	5.3	1.7	**83.9**

Experimental results and final comparison can be summarized by the following figures. Figure 6.4 depicts the comparison of classification accuracies of separate classifiers and their mult-expert combination for both databases.

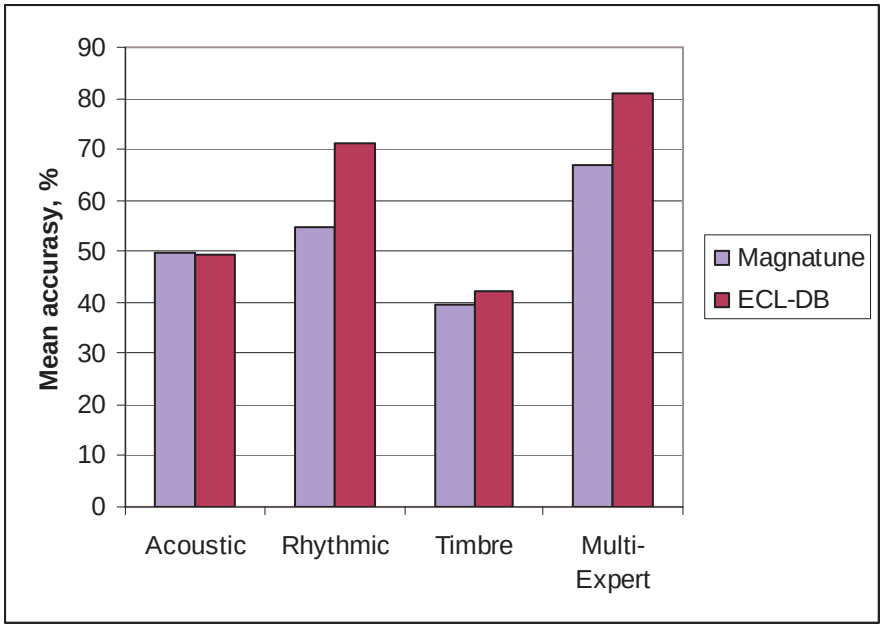


Figure 6.4. Comparison of classification results issued by different classifiers and their multi-expert combination for both databases.

All classifiers behave quite similarly in the case of both databases except the rhythmic classifier which performed better on the ECL database. In both cases there is a significant increase of classification rates with combined experts.

The next two figures (Figure 6.5 and Figure 6.6) show the performances of separate classifiers and their combinations for both databases according to genre.

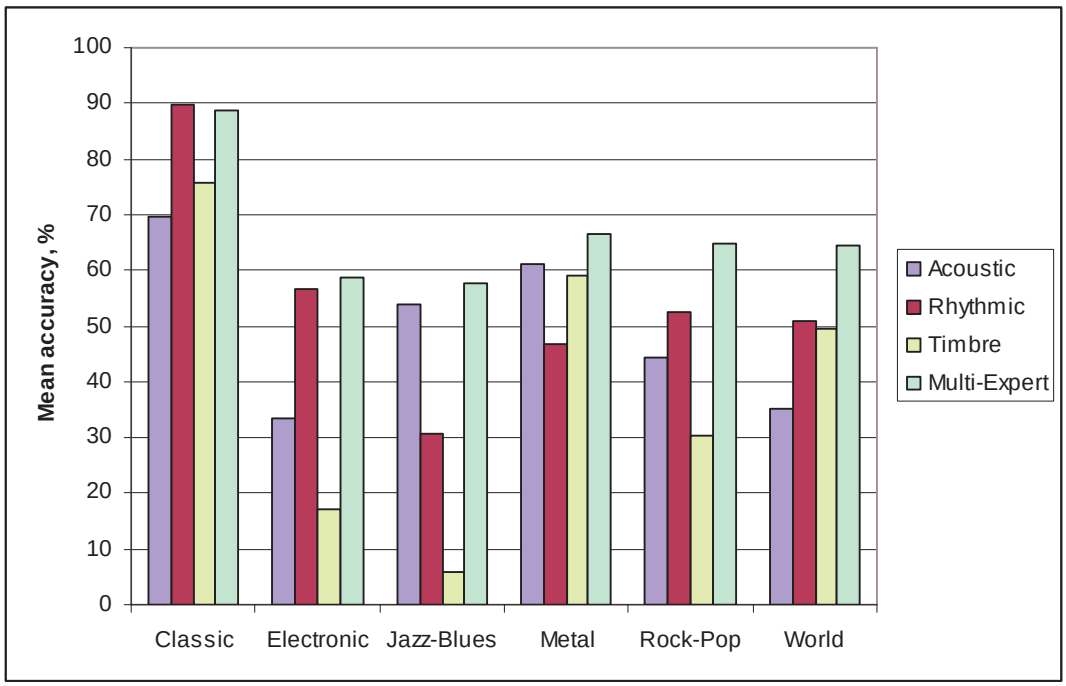


Figure 6.5. Performance of separate classifiers and their combination according to genre in the case of Magnatune database.

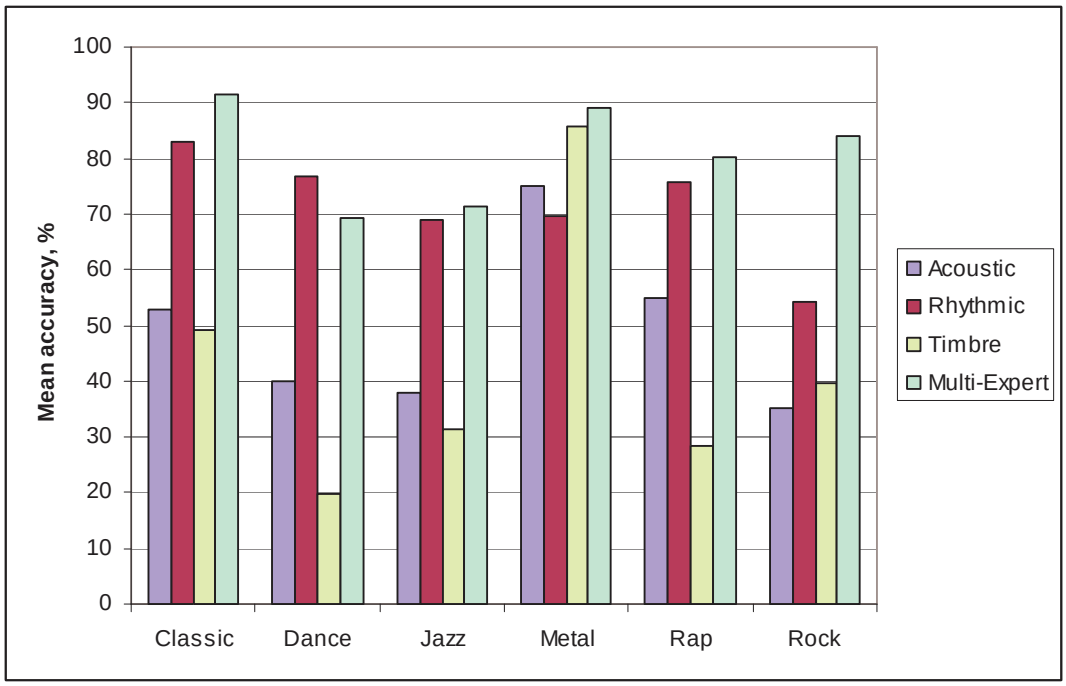


Figure 6.6. Performance of separate classifiers and their combination according to genre in the case of ECL database.

Generally a superiority of Multi-Expert classification results is observed for the majority of classes except just two cases – Classic of Magnatune database and Dance of ECL database. It can be explained by a tendency of the Multi-Expert configuration to average per-class accuracies together with a high Dance-to-Rock confusion in the latter case.

6.1.4.4 Discussion

In the previous experiments, some parameters need to be tuned in order to achieve the best genre classification accuracy. Using the ECL music genres dataset and the rhythmic similarity measure, we studied the impact of two particular parameters on the classification accuracy, namely k in the k-NN classifier and the window moving step in beat detection algorithm. Figure 6.7 illustrates the classification accuracy curve using different k values. As we can see, the best classification accuracy was achieved with k=9.

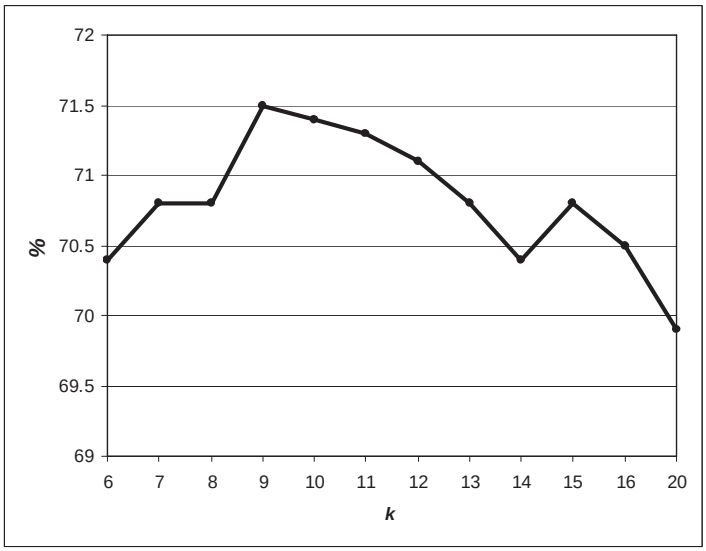


Figure 6.7. Dependency of the classification accuracy on the cluster size in the k-NN.

On the other hand, when studying the size of the shifting window, we can see from Figure 6.8 that the best classification rate was obtained for the shift equal to 15 ms. Further increasing the shift size almost linearly decreases the classification performance.

