CONCATENATIVE MUSIC COMPOSITION BASED ON RECONTEXTUALISATION UTILISING RHYTHM-SYNCHRONOUS FEATURE EXTRACTION

Diploma Thesis

written by

Luka Mikula

Institute of Electronic Music and Acoustics

University of Music and Dramatic Arts

Graz, Austria

Supervisor: DI Dr. Alois Sontacchi

Graz, September 2008

Acknowledgements

First and foremost, I would like to thank Alois Sontacchi for the many hours of creative, well-founded and often humorous advice and supervision.

I would also like to thank my family, and especially my parents, Kornelia and Johann Mikula, for the unwavering support over the years. Without you, none of this would have been possible.

A special thank you goes out to my nephew Konstantin Mikula, who has been a fountain of joy for me and the whole family.

Lastly, I also would like to thank my friends in Graz, Vienna and everywhere else for their friendship and support.

Abstract

The aim of this diploma thesis is to design a software tool that makes a new kind of transformation of audio signals possible. This can subsequently be used to synthesize music from audio signal fragments (“Concatenative Music Synthesis“). Many applications, e.g. genre classification of music or generation of playlists etc., analyse and compare audio signals using coefficients (“features”) describing, for example, timbre and harmony properties. These features are extracted from the audio signal itself. This diploma thesis attempts a new approach to the re-synthesis of audio signals. As in a mosaic, an existing song is newly constructed from small parts (“frames”) of other songs already stored in a database. The length of the frames corresponds to musically meaningful units. For the implementation of the tool, suitable onset- and beat tracking methods are evaluated and the selection of suitable parameters describing the subjective semantic similarities is determined by listening tests.

Zusammenfassung

Die vorliegende Diplomarbeit zielt auf die Entwicklung eines Software-Tools ab, das eine neuartige Klang- bzw. Musikstücktransformation ermöglicht. Damit soll in weiterer Folge eine musikalisch sinnvolle Synthese von Musikstücken aus kleinen Audiosignal-Einheiten („Concatenative Music Synthesis“) durchgeführt werden können. Bei vielen Anwendungen, wie z.B. Klassifizierung von Musik nach Genres, Generierung von Playlists etc., werden Musikstücke auf Basis des Audiosignals analysiert und verglichen. Als Maß dafür werden meist klangfarbenbeschreibende und harmoniebeschreibende Koeffizienten („Features“), die aus dem Signal selbst extrahiert werden, verwendet. In der vorliegenden Arbeit wird nun versucht, diesen Ansatz zur Resynthese von Audiosignalen zu nutzen. Vergleichbar einem Mosaik wird ein bestehender Song aus vielen kleinen, vorher analysierten und kategorisierten Teilen („Frames“) anderer, bereits in einer Datenbank gespeicherter Musikstücke, neu erstellt. Die Abgrenzung von Frames erfolgt anhand musikalischer Sinneinheiten. Für die Umsetzung werden geeignete Onset-Detection- bzw. Beattracking-Verfahren evaluiert und geeignete Parameter zur Beschreibung der subjektiven semantischen Entsprechungen und Ähnlichkeiten werden anhand eines Hörversuches festgelegt.

Table of Contents

1 Introduction

In this thesis, a software tool is presented that is able to re-synthesize a song or musical piece from audio signal fragments stored in a database. Audio signals are segmented according to their rhythmic structure and analysed in regard to their descriptive features and a new song is created by concatenating the existing database fragments.

This thesis is organised as follows: Chapter 1 serves as an introduction to the topics that are discussed in this thesis, including a short preview of the implemented synthesis algorithm, the used beat and onset detection system, the used descriptive features that characterise audio signals and the listening test that was carried out in order to evaluate subjective similarities between audio signals.

Section 2 is dedicated to the general concept of concatenative music synthesis, with an overview over the history of this branch of electro-acoustic music and the different approaches presented by a multitude of authors as well as the implementations of these systems.

In Chapter 3, the audio synthesis interface implemented in this thesis is presented. The organisation of the audio signals database is explained in detail, as well as the organisation of the synthesis algorithm that creates new music from audio frames contained in the aforementioned database.

For this thesis, a number of different onset detection algorithms were implemented and evaluated in order to ensure the best possible segmentation of audio signals. These algorithms, all based on some spectral feature of the audio signal, are described in Chapter 4. The method used to determine the actual onsets from local maxima extracted from the detection functions that are created in different ways are explained. Also, the inter-onset interval beat tracking system that evaluates and corrects the results of the onset detection algorithms is talked about. Lastly, the evaluation results of the different onset detection methods are presented.

Chapter 5 introduces the temporal, spectral and statistical low-level features characterising audio signals that are evaluated in this thesis. The correlation between the temporal evolution of these features and the positions of onsets is also investigated in this section.

In section 6, the listening test that was carried out in order to determine subjective similarities between audio signals and their relationships to features computed from the signals, as well as the test results and conclusions, are described in detail.

Section 7 discusses the results and provides a perspective on future tasks and research topics.

1.1 Algorithm Framework

The tool for concatenative music synthesis presented in this diploma thesis is implemented as a Graphical User Interface called “ConCat Music Synthesis Interface” under MATLAB. To run the tool, a working installation of MATLAB 7.x is required. Reasons for choosing MATLAB as development environment are the wide-spread use, its cross-platform compatibility and GUIDE, the development environment for Graphical User Interfaces. Another advantage of MATLAB is that it is widely used as a teaching tool and signal processing tool in the academic and industrial environment.

The basic structure of the algorithm is implemented as follows: a database is created by taking a number of songs or audio signals and analysing them. The data is divided into musically and psycho-acoustically meaningful segments by performing beat tracking and onset detection to determine fitting segment boundaries. A number of relevant audio features are extracted to describe the respective audio fragment. These are then re-sorted into clusters and sub-clusters, thereby arranging the database not in a chronological and therefore coincidental manner but according to basic signal parameters. Clustering takes into account the characteristics of stored data which correspond to human perception [1], which also has the added benefit of drastically reducing the computing time required for the algorithm. The fragments are first ordered according to their duration and then according to their pitch. Since the frames are segmented in relation to the detected onsets and the beat, the segment length is directly related to the rhythmic structure of the audio signal. This makes the organisation of the audio database psycho-acoustically valid since it takes into account both the rhythmic and the melodic structure of the analysed audio signals.

After creating and editing the database, the song that is to be re-synthesized (the *target song*) is loaded and analysed in the same way as the database audio data by segmentation and feature extraction. For every data frame, the database is searched and the cluster and sub-cluster where the most similar frames are stored are detected. On the basis of this information the frames in the found sub-cluster are compared to the current target song frame by matching the respective *feature vectors* and the information concerning the database frame with the best fit is returned. This information enables the re-synthesis algorithm to concatenate the found matching database frames to create a new song. The relevance of the selected low-level features for subjective similarity of audio segments is determined by a listening test.

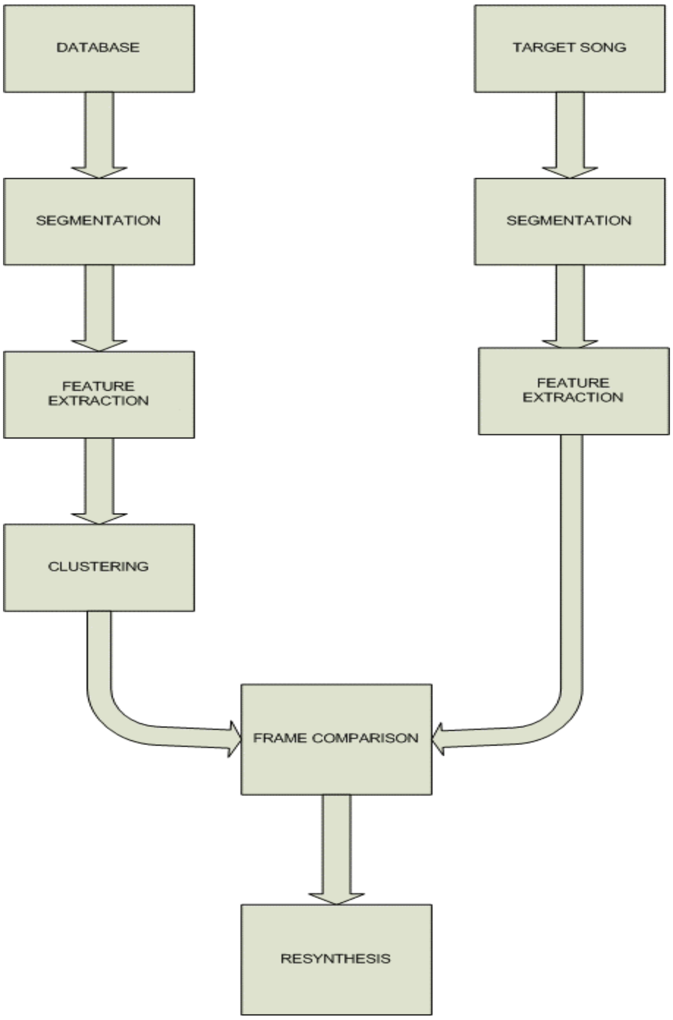


Figure 1.1 Re-synthesis algorithm structure

1.2 Beat Tracking

The automatic extraction of a structured pulse or rhythm from audio signals, called *beat tracking*, has been a major topic of research in Digital Audio in the last decade. Beat tracking algorithms attempt to create a symbolic representation of what human listeners experience as "beat" or "pulse". In this work, the *beat* of a signal is defined as a "sequence of equally spaced temporal units" [2] that describes the musical pulse or tempo. It should be noted that the grouping of beats into bars and the accents on strong beats (for example on the first note and in a lesser degree on the third note of pieces in 4/4 time signature) and its modelling is not in the scope of this work.

Existing research has focused on two approaches to beat tracking: some authors implemented methods where the beat is predicted directly via filter banks and resonators [2], while others based their method on the evaluation of onset detection results, grouping detected note onsets and extracting beat hypotheses from the discovered patterns [3]. While algorithms relying on onset detection are more flexible, they require the detection to be robust enough so that reliable beat estimates can be computed, thereby limiting their applicability.

1.2.1 Onset Detection

Music is an inherently non-stationary process – few people would classify a stationary tone or sound as music because there is no musical meaning or information without change. Music is an "event-based phenomenon" [3], which means that its attributes such as timbre, intensity or note length change with varying speeds during a music piece. *Onsets* are perceived and can be detected when a sound changes in this way.

Onset detection is a method to automatically detect these crucial "events" in audio signals and is used in a variety of applications including music analysis, segmentation ([5], [6]) – thereby enabling the cut-and-paste operations used by concatenative music synthesis –, audio/video-synchronisation [7], indexing [6] and many more. Many audio applications require accurate onset detection in order to work properly. For example, the improvement of low bit-rate audio quality that is sought after in newer compression standards requires accurate onset detection in order to segment audio files into regions with consistent statistical attributes, which in turn can be processed more efficiently [3]. This principle is also used in digital signal processing to adapt audio effects and transformations like pitch-shifting to the audio signal itself [8].

Since the term “onset” is used differently in different contexts, the terms used in this thesis will be defined as follows: “onset” is the point in time when the musical “event” appears, i.e. when there is significant and quick change in signal characteristics. It is the earliest time the event can possibly be detected.

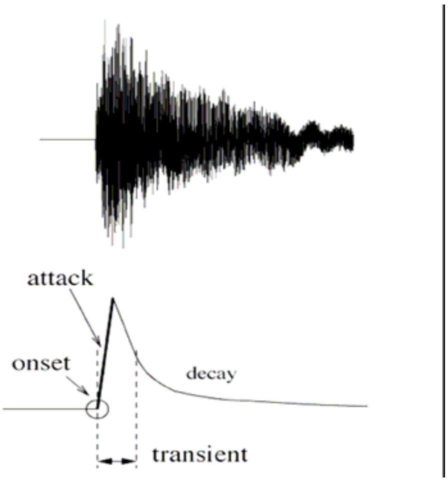


Figure 1.2 Onset definition [3]

When working with monophonic sound, the simplest solution for detecting onsets would be to differentiate the signal envelope and look for the signal areas where major positive changes reside. The onsets would then be the starting points of these changes.

This approach is not the optimal solution to this problem for a number of reasons: the amplitude of sounds, especially in the low frequency area, does not always increase monotonically, leading to more than one local maximum in the area of the actual onset and it may take some time for low-frequency sounds to reach the point where the increase in amplitude is the sharpest. Another difficulty arises when dealing with polyphonic audio data because the signals of the different instruments are superimposed.

Therefore, audio signals are processed in order to emphasise characteristics of the signal that can be used as indicators for onsets while still retaining the basic structure of the original signal. Afterwards, most existing approaches use down-sampling to reduce the data volume. The result of this processing is called a *detection function* or *novelty function* and could be described as a simplified version of the audio signal. After this, algorithms that pick out local peaks are applied to the detection function to locate the onsets.

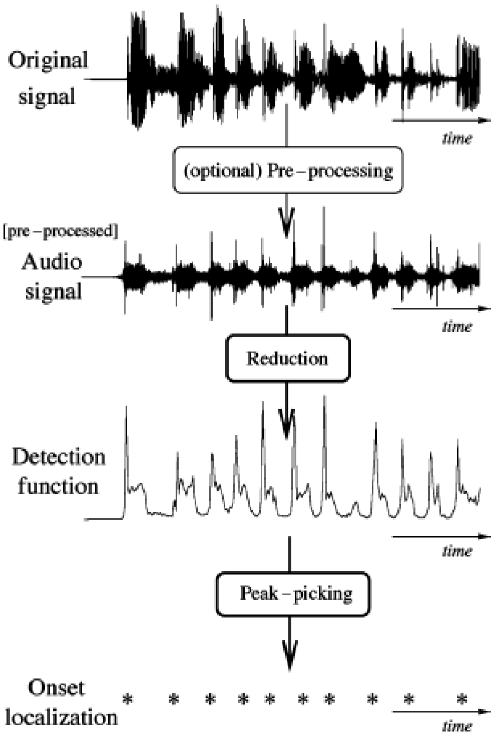


Figure 1.3 Onset detection systems workflow [3]

Pre-processing is used in some onset detection algorithms to accentuate aspects of the signal that are more relevant for the task at hand. Two approaches are widely used: the division of the signal into multiple frequency bands and *transient/stable separation*. Analysis over a number of frequency bands is used to increase the reliability of onset detection by combining results of filter bank outputs [9] or in combination with global tempo estimates [2]. In algorithms using transient/stable separation, the residual between an audio signal and a *Spectral Modelling Synthesis* (SMS) model is calculated. Sudden increases in the residual energy mark regions where the model and the original signal are mismatched and thus serve as indicators for onsets [10].

After the (optional) pre-processing stage the audio signal is down-sampled and transformed into a detection function. The approaches to this problem can be divided into two groups according to how data reduction is achieved. In the first group characteristic signal features are evaluated, the other bases its calculation on probabilistic signal methods.

Early onset detection methods based on signal features followed the amplitude envelope itself [3]. Variations included watching the energy envelope or the first-order difference of the amplitude envelope [10]. In the spectral domain, methods were proposed that use weighted short-time spectral energy measures or phase deviations as onset indicators [3] and methods that consider both [11].

Probabilistic signal models describe statistical features of audio signals. By using them, it is possible to match given audio signals to signal models and infer likely onset times. Of course, this approach is greatly dependent on the quality of the chosen signal model [3].

The last stage of onset detection algorithms consists of a peak-picking algorithm that chooses the most likely onset candidates from local maxima of the detection function. Often there is also an intermediate stage where the detection function is further processed to facilitate peak-picking, for example by *smoothing*[1] or normalising it. The peak-picker itself uses a threshold which can be fixed or adaptive. All local maxima that remain above the threshold are assumed to be indicative of onsets.

1.3 Features

Feature extraction is not only used in audio signal processing. It is essentially a way to reduce data volume by removing redundant information. This makes feature extraction an important tool in many fields other than audio, for example in image data processing [6] or pattern recognition [12].

The concept of describing the properties of audio signals with features was created for the purpose of characterising different audio signals and making statements about the similarity or dissimilarity between them. It has been attempted with varying results to use this information for purposes such as genre classification [13], thumbnailing [14], database organisation and searching [1] and many more.

The common approach to feature extraction is to use short-time analysis windows in the time and spectral domain. For meaningful results, it is important to regard only audio segments

[1] Using a low-pass filter or a moving average window, which eliminates fast signal changes

that are mostly stationary, because the inclusion of transient regions produces artefacts and analysis errors. The series of obtained low-level features can be combined to create high-level feature vectors [15].

Some features evaluated in this thesis are calculated “by hand” using either implementations of feature definitions or functions already existing in MATLAB[1], some of them are obtained by using the “MIR Toolbox” created by Olivier Lartillot [16]. This toolbox offers a wide variety of musical descriptors and features.

Features calculated in the time domain include *volume* (RMS) information and the *zero-crossing rate* describing the number of sign changes of the signal amplitude. In the spectral domain, a number of different features are computed. These include the *spectral centroid*, which contains information about the center of gravity of the energy distribution, two measures of high-frequency content (*spectral roll-off* and *brightness*), indicators for harmonic structures in the signal spectrum (*irregularity* and *roughness*), *pitch* and *MFCC* (*mel-frequency cepstral coefficients*) information as well as *chroma* estimation. In addition, statistical information about the signal spectrum is extracted by computing the statistical moments *skewness* and *kurtosis* as well as the *flatness* of the spectral distribution.

While there are countless audio features in literature [13], it is not advisable to use a high number of them for similarity or classification tasks. The computation time necessary to compare multi-dimensional feature vectors makes real-time implementations impossible. Also, many features are highly correlated, thus rendering a major number of them unnecessary.

1.4 Subjective Similarity Evaluation

To determine which audio features best correlate with subjective similarity perceptions, a listening test was carried out. Its goal was to arrive at a weighted combination of audio features that together give a meaningful measure of similarity between audio signals. The found feature combination is used by the concatenative music synthesis algorithm to determine the database fragments that best match the target song frames perceptually. By comparing the features extracted from audio signal segments that were judged to be similar by trained listeners, a six-dimensional feature vector that describes an audio signal is arrived at.

The test was implemented in the form of an A-B comparison. The subjects could all be classified as expert listeners. Most of them were Audio Engineering students with trained ears. They were asked to compare short audio fragments regarding their subjective similarity and to rate the similarity on a scale.

The results are evaluated using Multi-Dimensional Scaling (MDS), a technique used in statistics applications to reveal similarity relationships and to reduce data dimensions [17].

[1] Statistical features are calculated using existing functions from the pre-defined “Statistics Toolbox”

2 Concatenative Music Synthesis

The groundwork for sound synthesis in general and granular synthesis and micro-montage in particular was laid by the British physicist Dennis Gabor in the late 1940s, who postulated that musical tones are made up by a combination of elementary “grains” or atoms. Following the then recent discovery of discrete energy levels of atoms and atom nuclei, he described sound as a “succession of discrete units of acoustic energy” [18].

The practical origins of concatenative music synthesis – also called *Adaptive Concatenative Sound Synthesis* (ACSS) in literature [19] – as a tool for music composition can be traced back to the beginnings of electro-acoustic music in the second half of the 20th century. Back then, artists used the newly developed magnetic tape techniques to concatenate, splice and generally manipulate short sounds or music fragments. One of the most famous composers of this era, Pierre Schaeffer, presented his concept of “sound objects” which are similar to the fragments or segments used in concatenative synthesis as “clearly delimited segments in a source recording” and “the basic units of composition” [20]. At present, composers are able to create and transform sound fragments digitally, eliminating the need for written “notes” and scores. Digital synthesis “makes it possible to compose directly with sound, rather than by having to assemble notes” (Max Mathews in [21]).

The principle of sound synthesis has remained the same over the years – the desired sounds are created by concatenating segments synthesized from other sounds based on some measure of similarity, which could be defined as an “audio collage”. The segments are chosen by a “unit selection algorithm” to fit a specified “target” so as to minimise the difference between the synthesis product and the target, much like an adaptive system. In fact, some authors view concatenative synthesis not as a synthesis method but an “adaptive digital audio effect” [8] where synthesis or transformation tasks are controlled by features extracted from some sort of target, or a complex and automated remix apparatus. The similarity between database and target segment is mostly computed by comparing low-level or high-level features (“descriptors”). In contrast to “normal” composition and synthesis techniques which are “rule-based” this approach can be defined as “data-driven” – synthesis rules are induced from already existing data so as to preserve its attributes and details and not from abstract rules or sound models. Examples of concatenative synthesis taken from other artistic fields could include Arcimboldo’s paintings of faces composed of fruit – small picture “segments” are used to create a bigger picture – or the “pointilistic” paintings of Georges-Pierre Seurat.

At the present time, concatenative music synthesis can be used to effectively manage large sound databases and it also provides an intuitive and simple approach to sound fragment manipulation. Database management is an interesting research field nowadays because of the large amount of data that is used and manipulated by the average person, e.g. photos, audio tracks, films etc. The size of music databases is limited only by available disk space, which has increased exponentially over the past years while at the same time it has become much cheaper. The amount of time saved by a quick and efficient sound concatenation algorithm can be spent on experimenting, fine-tuning and on actual composition tasks, thus eliminating the need for time-consuming data or tape manipulation.

As Max Mathews wrote in 1969, “The two fundamental problems in sound synthesis are (1) the vast amount of data […] – hence the necessity of a very fast program – and (2) the need for a simple, powerful language in which to describe a complex sequence of sounds.” [22]. The first problem has become irrelevant due to the availability of cheap memory and due to the marked improvement in digital processor performance (processor power has grown 40% per year over the past decade), which allows real-time implementation of sound manipulation algorithms. The second one “cannot, in principle, ever be completely solved” [21] because manipulating individual sound samples is too time-consuming. To circumvent this problem, samples are synthesized algorithmically, where a small number of variables controls a very much larger number of samples.

2.1 Historical Overview

This chapter presents the history of concatenative music synthesis and Audio Montage as separate branches in electro-acoustic music. Their origins are traced back to the 1950s, when composers started experimenting with magnetic tape. A short introduction to digital audio signal montage is given.

2.1.1 Analog Montage

As mentioned above, the origins of concatenative music synthesis can be traced back to the 1950s where innovative artists started experimenting with short fragments of magnetic tape. The process of concatenating the segments was an arduous and time-consuming task and was performed by selecting, splitting and fusing sound segments recorded on magnetic tape. In this way it could take several months, if not years, to create musical pieces that were performed in a few minutes.

As the compositions of this period created new definitions of music and music performance the borders between the "styles" of electro-acoustic music were fleeting and were only later categorised as "micro-montage music", "musique concrète" etc. The one thing they had in common was the approach to the composition tasks, using techniques that can be defined as "granular synthesis".

Two well-known composers who worked with montage composition in the fifties were John Cage ("Williams Mix", 1952, where the musical "score" consists of graphical instructions for the splicing and gluing together of tape material [23]) and Iannis Xenakis, whose work includes "Diamorphoses" (1957) and "Concrèt PH" (1958, an introduction to the Varèse's "Poème electronique" written for the World Fair in Brussels) and "Analogique B" (1959, using electronically synthesized sounds [24]).

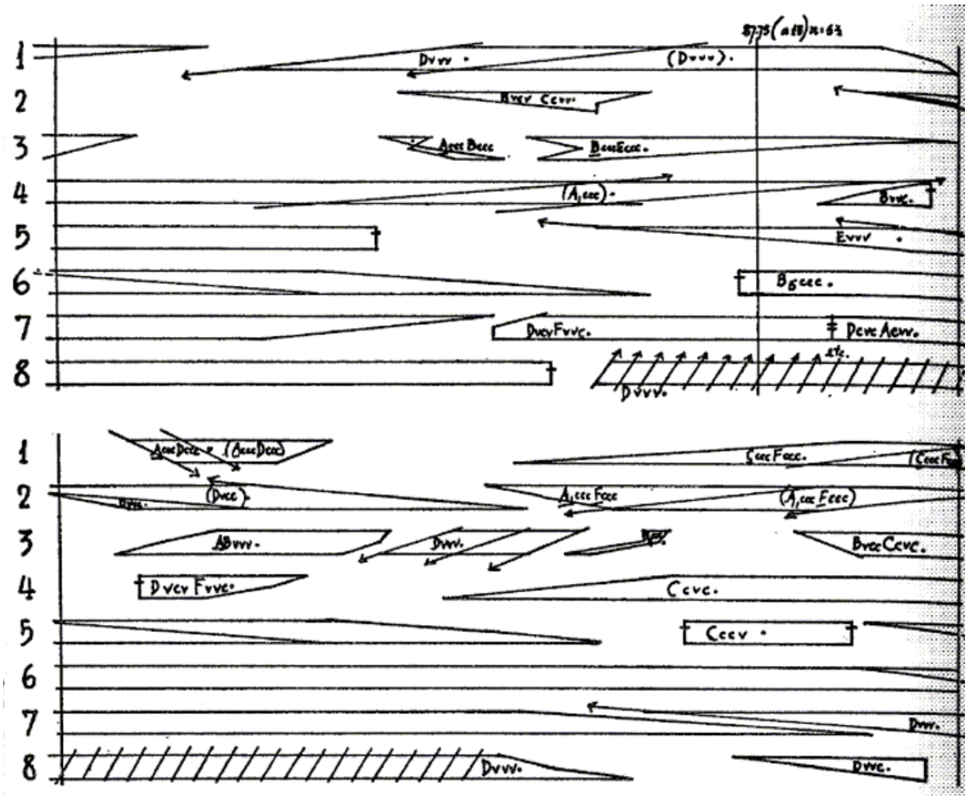


Figure 2.1 Score excerpt from John Cage’s „Williams Mix“ [23]

2.1.2 Digital Montage

The technique of montage composition began to be more widely used with the onset of the “Digital Era” when computers became affordable. The price reduction of memory led to large music and sound databases that could be manipulated with digital controllers – today, even ordinary personal computers are powerful and fast enough for these tasks.

Composers that used digital montage include famous names like Horacio Vaggione, Curtis Roads, Noah Creshevsky and Barry Truax.

Curtis Roads is very active in sound synthesis research and composition. He has worked on many algorithms and programs for music synthesis, for example a program for pulsar synthesis called “PulsarGenerator” and programs for granular synthesis – “Cloud Generator” (1995, with John Alexander) where time stretching and shifting is implemented by manipulating grains from sound files, or the “Creatovox” (1999, with Alberto de Campo), which is a synthesis engine built for the performance of granular synthesis and can be played via MIDI interface [18]. Roads is also a ground-breaking composer, as evidenced by “Klang-

1" (1974), the first granular synthesis composition realised by computer. He also pioneered automated granular synthesis – his work "Prototype" (1975) is the first music piece using this composition method. "Half-Life" (2004) uses audio material created by the above-mentioned Pulsar Synthesis.

Horacio Vaggione uses repeated and transformed piano sounds in his piece "Schall" (1995). He also experiments with multiple time scales and rearrangement of sound "particles", realised in "Agon" (1998), where Vaggione breaks up and rearranges percussion instrument sounds. He also composed pieces where instruments and electro-acoustic segments are played simultaneously, as in "MYR-11" (1997) [25].

It should be noted that both Vaggione and Roads used customised software programmed by themselves to generate sounds.

Noah Creshevsky describes his own style as "hyperrealism", a style where samples taken from everyday situations and acoustic environment are mutated by eliminating the characteristics that define them as ordinary. One typical example of his style is his work "Borrowed Time" (1995), where he combines fragments of vocal music from the 12th century until the present.

The first composer to implement real-time granular synthesis in an interactive environment was Barry Truax, who used signal processors for this time-consuming task. He incorporated this technique into the PODX computer music system at Simon Fraser University in Vancouver in 1986. He used sample-based granular synthesis for his famous work "Wings of Nike" (1987), an audio-visual composition. His partner Theo Goldberg created the images and Truax used only two phonemes (each about 170 milliseconds long) to create a twelve-minute soundscape.

Two composers who incorporate culturally meaningful and symbolic sounds into their works are James Tenney, who, using the same tape manipulation techniques as Xenakis, re-contextualises fragments from the Elvis Presley-sung "Blue Suede Shoes" in his work "Collage #1 (Blue Suede)" (1961) and John Oswald, who recombines short fragments taken from popular music of this decade, illustrated in his "Plexure" (1993).

2.2 Approaches to Concatenative Music Synthesis

Concatenative synthesis has been a popular technique in speech processing since the 1980s. Using a database of spoken characters, phonemes, syllables, words and sentences can be constructed by concatenating the recorded characters. By modifying the database waveforms, a higher degree of realism is introduced. Programs that concatenate spoken words stored in databases use this approach, called CTTS (*Concatenative Text To Speech*) [19], on a higher level. This technique can be useful when a limited number of words is combined in many different ways, for example for announcements at train stations or in car navigation systems.

CTTS is often judged to produce more realistic-sounding results than parametric models [19]. This alternative approach has the advantage of not needing large sound sample databases.

As mentioned above, composers of electro-acoustic and electronic music have used techniques similar to concatenative music synthesis.

Curtis Roads used the “Granulation” technique to segment audio signals and – after optional transformations and modifications – reassemble them later. However, his approach belongs more into the realm of granular synthesis or micro-montage due to the shortness of the used samples [18]. Trevor Wishart uses a similar approach termed “Brassage” which is the French word for “jumbling” or “mixing”. These methods have been used chiefly to create soundscapes, i.e. “dynamic audio environments” [19].

In the realm of (electronic) dance music, Nick Collins devised the “BBCut” extension libraries for SuperCollider, a high-level programming language for audio synthesis and algorithmic composition. His algorithm chops up audio files according to the beats found in the files and reassembles them, creating a “jumbled” but synchronous version of the original song [19].

Ian Simon developed an alternative method for concatenative synthesis. His algorithm is related to concatenative speech synthesis and includes concepts from image processing. It synthesizes audio material corresponding to a specified MIDI score from an existing monophonic audio file and the matching MIDI score [26].

Tristan Jehan introduced the concept of "music cross-synthesis". In his approach, he not only considers the raw audio data for analysis, he also considers the human perception and characterises audio files by their specific "audio DNA sequence" [27].

2.3 Existing Concatenative Music Synthesis Implementations

Beside the manual tape-based concatenative synthesis approaches in the 1950s, there have been quite a few computer-based or automated implementations of concatenative music synthesis, starting in the late 1990s. They can be sorted into a couple of categories according to their synthesis/selection methods. To illustrate this classification, examples will be given for the respective categories.

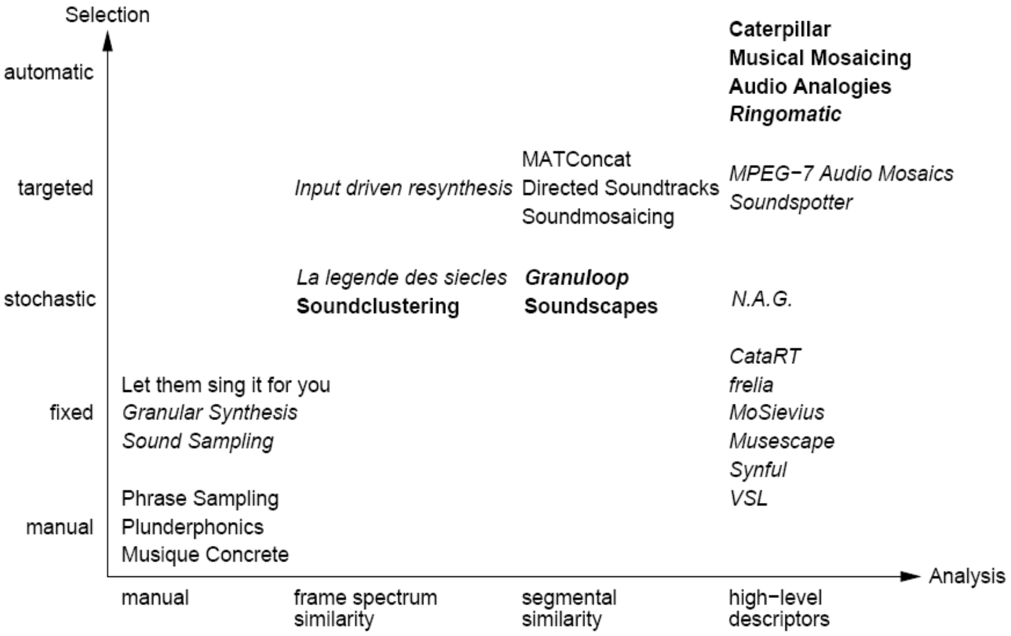


Figure 2.2 Concatenative music synthesis implementations [20]

2.3.1 Spectral Similarity

In this category, source frames are defined and matched to the target by analysing short-time spectra. Because the segments are so short (generally in the neighbourhood of a few milliseconds), there have to be selection rules governing the placement of fragments into the current context.

Kobayashi's "Sound Clustering Synthesis" achieves a consistent re-synthesis of classical music pieces by matching frame spectra using a vector-based function [20]. There are two constraints governing the re-synthesis placement of frames: the database frame must match the target frame and the transition between target frames and re-synthesized frames must be smooth.

2.3.2 Segmental Similarity

The segments in this category are selected by stochastic methods or by using similarity analysis based on low-level signal descriptors.

The freely available "Soundmosaic" interface [28] segments sounds by dividing them into two units and calculating their distance. Sound segments having the largest distance are switched, which implies a great amount of time spent searching and it slows the algorithm somewhat. The distance between segments is computed either by calculating the inner distance (the dot product) or by calculating the "Manhattan" or L1-distance.

Bob Sturm's "MATConcat" is a MATLAB-based Graphical User Interface [19]. It was the first system to compose electro-acoustic music pieces ("Concatenative Variations of a Passage by Mahler" and "Dedication to George Crum, American Composer"). The user can specify the desired segment length, after which a six-dimensional feature vector is computed which consists of low-level descriptors like RMS and spectral centroid values. The database frames are matched to the target frames by comparing the feature vectors.

2.3.3 High-Level Descriptors

The "MoSievius" system devised by Lazier and Crook [29] works in real-time and is based on looping sound segments. The user chooses descriptor ranges for classes like the spectral centroid, instrument or spectral flux manually and segments are picked when their descriptor values lie in the specified range. It also includes a control mechanism for real-time source selection called the "SoundSieve" that limits the search space by isolating sub-spaces of segments that have certain desired characteristics in common.

Given a monophonic recording, its MIDI score and a target MIDI score, the "Audio Analogies" [26] system attempts to synthesize audio to correspond to the target score. To ensure reaching exactly the right note pitch and duration, *pitch-synchronous overlap-add* techniques (PSOLA) are used to transform the audio material. The optimisation of the concatenation cost is solved by using a Viterbi algorithm to minimise a "match" cost function regarding the frame similarity and a "transition" cost function regarding the frame concatenation.

An elaborate implementation of concatenative music synthesis created by Schwarz [30], "Caterpillar" does not segment music by fixed analysis but by aligning the audio data with its score. Several continuous and discrete descriptors based on the MPEG-7 descriptor set [31] are computed. The unit selection algorithm uses a Viterbi algorithm to find the best frame match by trying to minimise two cost functions: a "target cost" describing the similarity of database and target segment and a "concatenation cost" specifying the "join quality" of two adjacent segments.

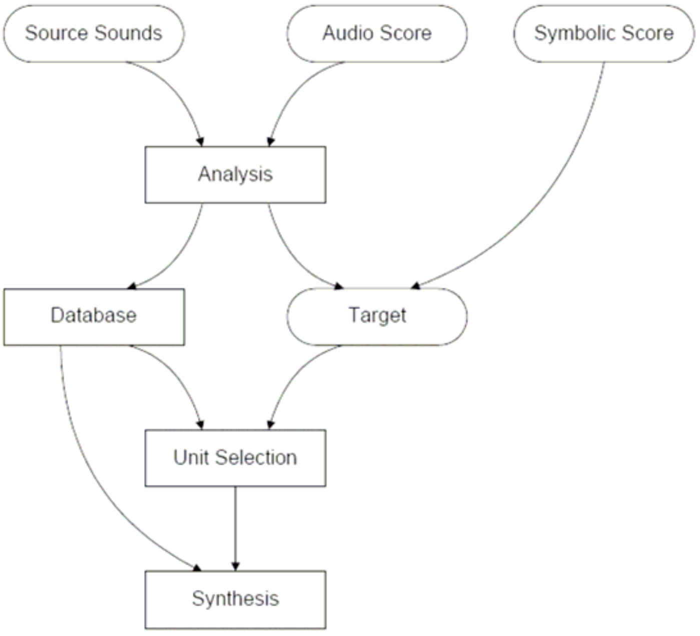


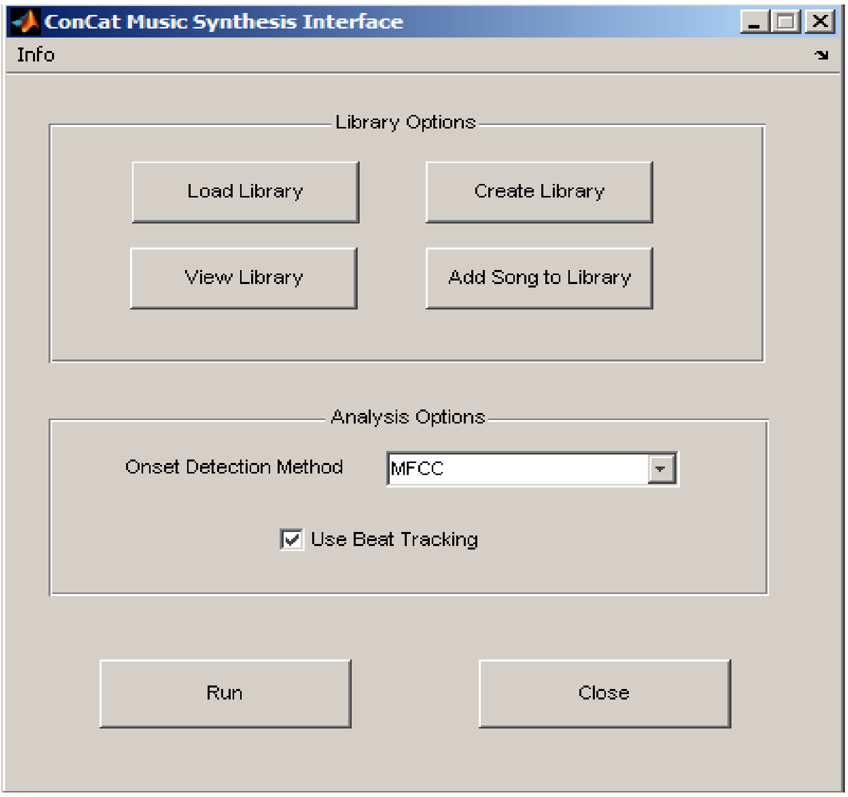
Figure 2.3 The "Caterpillar" system [30]

The "Caterpillar" data flow structure is representative for most concatenative music synthesis algorithms. While audio and symbolic scores are not needed by every algorithm, the principle of source sound analysis and selection and the subsequent synthesis and/or transformation always apply.

3 The ConCat Music Synthesis Interface

In this chapter, the ConCat Music Synthesis Interface is described in detail. The implementation in Matlab is discussed in Appendix C.

The ConCat Music Synthesis Interface is developed as a Graphical User Interface in MATLAB. This platform was chosen because of its platform compatibility and wide-spread use in academic institutions. The interface allows the user to create and edit audio databases and re-synthesize music from segments taken from these databases. It also allows the user to choose between different onset detection algorithms.



Onset Detection Method MFCC

☑ Use Beat Tracking

Figure 3.1 ConCat Music Synthesis Interface

The graphic tool depicted in Figure 3.1 serves as an interface to the underlying concatenative music synthesis algorithm. The user can choose the analysis options best suited to individual purposes and/or music styles.

3.1 Algorithm Structure

The first step necessary to perform concatenative synthesis is the creation of a sound database. The database can be hand-tailored to fit specific needs; for example, if the goal is to create a song that closely matches a target song, the database should be as large as possible, incorporating music from different genres and styles. If, on the other hand, a particular artistic concept is followed, it may be more useful to use smaller databases suited to the compositional needs. In any case, the size of the database is only limited by storage and computing time constraints.

Once a database is set up, it is possible to re-synthesize any given song or audio signal. This is done by searching the database for segments that are similar to the extracted segments of the target song. These segments are transformed if necessary and then concatenated to form a new song or audio signal that best matches the target song or signal.

3.1.1 Database Creation and Organisation

The database itself is created by analysis and subsequent segmentation in the time domain. The segment borders are located by performing a beat tracking analysis after a pre-processing stage where the signal is converted to mono format, filtered with a FIR (*Finite Impulse Response*) filter and down-sampled to 11,025 kHz in order to reduce computing time. To ensure correct onset detection at the start of the audio file, the file is zero-padded. The specific algorithm used for onset detection can be chosen among five possibilities, “Chroma” (based on the evaluation of energies in chroma bands, see section 4.2.3), “Complex” (an approach that evaluates amplitude and phase progression in the frequency domain [11], see section 4.2.4), “MFCC” (based on the trajectory of the first MFC coefficient, see section 4.2.5), “Modulation Spectrum 1” (based on short-time spectrum band trajectories [32], see section 4.2.6.1) and “Modulation Spectrum 2” (based on short-time spectrum band trajectories weighted according to characteristic drum frequencies [32], see section 4.2.6.2).

The following beat tracking algorithm based on a simple onset-interval histogram method can be turned off or on. This leads to musically meaningful audio fragments because the fragment length is highly correlated with the rhythmic structure of the audio signal. On the assumption that every frame starts with an onset and that therefore an "attack" phase exists where the signal is not deterministic, the audio segment is divided into two distinct regions.

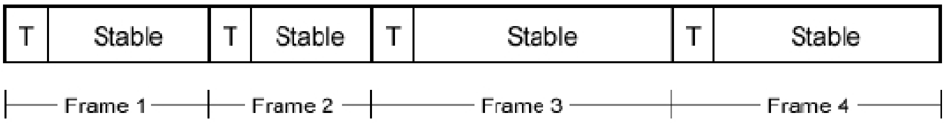


Figure 3.2 Frame segmentation with transient and stable regions [33]

The first region – the "transient" region - has a defined length in order to simplify calculations and is not analysed. The reason for this is that the following part of the audio signal is assumed to be the more characteristic and recognisable part of a sound. While it would be possible to extend the algorithm to include separate analysis of transient regions, this is omitted for simplicity and computing time reasons. The second region is defined as the duration between the end of the transient region and the start of the next frame. This region with variable length is assumed to be stable. After making sure the frame is at least 1024 samples (about 11 milliseconds) long to avoid dimension mismatch problems, it is windowed with a Hanning window to minimise spectral leakage effects and analysed in regard to descriptive audio features.

The segment start- and stop-time as well as the extracted features are stored in a data lookup table.

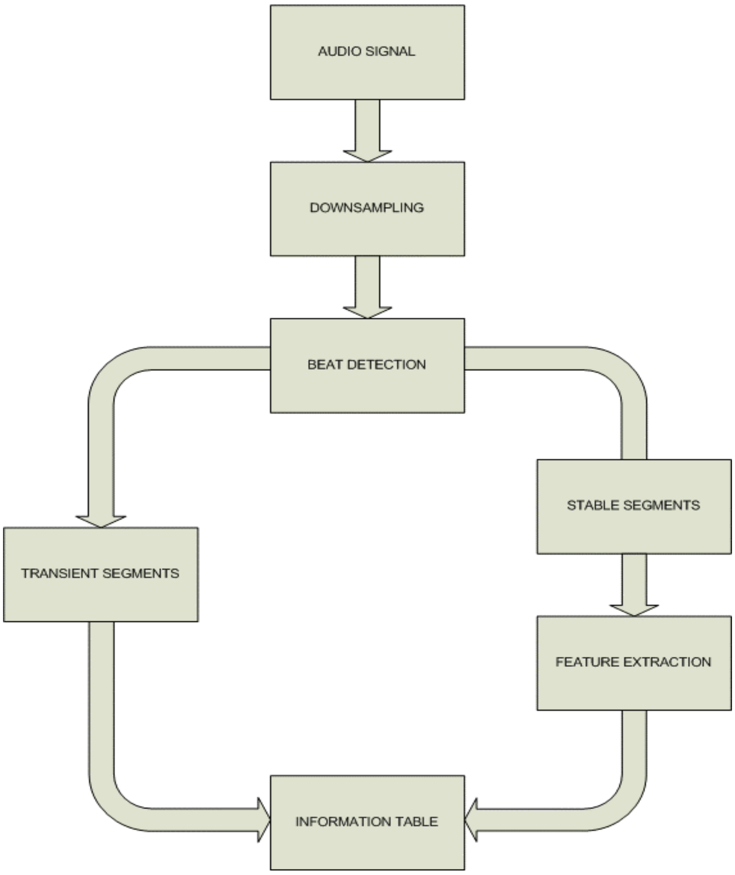


Figure 3.3 Segmentation of an audio signal into frames and extraction of signal parameters

Concerning the organisation of the database, the stored frames are sorted by perceptive criteria. The primary perceptive qualities of sounds are loudness, pitch, duration and timbre [34]. Since the loudness of frames is adjusted by the algorithm, the focus of the organisation criteria is on frame length and spectral attributes. To further reduce computation time and create a perceptually relevant database organisation, the stored frames that were sorted “chronologically” are rearranged by sorting them into *clusters* and *sub-clusters* according to the segment length and the segment pitch, thereby creating a three-dimensional cube containing the frame data. This means that sounds are represented not only by their mathematical and physical properties but also by their perceptual attributes [1]. This eliminates the need for time-intensive linear searching.

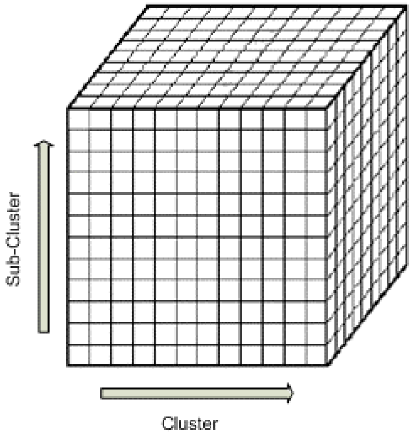


Figure 3.4 Database organisation with clusters and sub-clusters [33]

This shortens the re-synthesis computation time for a standard pop-song by a factor of about 10^3 to 10^4 (see e.g. [33] or section 7.2) while the time needed for the clustering is negligible in comparison. In this thesis, different clustering methods are evaluated to find the perceptually best-suited organisation method. The clusters are created according to either segment length or by pitch, the sub-clusters according to the spectral roll-off value [33] or by measures describing melodic content, such as chroma class, pitch and spectral centroid. Empirical evaluation of the re-synthesis results using the different organisation methods shows that the best results are obtained using length and a “melodic” parameter such as the pitch, chroma or spectral centroid parameter. Since one goal of the database organisation was to ensure uniform distribution of frames among clusters and sub-clusters, the chroma value was ruled out as organisation parameter. After evaluating a number of re-synthesized songs, the pitch value was chosen as the second organisation parameter.

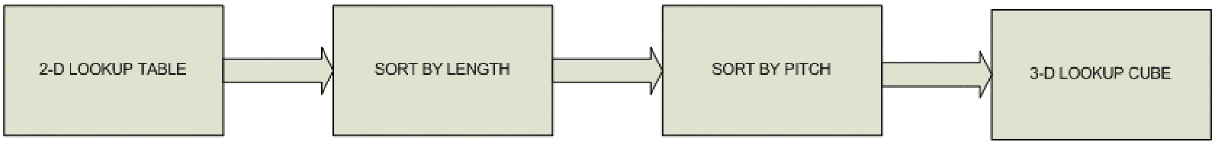


Figure 3.5 Database sorting

Ideally, the number of elements in the sub-clusters should be uniformly distributed to minimise searching time. For *n* clusters and *m* sub-clusters, the clustering algorithm finds *(n-1)* equally spaced boundary points between the shortest and the longest frame and *(m-1)* equally spaced boundary points between the frames with the biggest difference in pitch. The segments are then sorted into the respective clusters and sub-clusters by finding the boundaries nearest to the segment length (cluster) and the segment pitch (sub-cluster), respectively. The cluster and sub-cluster sizes can be varied depending on whether the goal is to minimise searching time or assignment errors - clustering represents a kind of pre-selection method with the above-mentioned advantages and disadvantages. In this thesis, the generic approach where the cluster and sub-cluster sizes are determined by the database size, thereby approximating uniform distribution, was selected.

It should also be mentioned that by using this database organisation, audio segments are stored in such a way that frames lie in close proximity to frames displaying similar characteristics, while different segments are separated by a greater distance. This mirrors the neurophysiologic organisation of the human and animal cortex [1].

Once a database is created or loaded, it is possible to re-synthesize a target song from existing database frames.

3.1.2 Synthesis

At the start of the synthesis algorithm, a target song representing the desired re-synthesis result is loaded. For obvious reasons, this target song cannot be part of the library.

The target song is analysed in the same way as the database songs. First, segmentation is performed by onset detection and - optionally - beat detection. The clustering algorithm detects the appropriate clusters and sub-clusters from where the re-synthesis audio frame should be taken.

The next task is to identify the frame in the selected cluster and sub-cluster that best matches the target song frame. This is done by computing a distance measure for every frame. The distance itself is computed by comparing the values of a feature set. A meaningful combination of features that are suited for this task was found by evaluating the subjective similarities of audio signals using a listening test which is described in chapter 6, resulting in a six-dimensional feature vector containing the first mel-frequency cepstral coefficient (MFCC), the zero-crossing rate, the pitch value and three features describing the statistical properties of the spectral distribution of the signal, namely flatness, skewness and kurtosis (for more information on these features, see chapter 5). The database segment that displays the highest similarity to the target song frame is selected and a pointer is created that shows the re-synthesis algorithm where to look when concatenating the audio segments. The Euclidean distance is used in this thesis for similarity evaluation. For two six-dimensional feature vectors, it is calculated as follows:

$$\Delta_s = \sqrt{\sum_{k=1}^{6} (x_k - y_k)^2} \tag{3.1}$$

where Δ_s is the segment dissimilarity (i.e. distance), *k* stands for the index of the feature, *x* for the database segment and *y* for the target song segment. If the database frame happens to be shorter or longer than the target song frame, it is scaled in a way that the segment durations match. This is achieved by using the fragment length adaptation algorithm described in section 3.1.2.2.

Target song frames that are purely transient[1] are handled separately. Those are not analysed and the re-synthesis algorithm skips them, leaving the frames themselves in place. Although this means that some segments of the target song are not replaced, which implies that no re-synthesis in the strict sense of the word takes place, this limitation was deemed necessary due to the reasons stated in chapter 3.1.1.

The information about the selected database fragments is gathered in a look-up table containing the frame position and current and desired length information as well as other necessary data.