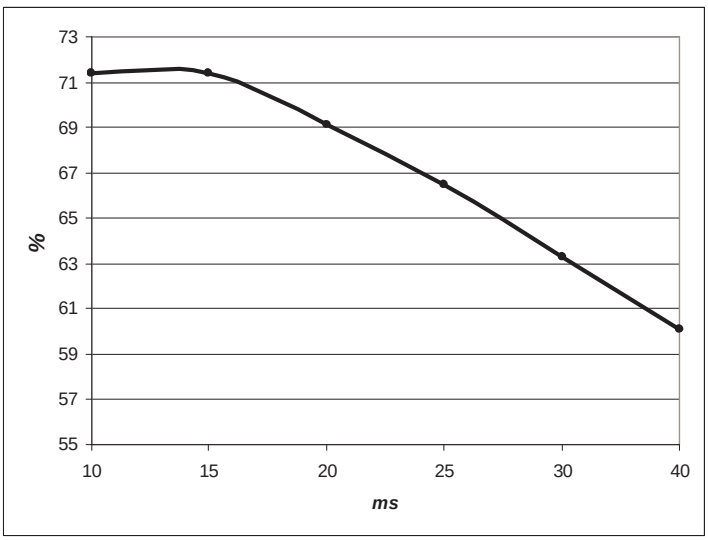


Figure 6.8. Dependency of the classification accuracy on the window shift in the beat detection algorithm.

However, these experimental results can be considered as very primary ones showing the utility of music based features in music genre classification in addition to purely acoustic based features. Several directions need to be further investigated. Indeed, while simple k-NN classifiers are used in these experiments, one can also consider some other popular classifier such as SVM, neural networks and Bayesian classifiers. These may display better classification accuracy when using an appropriate learning scheme. Moreover, we also need to deepen our study with respect to the fusion strategy. In the previous experiments, the outputs of single classifiers are fused by a neural network into a global classifier. However, other fusion strategies might give better results.

6.2. Music search by similarity

Search for music by similarity is an interesting but difficult direction of the music information retrieval research. It is at the heart of such applications as intelligent music navigation automatic, playlist composition, music recommendation and others.

6.2.1. The problem

Unfortunately, music similarity is rather subjective and personal according to cultural background. Precise definition of music similarity is thus impossible. Unlike well-defined task with ground truth data, such as, e.g., automatic classification of audio signal, the task of intelligent playlist composition by musical similarity lacks the reliable ground truth – construction of 100% ground-truth data cannot be possible.

However, several attempts on evaluating the musical similarity algorithms have been made. In a work on artist similarity [ELLI 02] the authors have constructed ground-truth data by artist survey. Visitors of their web site were proposed to find similar artist for a given one. The authors in another work [PAMP 03] have used statistical information. They computed an average distance between all musical pieces and an average distance in groups (artist, genre, album, etc.). The ratio between these two average distances was calculated. In that way a large-scale evaluation was performed without involving listening tests. MIREX evaluation contest[1] uses listening test together with statistical information analysis in order to evaluate various musical similarity retrieval algorithms.

In our work we based our evaluation on listening tests and analysis of statistical information. Reinterpreted compositions search was also performed and used as an objective evaluation.

---

[1] http://www.music-ir.org/mirex2006/index.php/Audio_Music_Similarity_and_Retrieval

6.2.2. Principle and architecture of our combination system of similarity measures

While pure similarity metrics could be interesting for exact matching of musical pieces by certain criteria, a combination of them have in goal of building "general" similarity metric like a human listener could do (e.g. finding a piece with the same rhythm and key type could issue two slow sad melodies which are judged similar by a human listener).

In the previous problem of genre classification tests, the final combination was done by a multi-expert system where the combining system was operating on class probabilities. In the case of similarity search, distances must be combined in order to get the overall similarity score including various aspects.

- **Fusing similarities by a Linear combination**

To combine 4 distances (rhythmic, tonality, timbre and note succession) obtained with algorithms described in Chapters 4 and 5, we use a liner combination which as used in many other works.

$$D = \sum k_i d_i \qquad \textbf{(6.2)}$$

Alternatively, when source distances have "incongruous" physical nature, another version of liner combining could be applied – namely, a weighted sum of ratings where a rating or position in a sorted list of similar titles is obtained for every kind of similarity being combined. Final distance is computed as a weighted sum of such ratings. In our experiments, these two kinds of linear combination are studied and compared.

- **Fusing Similarities by a Multi-layer perceptron**

These methods are non-linear methods that involve a neuron network trained on user feedback data. The choice of MLP is motivated by its relative simplicity and ability to approximate multi-variable functions. In our case the target function is a function of 4 distances described in previous paragraphs between any couple of music excerpts and the target values are similarity ratings (from 0 to 1) provided by listeners.

The MLP used in our work had 4 input neurons (the number of input variables) and 4 neurons in hidden layer. Training and testing sets consisted of approximately 800 listeners' votes. Training of the MLP was performed in two ways:

1) direct training with target values of listeners' votes;
2) threshold training when listeners' votes on similarity were divided into two category – similar (>=0.5) and dissimilar (<0.5), the target values for the MLP being 1 and 0 respectively.

In both cases the difference between classification results on the testing set was not statistically significant. The total error rate in both cases was around 20%. Further increase of number of neurons in the hidden layer produced instabilities in the classification of the testing set and resulted in an increase of classification errors.

The disadvantage of the MLP combining is its observed irregularity. Songs can hardly be sorted upon similarity measure issued by the neural network having a high error rate. In order to overcome this problem we propose to use a multi-cascade system where the linear similarity measure is controlled by MLP. The principle is the following: similar songs are searched using linear combination of distances and the final list is filtered by an MLP, which drops all non-similar songs according to its decision. In this case thresholds of the decision-making must be adjusted to have minimal false rejection rate. The principle of such architecture is depicted in Figure 6.9.

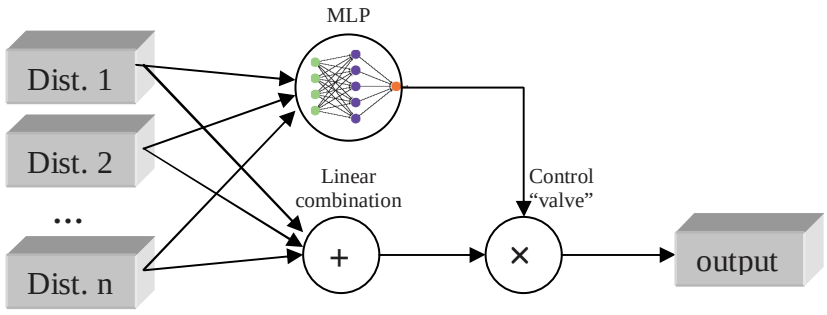


Figure 6.9. Architecture of the combining system with linear distance combination controlled by a neural network.

6.2.3. Experimental results

6.2.3.1 Evaluation method

We carried out preliminary experiments on musical similarity search. They consisted of three evaluation parts – listening test, statistical analysis and reinterpreted pieces search.

A database of approximately 1000 musical compositions of different artists, genres and rhythms has been collected. It was taken from private collections of music and contained musical pieces of various genres and different languages. The crucial point is to ensure existence of various close groups of music within the dataset. A small database containing musical pieces which are too different may result in unreliable evaluation results due to the absence of similar music excerpts. The Magnatune database could be used as a suitable one since it contains small variation of genres.

The evaluation proceeds as follows. For each type of similarity metric (rhythmic, tonality, timbre and melodic) a similarity matrix 1000x1000 had

been computed. Then the system retrieved 5 most similar songs from the database for a given example by different combinations of similarity metrics. People from our laboratory and third-party people (men and women, not necessarily working with music) were asked to act as listeners. A dedicated website was build which enables listeners to rate queries from the database with scores from 0 (not similar) to 5 (very similar) according to similarity type shown (for example, in the case of similarity by tonality, the listeners had to indicate if tonalities (tonality modes) of proposed songs were sounding similar or not). Neither songs' titles nor artist names were revealed to the listeners. Also with a probability of 50% the listeners were provided with randomly selected not similar music pieces and the listeners were not aware of this in order to avoid prejudice. The listeners had to rate the similarity for those random propositions as if they were retrieved by a true similarity algorithm. Further comparison of vote distributions of both similarity-based and random selections allows to rate the quality of search algorithms. The result is supposed to be reliable since such evaluation technique cannot be "tricked".

In our experiments we have used 4 pure similarity metrics: rhythmic, tonality, timbre and melodic; and 4 mixtures where ***comb1*** was a combination of tonality and rhythm metrics, ***comb2*** was timbre and rhythm combination, ***comb3*** was tonality + melody + rhythm respectively, ***comb4*** – timbre + melody + rhythm. For the mentioned mixtures two types of combinations were applied, namely, linear combination and combination by rating. We didn't use a linear combination of all 4 distances because it is quite difficult to say in advance if mixing coefficients should be equal for all distances or not since these 4 distance have different nature. In the case of Multi-layer perceptron combination all 4 similarity measures were used.

While the previous experiment is rather a subjective evaluation, the following ones are objective evaluations. Our second evaluation is reinterpreted pieces search. The aim of this experiment was the evaluation of melodic similarity measures. The test was based on composing of similarity playlists for musical titles that have multiple reinterpretations. For this purpose we have injected several musical pieces with multiple representations. The following reinterpreted pieces were included into the dataset.

1. Ennio Morricone – "Chi Mai", 3 interpretations
2. Roxette – "Listen to Your Heart", DHT – "Listen to Your Heart", DHT – "Listen to Your Heart" (dance)
3. Rednex – "Wish You Were Here", Blackmore's Night – "Wish You Were Here"
4. Tatu – "Not Gonna Get Us" (Eng), Tatu – "Nas Ne Dogonyat" (Rus)
5. Tatu – "All the Things She Said" (Eng), Tatu – "Ya Soshla s Uma" (Rus), Tatu – Remix
6. Tatu – "30 minutes" (Eng), Tatu – "Pol Chasa" (Rus)

7. Archie Shep, Benny Golson, Dexter Gordon, Mike Nock Trio, Ray Brown Trio – "Cry Me a River" (ver.1 jazz instrumental)
8. Diana Krall, Tania Maria, Linda Ronstadt, Bjork, Etta James, July London – "Cry Me a River" (ver. 2. vocal)

In this experiment the different interpretations of the same title are automatically considered as "similar". Playlists with 30 similar titles corresponding to each musical title from the list above were built. Appearance of our "a priori" similar titles at the top of playlist was considered as successful similarity output.

Finally as a third evaluation of our similarity system, we analyzed a relevance of the top 5 songs in playlists. We considered two types of relevance: the number of songs from the same genre and the number of songs from the same artists. As the database we took the ISMIR2004 genre classification database based on *Magnatune* collection since the musical pieces are classified by genres and there is no high variability in artist. The database contains totally 729 titles of 128 artists in 6 genres.

6.2.3.2 Listening test evaluation

Evaluation results obtained in our listening test experiments are presented in the Table 6.13. For each similarity type the mean and median values of totality of votes are given. The column "corresponding random" shows the mean and median of listeners' votes for those cases when the listeners were proposed random songs rather than similar. Since listeners were not notified about this fact, they still had to evaluate how similar were the proposed songs. These data are used as background "un-truth". All found multiple interpretations of songs were not filtered out and considered as 5 – very similar.

Table 6.13. Listening test results (mean / median).

Similarity type	Linear combination or single	Rating combination	Corresponding random
rhythmic	2.92 / 2	n/a	0.40 / 0
tonality	3.16 / 3		2.41 / 3
timbre	2.16 / 2		0.81 / 0
melodic	2.23 / 2		1.60 / 2
comb1	3.55 / 4	2.06 / 3	0.94 / 1
comb2	2.78 / 3	3.75 / 4	0.97 / 0
comb3	3.85 / 5	1.80 / 1	0.75 / 0
comb4	2.49 / 3	2.26 / 3	1.01 / 0

Figure 6.10 shows normalized distributions of votes for mentioned single similarity metrics. The upper histograms stand for distributions of votes for "random" similar songs.

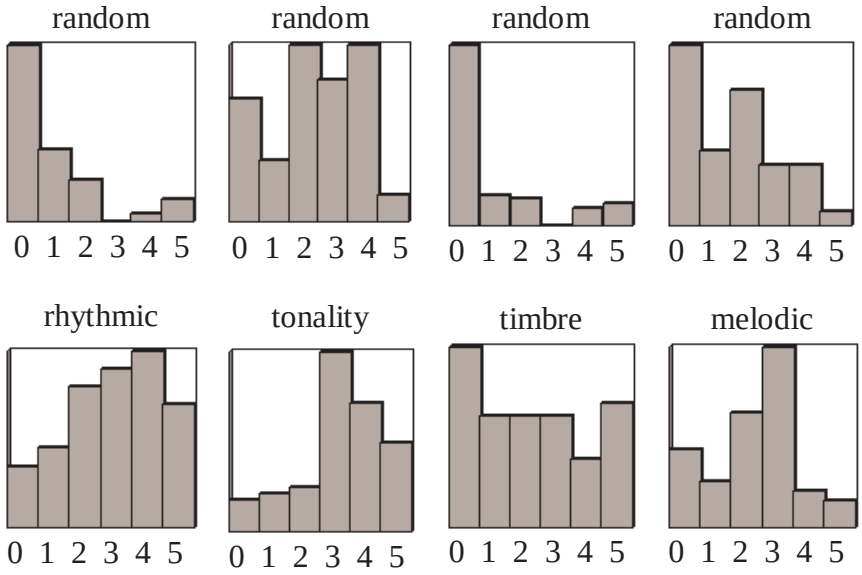


Figure 6.10. Histograms of listeners' votes for pure similarity metrics. The upper row contains histograms of votes for corresponding randoms.

Notice, that in the case of random similarity generation all histograms are different. It means that the notion of similar tonality class is not evident for all listeners. In addition, the mean error in the case of random selection should be around 50% since there are only two modes of tonalities − minor / major. The difference in melodic similarity-selection voting against the corresponding random-selection one is insignificant. This does not allow an immediate judgment of the similarity search performance. On the other hand, some of similarity measures already perform quite well, for example the rhythmic one.

All in all, examining histograms of the true similarity votes one can observe an evident positive bias of average score. It is also evident for the tonality case.

On the next figure the normalized histograms of votes for composite similarities are depicted. Here the upper row is showing histograms of votes for random songs. Two other rows include the results for linear (lin) and rating-based (rt) combinations, as described above. Figure 6.11 shows the vote histograms for MLP combinations. Here ***comb1*** is the combination of tonality and rhythm metrics, ***comb2*** − timbre and rhythm, ***comb3*** − tonality + melody + rhythm, ***comb4*** − timbre + melody + rhythm.

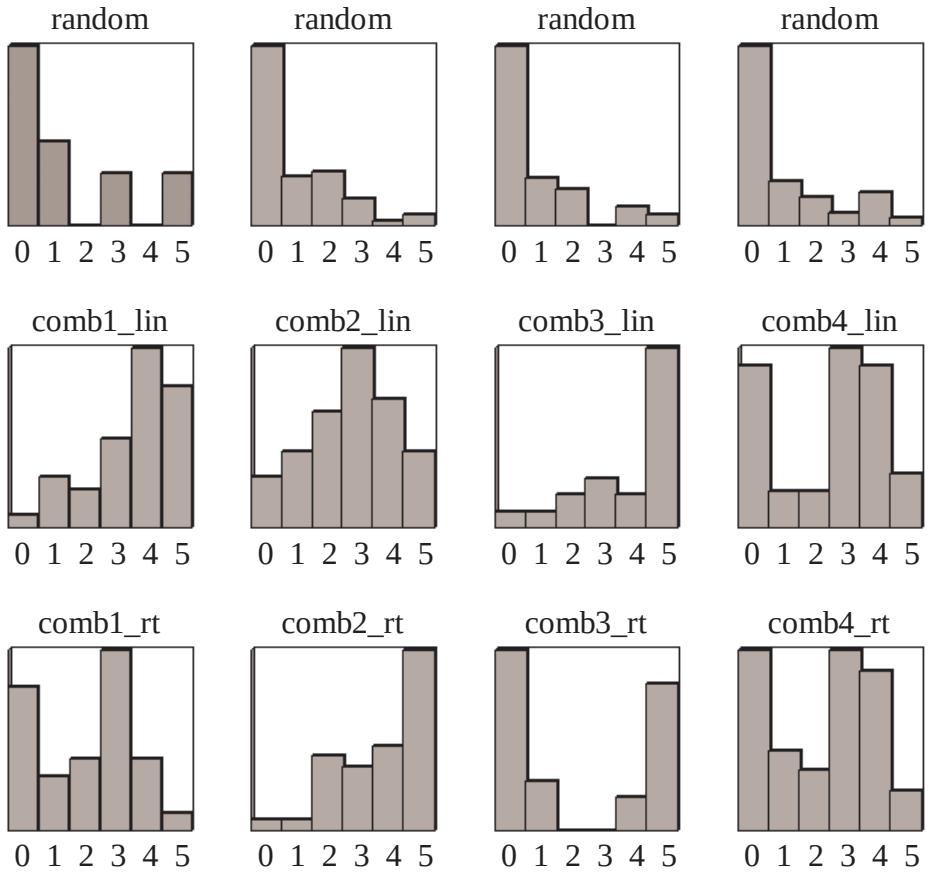


Figure 6.11. Histograms of listeners' votes for linearly and rating combined similarity metrics.

Considering the Figure 6.11 the following conclusion can be made:

- There is a remarkable positive bias of votes in the case of combined (we can call them general or true) similarities, which is in fact higher than for pure similarity measures. This can be seen by comparing comb3_lin versus pure rhythmic similarity rating histograms.

- Certain instability of results according to the type of combination is observed

- The hypothesis that the combination of various similarity types by its rating in generated sub-playlists could be better adapted for measures of different nature did not find confirmation (except the case of comb2 combination)

- In some cases of similarity combination as for instance comb3_rt, comb4_rt, a big number of completely dissimilar (voted as completely unsimilar) songs in resulting playlists were observed in comparison to pure similarity measures where this fact was only present in timbre similarity. This can be an evidence of nonapplicability of linear or rating combinations of various similarity types. For example, two musical pieces close to each other by the tonality or melodic passages may have a certain difference in their

rhythmic structure. Since one of similarity measures issues a very low value of distance, another similarity measure may not bring any significant dissimilarity. So, the pair of songs is considered to be close to each other. However, a human listener may not pay attention to the melodic aspect of similarity, but only to its rhythmic aspect (which is in fact has closer relation with the genre of music). As a result, a socre provided by the listener is considerably low.

- It can be stated, that the best two combinations for this particular dataset are comb3_lin and comb2_rt. The first case is a linear combination of rhythmic, tonal and melodic distances; the second case is a rating combination of rhythmic similarity and similarity by timbre. This can be explained by instability of combination of timbre similarity measure with all other measures.

Two other similarity fusing configurations are also investigated. The first one consists of using an MLP for a non linear combination of similarity measures. The second configuration is the one with the neuron network used as an agent for ensuring quality of similarity measure from the best linear combination Comb3, found to be the best in previous studies. The outputs from the linear combination Comb3, involving tonality, melody and rhythm similarity measures, are further controlled by a MLP discarding dissimilar music excerpts. Table 6.14 shows votes distributions for multi-cascade configuration and a single MPL. As we can see, no large statistical difference in the results was observed for these two configurations.

Table 6.14. Listening test results for MLP combinations.

Combination Type	Score
Comb3 + MLP	3.80 / 4
MLP	3.25 / 4

The decrease of the average similarity search quality in the case of pure MLP can be an evidence of a lack of statistical training data which makes the neuron network to behave in an unstable manner. The second source of instability is the subjectivity of the listener's votes. Having multiple listeners vote to be different for the same titles in the same playlist leads to ambiguities in training data.

Histograms of votes distributions are depicted in Figure 6.12.