Song	**T Start**	**T Stop**	**S Start**	**S Stop**	**Distance**	**Gain**	**Pad**	**Time-Scale**	**Target Length**
1	2755417	2756444	2756445	2769540	108,37	0,97	0	13,49	13095
7	26833	27860	27861	37116	0,65	2,13	0	2,31	9255
1	2371505	2372532	2372533	2373632	41,02	0,66	0	6,09	6695
12	7942617	7943644	7943645	7949060	15,60	0,68	0	2,12	5415
22	9333009	9334036	9334037	9339452	5,41	0,22	0	2,40	5415
3	879573	880600	880601	891136	10,29	1,35	0	44,83	10535
3	8357417	8358444	8358445	8359020	1546,18	2,26	0	11,64	6695
12	2371505	2372532	2372533	2373632	25,33	1,17	0	3,76	4135
4	2371505	2372532	2372533	2373632	24,25	0,65	0	4,93	5415

[1] in the segment, there is no stable region that can be analysed

3	7942617	7943644	7943645	7949060	3,99	0,79	0	2,12	5415
1	879573	880600	880601	884736	19,16	0,48	0	17,60	4135

Table 3.1 Excerpt from a song information look-up table

Table 3.1 shows such a generated look-up table. The data in the columns is used to re-synthesize the target song. Included are the database song number (“Song”), the start and stop points of the transient signal part (“T Start” and “T Stop”) in samples, the start and stop points of the stable signal part (“S Start” and “S Stop”) in samples, the calculated feature distance (“Distance”), the necessary gain adjustment factor (“Gain”), the number of needed zeros for padding (“Zero Pad”) as well as the time-scaling factor (“Time-Scale”), if necessary, and the target song length (“Target Length”).

After all data that is necessary for re-synthesis purposes is gathered, the found synthesis frames are processed to minimise synthesis artefacts and to ensure a consistent volume curve and a correct frame time adjustment.

3.1.2.1 Volume Adaptation

For consistent volume management, it is important to take into account the fact that the volume information about the segment is extracted from the stable region of the segment, but the frame is concatenated as a whole, including the transient region that was not analysed in the process. Therefore, a gain curve has to be used that adaptively adjusts the gain across the transient and stable region of the frame. This is realised by using a so-called “Smart Gain” curve that reaches its maximum at a position in the stable region where no damage to the overall audio signal can be done by over-emphasising or blurring the transient region [33]. The curve is designed with the idea in mind to place the volume curve slope between the transient region start and a position well after the stable region start. The gain is defined as 1 (= 0 dB) at the start of the transient region and reaches half the desired total gain at the start of the stable region. Thus the full gain is reached some time after the stable region start.

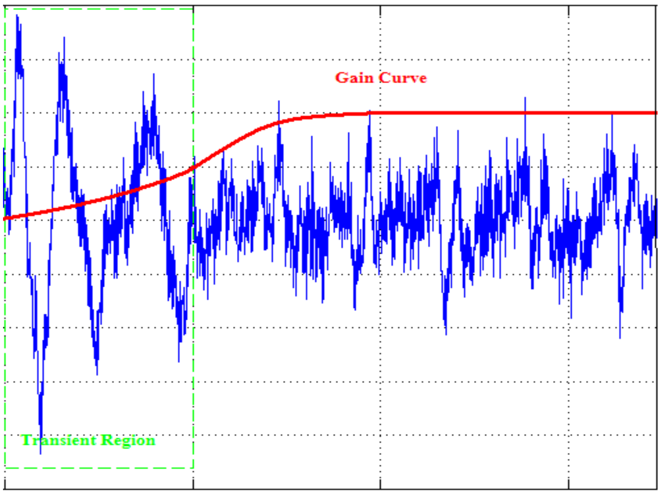


Figure 3.6 Smart Gain adjustment

3.1.2.2 Fragment Length Adaptation

In most cases, it will be necessary to adapt the database frame length to the target frame length. If the database segment is too short, two possibilities present themselves: the length can be changed by *looping* or by *stretching*. In this thesis, the “stretching” approach is implemented because simple looping of fragments leads to perceived discontinuities in the sound. This happens because listeners are generally accustomed to a “release” phase of continuous sounds that cannot be reproduced by simple looping methods and more complex looping approaches were found to be beyond the scope of this thesis. If the database segment is too long, it is simply cut at the appropriate points.

To ensure smooth segment transitions, the frames are faded in and out by applying half a Hanning window to the first and last millisecond. The segments are then concatenated with an overlap of one millisecond.

The time stretching algorithm is based on the *TimeScaleSOLA* algorithm presented in [35] adapted to work with stereo signals [33].

The algorithm is based on correlation methods. The audio signal to be stretched is divided into segments of equal length. These are shifted according to a *time scaling factor* which corresponds to variable overlap block lengths. The overlapping areas are then cross-correlated to find the position where the similarity between the blocks is highest. This corrects the primary positions of the blocks in respect to each other but makes a second comparison of database and target song frame lengths necessary. The blocks are faded in and out at the maximum similarity points and are superposed sample-wise.

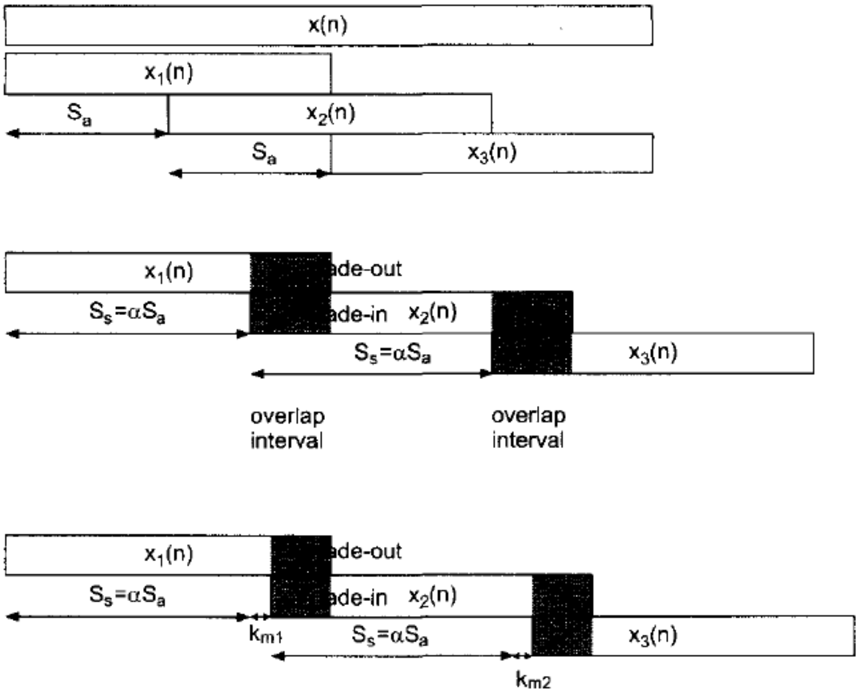


Figure 3.7 TimeScale SOLA algorithm [35]

Figure 3.7 shows an example of time-stretching applied to a signal, which is divided into three segments. These are moved subsequently into overlapping positions and the correlation between the overlapping areas is calculated.

4 Beat Tracking

Beat tracking is an important part of the concatenative music synthesis algorithm used in this thesis because it ensures a perceptually meaningful segmentation of database and target audio material.

In this chapter, some of the existing beat tracking methods are presented. The main focus will lie on beat tracking systems that are based on onset detection in the spectral and in the time domain, as well as approaches that are based on probabilistic signal models. The onset detection methods and the *inter-onset interval*[1] beat tracking system implemented in this thesis are explained in detail.

Music is a non-stationary process where different “events” in regard to melodic and rhythmic structure present themselves. Human listeners can detect these events easily - even untrained listeners are able to track beat structures by “tapping along” using hands or feet. Noticeable changes in pitch or intensity serve as indicators of such musical events happening. The changes have to happen in a significantly short time to be classified as events, thereby distinguishing them from simple and gradual changes in sounds that occur naturally as a result of tone decay or modulations. This thesis concerns itself only with the chain of events defined as the “tempo” of a music piece without taking into account the grouping into bars and higher-order structures and the relationships between the distinct events that are normally associated with the “rhythm” structure of a piece.

The need for reliable and fast beat detection systems arose with the advent of automatic music analysis systems and the areas where they are applied have increased considerably over the last years. These include harmonic analysis [5], database management and indexing [6] and audio signal transformations, including digital audio effects [8].

Many beat tracking systems use *transcriptive* methods, where the beat is estimated by first detecting discrete events ("onsets") and the results are used in a later stage to group together the distinct onsets to beat structures [2]. Another approach is the one devised by Scheirer [2], who tries to arrive at a beat estimate without having to rely on a "transcription" stage.

In this thesis, the term "onset" will be used when referring to the one time instant that marks the start of a transient event in an audio signal.

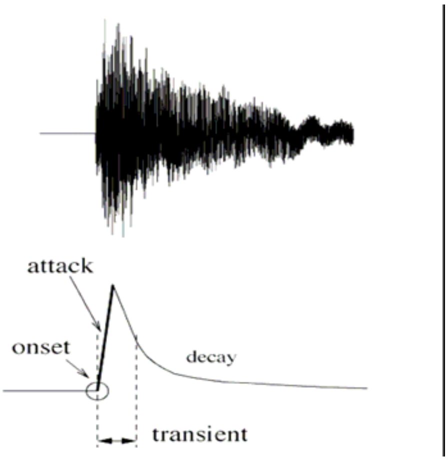


Figure 4.1 Onset definition [3]

4.1 Existing Beat Tracking Systems

In this section, an overview of existing beat tracking methods is presented. In the first part of the section, beat tracking systems that use signal information extracted in the time domain are presented, in the second part spectral domain methods are discussed. A brief overview over systems that are based on stochastic models is given at the end of this section.

The basic structure of the beat tracking algorithm is similar in many approaches. After an optional pre-processing stage, the original audio signal is transformed into a (for the most part highly down-sampled) version called a detection function that exhibits the characteristic behaviour of the signal. From this detection or novelty function, onsets are derived by peak-

[1] the time between successive onsets

picking algorithms. The found onsets are then analysed to yield beat information and, in some cases, rhythm information.

The pre-processing stage is used in some beat tracking algorithms to accentuate properties that are useful for finding onset information or to increase the reliability of the detection method. A widely used approach is to divide the audio signal into a number of sub-bands and to analyse the signal in the respective frequency bands. Examples of this technique can be found in [2], [9] or [36]. Some authors also use techniques derived from research concerning music signal modelling by separating transient from stable regions using *Spectral Modelling Synthesis* (SMS) or *Transient Modelling Synthesis* (TMS) ([37], [38]). This approach considers the difference (*residual*) between the audio signal and a signal derived from a SMS model fed with the signal parameters. Sudden energy increases in the residual are interpreted as a deviation of the signal from the spectral model, marking transient events [10]. The TMS model can be considered as an extension to the SMS model based on the *discrete cosine transform* (DCT) of the residual signal [38].

The post-processing stage generally consists of a peak-picking algorithm that extracts local maxima indicative of onsets from the detection function. Some systems also use an optional stage where the detection function is manipulated to facilitate this task, e.g. by removing noise with low-pass filters or normalising the function. By using a threshold, local maxima are picked out and assumed to be onsets. The threshold can be fixed, which means that all local maxima above the threshold are taken to be onsets. This approach only works when the signal does not exceed a certain dynamic range. By using a smoothed, i.e. low-pass filtered version of the detection function as an adaptive threshold, the dynamic of the signal does not impact the detection function [3].

4.1.1 Time Domain Approaches

Early approaches to beat tracking took advantage of the fact that, especially in music signals with percussive elements, the signal amplitude increases significantly in a short period of time when a transient event occurs. By following the amplitude envelope or, in other cases, the energy amplitude envelope (both of which can be easily calculated by rectification and subsequent low-pass filtering of the signal or the squared signal), onsets can be reliably detected when working with strongly percussive monophonic or non-complex music signals. Such an envelope follower for a signal *x[n]* can be described as follows:

$$E[n] = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} |x[n+k]| w[k] \tag{4.1}$$

where *w[k]* is a finite-length window function.

Variations on these systems include working with the time derivative of the signal energy or the logarithm of the signal energy or a combination thereof, resulting in sharper peaks (local maxima) in the detection function.

Klapuri evaluates the signal change in relation to the signal level [9], which corresponds to the differentiation of the logarithm of the amplitude envelope.

$$d[n] = \frac{d(\log(E[n]))}{dt} \tag{4.2}$$

This is psycho-acoustically motivated – according to Weber's law, the smallest change in the intensity of a signal that can still be detected is constant and related to the respective signal intensity:

$$\frac{\Delta I}{I} = const. \tag{4.3}$$

This transformation results in well-defined peaks in the detection function and resolves the problem of blurred peaks in a detection function that is based solely on the envelope information.

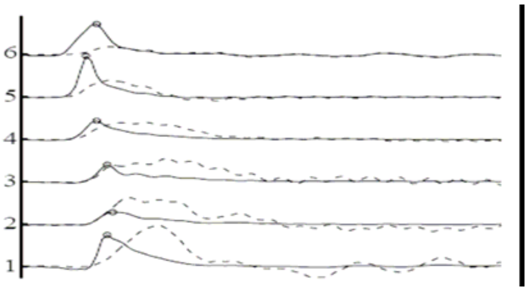


Figure 4.2 Differentiated envelope (dotted line) and differentiated logarithm of envelope (solid line) [9]

4.1.2 Frequency Domain Approaches

Approaches in the spectral domain have been found to be successful even with complex polyphonic audio signals. This approach has also proved to be computationally effective [3].

A special case of this approach is transient detection using wavelets. Daudet [39] uses *dyadic wavelet decomposition* of the residual of a signal. The transient events are linked to the larger wavelet coefficients and form “structures” across the dyadic plane. A *regularity modulus* describing the regularity of the signal is used as detection function.

4.1.2.1 Magnitude / Energy Information

The fact that transient events are generally linked to a broadband energy increase is exploited to extract onset information from the audio signal spectrum. The major part of audio signal energy is usually located in the lower part of the frequency spectrum. Therefore, energy increases in high frequency bands are indicative of onsets.

By weighting higher frequency bands proportionately higher than lower frequency bands, approaches based on instantaneous short-time spectra have been found to perform well when used to analyse strongly percussive signals (*High Frequency Content* approach, [12]). The mathematical expression of this approach is given by

$$E[n] = \frac{1}{N} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} |k| \cdot |X_k[n]|^2 \tag{4.4}$$

where *k* stands for the bin index. By using the absolute value of the bin index as a weighting function, the spectrum is linearly emphasised towards high frequencies.

Other methods use the differences in the spectra between successive short-term analysis frames to extract onset information from audio data. This can be done by considering short-time spectra as points in a multidimensional space and computing the “distance” between those points. Different distance measures have been used in literature: for example, Masri [12] uses the *L1* or *Manhattan* distance norm, Duxbury [36] the *rectified L2* distance norm.

$$L2[n] = \sum_{k=0}^{N} \left[H\left(|X_k[n]| - |X_k[n-1]| \right) \right]^2 \tag{4.5}$$

H stands for the rectifying function that is equal to zero for negative values. This has the effect that only increases in the signal energy are considered, while energy decrease information is discarded.

Foote [6] builds a *similarity matrix* by correlating short-time power spectra and applies a "checkerboard" kernel to the matrix, thereby detecting boundaries between areas of high and low similarity.

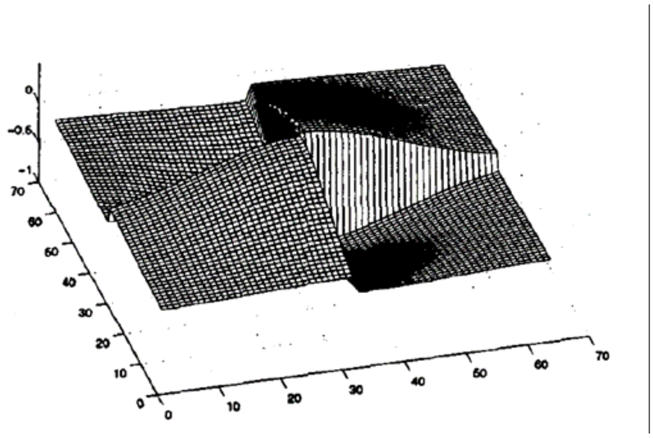


Figure 4.3 Checkerboard kernel with tapered edges to avoid edge effects [6]

An advantage of this method is that by varying the size of the kernel, different characteristics of the signal can be investigated. If the kernel is small, events on a small time-scale such as onsets are detected. Larger kernels work well to detect higher-level signal structures such as repeated bars or verse/chorus segments.

4.1.2.2 Phase Information

Recent approaches to onset detection have made use of the fact that information about the temporal structure of a signal is contained in the phase spectrum [11]. For a stable sinusoid signal, the change of phase of distinct spectral bins is expected to remain constant over adjacent short-time spectral frames. If the actual phase value deviates significantly from the expected value, a transient event probably has happened. By analysing the distribution of the deviations, transient events can be detected.

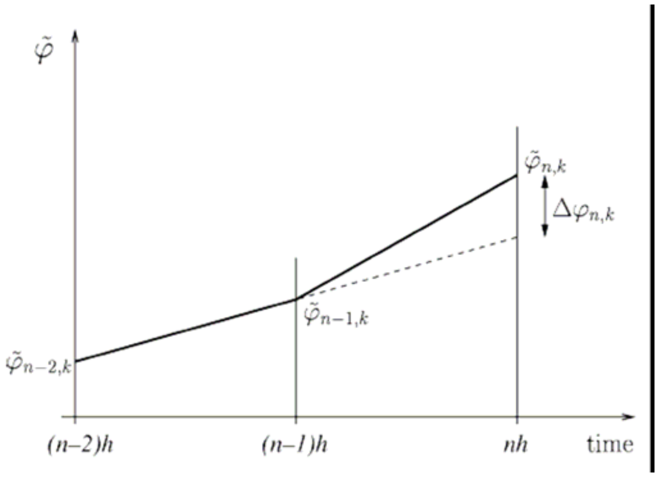


Figure 4.4 Deviation of expected and actual phase between successive short-time analysis frames [3]

4.1.2.3 Phase and Magnitude Information

An approach that takes into account phase information as well as magnitude information has been proposed by Bello [11] and is implemented in this thesis. The stationarity of separate spectral bins is determined by calculating the Euclidean distance between the observed complex Fourier coefficients and the coefficients of previous short-time analysis frames. These distances are summed over the frequency band to generate an onset detection function. The approach is explained in detail in section 4.2.4.

4.1.3 Probabilistic Approaches

Stochastic models of signals have been used to extract beat information from audio signals. By analysing the quality of "fit" of the model to the signal itself, i.e. the similarity between the assumed and the actual probability distributions, information about transient events can be gathered [3]. This approach is heavily dependent on a high-quality signal model.

The samples of the signal *x(n)* can be described as belonging to one of two signal models, in this case either a transient or a stable state [3]. The relationship between the respective probability density functions is used to define a *log-likelihood ratio*

$$s = \log \frac{p_b(x)}{p_a(x)} \tag{4.6}$$

The model expectations are calculated by computing the *Kullback-Leibler distance* between the model probability distribution and the observed probability distribution. Sign changes in the short-time average of the log-likelihood ratio are taken to be indicative of a model change. The log-likelihood ratio could therefore be described as a binary detection function.

Another approach [40] relies on one single global model to describe the signal. Once the system has been trained, it can identify stable regions while transient events come as a "surprise" to the system. The "surprise" can be mathematically evaluated by following the *negative log-probability* of the signal.

$$s = -\log p\left(x(n) \middle| \{x(j), j < n\}\right) \tag{4.7}$$

4.1.4 Comparison

According to their reduction method, the presented onset detection algorithms exhibit certain strengths and weaknesses, depending also on the audio signal type that is analysed. While methods in the time domain are simple to implement and take little time, their accuracy decreases when the audio signal is complex or polyphonic or when amplitude modulations take place.

Approaches in the spectral domain take a longer time to compute, yet can generally be described as more robust. When faced with percussive signals, magnitude-based methods work quite well, especially when the frequencies are weighted towards higher frequencies and the energy changes are detectable in the whole spectral band. However, the detection rate deteriorates when the signal contains parts without significant energy increases, such as soft onsets or string instruments playing *legato*. These can be quite reliably detected by onset detection methods that use phase information, where the tonal qualities of the signal are evaluated. The drawback of these methods is the susceptibility to phase distortions stemming, for example, from noisy signal components or distortions introduced in the recording process.

The onset detection method using wavelet regularity makes time resolutions of only a handful of samples possible, which is better than the time resolution of the human hearing system. However, the resulting detection functions are very noisy, thereby making extensive post-processing necessary.

Probabilistic onset detection methods work well for a wide range of signals when the used signal model is of high quality. On the other hand, the training process necessary to train the algorithm can be very time-consuming.

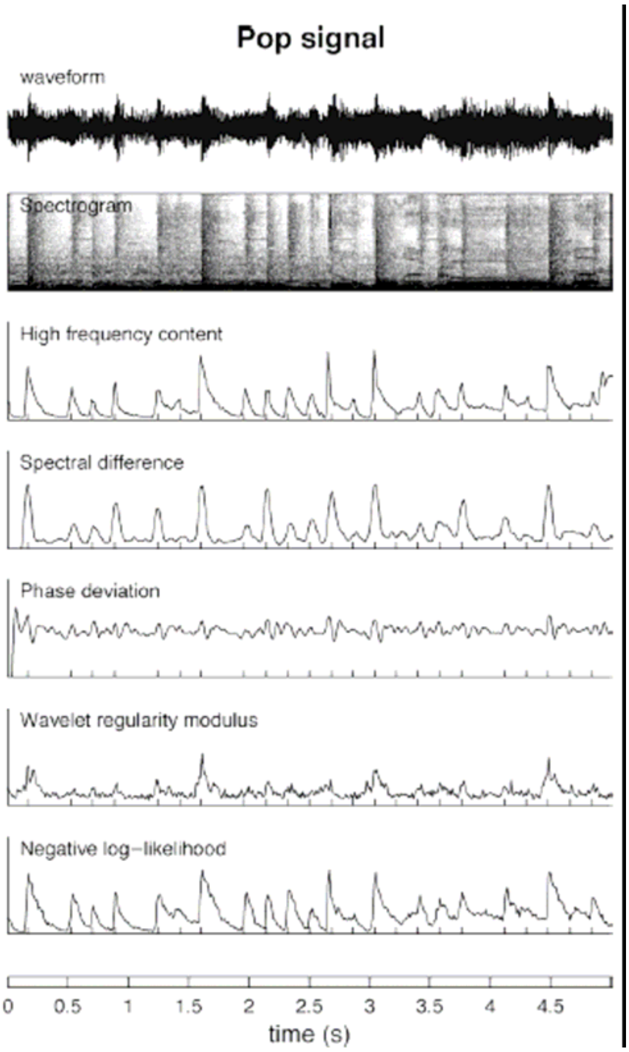


Figure 4.5 Comparison of different normalised detection functions for a pop song [3]

In the above picture, different onset detection functions are shown. The differences in the quality and noisiness of the detection functions are clearly visible. A smooth function with clearly defined peaks generally means a small number of false onset detections and a possibly higher number of missed onsets, while less smooth curves increase the probability of finding all onsets and at the same time the probability for false detections. This means that the onset detection function should be chosen according to the needs of the application that is to be implemented. For some applications, for example time-stretching algorithms, it may be more important to find all onsets, even if it means accepting a high number of false detections. For others, e.g. beat tracking, the minimisation of false detections is paramount.

4.2 Implemented Beat Tracking System

For a perceptually meaningful segmentation of audio signals, a robust beat tracking system is critical. Only by reliably detecting the boundaries of fragments that make sense musically can the concatenative music synthesis concept produce acceptable results.

In this thesis, a beat tracking system is implemented by finding onset times in audio signals and subsequently evaluating the resulting inter-onset interval times to form a beat hypothesis. This hypothesis is then used to enhance the results obtained by the onset detection algorithm.

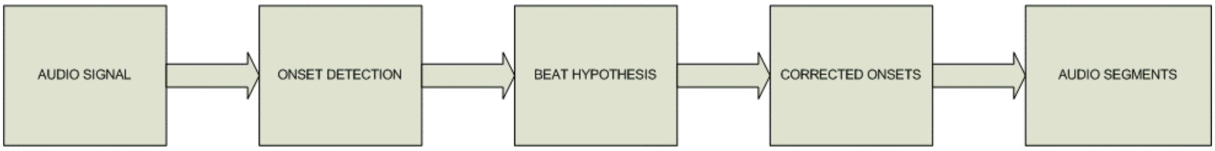


Figure 4.6 Beat tracking system

In this section, the principal structure of the beat tracking algorithm is presented. The individual onset detection algorithms as well as the beat-tracking algorithm employed are explained in detail.

4.2.1 Onset Detection Algorithms Structure

All the onset detection algorithms presented in this paper follow the same general structure: The audio signal to be analysed is pre-processed, afterwards the signal is reduced to an intermediate representation called detection function. From this, the exact onset time locations are extracted by a peak-picking algorithm that selects local maxima that fulfill certain criteria.

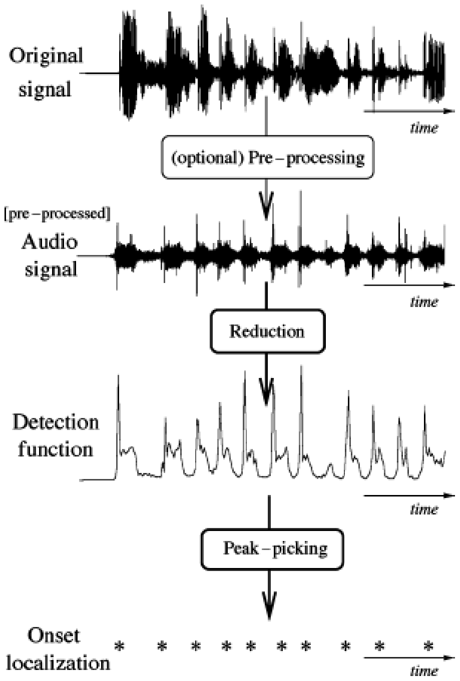


Figure 4.7 Onset detection systems workflow [3]

The pre-processing stage limits the original signal to one channel by converting stereo signals to mono format. The signal is also low-pass filtered with a standard FIR filter and down-sampled to reduce the data volume.