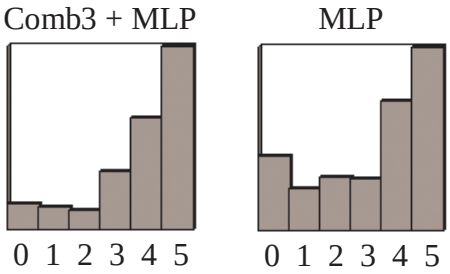


Figure 6.12. Histograms of listeners' votes for MLP combined similarities.

The configuration with a neural network as a quality agent controlling the output of linear combination is more preferable in comparison to the single NN mixing agent.

6.2.3.3 Objective evaluation

In our objective evaluation experiment we have generated 30-title playlists for each musical piece in the database. Appearance of "a priori" similar titles (see §6.2.3.1 for the list) at the top of playlist was considered as successful similarity output. The following Table 6.15 shows the result of playlist composition. It gives information about the position of similar titles in the associated playlist (1 – is the original music file at the top of playlist).

Table 6.15. Objective evaluation results of music comb3_lin similarity measurement.

Original music composition	Positions of appearance of similar titles
Chi Mai	(1), 2, 3
Listen To Your Heart	(1), 3, 12
Wish You Were Here	(1), 2
Not Gonna Get Us	(1), 2
All the Things She Said	(1), 2, 3
30 minutes	(1), 2
Cry Me a River (ver. 1)	(1), 2, 3, 4, 6
Cry Me a River (ver. 2)	(1), 2, 4, 7, 8, n/a (not appeared)

As we can see from the table, all interpretation of songs are taking first positions in corresponding playlists regardless of the fact that some of song groups had quite distant versions like dance versions (Listen To Your Heart), or different artist, language, or instrumentation (e.g. Wish You Were Here, Cry Me a River).

The second part of the objective evaluation consisted of playlist relevance analysis. For that purpose we moved on to analyze the relevance of the top 5 songs in the playlists generated for seed songs. We considered two

types of relevance: the number of songs from the same genre and the number of songs from the same artists. For the database we took ISMIR2004 genre classification database based on *Magnatune* collection. The database contained totally 729 titles of 128 artists in 6 genres.

The obtained results are as follows (Table 6.16).

Table 6.16. Average number of songs in the same genre or from the same artist.

Similarity type	Same genre	Same artist
Comb2_lin	3.58	0.99
Comb2_rt	3.48	0.89
Comb3_lin	3.07	0.86

The next picture (Figure 6.13) depicts distribution histograms of the number of songs in the same genre and from the same artist for the best combination which in this case is comb2_lin.

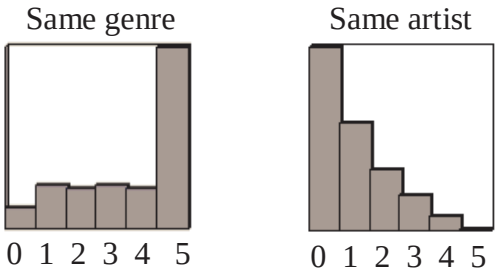


Figure 6.13. Histogram of the number of songs in the same genre in TOP-5 (left), and from the same artist in (TOP-5) (right).

Results of relevance analysis reported in literatures includes such numbers as average 1.43 songs in TOP-5 with the same genre as the query [AUCO 02], average 3.44 of similar genres and 1.17 of similar artist [LOG 01]. A result obtained from the same ISMIR'2004 database found in literature is an average 3.4 songs (67.9%) in TOP-5 with the same genre [POHL 06].

6.2.3.4 MIREX2007 Audio Music Similarity and Retrieval

Having 7000 30-second audio clips drawn from 10 genres (700 clips from each genre) the MIREX evaluation contest[1] has ran the audio music similarity and retrieval evaluation among the other MIR tasks. Two distinct evaluations were performed:

[1] http://www.music-ir.org/mirex2007/index.php/Main_Page

- Human Evaluation
- Objective statistics derived from the resulting lists

The primary evaluation involved subjective judgments of the retrieved sets by human evaluators. Given a search based on randomly selected queries, sets of results returned by all systems were provided to listeners. They had to rate the results by one of three classes (not similar, somewhat similar, very similar) and provide an indication on a continuous scale of 0 - 10 measuring how similar the track is to the query. The songs by the same artist as the query were filtered out of each result list (artist-filtering) to avoid biasing an evaluator's judgment.

The second evaluation was the objective statistics derived from the distance matrix which is:

- Average % of Genre, Artist and Album matches in the top 5, 10, 20 & 50 results - Precision at 5, 10, 20 & 50
- Average % of Genre matches in the top 5, 10, 20 & 50 results after artist filtering of results
- Average % of available Genre, Artist and Album matches in the top 5, 10, 20 & 50 results - Recall at 5, 10, 20 & 50 (just normalizing scores when less than 20 matches for an artist, album or genre are available in the database)
- Always similar - Maximum # times a file was in the top 5, 10, 20 & 50 results
- % File never similar (never in a top 5, 10, 20 & 50 result list)
- % of 'test-able' song triplets where triangular inequality holds

The algorithm we have submitted to the MIREX evaluation was the system, which toke into account only rhythmical distance.

The final subjective judgment results are provided in Figure 6.14 where the system #9 is the system we have presented.

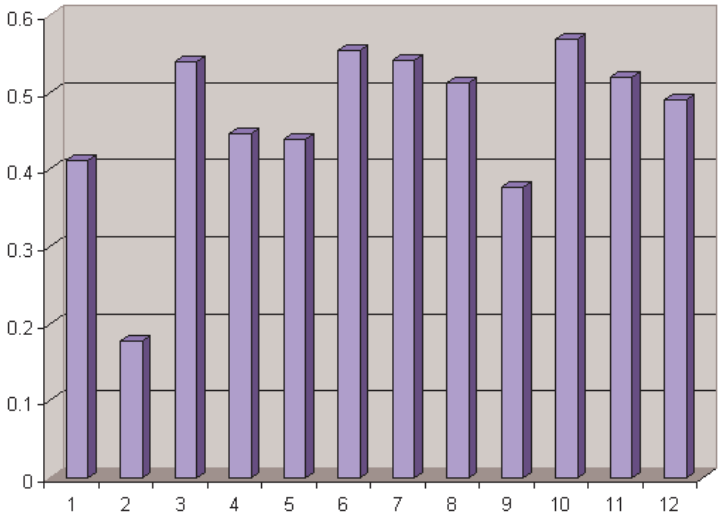


Figure 6.14. MIREX 2007 Audio Music Similarity and Retrieval subjective evaluation results (x-axis – team number, y-axis – average vote divided by 10).

The histogram of listeners' votes expanded to fit 5-level scale as we used in our work is given on the Figure 6.15

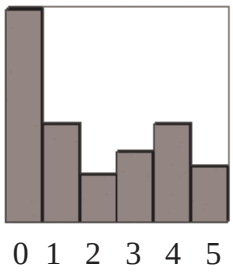


Figure 6.15. Histogram of listeners' votes resulted by our system at MIREX contest.

The experiment shows that rhythmic similarity plays an important role in global music similarity. However, both experiments (see also §6.2.3.2) prove that perceptual similarity is not limited to rhythmic similarity.

Another observation was made from the MIREX results. For certain genres the rhythmic similarity was more important than for other genres. These genres are for example Jazz/" luez and Country. The complete ratings by genre of algorithms presented at MIREX are given on the Figure 6.16.

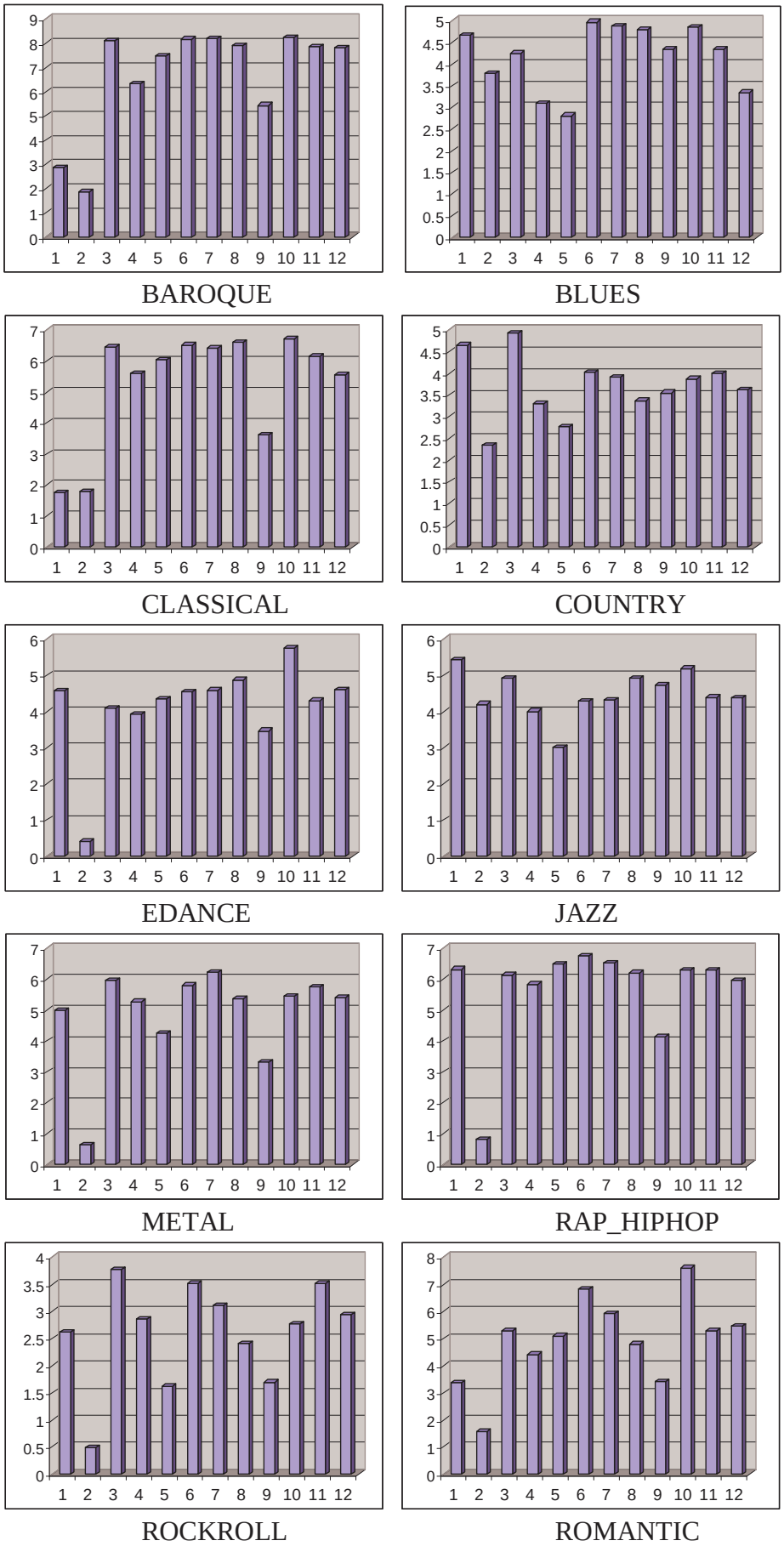


Figure 6.16. MIREX similarity contest result distribution by genres.