The processed signal is then reduced or transformed into a representation that exhibits the characteristics of the original audio signal while emphasising the data traits that facilitate onset detection. This detection function exhibits a number of local peaks that are evaluated in the peak-picking stage to find the exact onset locations. The detection function is created by evaluating the energy changes in chroma classes ("Chroma-based Onset Detection", section 4.2.3), by calculating the change of Fourier coefficients in the complex frequency domain ("Onset Detection in the Complex Frequency Domain", section 4.2.4), by following the mel-frequency cepstral coefficients ("MFCC-based Onset Detection", section 4.2.5) and by locating changes in the *modulation spectrum* sub-band trajectories ("Onset Detection based on Modulation Spectra", section 4.2.6).

The peak-picking algorithm then chooses the peaks that are most likely to be onset events. The locations of the peaks are forwarded to the beat tracker, which in turn determines the likely beat of the signal and processes the onset time information accordingly. This is done by evaluating the inter-onset intervals.

4.2.2 Pre-Processing

The first stage in onset detection algorithms consists of a pre-processing stage where the audio signals are transformed in order to conform to a standard signal representation. This ensures that onset and beat information about signals can be compared even when the signals' characteristics like channel number and sampling rate are different.

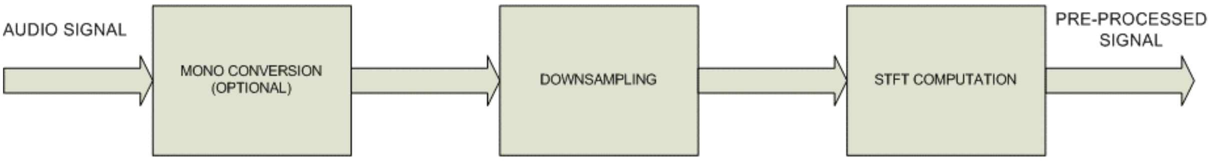


Figure 4.8 Pre-processing

The audio signal is limited to one channel by converting the information from stereo channels into a mono channel, if necessary.

In order to reduce calculation time while preserving most of the information about the signal, it is down-sampled to 11.025 kHz. An added benefit of this computation is the standardisation of the sampling rate for all audio signals to the above rate. This is achieved by using the pre-defined Matlab function *decimate* (from the *Signal Processing Toolbox*). This function uses an 8th-order Chebyshev low-pass filter before the actual decimation stage.

A well-known and widely used representation of a signal in the time-frequency domain is the *Short-Time Fourier Transform* (*STFT*), a method that computes the signal spectrum during short time windows. The window is set to a fixed length of 11.6 milliseconds (128 samples at 11.025 kHz) and a hop-size[1] of 5.8 milliseconds (64 samples at 11.025 kHz). To avoid spectral leakage, the signal is windowed using a Hanning window. For better spectral resolution, the STFT length is expanded to 23.2 milliseconds (256 samples at 11.025 kHz) by simple zero-padding.

[1] The duration between the start of successive time windows

4.2.3 Chroma-based Onset Detection

This chapter introduces the concept of *chroma* as opposed to *tone height* or pitch. The energy change in different chroma bands or classes is used for an onset detection function implemented in this thesis that will be presented thereafter.

The term "chroma" is used to distinguish the cyclic frequency relationships illustrated in the circle of fifths from the tone height itself. Tone height maps an increase or decrease in perceived pitch caused by an increase or decrease in signal frequency to a linear scale, while the chroma or *pitch class* describes the circular nature of interval relationships that repeats itself for every octave [14]. The difference between the two concepts is illustrated by the so-called *Shepard scale*, where sinusoidal signals, separated by octaves, are superimposed. By progressively changing the base pitch of the signal, an auditory illusion of constantly rising or falling pitch is created. Another example of this phenomenon is that a chord that is played in different ways is still perceived as the same chord because the discrete tones that make up the chord stay in the same pitch or chroma class.

The chroma scale itself consists of twelve distinct values corresponding to the twelve semitones without enharmonic equivalents represented in the circle of fifths. When two or more tones are found to belong to one chroma value, this implies that the tones are one or more octaves apart from each other. This leads to a *chroma helix* that looks like the circle of fifths from above. The pitch height is represented by the vertical changes in the helix while the chroma is constituted by its rotation.

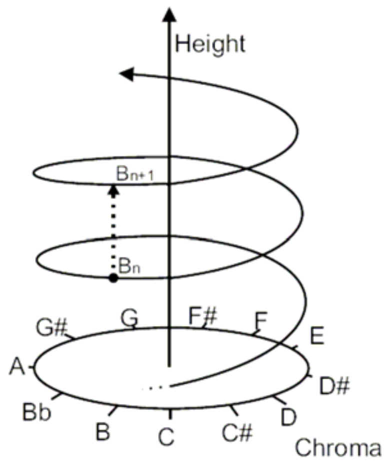


Figure 4.9 Chroma helix [41]

The tracking of chroma values as a twelve-dimensional vector over time leads to a time-frequency representation of the signal called *chromagram* [41] or *Harmonic Pitch Class Profile* (HPCP).

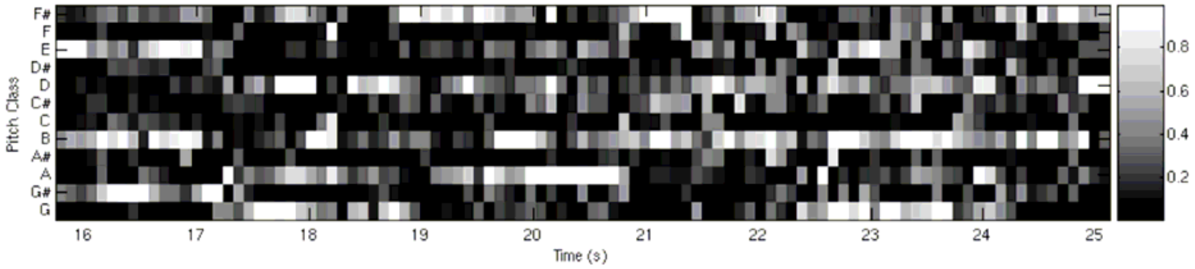


Figure 4.10 Semitone-quantised chromagram (Beatles, „Eight Days a Week") [41]

So, by mapping short-time signal spectra onto a chroma scale, the energy of the signal is compressed from the number of STFT bins used to 12, leading to a very compact representation of the signal. This can also be used for similarity evaluation purposes by comparing the chroma values concentrated in 12-dimensional feature vectors to compute a measure of similarity. Bello [5] combines the chroma representation with *Hidden Markov Models* (HMM) to determine harmonic content and larger-scale segments in music signals. Other authors rely on chroma detection to estimate the key or the chord structure of a music piece [41] or to create similarity matrices from which large-scale song segments such as chorus or verse can be inferred [14].

4.2.3.1 Approach Overview

The fact that the circularity of the chroma definition is intuitively clear for every musician makes it a useful mid-level representation of audio [5]. Changes in the chroma structure of an audio signal indicate the presence of onsets. Even if the same note is played twice in succession, there will be a short time during the attack of the second tone that the chroma value will jump from the defined value of the first note to a value determined by the sharpness of the onset event, thus evidencing the presence of an onset. The implementation of a chroma-based onset detection function presented in this thesis takes advantage of this fact by following the energy distribution in the respective chroma bands over time. After some pre-processing, the STFT of the signal is calculated and mapped onto a twelve-point chroma scale. The energies in the chroma bands are summed and their change is tracked to create a detection function which has to undergo some post-processing, after which the local maxima are taken as onsets.

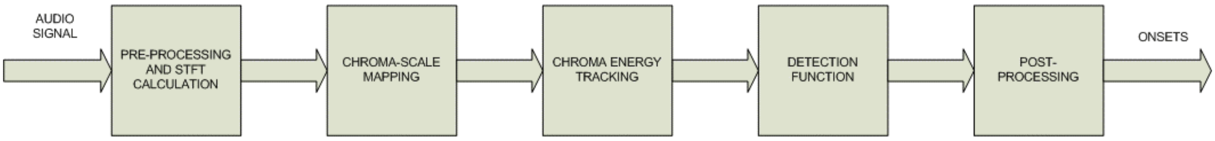


Figure 4.11 Chroma-based onset detection

The first stage of the onset detection function is constituted by the pre-processing and STFT calculation stage that is explained in detail in section 4.2.2.

A chroma axis reaching from the E0 tone at 41 Hz up to 5 kHz is then created by using the just temperament to create the respective chroma boundaries. The upper boundary is set at 5 kHz because the majority of the signal energy is concentrated in the lower frequency bands. The STFT bins are then mapped to this chroma scale according to their frequency value while taking care that the difference between the STFT bin frequency and the chroma frequency scale does not exceed 15 cents (this corresponds to a deviation of approximately $1/7$ of a semitone).

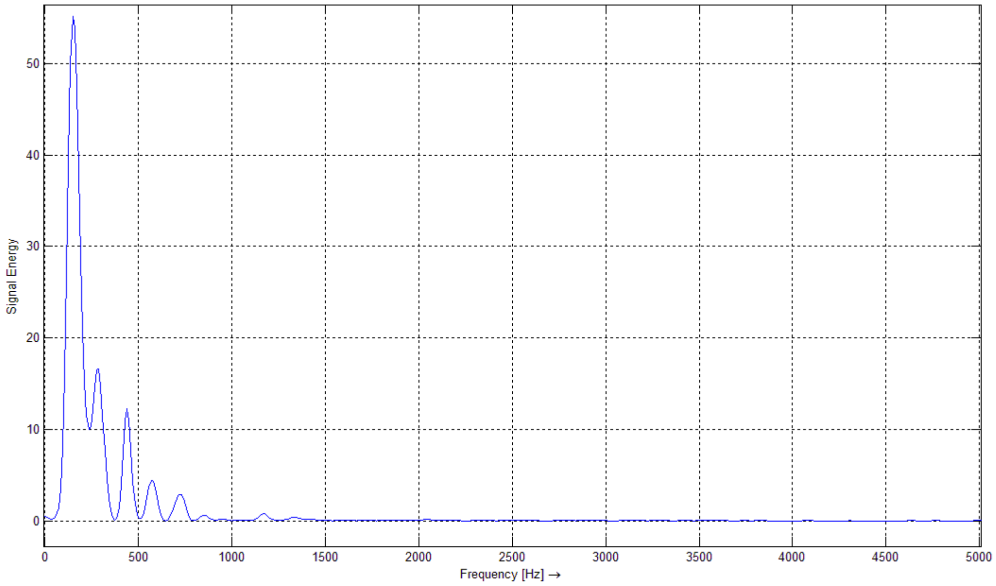


Figure 4.12 Energy spectrum of one STFT frame (Cream, „Sunshine Of Your Love“) showing the concentration of the signal energy in the low frequency range

The above figure shows the energy spectrum, i.e. the squared magnitude spectrum of an audio signal. The energy is visibly concentrated at the lower end of the spectrum. By sorting the energies at distinct frequency bands into the appropriate pitch class, the chroma energy distribution is obtained.

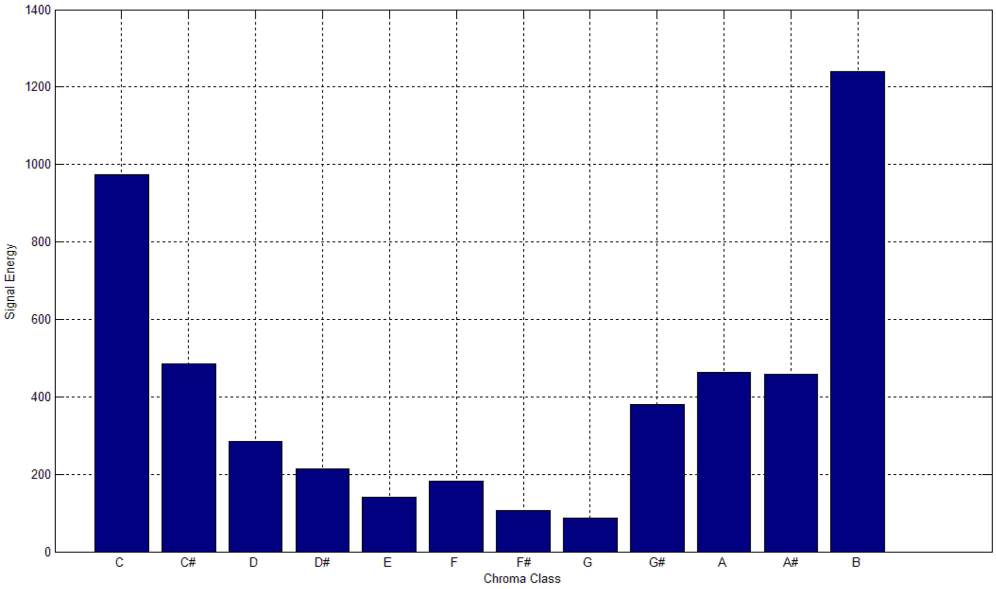


Figure 4.13 Energy distribution in chroma classes for one STFT frame (Cream, „Sunshine Of Your Love“)

By evaluating the energy distribution using musical knowledge about harmonies and chord formations, assumptions concerning the harmonic structure of the analysed signal can be made [41]. For example, Figure 4.13 shows a concentration of the signal energy around the B tone, suggesting a well-defined tonal center of the current STFT frame.

To avoid false detections due to signal segments where there is little energy, i.e. pauses or silent regions, frames containing less energy than a fixed threshold (in this case, a threshold of 1% of the maximal energy found in a frame of the signal works quite well) are disregarded by the algorithm.

In a later stage, a *chroma vector* is created where the name of the chroma class containing the maximum of the energy present in every frame is stored, thus leading to a description of the tonal progression of the signal over time.

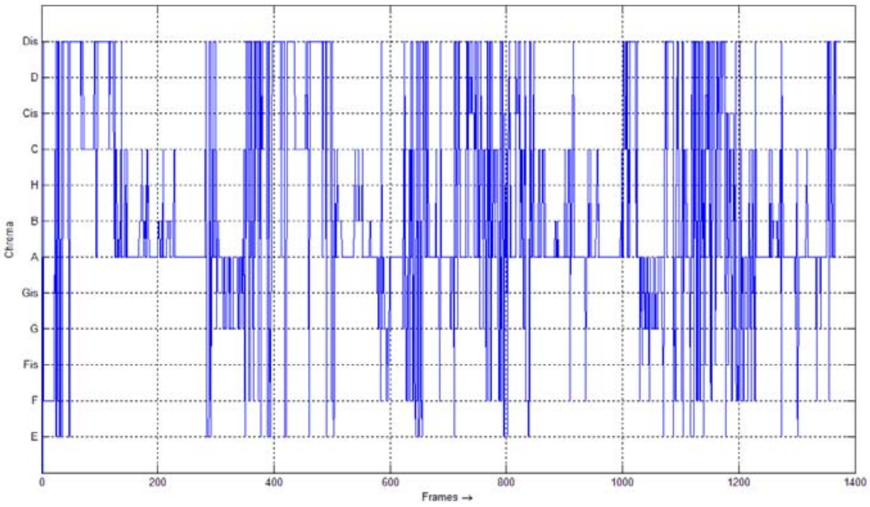


Figure 4.14 Chroma progression (Cream, „Sunshine Of Your Love“)

The detection function itself is created by calculating the ratio of the energy present in tonal components to the total energy contained in every STFT frame.

$$r[m] = \frac{E[m]}{E_c[m]} \tag{4.8}$$

with m as the frame number, $E[m]$ as the total frame energy and $E_c[m]$ as the tonal energy contained in the detected chroma class. Highly tonal components of audio signals, for example instruments playing sustained notes, will tend to have a ratio close to one because a major part of the signal energy will be concentrated around the fundamental frequency, while onsets that are characterised by a flat, broadband spectrum will tend to have much larger ratios as the energy of the spectrum will be equally divided across all chroma values. From this information, a detection function that shows maxima at probable onset locations can be created.

Two ways to create a detection function are explored: in the first, the ratio of tonal energy is tracked over time. After some low-pass filtering to eliminate noise and spurious peaks, the ratio is differentiated and half-wave rectified, which amounts to determining the positive changes in the energy ratio. The second method uses an approach presented in [42] where the ratio is first compressed with a logarithmic function motivated by the fact that humans detect signal intensity changes in relation to the signal intensity itself. Then a weighted sum is calculated of the compressed, low-pass filtered ratio and a half-wave rectified differential of the compressed, low-pass filtered ratio.

The first approach to the detection function calculation uses a simple Butterworth low-pass filter[1] to smooth the signal, thereby removing many spurious maxima. This results in a lower number of falsely detected onsets but has the drawback of missing some onsets. As onsets are accompanied by positive energy changes, only large positive changes in the filtered ratio are of interest. The changes are determined by calculating the first-order difference of the energy ratio, i.e.

$$r'[m] = r[m] - r[m-1] \tag{4.9}$$

This differential ratio is then half-wave rectified to create the final detection function as follows:

$$df[m] = \frac{\left(|r'[m]| + r'[m]\right)}{2} \tag{4.10}$$

[1] The Matlab function *filtfilt* filters signals with the previously determined filter coefficients with no phase distortion by running the reversed filtered signal back through the filter.

The second approach to the creation of the detection function follows the compression and combination of ratios used in [42]. To create a perceptually relevant representation, the data is first compressed using μ-law compression:

$$r_c[m] = \frac{\ln\left(1+\mu r[m]\right)}{\ln\left(1+\mu\right)} \tag{4.11}$$

with μ as the compression factor that can be adjusted to provide near-linear transformation for small values of μ and near-logarithmic transformation for large values of μ. The compression behaves generally linear near zero but logarithmic for higher values. This is due to the fact that the human hearing system can discriminate changes in a signal in proportion to the signal itself. As in the previous approach, the compressed ratio is then low-pass filtered to smooth the ratio curve by removing irrelevant peaks, resulting in a smoothed ratio $r_{c,s}$. The same differentiation and half-wave rectification described above is also performed in this approach.

$$\tilde{r}_{c,s}[m] = \frac{\left(\left|r'_{c,s}[m]\right| + r'_{c,s}[m]\right)}{2} \tag{4.12}$$

The overall detection function is computed by performing a weighted summation of the half-wave differential version and the previous, smoothed ratio curve:

$$df[m] = \left(1-\lambda\right) r_{c,s}[m] + \lambda \frac{f_r}{f_{LP}} \tilde{r}_{c,s}[m] \tag{4.13}$$

where f_r is the frame rate (i.e. the decimated sampling frequency) and f_{LP} is the cut-off frequency of the low-pass filter used to smooth the signal. λ is the factor that determines the balance between the two transformed energy ratios and the factor $\frac{f_r}{f_{LP}}$ compensates the small amplitude of the differential energy ratio [42].

The balance factor λ was set to 0.8, however, any value above 0.5, i.e. with a higher emphasis on the differentiated, rectified ratio works well. The onset detection method is evaluated with and without this compression and combination approach. The results are detailed in section 4.3.

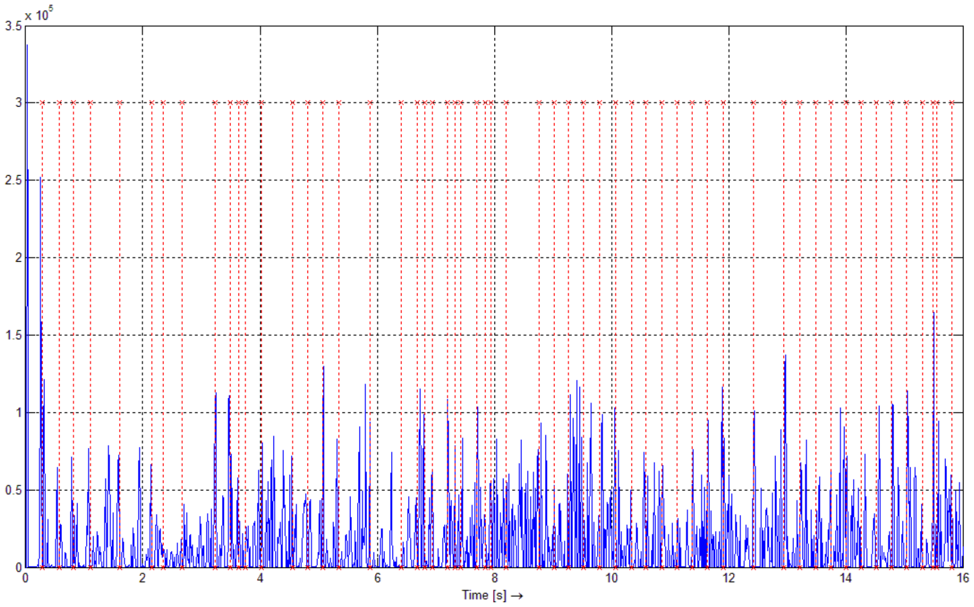


Figure 4.15 Chroma-based onset detection function (blue) with labelled onsets (red) (Cream, „Sunshine Of Your Love")

The last stage of the algorithm consists of a peak-picking part where local maxima are picked out using an adaptive threshold to determine likely onsets. For more information on peak-picking refer to section 4.2.7.

4.2.4 Onset Detection in the Complex Frequency Domain

In most cases, an onset event is accompanied by an increase in the energy of the signal. The phase values tend to change abruptly as well. Some onset detection methods track the signal energy, others the phase curve to detect transient events.

In this section, the relationships between energy and phase changes during transient events and stable periods are briefly investigated. Afterwards, the implementation of Bello's [11] approach to onset detection used in this thesis is presented.

4.2.4.1 Spectrum Change at Transient Events

To investigate the relevance of changes in the energy and phase spectrum in regard to transient events, different audio signals are analysed accordingly.

Every audio signal is analysed using the STFT. The signals are segmented into 23.2 millisecond frames. These are then windowed with a Hanning window to minimise spectral leakage and zero-padded to a length of 46.4 milliseconds for better frequency resolution. Then the *Fast Fourier Transform* (FFT) of the frames is computed, with an overlap of 11.6 milliseconds between successive frames.

The energy changes as well as the unwrapped phase changes that happen from frame to frame are calculated and the changes are combined by computing the mean energy and phase changes in ten octave frequency bands covering the whole spectrum. The changes that take place between frames that contain no onsets are compared to the changes between a stationary and the successive transient frame.

The onsets are labelled by hand, using the Sound Onset Labeliser [43] and the resulting onset times are used to mark frames where onset events take place. The used audio material is taken from popular music. The following figures illustrate the differences in the spectrum change when a transient event is present and when there is no onset, respectively.

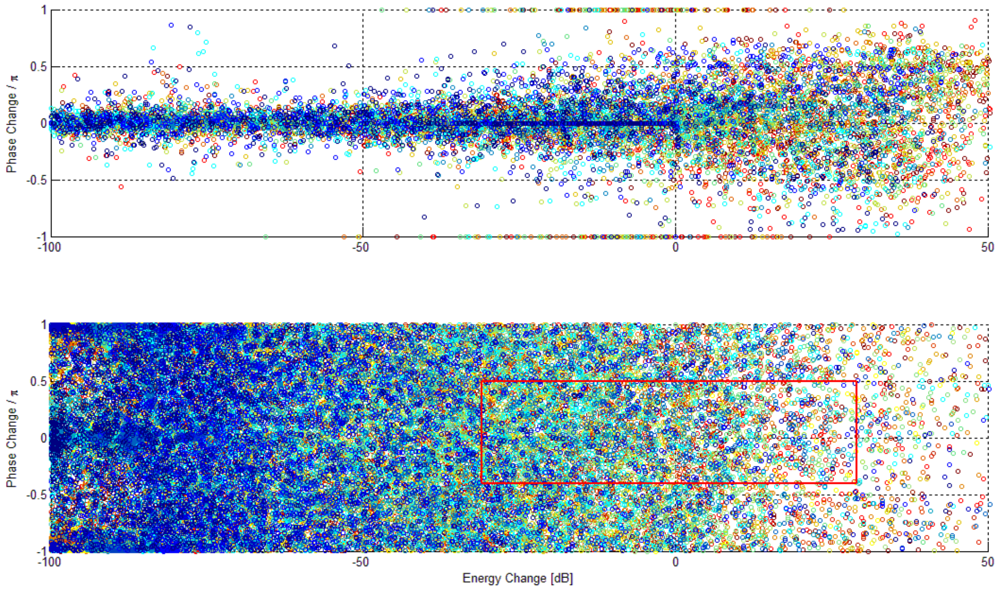


Figure 4.16 Mean energy and phase change between stationary frames (top) and between frames containing transient events (bottom) (Cream, “Sunshine Of Your Love”)

Figure 4.16 (top) shows the normalised unwrapped phase change and the logarithmic energy change when there are no transient events present. Every point in the figure stands for the mean change in an octave band of the signal spectrum between successive frames containing no onsets and every colour stands for such a frame pair. The majority of changes is concentrated around 0 dB energy change and 0 rad/π phase change. In contrast, Figure 4.16 (bottom) shows the spectrum changes when a stable frame is followed by a frame containing an onset. Again, the points denote the mean changes in octave bands between frames and the different colours stand for the respective frame pairs. The red rectangle represents the region where 90% of the change values of frames lie when there is no onset present. The spectrum changes are much more pronounced and spread over a larger area. This supports the assumption that onset events are for the most part accompanied by major changes in the phase and energy spectrum, respectively. As the size and location of the red rectangle shows, the spectrum change is much less evident when a stable frame follows another, while there is still some energy and phase change due to, for example, modulations and decay effects. While 90% of change values in regard to energy and phase change between stable frames lie in the rectangle, the percentage of change values between stable and onset frames lie at 17% for energy changes and 32% for phase changes.

For further illustration of this fact, another example of a modern pop song analysed in regard to energy and phase changes in the spectrum is given below.

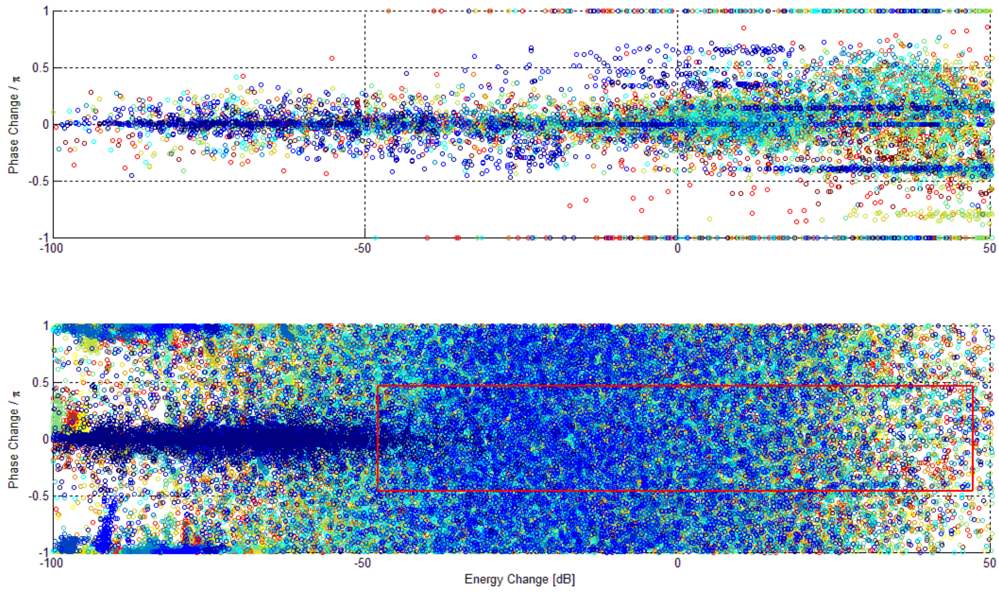


Figure 4.17 Mean energy and phase change between stationary frames (top) and between frames containing transient events (bottom) (Muse, “Muscle Museum”)

As in Figure 4.16, the unwrapped phase change between frames containing no transient events (top) and between frames containing transient events (bottom) is plotted versus the logarithmic energy change. As in Figure 4.16 (top), few changes in the phase spectrum are higher than $\left[-\frac{\pi}{2},\frac{\pi}{2}\right]$. While there are some outliers concerning the energy spectrum above 50 dB, the major part is concentrated between 0 dB and approximately 30 dB.

In Figure 4.17 (bottom), the space taken up by the majority (90%) of frame change values without an onset - again denoted by the red rectangle - is significantly smaller than the space occupied by the change values when an onset event is preceded by a stable frame. However, in contrast to the first example (Cream's "Sunshine Of Your Love"), the energy changes between stable frames are not markedly different from those between a stable and a transient frame – 58% of energy change values of the latter also lie in the red rectangle. This suggests that, at least for this particular audio signal, a phase-based onset detection method is likely to work better than an energy-based method. However, this fact cannot be generalised, as the following example shows.

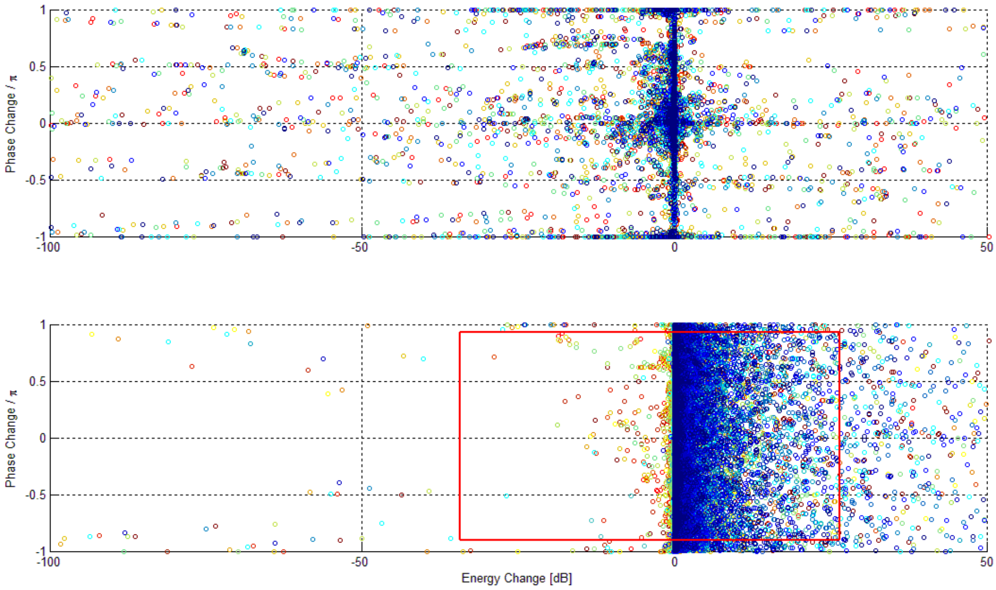


Figure 4.18 Mean energy and phase change between stationary frames (top) and between frames containing transient events (bottom) for a simple drum track

The upper part of Figure 4.18 depicts the change between stationary frames. The spread of the energy and the phase of stationary frames is much more pronounced than in the previous pictures.

The lower figure depicting the change between stable frames and frames containing transient events shows that the majority of phase change between stable frames reaches up to $\left[-\pi, \pi\right]$. This implies that a phase-based onset detection will probably not yield satisfactory results because phase change is, in this case, not a reliable indicator for transient events. In fact, the region where 90% of the phase changes between stable frames lie (denoted by the red rectangle) is also occupied by 88% of the phase changes of stable-to-transient regions. The ratio concerning energy changes lies at 90% for stable frames to 66% for stable-to-transient regions. The figure suggests that for this audio signal, an energy-based onset detection method should produce better results than a phase-based method.

This section shows that onsets are accompanied by significant changes in the energy and phase values of the STFT spectrum. It also demonstrates that for some audio signals, the phase changes are more pronounced than the energy changes while for other signals it is the other way round. So, while for certain audio signals phase-based onset detection methods produce good results, for others energy-based methods seem better suited. To avoid having to adapt the onset detection algorithm to the signal, it seems a good solution to combine the two approaches by considering the phase as well as the energy of the spectrum for onset detection. This was proposed in [11] and implemented in this thesis and is presented in the next chapter.

4.2.4.2 Approach Overview

In many types of music, especially in modern pop and rock music, the introduction of a new note (onset) will be accompanied by a sharp increase in the signal energy. This is especially true for signals with strong percussive elements.

By computing the first-order difference of the signal energy, which corresponds to the energy change over time, onsets can be identified by picking out local maxima. Another possibility for detecting changes in the magnitude or energy of the signal is given by computing the STFT of a signal *x[n]* weighted with a sliding window *w* as

$$S_k[m] = \sum_{n=-\infty}^{\infty} s[n]w[mh-n]e^{-j\frac{2\pi nk}{N}} \tag{4.14}$$

with k as the frequency bin index, $k = 0,1,...,N-1$ and m as the STFT frame index, and comparing the difference between successive frames. These approaches work well for simple, percussive signals (see also section 4.2.4.1) and are fast and easy to implement.

For other types of signals, it may be more suitable to find onsets by working with the phase information of the signal spectrum. By evaluating the difference between the expected and the actual phase value, onsets can be detected when the deviation between these values is high. This method works well with non-percussive signals containing so-called “soft” onsets, i.e. onsets that do not exhibit major energy changes, such as a violin playing *legato*. However, phase distortions present in the signal can lead to the deterioration of detection results.

Bello [11] presents a method where phase and magnitude information of a signal are combined in the complex frequency domain to create an onset detection function. In doing so, the advantages of the respective methods are combined – the ability of the phase approach to find “soft” onsets is combined with the robustness of the magnitude approach and its ability to find percussive onsets.

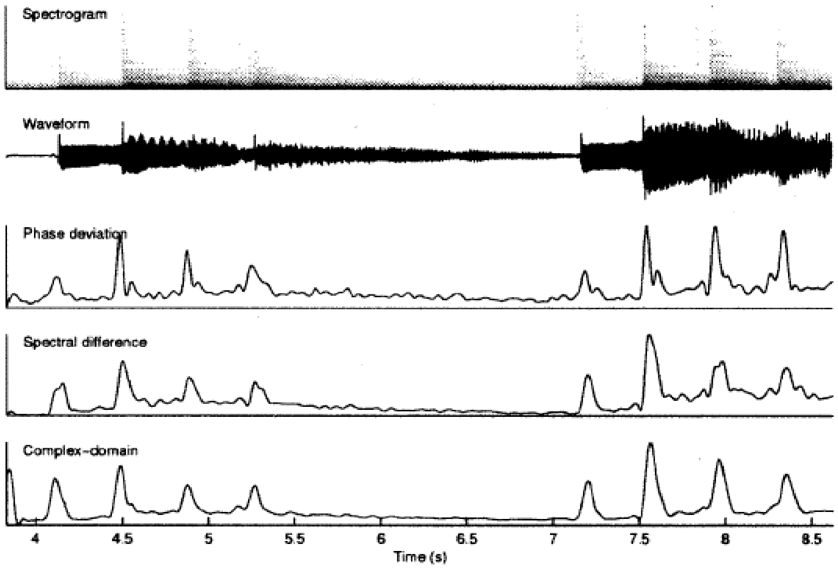


Figure 4.19 Spectrogram and waveform of an audio signal with the onset detection functions of the phase-based approach (upper middle), the magnitude-based approach (lower middle) and the combined approach in the complex frequency domain (bottom) [11]

As seen in Figure 4.19, this approach leads to well-defined local maxima in the detection function corresponding to onsets.

The algorithm implemented in this thesis uses a pre-processing stage to standardise the signal representation. In later stages, the STFT of the signal is computed, which serves as a basis for the onset detection algorithm. From the STFT values, the signal representation in the complex frequency domain using magnitude and phase is extracted. A detection function is then created by considering the frame-wise changes in the real and the imaginary parts of the expected and the detected magnitude and phase values of the spectrum. The detection function then is post-processed and local maxima in the function are accepted as onset events.

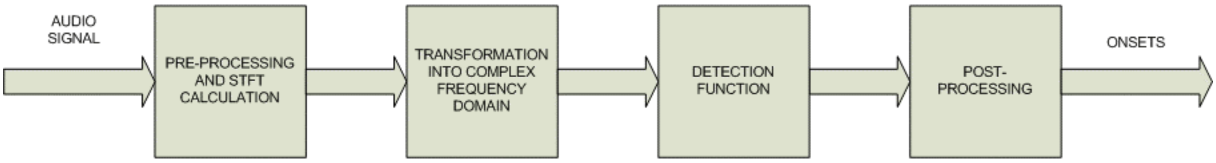

AUDIO SIGNAL
PRE-PROCESSING AND STFT CALCULATION
TRANSFORMATION INTO COMPLEX FREQUENCY DOMAIN
DETECTION FUNCTION
POST-PROCESSING
ONSETS

Figure 4.20 Onset detection in the complex frequency domain

The audio signal that is to be analysed is first pre-processed and its STFT representation is calculated (for more information, refer to section 4.2.2).

The expected combination of spectral magnitude and phase for the *k*-th bin of the STFT is given by

$$\hat{S}_k[m] = \hat{R}_k[m]e^{j\hat{\phi}_k[m]} \tag{4.15}$$

where $\hat{R}_k[m]$ is the expected magnitude value and should, for stationary frames, equal the magnitude of the previous frame

$$\hat{R}_k[m] = \left|S_k[m-1]\right| \tag{4.16}$$

and $\hat{\phi}_k[m]$ is the expected phase value, calculated as the sum of the unwrapped phase of the previous frame with the unwrapped phase difference of the preceding frames

$$\hat{\phi}_k[m] = princ\arg\left(2\varphi_k[m-1] - \varphi_k[m-2]\right) \tag{4.17}$$

On the other hand, the actual magnitude and phase for the *k*-th bin is given by

$$S_k[m] = R_k[m]e^{j\phi_k[m]} \tag{4.18}$$

The measure for the stationarity for the *k*-th bin of a signal between two successive frames can be computed by calculating the (Euclidean) distance between the actual and the expected complex vectors:

$$\Gamma_k[m] = \sqrt{\left\{\left[\Re(\hat{S}_k[m]) - \Re(S_k[m])\right]^2 + \left[\Im(\hat{S}_k[m]) - \Im(S_k[m])\right]^2\right\}} \tag{4.19}$$

This equation can be simplified by rotating $\hat{S}_k[m]$ onto the real axis, which means setting the expected phase value $\hat{\phi}_k[m]$ to zero.

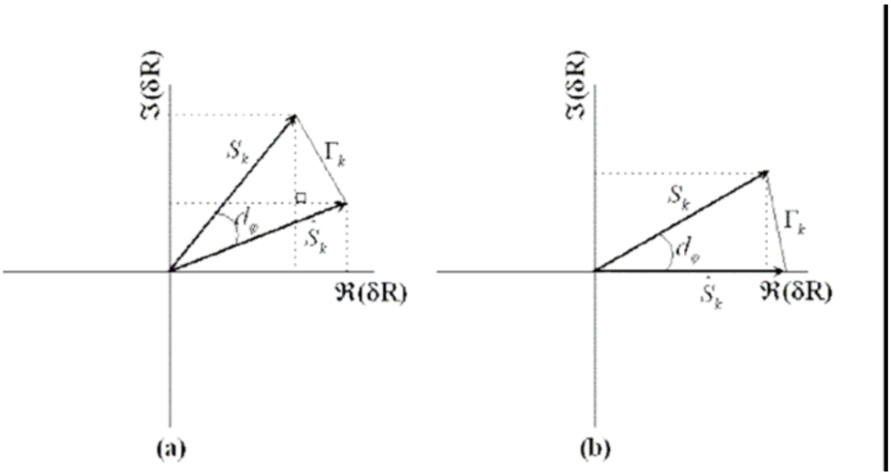


Figure 4.21 Deviation between actual and expected bin in the complex frequency domain, (a) normal and (b) rotated on the real axis [11]

This means that $S_k[m]$ can be rewritten using the phase deviation value $\Delta_{\phi,k}[m]$ explained above:

$$S_k[m] = R_k[m]e^{jd_{\phi,k}[m]} \tag{4.20}$$

It can be shown that $\Gamma_k[m]$ equals the spectral difference measure $\Delta S[m]$ if and only if the phase deviation value is equal to zero [11]. This means that the phase behaves as expected (the difference between successive frames remains constant) and only the energy difference is used to create the detection function. Otherwise, the phase value also influences the detection function.

The detection function itself is constituted by concatenating the complex distance measures for each frame into a continuous function as described by

$$df[m] = \sum_{k=1}^{N} \Gamma_k[m] \tag{4.21}$$

The detection function is then processed using a low-pass filter to smooth the function and remove any unwanted spurious peaks. Then local maxima are picked out by an adaptive peak-picking algorithm. For more information on post-processing and peak-picking see chapter 4.2.7.

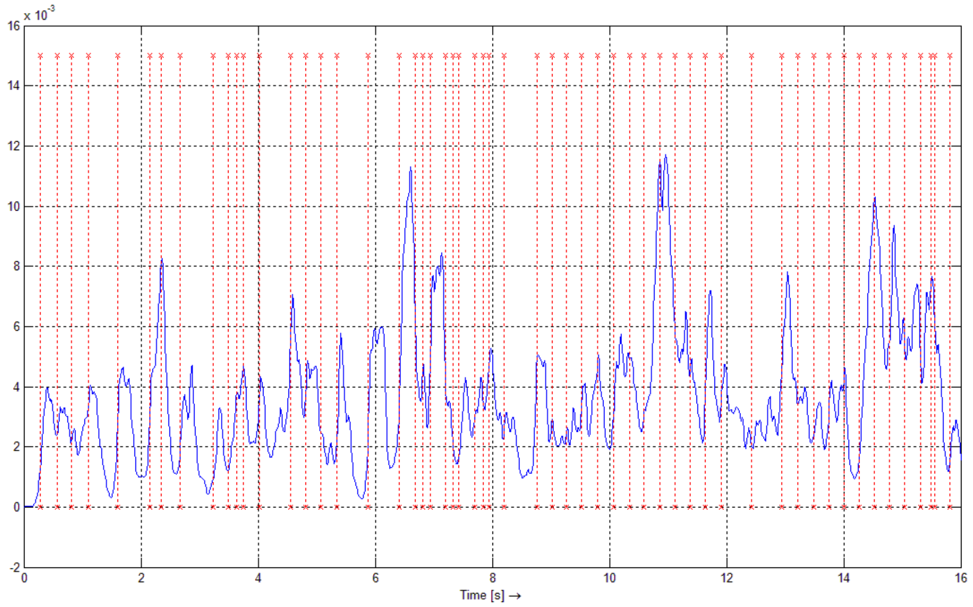


Figure 4.22 Complex frequency domain detection function (blue) with labelled onsets (red) (Cream, „Sunshine Of Your Love“)

4.2.5 MFCC-based Onset Detection

In this chapter, a brief explanation of Mel-Frequency Cepstral Coefficients (MFCCs) is given. For a more detailed description, refer to section 5.2.1. Then the implemented onset detection that works by following the MFCCs over time is presented.

Mel-Frequency Cepstral Coefficients represent the spectral characteristics of a signal in a very compact way. They are used heavily in speech processing applications, mainly in speech recognition and speech coding algorithms where the cepstral domain is useful for the extraction of the spectral envelope and the separation of signals [44]. The resolution of the MFCCs varies according to the chosen number of coefficients.

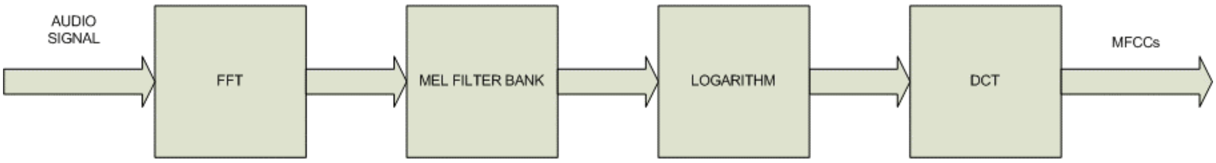


Figure 4.23 Mel-Frequency Cepstral Coefficient calculation

The calculation of the coefficients is realised as follows: first, the magnitude spectrum of the signal is determined, for example by using the FFT. The magnitude spectrum is then filtered by a *Mel filter bank*, which is a group of triangular filters that fulfills the purpose of grouping together frequency components according to the Mel scale, which is based on the human perception of pitch distances. The logarithm is then computed over the summation of the frequency groups. This mirrors the behaviour of the human cochlea, where neuronal impulses are evaluated in frequency groups, resulting in an integration of the impulses. In the last stage of the calculation, the values obtained from the filter bank are transformed into the *cepstral domain* using the *Discrete Cosine Transform* (DCT) [45].

4.2.5.1 Approach Overview

The idea behind the onset detection algorithm is to track MFC coefficients over time. Two methods were implemented: in the first one, only the first MFCC is tracked (see section 5.2.1); in the second one, *linear regression* is used to determine a weighting vector which is applied to the MFCCs to generate a detection function.

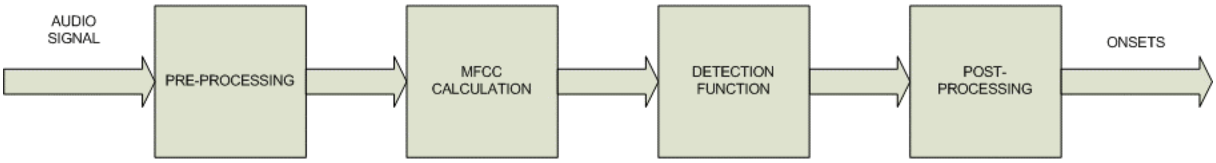


Figure 4.24 MFCC-based onset detection function structure

The standard pre-processing stage consists of converting the signal to mono format if necessary and down-sampling to 11.025 kHz (for more information on the pre-processing stage, refer to section 4.2.2).

The signal is then divided into overlapping equal-length frames, for each of which the MFCCs are calculated. The MFCC computation is explained in detail in section 5.2.1.

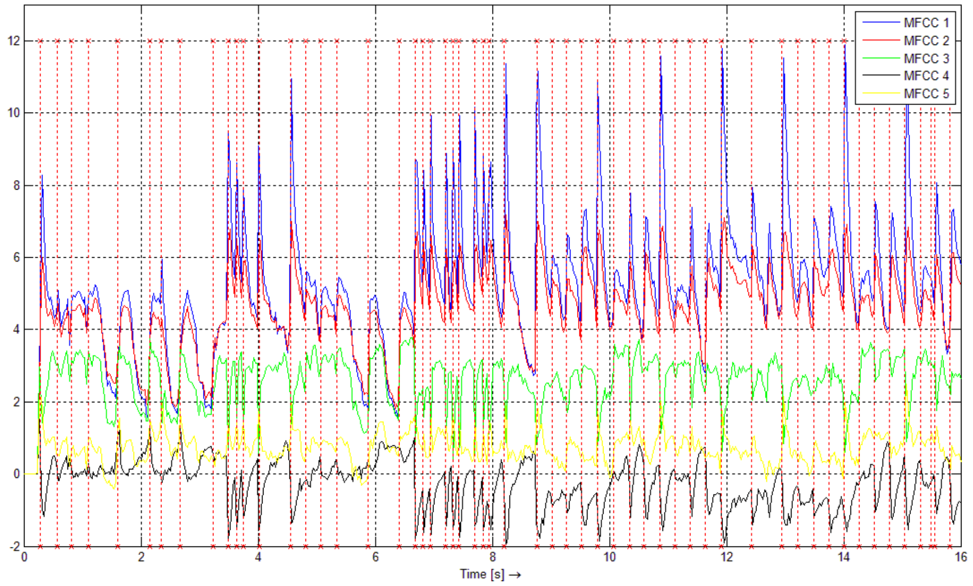


Figure 4.25 First 5 MFC coefficients and labelled onsets (red markers) (Cream, „Sunshine Of Your Love“)

Two different approaches are explored in order to create a detection function from the MFCC data: by using signals where the onsets are known (they were hand-labelled using the Sound Onset Labeliser GUI [43]), linear regression is employed to try to find a suitable weighted combination of MFCCs that represent a well-tuned detection function; the second method used is to follow the first MFCC over time.

Linear regression is a mathematical method that describes the relationships between a dependent variable and any number of independent variables, which is assumed to be a linear function. A simple example of linear regression could be

$$y(x) = c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_n x_n + \varepsilon \qquad (4.22)$$

where *y(x)* are the dependent variables, $c_1...c_n$ stand for constant values, *ε* for a random term, $x_1...x_n$ are the independent variables and *n* determines the number of parameters to be estimated. More complex linear regression models also incorporate mixed and quadratic terms of x_n. However, for this thesis the reliance on only first-order independent variables was deemed to suffice.

In matrix notation, the above equation can be written as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1n} \\ 1 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{4.23}$$

or, in simplified notation,

$$\vec{y} = \vec{x} \cdot \vec{c} + \vec{\varepsilon} \tag{4.24}$$

To estimate the weighting coefficients $\vec{c}$, following equation is used:

$$\vec{c} = \left(\vec{x}^T \vec{x}\right)^{-1} \vec{x}^T \vec{y} \tag{4.25}$$

This means that by knowing the independent variables, which in our case are the trajectories of the MFCCs over time, and the dependent variables - in our case the exact points in time when onsets appear - it should be possible to find a good weighting vector that can be used to create a detection function. The dependent variable $\vec{y}$ is modelled by creating a pulse train with pulses at the times of labelled onsets and convolving the result with a Hamming window of 50 millisecond length. However, it was found that the first approach using only the first MFC coefficient works as well or better with most signals, especially percussive ones, whereas the second approach using a weighted sum of MFC coefficients has the drawback that it has to be “trained” before use because the weighting factors change when using different signals, therefore a large amount of training time and computation time is needed for this algorithm to work well.

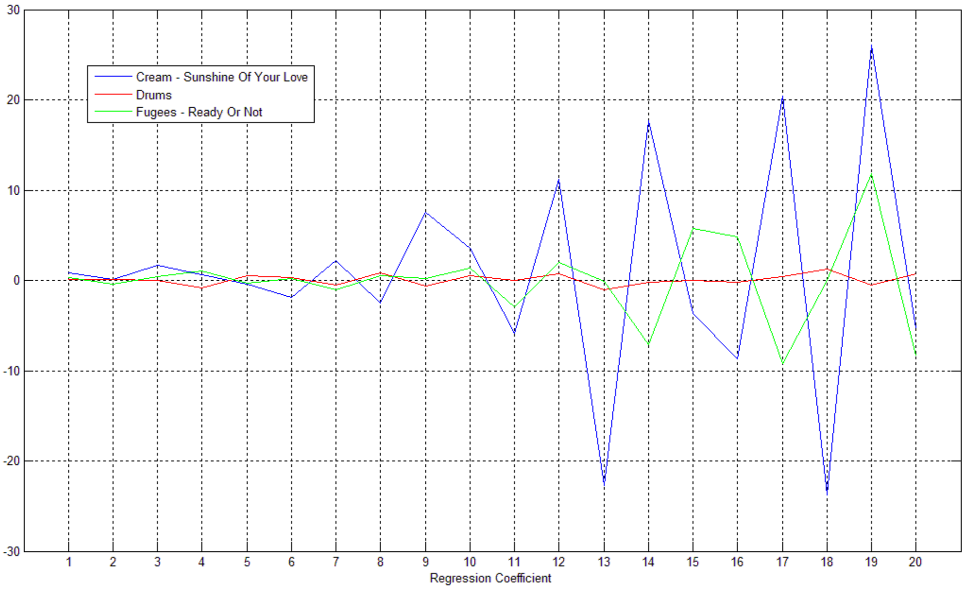


Figure 4.26 Weighting factors determined by linear regression for different audio signals

Figure 4.26 shows that the weights for different signals show a large amount of variation. The weighting factors for the first 20 MFC coefficients for an excerpt from a pop song with percussive elements (Cream, “Sunshine Of Your Love”), an excerpt from a pop song containing no percussive elements (Fugees, “Ready Or Not”) and a short drum loop are depicted. While the first few coefficients are weighted similarly, the following weights differ greatly, which indicates that the first coefficients behave similarly in audio signals of different genres. Another fact that speaks for the choice of the first MFCC as detection function is the fact that for the most evaluated audio signals, only a small number of MFCCs carries meaningful information.