We have analyzed the correlation between equality of genres in query vs. retrieved samples and users votes. The average the votes for those queries which returned a piece of the same genre was 6.42 while for non-similar genres the average was 3.55 (Figure 6.17).

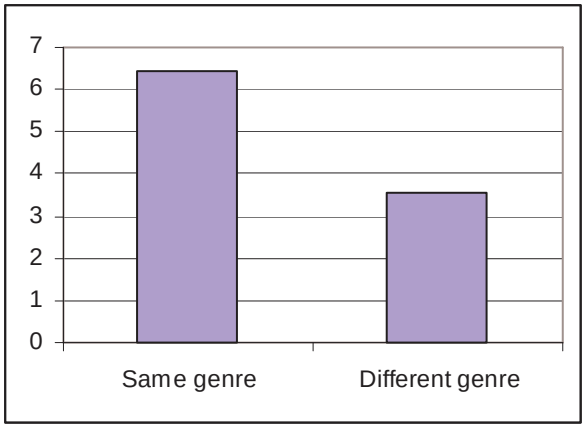


Figure 6.17. Average vote for query and result of the same or of different genres.

According to our observation (6000 queries from all algorithms have been analyzed), the correlation between genre and similarity is quite strong, but the average of votes for non-similar genres in query/result pairs is high enough to conclude that the perceptual similarity is not only limited to the similarity of genre.

6.2.3.5 Discussion

The problem of "hubs" described in numerous works [AUCO 04; PAMP 06a] was also analyzed for the proposed similarity metrics. A hub is a song which appears to be similar to a large number of other songs. In the results obtained by our similarity combination algorithm no extreme hubs were observed. The following Table 6.17 presents the maximum number of appearances of any single song in the TOP-5 rankings of all other songs showing the issue of "hubs" insignificant for our similarity measurements.

Table 6.17. Maximum number of appearances of a song in top-5 rankings

Similarity metric	Rhythmic	Tonality	Timbre	Melodic
Number of appearances	24	19	124	101
Similarity metric	Comb1	Comb2	Comb3	Comb4
Number of appearances	25	41	39	66

The maximum number of song appearance reported at MIREX'06 was 24 to 61 [PAMP 06a]. One irregular result of 1753 appearances was also reported for one of the algorithms indicating the presence of hubs.

6.3. Conclusions

In this chapter we have described two direct applications of music features and the associated similarity measures. The first application consisted of automatically classifying songs by genres. We have shown that single application of acoustic similarity or musical similarity features issues quite low classification accuracies while a combination of both approaches leads to a significant gain of classification performance.

The second application, namely, music search by similarity was based solely on musical similarity features. The evaluation we have carried out consisted of subjective judgment (human feedback) and objective evaluation such as relevance analysis. Objective evaluation showed quite good, but rather unstable results when using linear or rating combination of similarity measure. We have also found the best two combined similarity measures which were combinations of rhythm/tonality/melody and rhythm/timbre. A surprising result was observed when putting timbre similarity measure instead of tonality in the first combination, producing lower results. However, putting all distances together in a neuron-network combination mechanism showed stable results but not higher than in the case of linear combinations.

The objective analysis of similarity retrieval algorithm have shown very good similar genre rate – 3.58 against the best rate found in literature (3.4) in TOP-5 playlists analysis based on the ISMIR'04 corpus. Promising results were also achieved in search for pieces with multiple interpretations.

Conclusions and Outlook

7. Conclusions and outlook

In this thesis we have considered the problem of automatic music analysis within such music information retrieval applications as music search by similarity (intelligent navigation) and automatic genre classification.

We have started in Chapter 3 by developing an appropriate tool of musical signal analysis. We presented the variable resolution transform as such a tool. We have shown it to be better suited for our applications. The goal we have achieved is to obtain a single transform which can simultaneously cover the whole time-frequency scale in such a way that both pitch and rhythm information is gathered at the same time. The advantage of the tool we have proposed is that it has logarithmic frequency sampling in order to follow musical notes. In comparison to some classical approaches where the frequency sampling is also logarithmic, we have an improved frequency resolution in high frequency area, allowing us to better distinguish the high-order harmonics of the signal.

Starting from Chapter 4 the questions of music similarity measures were discussed. In Chapter 4 we have presented and described an algorithm of beat and strong note onset detection based on the VRT and image treatment technique. In comparison to classical resonator-based or autocorrelation-based approaches, our algorithm is suitable for detection of any kind of beats with or without periodicities. The extracted beat information can be used to construct a rhythmic similarity measure in the form of 2D beat histograms allowing their direct comparison. A tempo induction method based on the 2D beat histogram was presented as evaluated as well.

In Chapter 5 we have followed with a description of an algorithm for multiple fundamental frequency (f_0) estimation based on the VR Transform. Like most of the classical approaches, this algorithm was based on harmonic pattern matching. Our contribution in this case is an application of note-adapted variable resolution transform. An evaluation of the aforementioned algorithm was performed on waveteable synthesized musical signals. The chapter also discusses the aspects of musical similarity. For this purpose various f_0-estimation derived or melodic features such as note succession histogram, note profile or timbre histogram were proposed.

Direct applications of musical similarity features and estimation of their performance is presented in Chapter 6. In the case of automatic genre classification it was proved that combining musical and spectral similarity features leads to a considerable performance gain. The objective of the second application was an automatic construction of similar music playlists. The algorithms were based only on musical features and were evaluated by human feedback giving encouraging results. The objective analysis of

similarity retrieval algorithm have shown superior genre rate in TOP-5 playlists analysis compared to the genre rate found in literature.

The techniques described in the thesis are for the most part ready to be implemented and deployed in real-life applications. Music similarity search produces acceptable results and can be integrated in online music stores or media players. However, the work opens a large number of scientific questions. Of course, the first research direction is the improvement of musical similarity features as well as of methods of their combination. For instance, the k-nearest neighbors (kNN) classifier we applied in the problem of genre classification is very content-dependent and requires training data to be available all the time. Another disadvantage of the kNN classifier is its increasing complexity with increasing training data set size. Thus, an important direction for future work is to employ other types of classifiers potentially capable of producing better and more stable results. This also concerns the use of neuron network in combining of different experts in both genre classification and music similarity problem. Using stochastic approaches can also be promising in all kind of algorithms we have presented.

Improving the musical features may require such algorithms as automatic instrument recognition which are outside the scope of in our work.

Another question remaining open is the invention of truly efficient note transcription algorithms working well with all kinds of music. As a subject of future work we can envisage a development of musical object detection-based approaches which would enable high semantic modeling and analysis, like it is presently done in image and video processing.

References

8. REFERENCES

[ABE 96] Abe T. et al.,. Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency. *In proceedings of ICSLP'96* (1996) : 1277-1280.

[AHA 91] Aha D., Kibler D., Albert M.. Instance-based Learning Algorithms. *Machine Learning* (1991) vol. 6: 37-66.

[ALGH 99] Alghoniemy M., Tewfik A.H.. Rhythm and Periodicity Detection in Polyphonic Music. *IEEE 3rd Workshop on Multimedia Signal Processing* (1999) : 185-190.

[ALON 03] Alonso M., Badeau R., David B., Richard G.. Musical Tempo Estimation Using Noise Subspace Projections. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (2003).

[ALON 06] ALONSO–AREVALO M.A.. Extraction d'Information Rhythmique a Partir d'Enregistremets Musicaux. PhD thesis at l'Ecole Nationale Superieure des Telecommunications.2006.

[ALON 07] Alonso M., Richard G., David B.. Tempo Estimation for Audio Recordings. *Journal of New Music Research* (2007) 36: 17-24.

[AREN 01] Arentz W.A.. Beat Extraction from Digital Music. *NORSIG* (2001).

[AUCO 02] Aucouturier J.J., Pachet F.. Music Similarity Measures : What's the use?. *Proceedings of ISMIR* (2002).

[AUCO 04] Aucouturier J.J., Pachet F.. Timbre Similarity: How high is the sky?. *In JNRSAS* (2004).

[AUCO 05] Aucouturier J.J. Pachet F., Sandler M.. "The Way It Sounds": Timbre Models for Analysis and Retrieval of Music Signals. *IEEE Transacations on Multimedia* (2005) vol. 7.

[BADE 02] Badeau R, Boyer R. David B.. EDS Parametric Modeling and Tracking of Audio Signals. *Proc. of the 5th Int. Conference on Digital Audio Effects (DAFx-02)* (2002).

[BAUM 03] Baumann S.. Music Similarity Analysis in a P2P Environment. *In proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services WIAMIS03* (2003) : 122-128.