For further information on the evaluation results concerning MFCC-based onset detection see section 4.3.

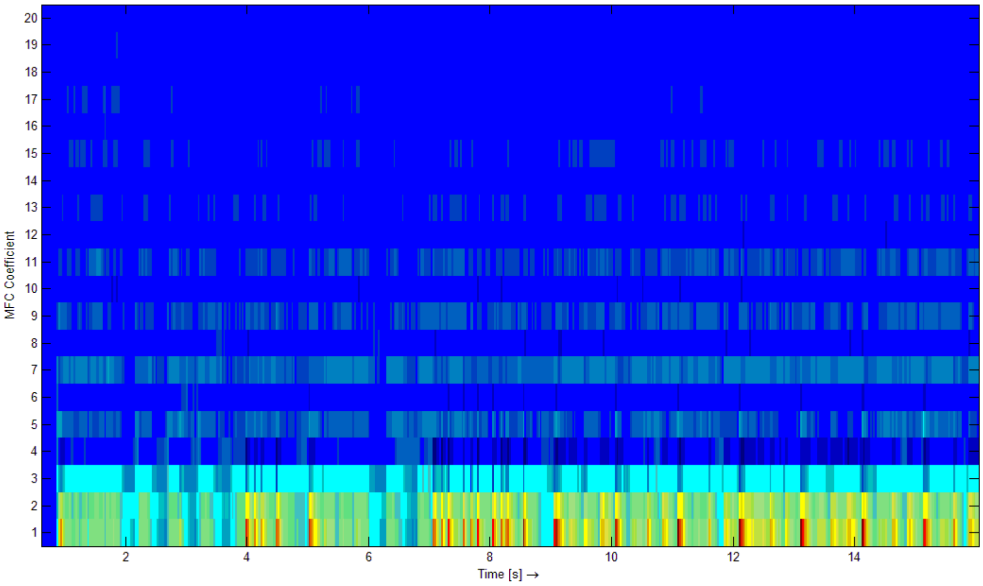


Figure 4.27 First 20 MFCCs over time (Cream, „Sunshine Of Your Love“)

Figure 4.27 shows that only the first three MFC coefficients carry meaningful spectral information. This is due to the compression of the spectrum. While there is still some information in MFCCs 4 through 11, it is visibly less than in the first ones. Coefficients with a higher number can be said to be ineffective in describing the audio signal spectrum.

As in the onset detection method based on chroma values (section 4.2.3.1), the chosen detection function is processed in order to remove any irrelevant peaks and enhance the relevant peaks. Again two methods are evaluated: one using the half-wave rectified differential of the smoothed detection function, the other using a combination of the smoothed and the half-wave rectified differential smoothed version of the detection function [42]. Both methods are described in detail in section 4.2.3.1, and evaluation results for both methods are given in chapter 4.3.

In the final algorithm stage, the local maxima of the created detection function are picked out and taken as onset times. For more information on peak-picking, see section 4.2.7.

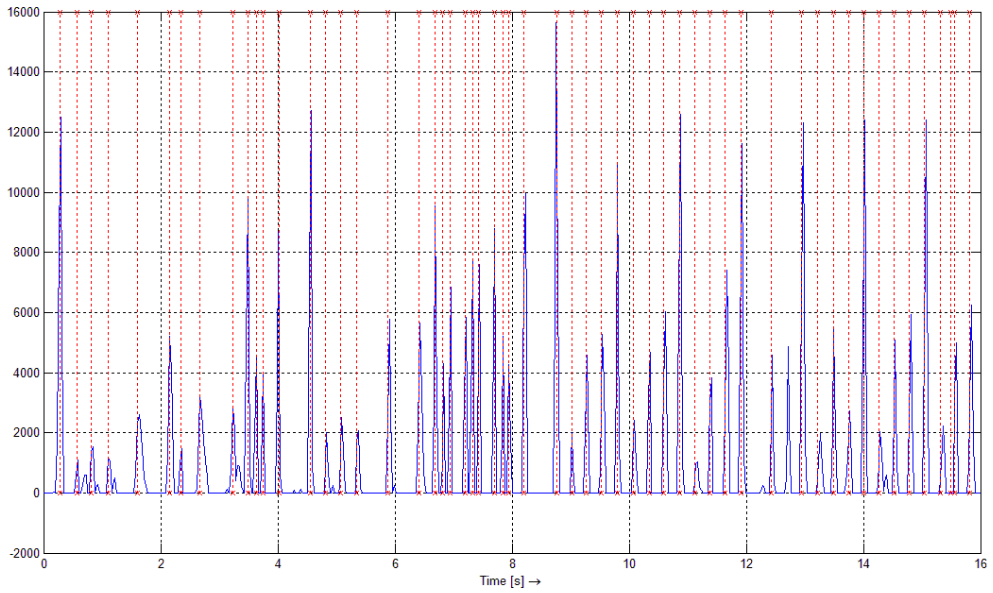


Figure 4.28 MFCC-based onset detection function (blue) with labelled onsets (red) (Cream, „Sunshine Of Your Love“)

4.2.6 Onset Detection based on Modulation Spectra

This chapter focuses on the implementation of onset detection functions based on the evaluation of *modulation spectra*, i.e. the STFT-transformed trajectories of STFT sub-bands. The first part of the chapter will present the concept of modulation spectra. The following two subsections will describe in detail the implementations of two different approaches to creating a detection function from modulation spectrum information. The methods differ in the creation of the detection function itself; while the first sums the modulation spectra weighted toward higher frequencies, the second uses a weighting method based on the characteristic frequencies of drum and percussion signals.

The usual way to interpret the STFT is as a frame-by-frame evaluation of the spectral characteristics of a given signal, i.e. a “series of time-localised spectra” [32]. But from another point of view, the STFT can be interpreted as the sub-band output of a filter bank. Every sub-band output can in turn be viewed as the temporal evolution of the signal amplitude and phase in the frequency range determined by the sub-band center frequency and sub-band width called the sub-band trajectory. Therefore, the STFT corresponds to a combined representation of sub-band trajectories.

Every sub-band trajectory can now be analysed in regard to its spectrum, just like any signal in the time domain. This new domain, which could be described as the spectrum of spectrum, is called modulation spectrum or *modulation frequency domain* [32]. Computing the STFT of $X_k[n]$[1] results in the transformed signal $\tilde{X}_k[p,q]$, the three-dimensional representation of the modulation spectrum with the modulation frequency p and the new time q. The time is decimated because the new time axis is determined by the original STFT frame rate.

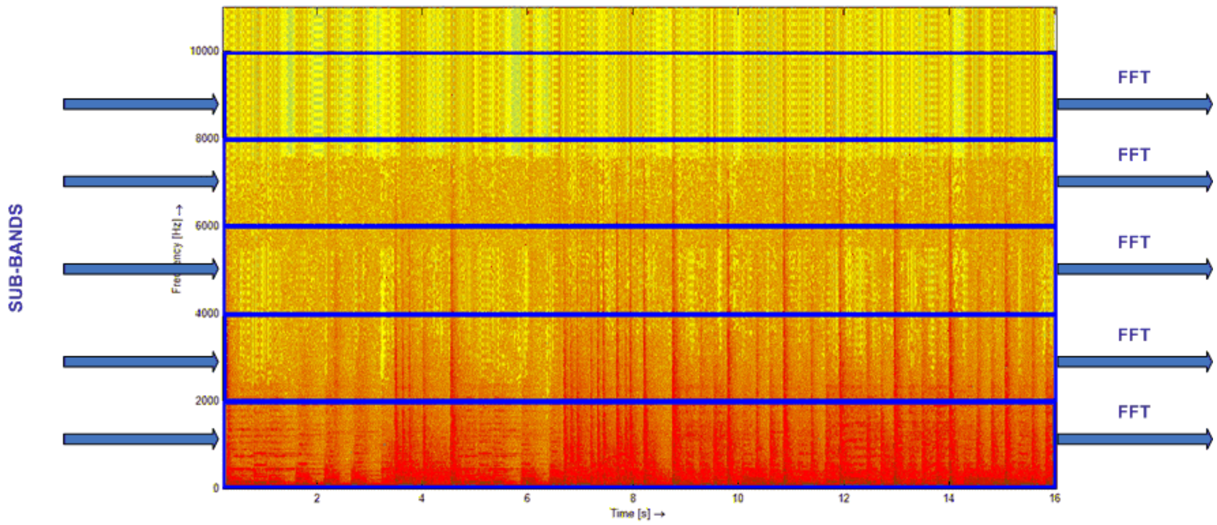


Figure 4.29 Modulation spectrum principle

Figure 4.29 shows the principle of modulation spectrum calculation. In order to obtain a single representation of the modulation spectrum (much like the spectrogram) and not the modulation spectra of the respective sub-bands, the sub-bands are added together. In this thesis, the sub-bands are weighted towards higher-frequency sub-bands before summation in order to amplify the spectral parts that have a greater influence on onset detection.

Modulation spectra are used for describing and improving speech intelligibility [32] by removing noise disturbances or acoustic reverberation effects. The modulation spectrum domain is also very useful for applications that concern themselves with evaluating the amplitude modulation of the respective sub-band trajectories. For example, a sinusoidal signal with constant amplitude will exhibit no changes in the modulation spectrum, there will only

[1] The short-time spectrum of *x[n]* of the *k*-th sub-band

be a DC component visible. In contrast, any other signal with changing amplitude or frequency will exhibit clearly visible changes in the modulation spectrogram. The changes will be more or less pronounced according to the sub-band that is analysed.

This quality is clearly visible in Figure 4.30, where the modulation spectra of different sub-bands are shown. The analysed signal is a simple drum recording using only bass drum, snare drum and a hi-hat. In the lowest analysed sub-band around 40 Hz, only the signal changes due to the bass drum are visible, the second sub-band shows the changes due to the snare drum and the third sub-band is composed of snare-drum and hi-hat influences. The hi-hat starts in the second part of the recording and is visible in the fast but not very pronounced changes starting at approximately 12 seconds. While the structure of more complex recordings cannot be resolved as easily as in this example, the principle still remains the same.

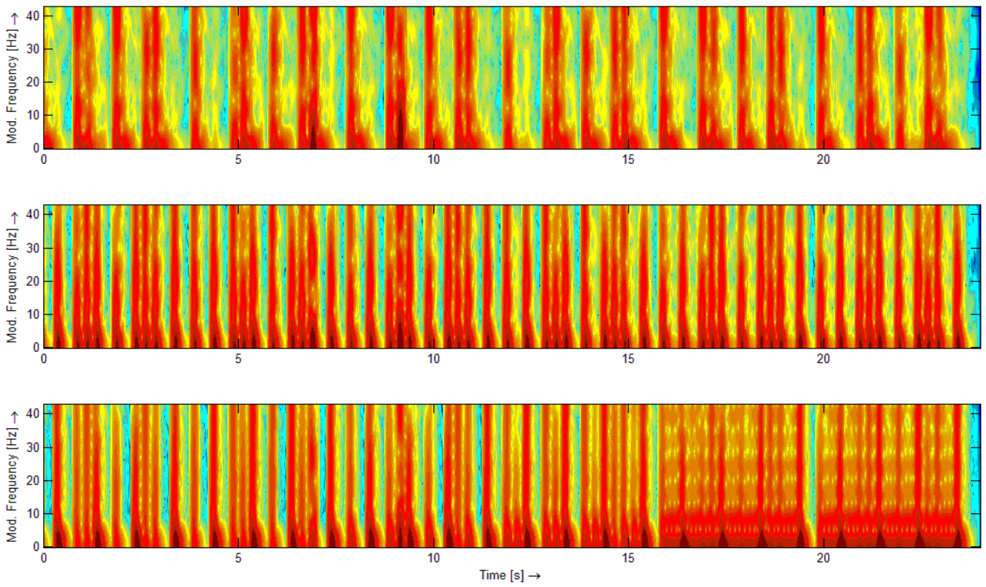


Figure 4.30 Modulation spectra of different sub-bands (top: 40 Hz, middle: 340 Hz, bottom: 5 kHz for a simple drum track)

In order to be able to create a detection function from the sub-band modulation spectra, these have to be grouped together. The spectra can simply be added or some kind of weighting can be used to enhance sub-bands that have a greater impact on onset detection, leading to a single modulation spectrum representation that can be evaluated.

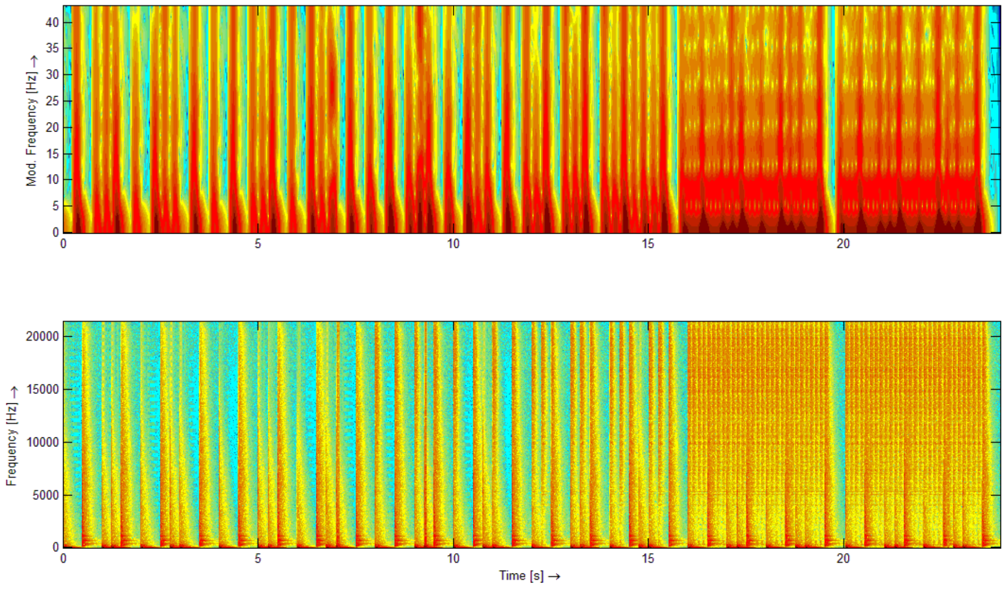


Figure 4.31 Grouped modulation spectrum (top) and spectrogram (bottom) of a simple drum track

Figure 4.31 shows the grouped modulation spectrum and the spectrogram of a simple drum track. The advantage of the modulation spectrum is apparent – changes in the signal energy are much more pronounced and visible than in the spectrogram of the signal.

In this thesis, an attempt is made to create an onset detection function by following the weighted modulation spectrum sub-bands over time. The assumption is made that when onsets appear in a signal, there will be a broad-band increase in energy in the modulation spectrum of the signal. The next subsections present the two approaches considered for this application where the major difference consists of the weighting method of the sub-band modulation spectra. The basic principle of both approaches is shown in Figure 4.32.

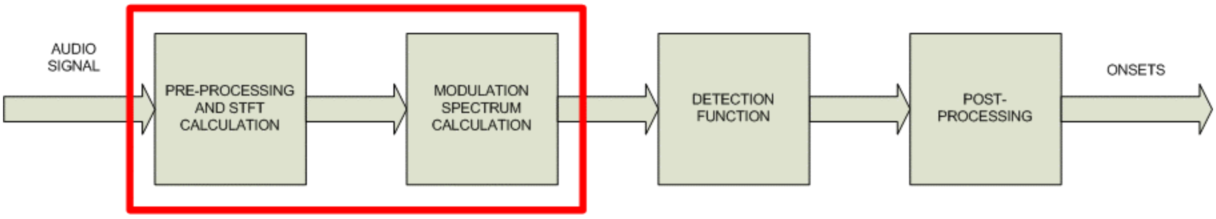

AUDIO SIGNAL
PRE-PROCESSING AND STFT CALCULATION
MODULATION SPECTRUM CALCULATION
DETECTION FUNCTION
POST-PROCESSING
ONSETS

Figure 4.32 Onset detection based on modulation spectra

After pre-processing, the STFT is calculated, from which the modulation spectrum is derived. This in turn is transformed to create the detection function, from which the onsets are determined after a post-processing stage. The difference between the two approaches mentioned above lies in the computation of the combined modulation spectrum from the sub-band modulation spectra using different weighting methods.

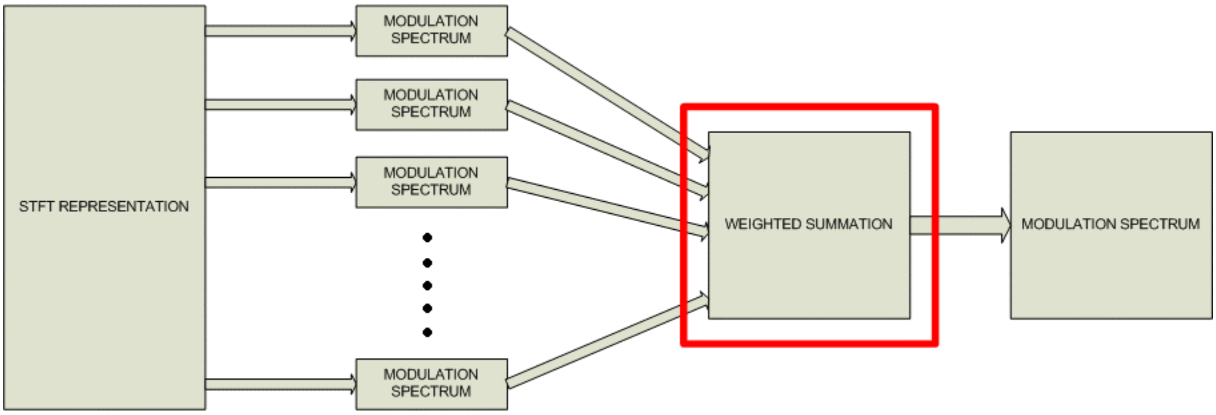


Figure 4.33 Modulation spectrum calculation

The first approach presented in chapter 4.2.6.1 uses weighting of sub-bands towards higher frequencies, emphasizing changes in the upper part of the spectrum, while the second approach described in chapter 4.2.6.2 uses musical knowledge to place higher emphasis on frequency regions where characteristically high energies from drum sounds are found.

4.2.6.1 Sub-Band Energy Trajectories

As explained in the previous section, broad-band changes in the modulation spectrum can be interpreted as onsets in the signal. The modulation spectrum is calculated by STFT-analysing the sub-band trajectories of a STFT representation of a time-domain signal *x[n]* and describes the modulations in the spectral amplitudes of a signal over time. This section will present the basic approach taken to create a detection function from modulation spectrum data. The next section will present an alternative approach to the same problem.

The first stage of the algorithm consists of the pre-processing and STFT calculation section described in chapter 4.2.2.

In the second stage, the modulation spectrum is calculated from the STFT representation. The sub-band representation of the signal spectrum is created by grouping the trajectories of the spectral bin magnitudes in octave bands around a center frequency k_0 by summation. This results in 10 sub-bands with center frequencies reaching from 30 Hz to 8.5 kHz. Each sub-band is in turn analysed using a second STFT stage.

This second analysis stage uses an 18-point frame length corresponding to approximately 420 milliseconds and a hop-size of 1, i.e. no time decimation. The frame length of 18 points was determined after extensive testing (for more information, see chapter 4.3). Finally, the frames are again multiplied with a Hanning window and zero-padded to 512 points. By computing the FFT of the signal segments the modulation spectral domain representation of the signal is obtained.

In order to be able to create a single representation of the signal, the sub-band modulation spectra are combined. This is achieved by summing the octave-band modulation spectra weighted towards higher-frequency bands.

$$\tilde{X}[q] = \sum_{p=1}^{10} p\tilde{X}_p[q] \tag{4.26}$$

where p stands for the sub-band index, $\tilde{X}_p[q]$ for the modulation spectrum of the p-th sub-band and $\tilde{X}[q]$ for the combined modulation spectrum representation. This is a weighting biased towards high frequencies similar to the weighting used in [12]. The idea behind this weighting approach is to take advantage of the fact that between onsets, the major part of the signal energy is contained in the lower part of the modulation frequency spectrum. Between onsets, the audio signal energy changes only slightly due to the mostly stationary nature of the signal in that segment compared to the more abrupt changes with uniform distribution over the modulation frequency bandwidth when onsets are present.

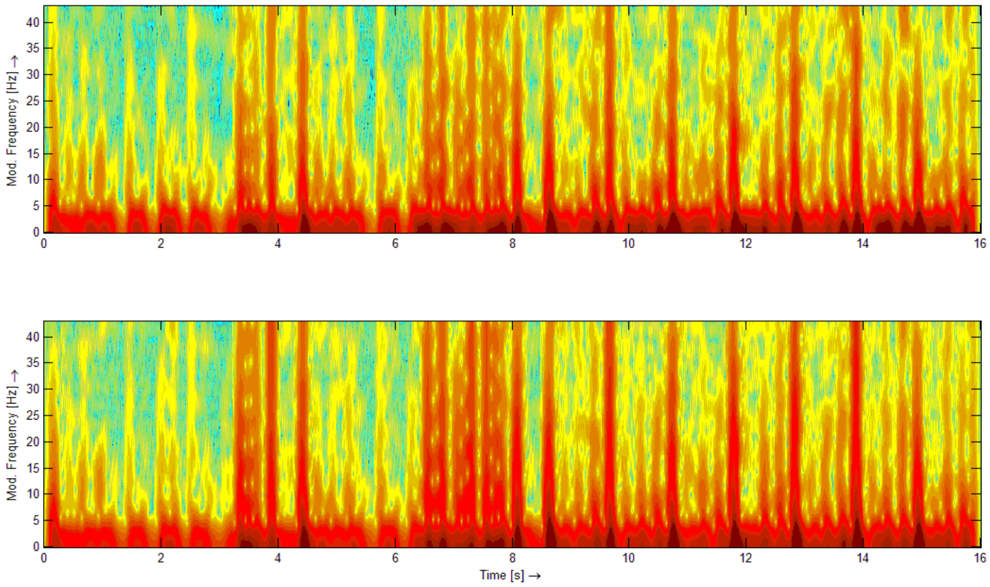


Figure 4.34 Modulation spectrum (top) and weighted modulation spectrum (bottom) (Cream, „Sunshine Of Your Love“)

Figure 4.34 shows the effect of weighting towards higher frequencies. The modulation spectrum of up to 5 Hz changes only slightly throughout the signal, while the upper modulation spectrum changes due to the onsets present in the signal. The weighted approach depicted in the lower part of the figure displays a more broad-band change in the modulation spectrum, thus making it easier to identify the onsets.

The next algorithm stage consists of creating a detection function from the modulation spectrum data. Two methods are evaluated to this end: firstly, the modulation spectra are divided into modulation frequency sub-bands which are weighted towards higher modulation frequencies and summed; a second approach uses linear regression to determine the optimal combination of modulation frequency sub-bands for the creation of a detection function.

Independently of the detection function creation mode, the weighted and summed modulation spectrum described above is divided into eight octave bands covering the complete modulation frequency bandwidth, in our case up to a little above 40 Hz. This is done because, as seen in Figure 4.34, the energy in the low modulation frequencies tends to be uniformly high, in contrast to the higher modulation frequency bands where there are numerous sharp increases in the signal energy mainly due to onset events.

In the first approach, a weighting similar to the method used to form the complete modulation spectrum presented previously is used:

$$df[q] = \sum_{p=1}^{8} p\tilde{X}_p[q] \tag{4.27}$$

where *df[q]* describes the detection function, *p* the modulation spectrum octave band index and $\tilde{X}_p[q]$ stands for the p-th modulation frequency octave band.

The second detection function creation method uses linear regression to determine a suitable sub-band weighting function (for more information about linear regression, refer to section 4.2.5.1). By using the sub-band modulation spectrum trajectories over time as independent variables and the convolution of a pulse train with pulses at the previously determined onset times with a 50-millisecond Hanning window as the dependent variable, a weighting vector is calculated.

After evaluating the onset detection results for either weighting method, it was decided to use the first approach that uses linear weighting of modulation-spectrum sub-bands towards higher modulation frequencies in the onset detection algorithm. While in some cases the second approach works as well or better as the first one, the drawbacks of the linear regression approach, including the additional computation time and the dependency on similar audio signals which arises from the differences in the weighting function for different signals, are felt to be too severe for this application.