[BERE 03] Berenzweig A., D.P.W. Ellis & S. Lawrence. Anchor Space for Classification and Similarity Measurement of Music. *In Proceedings of the IEEE International Conference on Multimedia and Expo, ICME 03, Baltimore* (2003).

[BERG 06] Bergstra J., Casagrande N., Erhan D., Eck D., Kegl B.. Aggregate Features and AdaBoost for Music Classification. *Nachine Learning* (2006).

[BROW 91] Brown J. C.. Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am.* (1991) 89(1): 425-434.

[BROW 93] Brown J.C.. Determination of the meter of musical scores by autocorrelation. *J. Acoust. Soc. Am.* (1993).

[CAS 05] Casagrande N., Eck D., Kegl B.. Frame-Level Audio Feature Extraction Using AdaBoost. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : 345-350.

[CHEV 02] de Cheveigne A., Kawahara H.. YIN, a Fundamental Frequency Estimator for Speech and Music. *Journal of the Acoustic Society of America,* (2002) : 111:1917-1930.

[CHUA 05] Chuan C.-H., Chew E.. Polyphonic Audio Key Finding Using the Spiral Array CEG Algorithm. *Proceedings of ICME* (2005) : .

[CODY 94] Cody M.A.. The Wavelet Packet Transform. *Dr. Dobb's Journal* (1994) 19: 44-46, 50-54.

[COHE 92] Cohen A., Daubechies I. Feanveau J.-C.. Biorthogonal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics* (1992) XLV: 485-560.

[COLL 05] Collins N.. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. *Proc. AES 118th Convention Barcelona, Spain* (2005).

[COVE 91] Cover T.M., Thomas J.A.. Elements of Information Theory. *Wiley Series in Telecommunications. John Wiley and Sons* (1991).

[DAN 02] Dan-Ning J., Lie L., Hong-Jiang Z., Lian-Hong C., Jian-Hua T.. Music Type Classification by Spectral Contrast Features. *In proceedings of ICME* (2002).

[DAUB 92] Daubechies I.. Ten Lectures on Wavelets. *CIAM. Philadelphia PA.* (1992).

[DINI 07] Diniz F.C.C.B, Kothe I, Netto S.L., Biscainho L.P.. High-Selectivity Filter Banks for Spectral Analysis of Music Signals. *EURASIP Journal on Advances in Signal Processing* (2007).

[DOVA 91] Doval B., Rodet X.. Fundamental Frequency Estimation Using a New Harmonic Matching Method.. *In proceedings of ICMA* (1991) : 555-558.

[ECK 05] Eck D., Casagrande N.. A tempo-extraction algorithm using an autocorrelation phasematrix and shannon entropy. *Proc. International Conference onMusic Information* (2005) : 504–509.

[ELLI 02] Ellis D.P.W., Whitman B., Berenzweig A., Lawrence S.. The Quest for Ground Truth in Musical Artist Similarity. *In proceddings of ISMIR* (2002).

[ESSI 05] Essid S.. Classisfication automatique des signaux audio-fréquences : reconnaissance des instruments de musique. PhD thesis Informatique, Telecommunications et Electronique, ENST.2005.

[FOOT 02] Foote J., Cooper M., Nam U.. Audio retrieval by rhythmic similarity. *Proceedings of ISMIR* (2002).

[FOOT 97] Foote Jonathan T.. Content-Based Retrieval of Music and Audio. *Proceedings of SPIE Multimedia Storage and Archiving Systems II (Bellingham,WA) vol. 3229, SPIE* (1997) : 135-147.

[GOME 06] Gomez E.. Tonal Description of Music Audio Signals. PhD thesis in University Pompeu Fabra, Spain.2006.

[GOTO 01] Goto M.. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. *Journal of New Music Research* (2001) Vol. 30, No. 2: 159-171.

[GOTO 01a] Goto M.. A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation Using EM Algorithm for Adaptive Tone Models. *In proceedings of ICASSP* (2001).

[GOUY 03] Gouyon F., Herrera P.. A Beat Induction Method for Musical Audio Signals. *Proceedings of 4th WIAMIS-Special session on Audio Segmentation and Digital Music* (2003).

[GRAC 02] Grachten M., Arcos J.-L. et al.. A Comparison of Different Approaches to Melodic Similarity. *Proceedings of ICMAI* (2002).

[GRIM 02] Grimaldi M., Kokaram A., Cunningham P.. Classifying Music by Genre Using the Wavelet Packet Transform and a Round-Robin Ensemble. (2002).

[GROS 84] Grossman A., Morlet J.. Decomposition of Hardy into Square Integrable Wavelets of Constant Shape. *SIAM J. Math. Anal.* (1984) 15: 723-736.

[HARB 01] Harb H., Chen L, Auloge J.Y.. Segmentation du son en se basant sur la distance de KulBack-Leibler. *In proceedings of CORESA, France* (2001) : 63-68.

[HARB 03] Harb Hadi. Classification d'un signal sonore en vue d'une indexation par le contenu des documents multimédias. PhD Thesis, Ecole Centrale de Lyon.2003.

[HARB 04] Harb H., Chen L., Auloge J-Y.. Mixture of experts for audio classification: an application to male female classification and musical genre recognition. *In proceedings of ICME* (2004).

[HARB 05] Harb H., Chen L.. A General Audio Semantic Classifier based on human perception motivated model. *Multimedia Tools and Applications, Eds. Springer Netherlands , ISSN 1380-7501 (Print) 1573-7721 (Online), http://dx.doi.org/10.1007/s11042-007-0108-9,* (2005) : .

[HAWL 93] Hawley M.. Structure out of Sound. Massachusetts Institute of Technology.1993.

[HOFM 02] Hofmann-Engl L.. Rhythmic Similarity: A Theoretical and Empirical Approach. *Proceedings of ISMIR* (2002).

[HOUT 73] Houtgast T., Steeneken H. M.. The modulation transfer function in room acoustics as a predictor of speech intelligibility. *Acustica* (1973) 28: 82-108.

[HU 01] Hu J., Sheng Xu., Chen J.. A Modified Pitch Detection Algorithm. *IEEE COMMUNICATIONS LETTERS* (2001) Vol. 5, No 2.

[Intel 03] Intel Corp.. IA-32 Intel(R) Architecture Software Developer's Manual. . , 2003.

[JORD 94] Jordan M.I., Jacobs R.A.. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* (1994) 6(2): 181-214.

[KADA 92] Kadambe S., Faye Bordreaux-Bratels G.. Application of the Wavelet Transform for Pitch Detection of Speech Signals. *IEEE Transactions on Information Theory* (1992) 38, no 2: 917-924.

[KASH 85] Kashimand K. L., Mont-Reynaud B.. The bounded-Q approach to time-varying spectral analysis. *Stanford Dept. of Music Tech. Rep.* (1985) STAN-M-28: .

[KLAP 04] Klapuri A.. Signal Processing Methods for the Automatic Transcription of Music. PhD thesis at Tampere University of Technology.2004.

[KLAP 06] Klapuri A., Eronen A., Astola J.. Analysis of the meter of acousticmusic signals. *IEEE Trans. on Speech and Audio Processing* (2006) 14(1): p. 342–355.

[KLAP 99] Klapuri A.. Pitch Estimation Using Multiple Independent Time-Frequency Windows. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (1999).

[KLAP 99a] Klapuri A.. Sound onset detection by applying psychoacoustic knowledge. *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* (1999).

[KOTO 07] Kotov O., Paradzinets A., Bovbel E.. Musical Genre Classification using Modified Wavelet-like Features and Support Vector Machines. *In proceedings of EuroIMSA* (2007).

[KRON 87] Kronland-Martinet R., Morlet J. and Grossman A.. Analysis of sound patterns through wavelet transform. *International Journal of Pattern Recognition and Artificial Intelligence,Vol. 1(2)* (1987) : 237-301.

[LANG 98] Lang W.C., Forinash K.. Time-frequency Analysis with the Continuous Wavelet Transform. *Am. J. Phys.* (1998) 66(9): 794-797.

[LAO 04] Lao W., Tan E.T., Kam A.H.. Computationally inexpensive and effective scheme for automatic transcription of polyphonic music. *Proceedings of ECME* (2004).

[LARG 94] Large E. W., Kolen J. F.. Resonance and the perception of musical meter. *Connection Science* (1994) 6(2).

[LI 07] Li Y., Wang D.. Pitch Detection in Polyphonic Music Using Instrument Tone Models. *In proceedings of ICASSP* (2007).

[LIM 92] Lim Y. C., Farhang-Boroujeny B.. Fast filter bank (FFB). *IEEE Transactions on Circuits and Systems II: Analog and Digital* (1992) vol 39: 316-318.

[LIPP 04] Lippens S., Martens J.P., Leman M., Baets B., Meyer H. and Tzanetakis G.. A Comparison of Human and Automatic Genre Classification. *In proceedings of the ICASSP* (2004).

[LOG 00] Logan B.. Mel Frequency Cepstral Coefficients for Music Modeling. *Proceedings of the ISMIR International Symposium on Music Information Retrieval (Plymouth, MA)* (2000).

[LOG 01] Logan B., Salomon A.. A music similarity function based on signal analysis.. *In Proceedings of IEEE International Conference on Multimedia and Expo ICME 01* (2001).

[MALL 89] Mallat S.G.. Multiresolution Approximations and Wavelet of orthonormal Bases of L2(R). *Transactions of the American Mathematical Society* (1989) 315: 69-87.

[MALL 99] Mallat S.G.. A Wavelet Tour of Signal Processing. . Academic Press, 1999.

[MAN 05] Mandel M., Ellis D.. Song-Level Features and Support Vector Machines for Music Classification. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : 594-599.

[MCAD 92] McAdams S., Cunibile J.C.. Perception of timbral analogies. *Philosophical Transactions of the Royal Society* (1992) vol. 339.

[MCK 03] McKinney M.F., Breebaart J. Features for Audio and Music Classification. *Proceedings of the ISMIR International Conference on Music* (2003) : 151-158.

[MCKA 06] McKay C., Fujinaga I.. Musical Genre Classification: Is it Worth Pursuing and how can it be Improved. *In proceeding of ISMIR* (2006).

[MCKI 03] McKinney M.F., Breebaart J.. Features for Audio and Music Classification. *Proceedings of ISMIR* (2003).

[MEN 05] Meng A., Shawe-Taylor J.,. An Investigation of Feature Models for Music Genre Classification Using the Support Vector Classifier. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : 604-609.

[MOEL 04] Moelants, D. and McKinney, M.F.. Tempo perception and musical content: What makes a piece slow, fast, or temporally ambiguous?. *International Conference on Music Perception & Cognition, Evanston, IL* (2004).

[NAVA 04] Nava G.P., Tanaka H.. Finding Music Beats and Tempo by Using an Image processing Technique. *In proceedings of ICITA* (2004).

[NAWA 01] Nawab S.H., Ayyash S.H., Wotiz R.. Identification of musical chords using constant-Q spectra.. *In Proc. ICASSP* (2001).

[PACH 00] Pachet F., Cazaly D.. A Taxonomy of Musical Genres. *In proceedings of RIAO* (2000).

[PACH 03] Pachet F., Laburthe A., Aucouturier J.J.. The Cuidado Music Browser: an end-to-end electronic music distribution system. *In INRIA, editor, Proceedings of CBMI 03, IRISA Rennes, France* (2003).

[PACH 07] Pachet F., Roy P.. Exploring Billions of Audio Features. *In proceedings of CBMI* (2007).

[PAMP 03] Pampalk E., Dixon S., Widmer G.. On the Evaluation of Perceptual Similarity Measure for Music. *In proceedings of DAFx* (2003).

[PAMP 05] Pampalk E., Flexer A., Widmer G.. Improvements of Audio-based Music Similarity and Genre Classification. *In proceedings of ISMIR* (2005) : .

[PAMP 06] Pampalk E.. Computational Models of Music Similarity and their Application in Music Information Retrieval. PhD thesis at Technishen Universitaet Wien, Fakultaet fuer Informatik.2006.

[PAMP 06a] Pampalk E.. Audio-Based Music Similarity and Retrieval: Combining a Spectral Similarity Model with Information Extracted from Fluctuation Patterns. *MIREX* (2006).

[PAR 05] Parshin V., Paradzinets A., Chen L. Multimodal Data Fusion for Video Scene Segmentation. *VIS* (2005).

[PARA 06] Paradzinets A., Harb H., Chen L.,. Use of Continuous Wavelet-like Transform in Automated Music Transcription. *Proceedings of EUSIPCO* (2006).

[PARA 07] Paradzinets A., Kotov O., Harb H., Chen L.,. Continuous Wavelet-like Transform Based Music Similarity Features for Intelligent Music Navigtion. *In proceedings of CBMI* (2007).

[PAUL 02] Paulus J., Klapuri A.. Measuring the similarity of rhythmic patterns. *Proc. Int.* (2002).

[PAUW 04] Pauwls S.. Musical Key Extraction form Audio. *In proceedings of ISMIR* (2004).

[PEET 04] Peeters G.. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. (2004).

[PEET 05] Peeters G.. Time Variable Tempo Detection and Beat Marking. *in proceedings of ICMC* (2005).

[PEET 06] Peeters G.. Chroma-based Estimation of Musical Key from Audio-signal Analysis. *Proceedings of ISMIR* (2006).

[PERR 99] Perrot D., Gjerdigen R.. Scanning the Dial: An Exploration of Factors in the Identification of Musical Style. *In proceeedings of the Society for Music Perception and Cognition* (1999) : p. 88 (abstract).

[PISZ 79] Piszczalski M., Galler B.A.. Predicting musical pitch from component frequency ratios.. *Journal of the Acoustic Society of America* (1979) 66: 710-720.

[POHL 06] Pohle T.. Post Processing Music Similarity Computation. *MIREX* (2006).

[PYE 00] Pye D.. Content-based Methods for the Management of Digiral Music. *In proceedings of ICASSP* (2000) vol. 4: 2437-2440.

[RIOU 91] Rioul O.. Fast Algorithms for the Continuous Wavelet Transform. *Proceedings of the 1991 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-91)* (1991) 3: 2213-2216.

[RUBN 00] Rubner Y., Tomasi C., Guibas L.J.. The Eartch Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* (2000) 40(2): 99-121.

[RUME 86] Rumelhart D.E., Hinton G.E., Williams R.J.. Learning internal representations by error propagation. *Distributed Processing - Explorations in the Microstructure of Cognition* (1986) vol. 1, chapter 8: 318-362.

[SAUN 96] Saunders J.. Real Time Discrimination of Broadcast Speech/Music. *In proceedings of ICASSP* (1996) 2: 993-996.

[SCAR 05] Scaringella N., Zoia G.. On the Modeling of Time Information for Automatic Genre Recognition Systems in Audio Signals. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : 666-671.

[SCAR 05a] Scaringella N., Mlynek D.. A Mixture of Support Vector Machines for Audio Classification. *In MIREX* (2005).

[SCHE 00] Scheirer E.. **Music Listening Systems.**. PhD thesis at the Massachusetts Institute of Technology.2000.

[SCHE 97] Scheirer E., Slaney M.. Construction and Evaluation of a Robust Multifeature Seech/Music descriminator. *In proceedings of ICASSP* (1997) : .

[SCHE 97a] Scheirer E.. Tempo and beat analysis of acoustic musical signals. *Machine Listening Group, E15-401D MIT Media Laboratory, Cambridge, Massachusetts* (1997).

[SMIT 07] SMITH J.O.. Spectral Audio Signal Processing. (http://ccrma.stanford.edu/~jos/sasp/). , 2007.

[SOBE 90] Sobel I.. An isotropic image gradient operator.. *Machine Vision for Three-Dimensional Scenes* (1990) : 376-379.

[SOLT 98] Soltau H., Schultz T., Westphal M.. Recognition of Music Types. *In proceedings of ICASSP* (1998).

[STEV 40] Stevens S., Volkman J.. The relation of pitch to frequency. *American Journal of Psychology* (1940) 34: 329.

[TALK 95] Talkin D.. Robust Algorithm for Pitch Tracking.. *In Speech Coding and Synthesis. Elsevier Science B.V.* (1995).

[TANG 93] Tanguiane A.S.. Artificial Perception and Music Recognition (Lecture Notes in Computer Science). . Springer, October 1993.

[TOIV 02] Toiviainen P., Eerola T.. A Computational Model of Melodic Similarity Based on Multiple Representations and Self-Organizing Maps. *Proceesings of the 7th International Conference on Music Perception and Cognition* (2002).

[TYPK 03] Typke R., Giannopoulos P. et al.. Using Transportation Distances for Measuring Melodic Similarity. *Proceedings of ISMIR* (2003).

[TZAN 01] Tzanetakis G., Essl G., Cook P.. Audio Analysis using the Discrete Wavelet Transform. . *WSES Int. Conf. Acoustics and Music: Theory 2001 and Applications (AMTA), Skiathos, Greece* (2001).

[TZAN 02] Tzanetakis G., Cook P.. Automatic Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing 10* (2002) : no 5, 293-302.

[TZAN 02a] Tzanetakis G.. Manipulation, Analysis and Retrieval Systems for Audio Signals. PhD thesis at Princeton University.2002.

[TZAN 03] Tzanetakis G., Gao J., Steenkiste P.. A scalable peer-to-peer system for music content and information retrieval. *Proceedings of International Conference on Music Information Retrieval (ISMIR)* (2003).

[WEST 04] West K., Cox S.. Features and classifiers for the automatic classification of musical audio signals. *Proceedings of the ISMIR International Conference on Music Information Retrieval (Barcelona, Spain)* (2004) : 531-536.

[WOLD 96] Wold E., Blum T., Keislar D. and Wheaton J.. Content-based Classification Search and Retrieval of Audio. *IEEE Multimedia Magazine* (1996).

[YANG 02] Yang C.. Acoustic Index for Music Retrieval with Various Degrees of Similarity. *In Proceedings of ACM Multimedia* (2002).

[YEH 05] Yeh C., Roebel A., Rodet X.. Multiple fundamental frequency estimation of polyphonic music signals. *in Proc. IEEE, ICASSP* (2005).

List of figures

Figure 2.1. 5-level music-based similarity analysis dataflow model compared to audio-based similarity dataflow. 14

Figure 2.2. 5-level music similarity analysis dataflow and chapter coverage. 15

Figure 3.1. Typical music pattern. 17

Figure 3.2. Fourier transform of two test signals. The signal a) is composed with two superposed waves with different frequencies, b) the same waves, but concatenated one after another, c) d) their Fourier spectrum 21

Figure 3.3. Mismatch of note frequencies and frequency resolution of the FFT. 22

Figure 3.4. Block diagram of DWT filter cascade. 25

Figure 3.5. Example of DWT filterbank. 26

Figure 3.6. Example of a valid wavelet packet tree. 26

Figure 3.7. Fourier basis functions, time-frequency tiles, and coverage of the time-frequency plane. 28

Figure 3.8. Morlet wavelet basis functions and time-frequency coverage. 28

Figure 3.9. Small-windowed Fourier transform (512 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top. 29

Figure 3.10. Large-windowed Fourier transform (≥1024 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top. 30

Figure 3.11. Wavelet transform (Morlet) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top. 30

Figure 3.12. Our mother wavelet function. A flat wave modulated by Hann window with $l=20$. 32

Figure 3.13. Harmonic structure in logarithmic frequency scale. 33

Figure 3.14. Equivalent central frequency of wavelet according to its bin number. $f_{min}=50$, $f_{max}=8000$. 34

Figure 3.15. Wavelet spectrogram of a piano recording (wavelet (3.18)). Single notes on the left and chords on the right. Up to 5 harmonics are resolvable. Higher harmonics after the 5^{th} one become indistinguishable especially in the case of chords where the number of simultaneous frequency components is higher. The main window illustrates our wavelet analysis tool developed within this thesis work. 35