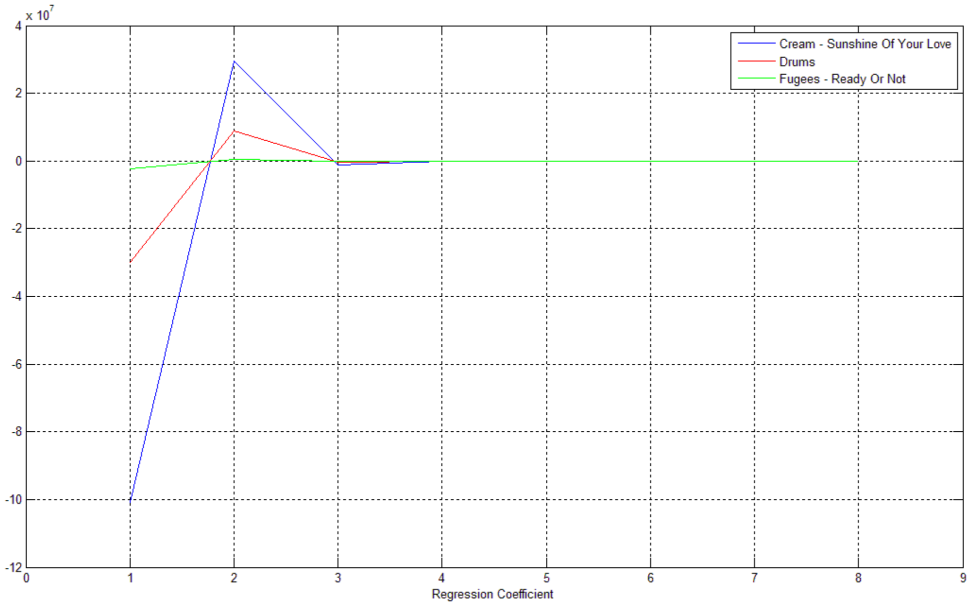


Figure 4.35 Weighting factors determined by linear regression for different audio signals

In Figure 4.35, the optimal weights determined by linear regression for the weighting of the modulation spectrum sub-bands for different signals are shown. While the major part of the weights has similar values, the differences in the first few weighting coefficients is striking – the nearly linear weighting calculated for a non-percussive fragment of The Fugees' "Ready Or Not" cannot be compared to the weighting deemed suitable for the two other, more percussive signals.

The detection function is post-processed in the next algorithm stage in order to de-noise it and to emphasise the local maxima, thereby facilitating the peak-picking operations that are carried out afterwards. To enhance the relevant peaks and remove irrelevant ones, two different approaches are used: one uses the half-wave rectified differential of the smoothed detection function, the other uses a combination of the smoothed detection function and the half-wave rectified differential smoothed detection function. Both methods are described in chapter 4.2.3.1.

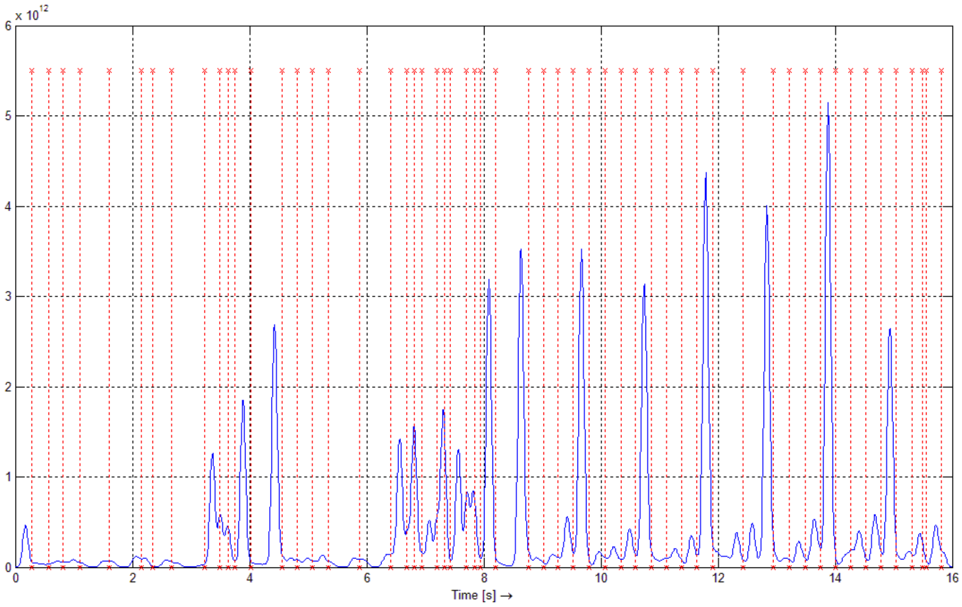


Figure 4.36 Modulation spectrum-based onset detection function (blue) with labelled onsets (red) (Cream, „Sunshine Of Your Love“)

The detection function is then post-processed and the function maxima are selected and taken as onset times. For more information on post-processing and peak-picking, see section 4.2.7.

4.2.6.2 Weighted Sub-Band Energy Trajectories

The second algorithm implementation that uses modulation spectrum data to find onsets in audio signals differs from the first only in the manner of the weighting of STFT spectra. Therefore, this weighting method will be explained in detail while the rest of the algorithm will be only outlined broadly. For more information about the general structure of this onset detection algorithm, see section 4.2.6.1.

The previous chapter explains in detail how changes in the modulation spectrum can be interpreted as onsets in the signal. When the temporal sequence of STFT representations of a signal is in turn analysed by calculating the STFT of the sub-bands, this leads to the modulation spectrum or modulation frequency domain representation that describes the modulations in the spectral signal amplitude over time. While the modulation energy changes only slowly when the audio signal is stationary, i.e. between onsets, abrupt changes in the energy distribution can be observed when an onset is present. When a musical note is sustained, the major part of the signal energy will be concentrated in the lower end of the modulation spectrum (below 14 Hz) [32], while transient events such as onsets lead to a constantly high energy level across the whole modulation spectrum. This fact is exploited to find the onsets in audio signals by evaluating the energy present in modulation spectrum sub-bands.

As in the onset detection approach presented in the previous chapter, the audio signal is first pre-processed and the STFT is calculated as described in section 4.2.2.

The next step consists of summing the spectral magnitudes in octave frequency bands, thereby compressing the STFT representation into 10 sub-band trajectories with center frequencies ranging from 30 Hz to 8.5 kHz. Each of these sub-bands is viewed as a pseudo-temporal signal upon which the STFT is performed using a frame length of 18 points, which corresponds to a resolution of about 480 milliseconds, and a hop-size of 1, i.e. no time decimation. The frames are again windowed with a Hanning window and zero-padded to a length of 512 points. The result of these calculations is the modulation spectrum representation of the signal for 10 different sub-bands.

The modulation spectra of the respective sub-bands are combined to form the complete modulation spectrum representation. This is done by summing the sub-bands weighted according to the characteristic high-energy frequency bands of drum recordings.

Standard drum sets have at least the following components: a bass drum, a snare drum and a hi-hat. However, as the hi-hat tends to have a very broad-band energy distribution across the frequency spectrum and the bass drum and snare drum often have a clear spectral centroid in the very low frequency regions, it was decided to concentrate on the latter drum components. By emphasizing the frequency bands that are characteristic of these instruments, detection functions that rely on spectral information to find onsets should be able to produce better results.

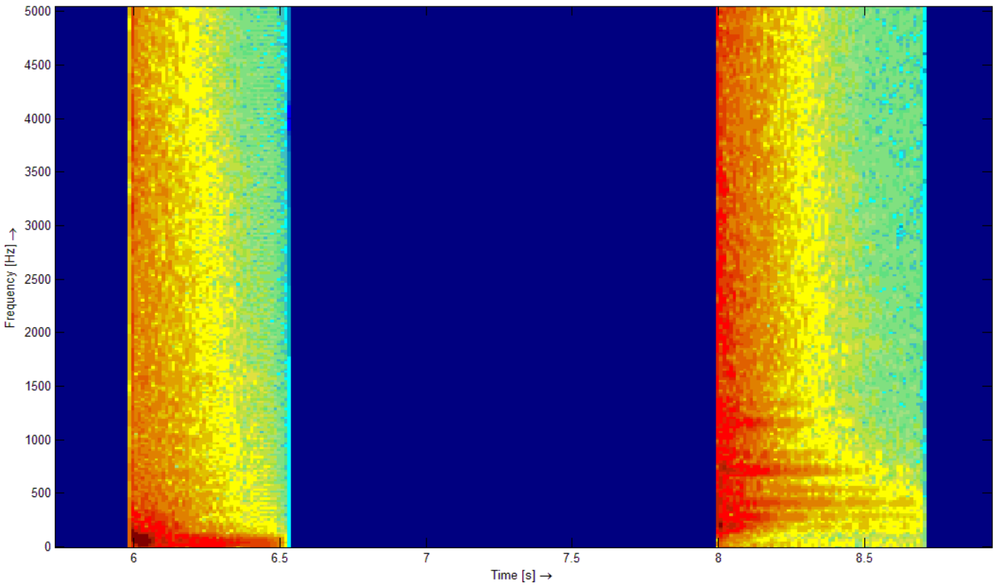


Figure 4.37 STFT representation (detail) of a simple drum signal with a bass drum (left) and a snare drum (right)

Figure 4.37 shows the temporal evolution of the spectrum of a modern bass drum and snare drum. The major part of the bass drum's energy is clearly concentrated below 200 Hz. The snare drum spectrum shows a broader energy distribution, but there is still a concentration of energy to be found between 200 Hz and approximately 1 kHz.

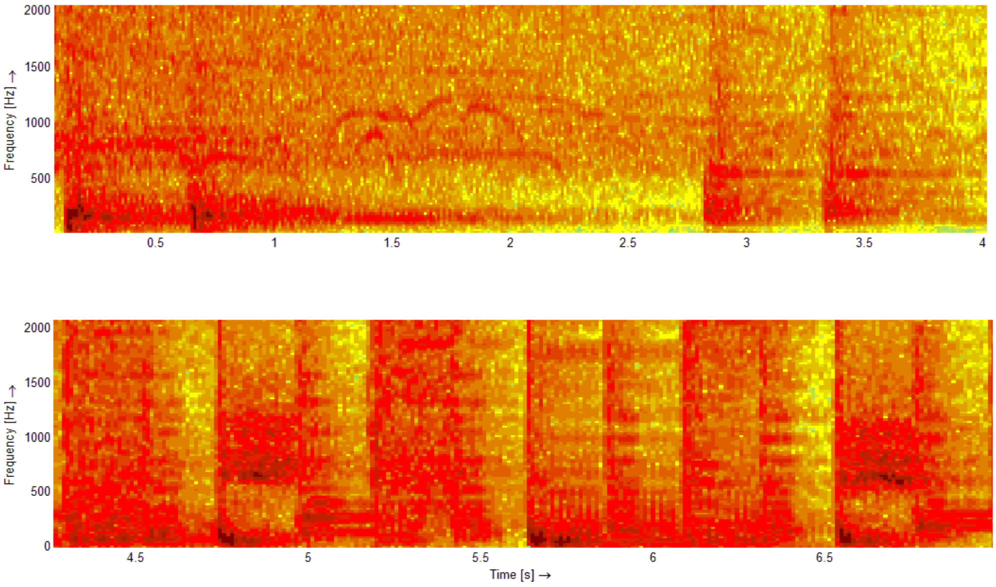


Figure 4.38 Spectrograms for percussive excerpts from a Phil Collins live recording (top) and E.A.V.'s „Ding Dong" (bottom)

The assumption about the spectral characteristics of bass drums and snare drums is confirmed after the analysis of a number of modern pop songs. Figure 4.38 shows two spectrograms that show the spectral distribution of bass drum and snare drum sounds. The upper figure, taken from a Phil Collins live recording, contains two bass drum sounds at the beginning and two snare drum sounds at the end. The bass drum shows a narrow-band energy concentration around 250 Hz while the snare drum has a more broad-band energy distribution between 200 and 700 Hz. The lower figure, an excerpt of E.A.V.'s "Ding Dong", shows similar characteristics with narrow-band bass drum energy at up to 200 Hz and more broad-band snare drum energy above that.

This knowledge is used to determine a weighting function for the sub-bands of the frequency spectrum for use in the modulation spectrum calculation.

$$w[p]=\begin{cases}1.5 & for \quad p\in p_{bd}\\ 1 & for \quad p\in p_{sd}\\ 0.5 & else\end{cases} \tag{4.28}$$

where p is the octave band index, p_{bd} stands for the indices of octave bands that are part of the characteristic bass drum frequency region and p_{sd} stands for the indices of bands in the snare drum frequency region. The higher weighting of the bass drum is due to the higher energy concentration and also due to the fact that the bass drum is always a reliable indicator of a change in the signal.

The complete modulation spectrum representation of the signal is then calculated by summing the weighted octave bands together.

$$\tilde{X}[q]=\sum_{p=1}^{10}\tilde{X}_p[q]w[p] \tag{4.29}$$

where p stands for the sub-band index, $\tilde{X}_p[q]$ for the modulation spectrum of the p-th sub-band and $\tilde{X}[q]$ for the combined modulation spectrum representation.

In the next step, the detection function is created from which the onsets are picked out. The modulation spectrum is segmented into eight octave frequency bands covering the whole spectrum bandwidth. The reason for this repeated sub-band division is that the energy in the lower modulation frequency regions will tend to be uniformly high because even when the signal is assumed to be stationary, as is the case with, for example, sustained notes, there will be some measure of amplitude decay. However, as seen in Figure 4.34, when an onset is introduced in the signal, there will be a noticeable increase in the energy over the whole modulation spectrum bandwidth. After this segmentation, the weighted sum of the octave bands is calculated to form the detection function.

$$df[q] = \sum_{p=1}^{8} p\tilde{X}_p[q] \tag{4.30}$$

where *df[q]* describes the detection function, *p* the modulation spectrum octave band index and $\tilde{X}_p[q]$ stands for the *p*-th modulation frequency octave band. This corresponds to the linear weighting method presented in [12] that takes advantage of the fact that onsets tend to appear as broadband events in the modulation spectrum while stationary regions show a concentration of energy in the low modulation frequency bands.

In the next stage, the detection function is post-processed in order to remove noise components and enhance relevant peaks. For more information about this algorithm stage see section 4.2.3.1.

The two different approaches that are used for the enhancement of relevant peaks and the removal of irrelevant ones are the same as in the previous approach.

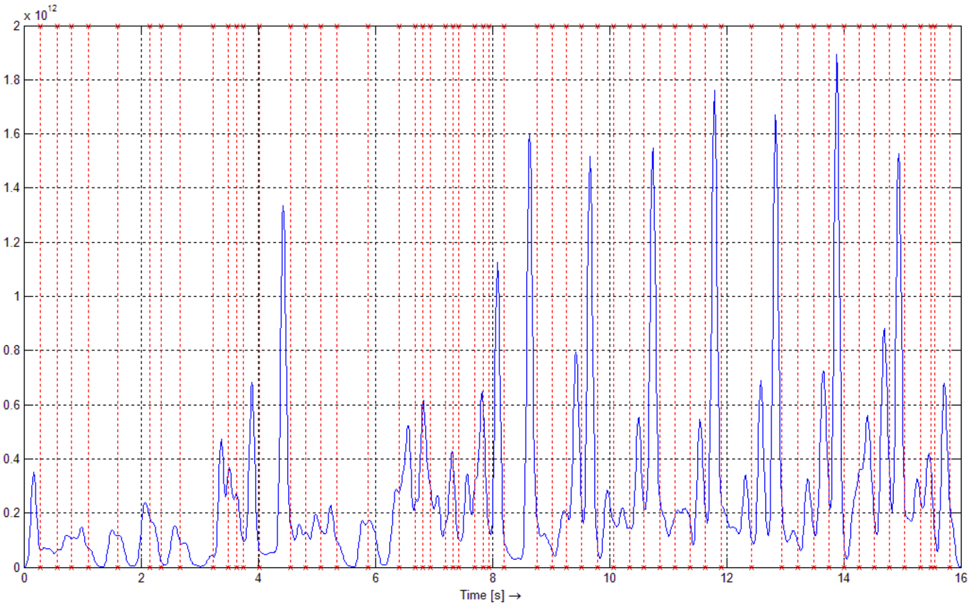


Figure 4.39 Modulation spectrum-based onset detection function with weighted sub-bands (blue) with labelled onsets (red) for Cream, „Sunshine Of Your Love“

In the last stage of the algorithm, the detection function is searched for remaining maxima over an adaptive threshold. For more information on the peak-picking stage, see section 4.2.7.

4.2.7 Peak-Picking

The previous chapters present the implementation of different algorithms that create an onset detection function from an audio signal. These detection functions are intermediate representations of the signal that are calculated according to certain signal characteristics that are assumed to facilitate the detection of onsets present in the signal. This section is concerned with the implementation of an algorithm that reliably detects the onsets by finding local maxima in the detection function using a locally adaptive threshold. This algorithm is used with all onset detection function implementations. It consists of three stages: in the first stage suitable threshold criteria are estimated, in the second stage local maxima are identified and in the final stage multiple detections within a certain time slot are removed.

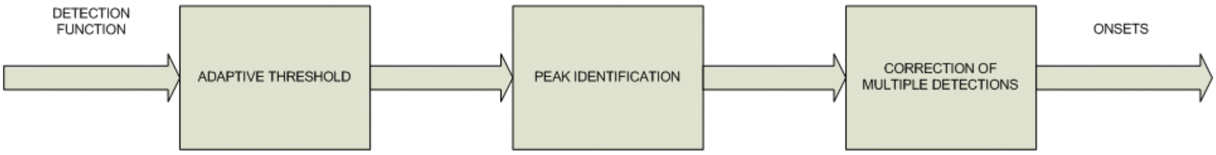


Figure 4.40 Peak-Picking stage structure

The task of finding function values that exhibit certain characteristics has been widely researched in many fields, including information theory (e.g. block-switching coder applications [32]), audio applications such as de-noising [32] as well as in speech processing, for example in speaker identification tasks [46]. For most applications, there will have to be some kind of threshold, be it fixed or adaptive, which leads to a binary decision about whether some part of the signal fulfills the required characteristic or not. While in some cases, for example concerning modifications of perceptual attributes, this decision can be avoided by using continuous adaptation curves [32], in most cases a yes-no judgment concerning the presence of specific signal attributes is necessary. The downside of this decision-making is that there will be some cases when a wrong decision will be made, i.e. there will be missed detections as well as false detections. The key is therefore to find an algorithm that maximises the number of correct detections while minimising the number of false detections.

For simple applications, a fixed threshold might be adequate. However, in the case of onset detection algorithms, the range of values of the detection function depends on a number of variables such as the signal level, the frame length and the STFT size. Therefore, if the threshold is chosen as a small value, there will be numerous false detections, while if the threshold is set at a higher value, many genuine onsets will be overlooked. Also, while in the case of standard pop songs the dynamic range of the audio signal is not very high, most classical pieces display many changes in the dynamic range and therefore in the signal loudness, which can lead to the algorithm missing onsets in quiet passages while detecting wrong onsets during loud passages.

Because of these limitations, most audio applications use an adaptive threshold. In most cases, this threshold is computed as a transformed version of the signal itself [3] by using linear or non-linear smoothing. Still better results are obtained with a smoothing method relying on *local medians*, where smaller peaks in the vicinity of larger peaks are not masked. The median of a set of values is defined as the value in the middle of an ordered table of the set. If the length of the signal window from which the median is obtained is longer than the width of the onset peak, this will not cause the adaptive threshold to rise at the peak position, which results in better adaptation of the threshold curve [47].

For the peak-picking stage implemented in this thesis, the main goal is to detect significant peaks in the detection function while eliminating noise and spurious peaks that result in false detections. This is achieved by using a constant threshold in combination with an adaptive threshold obtained by calculating the local median of detection function segments [11].

$$thr[m] = \delta + \lambda \cdot median\left(df[k_m]\right), k_m \in \left[m - \frac{H}{2}, m + \frac{H}{2}\right] \tag{4.31}$$

This results in a frame-wise detection function threshold *thr[m]* that can be described as local adaptive filtering. δ is a constant value that represents the fixed threshold, λ is a scaling factor that determines the influence of the surrounding frames on the threshold and *H* stands for the window length of the median computation. The constant δ should be chosen with care since it has considerable influence over the ratios of correct and false detection while the scaling factor λ is of less importance. Values of 0.01 for the threshold δ and 0.5 for the scaling factor λ worked well in most cases.

While the approach mentioned above works fairly well for selecting probable onsets from a detection function, additional steps are taken to further improve detection results. After the adaptive threshold has filtered out a number of potential onset candidates, two separate processing stages eliminate multiple or incorrect onset detections.

The first correction stage makes sure that only the maximum of the values of the detection function over the threshold is actually recognised as an onset. This is achieved by comparing the detection function values to the two previous and the two ensuing values.

$$df[q-2] < df[q-1] < df[q] > df[q+1] > df[q+2] \tag{4.32}$$

A real peak will have a detection function value higher than the value in the previous and following frames. Therefore, only detection function values that match this criterion are processed further.

The last stage of the peak-picking algorithm deletes any multiple detections that are possibly still present.

A sixteenth note at 208 BPM is taken to be the smallest beat that is still perceptually meaningful. Therefore, it is assumed that if there are multiple onset detections during that time span, each but one will be a false detection. At a sampling rate of 11.025 kHz, the length of this time segment is approximately 800 samples, which corresponds to roughly 10 analysis frames. A moving window is applied to the detection function in order to identify regions with multiple detections during that time span. In every such region, only one peak is assumed to be a correct detection, so only the point with the maximal detection function value within the region, and not the first possible detection, is selected. This is done because a maximum in the detection function indicates the point in time where the biggest change in the signal characteristic takes place.

4.2.8 Inter-Onset Interval Beat Tracker

The previous chapters give an overview of the different methods used in this thesis to find onsets in audio signals. However, the information about onset times alone is not psycho-acoustically meaningful. By tapping along with a signal, human listeners automatically infer a beat hypothesis consistent with the spacing of the onsets in the music. This can be described as finding musical accents and subsequently filtering them in order to find underlying periodic structures [42]. By detecting the beat structure of a signal, the results of the onset detection function can be further improved because the found onset positions can be compared to likely beat hypotheses. This section will introduce the concept of beat structures and present the inter-onset interval beat tracking system implemented in this thesis.

The perception of musical beat is founded on different time scales which are summed to form an idea of the rhythmical structure of the signal [42]. What is commonly described as the "beat" of a signal corresponds to the so-called *tactus* level or *foot-tapping rate* and is intuitively understood as the rate a human listener would tap along with the music. Also, the *tempo* of a piece is defined as the rate of tactus pulses. The *tatum* ("temporal atom") level of rhythmical understanding describes the rate of the shortest meaningful pulse periods that appear in a music signal, while the larger-scale harmonic change rates are described on the *measure* level. In most cases, the tactus pulses will be found at integer multiples of tatum pulses and measure pulses at integer multiples of tactus pulses.

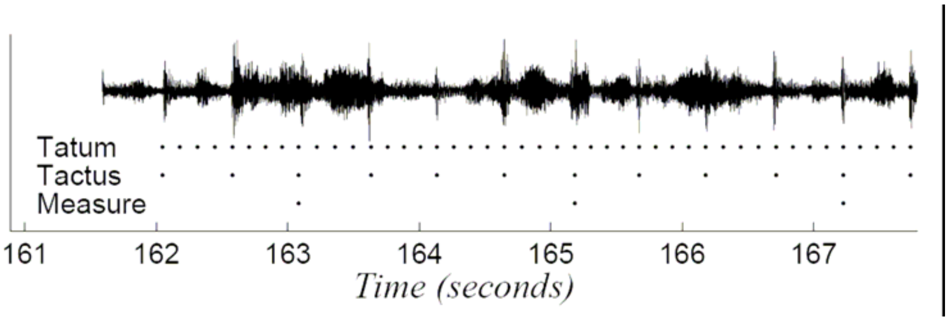


Figure 4.41 Rhythmic levels [42]

In this thesis, the underlying tactus and tatum levels are estimated to form a beat hypothesis. This is achieved by evaluating the *inter-onset intervals* (IOI), defined as the time span between successive onsets, a procedure that is often used ([48], [49], [50], [51]). The distribution of the IOIs is evaluated separately for the tactus and the tatum level by creating histograms that show the aforementioned distribution. From these histograms, the most likely beat hypothesis is calculated by evaluating the histogram maxima and comparing the maxima in the tactus and the tatum IOI distribution. If this yields a clear maximum, the onset detection results are compared to the beat hypothesis and corrected if necessary.

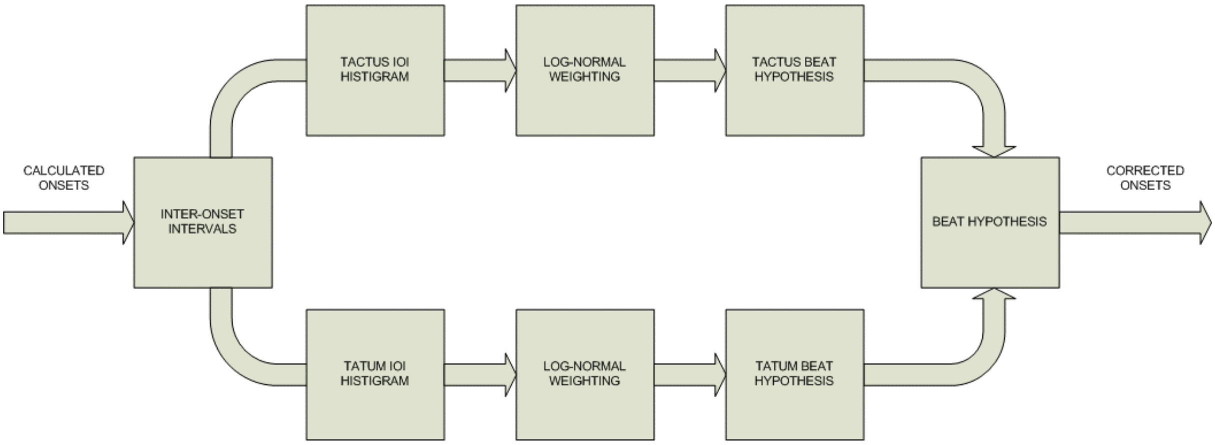


Figure 4.42 Inter-onset interval beat tracker

The beat-tracking algorithm is fed with the onset detection results, i.e. the temporal positions of the found onsets. The time spans between onsets is determined by calculating the first-order difference between the onsets:

$$\tau_i = t_i - t_{i-1} \tag{4.33}$$

with t_i as the time position of the i-th onset. From these differences the histograms of the tactus and tatum time scale are computed.