Figure 3.16. Wavelet transform of a signal containing 5 delta-impulses. The distance between two impulses on the right is only 8 ms. 35

Figure 3.17. Human auditory system frequency resolution histogram for frequencies around 500Hz. 36

Figure 3.18. Fourier transform of the signal with notes played by a piano (the same signal with was used in previous wavelet experiment on Figure 3.15). Neither fundamental frequencies nor partials can be extracted. 36

Figure 3.19. Various *l(n)*, depending on parameters. From linear (left) to exponential (right). 37

Figure 3.20. Piano recording VRT spectrogram where fundamental frequency and partials are distinguishable (k1=0.8, k2=2.1). 38

Figure 3.21. Dependency of the distance between partials form partial number. 39

Figure 3.22. Spectral dispersion for a wavelet transform. It is taken in terms of wavelet number n on x axis. 40

Figure 3.23. Morlet wavelet transform with $\alpha^2=200$ of a delta-impulse. The maximum time localization exceeds one second bounds. 40

Figure 3.24. Spectral dispersion graph of VR transform with $k_2=2.1$ *and* $k1=0.8$ for bins numbers from 30 to 900. 41

Figure 3.25. Time resolution dependency of VR transform with $k_2=0.8$, $k_2=2.1$. 41

Figure 3.26. Practically calculated time resolution grid of our VR transform ms/Hz. 42

Figure 3.27. Practically calculated frequency resolution grid of our VR transform. The red trace stands for dispersion graph according to frequency; blue graph signifies a frequency step (Hz/bin) of the transform. Green curve on the picture stands for equivalent FFT frequency resolution (Hz/sample) as if it would have window size equal to that one from VRT (from Figure 3.26). 42

Figure 3.28. Intersection of two close frequencies on VR spectrogram. 43

Figure 3.29. VRT spectrogram of an excerpt from *Era – Flowers of the Sea*, piano. 43

Figure 3.30. 2048-point FFT spectrogram of the excerpt from *Era – Flowers of the Sea*. 44

Figure 3.31. Example of integer SIMD instruction from Intel's MMX instruction set. 45

Figure 3.32. Spectral feature extraction procedure. Wavelet or VRT spectrum is divided into time-frequency tiles. Histograms of values are computed in each tile. Histograms are then serialized in form feature vectors. 48

Figure 4.1. Typical energy envelop of a note (piano-like instrument) 51

Figure 4.2. Beat detection from the raw WAV. The signal's waveform is cut at certain energy level. Obtained peaks are considered as detected beat or high-energy events. 52

Figure 4.3. VR transform representation of a musical excerpt (of metal genre). 55

Figure 4.4. Sobel gradient operator in horizontal plane. 55

Figure 4.5. Sobel-filtered VRT spectrogram for a muscal excerpt with detected beat strength curve at the bottom. 56

Figure 4.6. Beat curve extraction procedure diagram. 56

Figure 4.7. Beat detection windowing procedure. Beat probability or beat strength curve (at the bottom) is obtained by moving a sliding window and calculating number of spectrum points whose values are higher than the average value for bigger one. 57

Figure 4.8. FFT spectral image (excerpt from “Nightwish – Come Cover Me”) 58

Figure 4.9. VRT spectral image (excerpt from “Nightwish – Come Cover Me”) 58

Figure 4.10. Classical one-dimensional beat histogram from the work of *G. Tzanetakis*. 59

Figure 4.11. Example of two-dimensional beat histogram. Here the x axis represents beat period, y axis stands for threshold value used in peak detection on the beat curve. 59

Figure 4.12. Two-dimensional beat histograms for dance (left) and classic (right) musical pieces. 60

Figure 4.13. Beat histogram for *“Rammstein – Mein Herz Brennt”* without filtering (top) and after bass/treble cut filter (bottom). 61

Figure 4.14. Small rhythmic changes lead to large bin-by-bin beat histogram distance (at strength plane). 62

Figure 4.15. Illustration of EMD between suppliers and consumers. 63

Figure 4.16. Tolerance area used in the similarity computation of two beat histograms. 65

Figure 4.17. A 2D beat histogram. Main tempo and its 2x alias are marked on the image. 65

Figure 4.18. Performance comparison of two tempo estimation algorithms – EC-Lyon and ENST-Paris. 66

Figure 4.19. Distribution of the tempo estimation accuracy as function of genre for ECL algorithm. 67

Figure 4.20. Distribution of the tempo estimation accuracy as function of genre for ENST algorithm. 67

Figure 4.21. Tatum estimation accuracy of two algorithms. 68

Figure 4.22. Tatum estimation performance by musical genre. EC-Lyon algorithm. 68

Figure 4.23. Estimation performance by musical genre. ENST-Paris algorithm. 69

Figure 5.1. Harmonic structure. 75

Figure 5.2. Matching of harmonic models to spectrum. 75

Figure 5.3. Procedure of extraction of harmonic amplitude vector. 76

Figure 5.4. Block diagram of note detection procedure. 78

Figure 5.5. Note recognition and evaluation program. 79

Figure 5.6. Note detection algorithm performance according to underlying spectral analysis approach. 81

Figure 5.7. Tonality. Do-Major and Do-Minor (natural) tonalities. 82

Figure 5.8. Note profiles for major (C-dur, left) and minor (C-mol, right) tonalities (approximate). 83

Figure 5.9. Comparison of note histograms taking into account all possible transpositions. 83

Figure 5.10. Procedure of note succession histogram calculation. 84

Figure 5.11. 3D note successions histogram example. 85

Figure 5.12. Computing of timbre histogram. 85

Figure 6.1. General principle of classification 91

Figure 6.2. The k-NN classification principle. 91

Figure 6.3. Multi-expert architecture of the classification system. Individual experts issue genre probabilities which are then mixed using the final combining expert. 93

Figure 6.4. Comparison of classification results issued by different classifiers and their multi-expert combination for both databases. 102

Figure 6.5. Performance of separate classifiers and their combination according to genre in the case of Magnatune database. 103

Figure 6.6. Performance of separate classifiers and their combination according to genre in the case of ECL database. 103

Figure 6.7. Dependency of the classification accuracy from the size of cluster in the k-NN. 104

Figure 6.8. Dependency of the classification accuracy from the window shift in beat detection algorithm. 104

Figure 6.9. Architecture of the combining system with linear distance combination controlled by neuron network. 107

Figure 6.10. Histograms of listeners' votes for pure similarity metrics. The upper row contains histograms of votes for corresponding randoms. 110

Figure 6.11. Histograms of listeners' votes for linearly and rating combined similarity metrics. 111

Figure 6.12. Histograms of listeners' votes for MLP combined similarities. 113

Figure 6.13. Histogram of number of songs in the same genre in TOP-5 (left), and histogram of number of songs from the same artist in (TOP-5) (right).....114

Figure 6.14. MIREX 2007 Audio Music Similarity and Retrieval subjective evaluation results (x-axis – team number, y-axis – average vote divided by 10).....116

Figure 6.15. Histogram of listeners' votes resulted by our system at MIREX contest. 116

Figure 6.16. MIREX similarity contest result distribution by genres.....117

Figure 6.17. Average vote for query and result of the same or of different genres..118

List of tables

Table 3.1. Comparison of VRT to other approaches according to their properties....46

Table 3.2. Comparison of VRT to other approaches according their applicability....47

Table 4.1. Tempo estimation results for 10ms analysis window shift.....70

Table 4.2. Tempo estimation results for 15ms analysis window shift.....70

Table 4.3. Tempo estimation results for 30ms analysis window shift.....70

Table 4.4. Tempo estimation results for 30ms analysis window shift, 10% confidence area.....70

Table 4.5. Complete table of reference results from ISMIR2004 tempo estimation contest. 70

Table 5.1. Note detection performance in monophonic case. Sequences are played manually using keyboard. 80

Table 5.2. Note detection performance in polyphonic case. Sequences of chords are played manually using the keyboard. 80

Table 5.3. Note detection performance in polyphonic case. Classical music titles (single- and multi-instrument, no percussion).....80

Table 5.4. Note detection performance in polyphonic case. Popular and other music (multi-instrument with percussion).....80

Table 5.5. Comparison of transcription performance based on different time-frequency transforms (the FFT with various window sizes versus VRT). 81

Table 6.1. Database details – number of songs per class, number of various artists per class etc. 96

Table 6.2. *Magnatune* database details. 96

Table 6.3. Beat expert classification result on *Magnatune* (normalized mean accuracy 54.6%, raw accuracy 68.1%).....97

Table 6.4. Timbre expert result (normalized mean accuracy 39.6%, raw accuracy 52.2%).....98

Table 6.5. Acoustic expert on *Magnatune* (normalized mean accuracy 49.6%, raw accuracy 53.6%) 98

Table 6.6. Dan Ellis & Brian Whitman's system classification result (51.48% mean and 64% raw).....99

Table 6.7. Classification accuracy by a reference system (E. Pampalk) (78.78% normalized accuracy and 84.07% raw).....99

Table 6.8. *ECL_genres* database classification results with rhythmic classifier (normalized mean accuracy 71.4%) 100

Table 6.9. *ECL_genres* database classification results by timbre (normalized mean accuracy 42.4%) 100

Table 6.10. *ECL_genres* database classification results by acoustic expert (normalized mean accuracy 49.3%).....100

Table 6.11. All experts combined by MLP, (normalized mean accuracy 66.9%, raw accuracy 74.2%) 101

Table 6.12. *ECL_genres* database classification results (normalized mean and raw accuracy 80.9%) 102

Table 6.13. Listening test results (mean / median). 109

Table 6.14. Listening test results for MLP combinations. 112

Table 6.15. Objective evaluation results of music comb3_lin similarity measurement. 113

Table 6.16. Average number songs in the same genre or from the same artist.....114

Table 6.17. Maximum number of appearances of a song in top-5 rankings.....118