The two-parameter log-normal distribution is applied to the histogram data to determine the distribution of IOIs across the BPM scale.

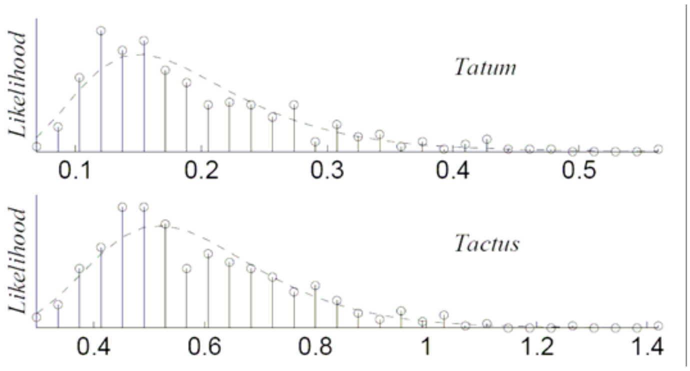


Figure 4.43 Tatum and tactus inter-onset interval histogram and corresponding probability distributions [42]

Figure 4.43 shows the number of occurrences for different inter-onset interval times for the tactus and tatum level as well as the log-normal distribution that corresponds to those.

Although the parameters of the distribution depend to some extent on the musical genre of the signal, fixed parameters are chosen for the respective distribution since the algorithm should work independently of the musical genre. The scale and shape parameters are chosen by empirical estimation from hand-labelled data [42]. The log-normal distribution proposed by Parncutt [42] is given by

$$p(\tau) = \frac{1}{\tau\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}\left[\ln\left(\frac{\tau}{m}\right)\right]^2} \tag{4.34}$$

with τ as the IOI and the scale and shape parameters σ and m. For the tactus level, σ is chosen as 0.28 and m as 0.55, while for the tatum level σ is estimated by 0.39 and m by 0.18 [42]. For graphical representation, the scale is defined in BPM (*beats per minute*).

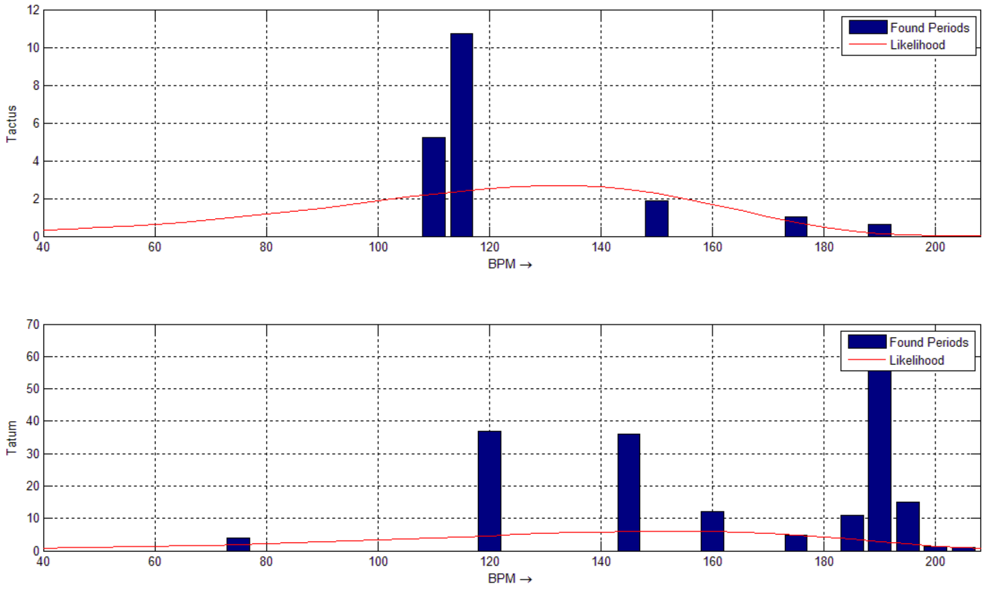


Figure 4.44 Tactus (top) and tatum (bottom) histograms and probability distributions (Cream, "Sunshine Of Your Love")

Figure 4.44 shows the histogram of inter-onset intervals and the corresponding log-normal distribution across the BPM scale for a pop song containing percussive elements. The tempo of the audio signal is located around 115 BPM, the tempo that was estimated by the algorithm. One thing that stands out is that the peaks in the histogram are well-defined, which indicates that the majority of IOIs lie around only a few BPM values, which shows that the onset detection works correctly. The tatum histogram also shows that major peaks lie around the estimated tempo as well as more or less around double the estimated tempo, which corresponds to the above-mentioned fact that tactus pulses mostly lie at integer multiples of tatum pulses.

The histogram will, however, not always show a similar distribution concentrated on a few major peaks. In some cases there will be too few onsets detected to reliably estimate the beat of the signal, while in other cases there will be numerous false detections that falsify the histogram.

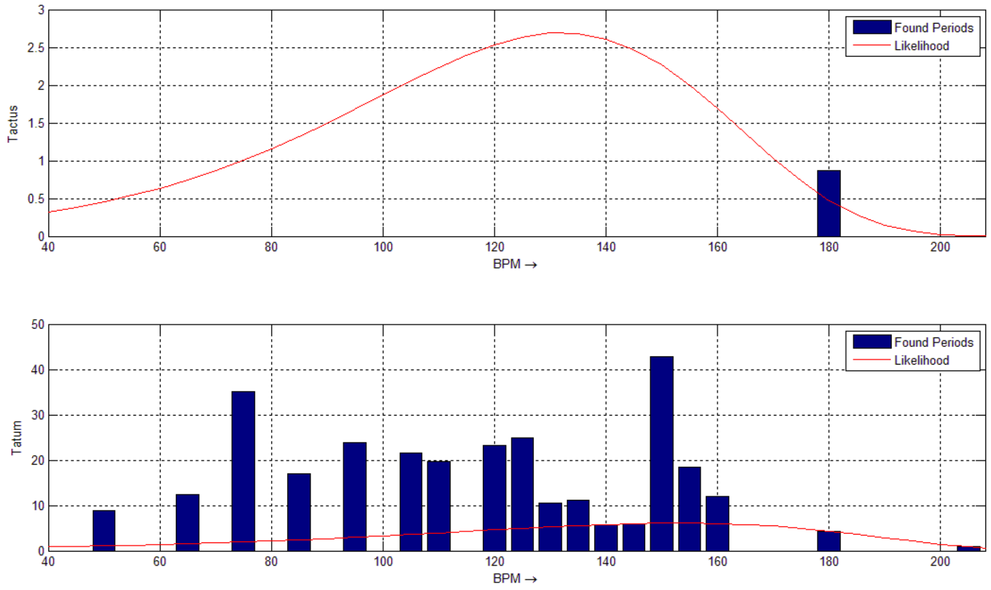


Figure 4.45 Tactus (top) and tatum (bottom) histograms and probability distributions (Fugees, “Ready or Not”)

Figure 4.45 shows the tactus and tatum histograms of IOIs and the corresponding probability distribution for an excerpt from a pop song that contains no percussive parts. In contrast to Figure 4.44, there are no clear beat candidates for this audio signal. The inter-onset intervals are distributed over a wide range of values and creating a beat hypothesis out of this data will not have the positive effects on the onset detection results as described earlier. In order to avoid making assumptions about the beat structure of a signal when there are no viable beat candidates, a threshold is introduced. The threshold is computed by comparing the arithmetic and the geometric mean of the beat histogram distributions.

The arithmetic mean of a value set *x* is defined as

$$\overline{x}_A = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (4.35)$$

while the geometric mean of the same value set is given by

$$\overline{x}_G = \sqrt[N]{\prod_{i=1}^{N} x_i} \qquad (4.36)$$

By comparing the two means, a statement about the plausibility of a beat hypothesis extracted from the histograms can be made. The geometric mean is always smaller than the arithmetic mean except in the case when all elements of the analysed value set are equal. This means that if there are many similarly small peaks in the histogram, as is the case in Figure 4.45, the value of the two means will lie close to each other, while if there are only a few sharp and distinct peaks in the histogram, as seen in Figure 4.44, the two means will diverge by a large margin. Therefore, a beat hypothesis is only constructed if the two means show significant divergence:

$$\nu \cdot \overline{x}_G < \overline{x}_A \tag{4.37}$$

The threshold parameter *v* is set to 1.5 after comparing different beat histograms and their respective means. If the condition described above is met by the tactus histogram values, all histogram values that lie above the arithmetic mean are selected as possible beat candidates. If that is not the case, no beat hypothesis is formed.

The next task of the beat tracking algorithm is to select the most likely beat from the candidates selected in the previous stage. The assumption is made that if beat hypotheses are formed on the tactus as well as the tatum level, every viable beat candidate on the tactus level should have a counterpart on the tatum level. By comparing the tactus beat candidates with the tatum candidates and their integer multiples with a tolerance of ±10 BPM, the beat candidates are narrowed down to a smaller number because any erroneous detections having no counterpart on the other beat level are eliminated. In the next stage, if there is still more than one beat candidate, the candidate with the highest value, i.e. the highest number of inter-onset intervals of that duration, is selected as the final beat hypothesis.

If the algorithm successfully forms a beat hypothesis, this hypothesis is used to further correct and enhance the onset detection results. This is achieved by setting up weighting functions that leave onsets found in the vicinity of probable beats untouched while deleting detected onsets that do not conform to the beat hypothesis.

The first step of the onset correction algorithm consists of calculating the note durations corresponding to the beat hypothesis. Since human listeners tend to tap to music in quarter notes and most western pop songs and a big part of classical music are based on quarter beats, i.e. either two, three or four beats to the bar, the probable tempo is assumed to be based on quarter notes. From the duration of a quarter note at the assumed beat, the durations of half notes, full notes, eighth notes, sixteenth notes as well as quarter trioles, eighth trioles and sixteenth trioles are extrapolated. These rhythmic units cover most of the rhythmic spectrum present in musical signals.

In the main stage of the correction algorithm, the intervals between the detected onsets are compared to the note durations calculated earlier. For every note, the inter-onset intervals are checked to see if at least one of them is roughly as long as the note duration, with a tolerance of 25 milliseconds either way. If that is the case, a weighting function for every note duration mentioned above is set up. The function is created by setting the area around the probable beats to one while setting the rest of the function to zero:

$$w_d(t) = \begin{cases} 1 & for \quad t_d - 25ms < t < t_d + 25ms \\ 0 & else \end{cases} \tag{4.38}$$

with t_d as the positions where the beats fall and d stands for the current note duration. This leads to eight different rectangular weighting functions. These weighting functions are convolved with a pulse train where the pulses are positioned at the detected onset times. This eliminates any onsets that do not conform to the computed beat hypothesis.

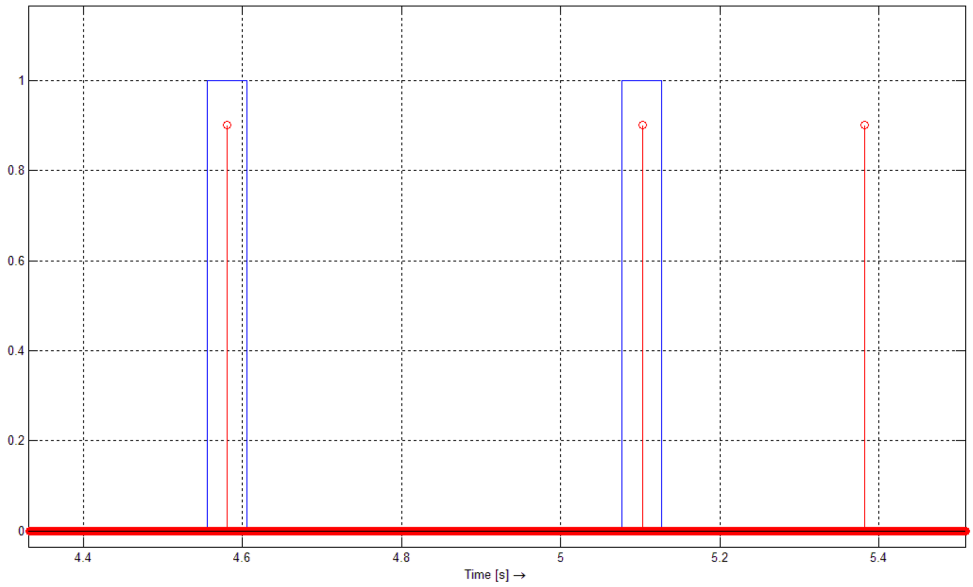


Figure 4.46 Onset detection results correction (detail) with correction function (blue) and found onsets (red)

Figure 4.46 shows such a correction of onset detection results. The blue rectangles show the course of the weighting function for, in this case, quarter notes. The red lines represent detected onsets at that point in time. The first two onsets fall nicely into the computed beat hypothesis while the third onset was probably detected incorrectly. This last onset is deleted by multiplication with the weighting function, which has the value of zero at that temporal point, while the first two onsets remain untouched.

4.3 Evaluation

In the previous sections, different methods for finding onsets in audio signals are described. This chapter presents the results of the evaluation of these algorithms using a database of simple monophonic sounds and more complex pop songs from the last decades.

The audio files used for this purpose are all sampled at 44.1 kHz with 16-bit resolution. They are grouped into four categories: *non-pitched percussive* (NPP), *pitched percussive* (PP), *pitched non-percussive* (PNP) and *mixed* (M). The NPP files contain solely percussive sounds extracted from drum and sequencer tracks, the PP files contain songs using instruments such as bass guitars with clearly percussive attacks marking onsets, the PNP files are made up mostly from songs using soft synthesized sounds or instruments played *legato*, i.e. the transition between successive notes is smooth. Lastly, the sound files containing complex mixtures (M) are taken from pop songs. A total of 26 different sound files containing 1514 onsets is used in the evaluation process. All used sound files are listed in Appendix 9.2.

The reference onsets are labelled by hand using the Sound Onset Labeliser interface from [43]. In order to fairly compare the different detection function implementations, one standard peak-picking algorithm is used throughout, namely the algorithm presented in [11] and explained in section 4.2.7.

A detected onset is defined to be a correct detection if it falls within 35 milliseconds on either side of the according reference onset. While some authors use different tolerance regions for different signal types [52], in the interest of results compatibility across the databases a fixed tolerance time value is chosen. For comparison purposes, an evaluation score that combines the number of correctly detected onsets, false detections and missed detections is computed from the results [52]:

$$R = \frac{TP}{TP + FP + FN} \cdot 100\% \tag{4.39}$$

where *TP* stands for the number of correct detections (true positives), *FP* for the number of incorrect detections (false positives) and *FN* for the number of missed detections (false negatives). The resulting percentage *R* provides information about how well the onset detection algorithm works.

The influence of different hop-sizes on onset detection results is evaluated for all onset detection methods. For the chroma-based approach, the MFCC-based approach and both approaches using modulation spectra, the influence of Klapuri's combination and compression method [42] is evaluated. The parameters for this approach are set to fixed values of 0.8 for the balance factor λ and 100 for the compression factor μ (for more information about this method, see chapter 4.2.3.1). The onset detection results of modulation spectrum-based methods with different frame lengths for the modulation spectrum calculation is also investigated.

4.3.1 Chroma-based Onset Detection

The onset detection method based on the tracking of signal energy in chroma classes is described in chapter 4.2.3. It is evaluated with different audio signals, two different STFT-analysis hop-sizes (256 and 512 samples, which amount to 23 and 46 milliseconds) and different detection function transformations (one using simple differentiation and half-wave rectification, the other using the compression method proposed in [42]).

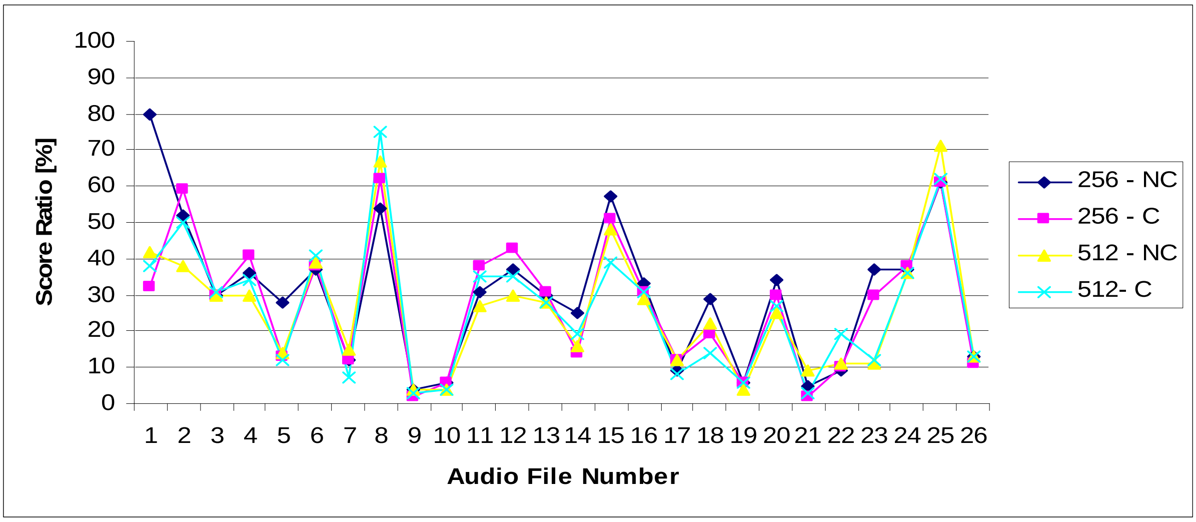


Figure 4.47 Chroma-based onset detection performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

As can be seen in Figure 4.47, there is a large fluctuation in onset detection results according to the audio file that is analysed. While there are visible differences in performance according to hop-size and compression method differences, these are markedly less obvious than those that are related to the audio file characteristics – the general trend stays the same.

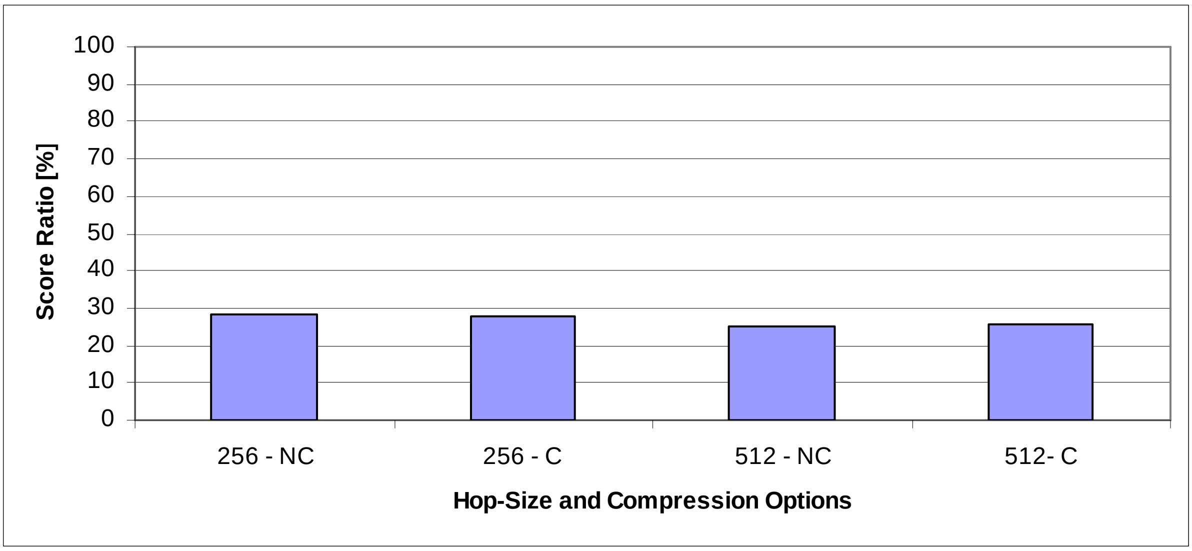


Figure 4.48 Mean chroma-based onset detection performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

Figure 4.48 shows that a small hop-size combined with a simple half-wave rectification and differentiation of the chroma energy data leads to the best results. Compressing the data with Klapuri's approach leads to worse results when using a small hop-size while it improves detection results when using a bigger hop-size. This may be due to the fact that the smaller amount of data that is computed using a hop-size of 512 samples leads to a less-defined onset detection function curve which is "sharpened" by the compression.

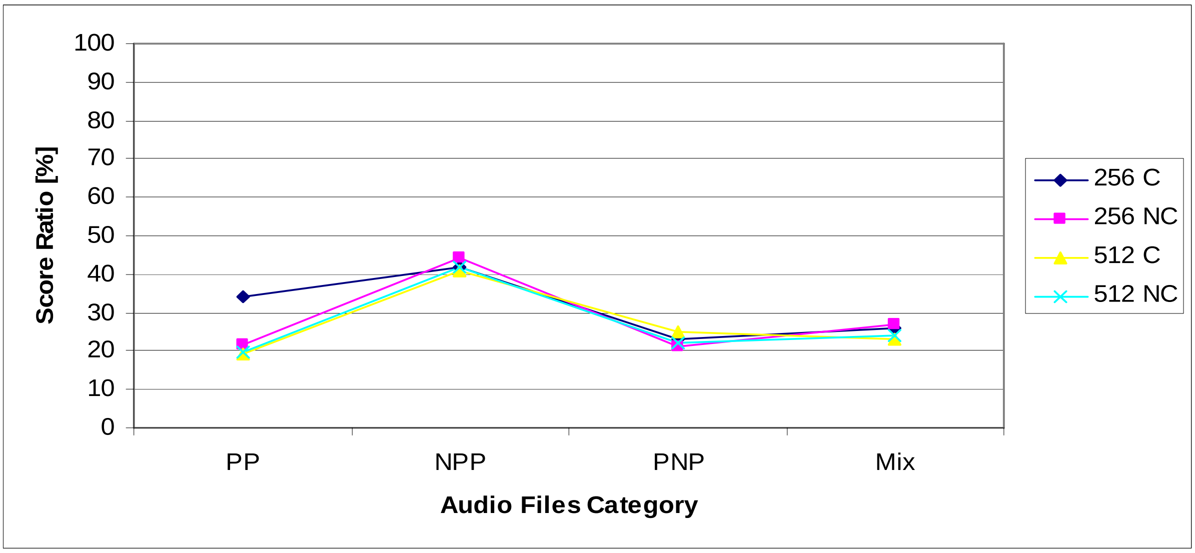


Figure 4.49 Mean chroma-based onset detection performance for pitched percussive, non-pitched percussive, pitched non-percussive and mixed audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

As Figure 4.48 and Figure 4.49 show, the chroma-based onset detection approach does not yield good results over a number of different audio signals. While there are some signals where the results are acceptable (signals 1, 8, 15, 25 in Figure 4.47), the results are not satisfactory for the major part of the evaluated audio signals. The detection method does not work well for any type of signal (Figure 4.49), even pitched signals which in theory should produce good results due to the fact that this method is pitch-based. In fact, the best results are achieved with non-pitched percussive signals (Figure 4.49). This method also has the drawback that it does not detect "soft" onsets because there is no major change in the relationship between tonal energy and total energy, which is the basis of the detection algorithm.

4.3.2 Onset Detection in the Complex Frequency Domain

The approach to onset detection in the complex frequency domain is explained in chapter 4.2.4. Since no rectification and compression is used in this onset detection method, only the influence of two different hop-sizes is looked at closely.

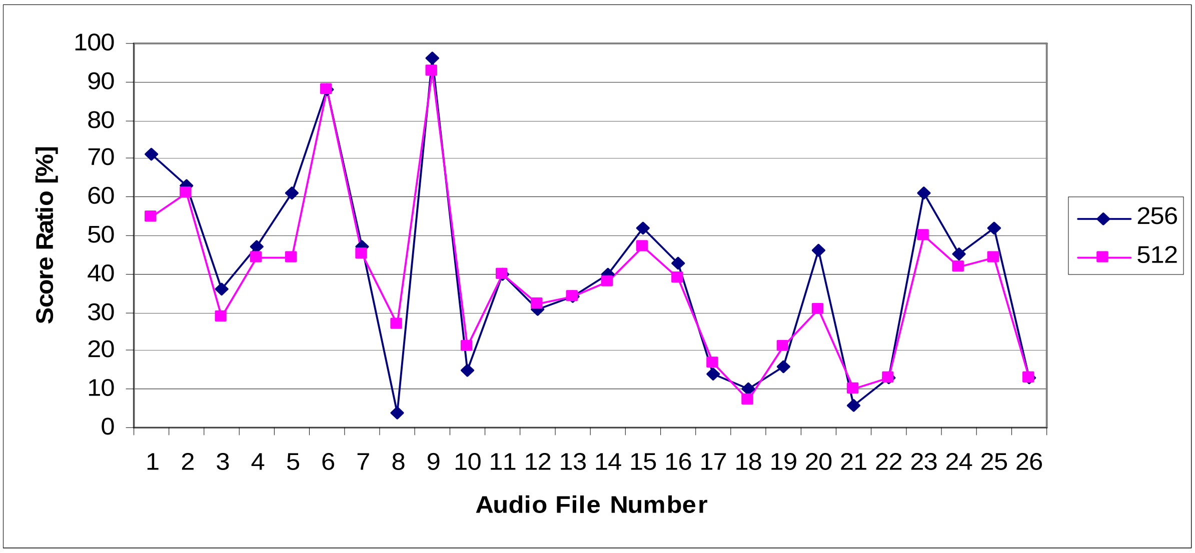


Figure 4.50 Performance of onset detection in the complex frequency domain for all audio files, with hop-sizes of 256 and 512 samples

As is the case with the chroma-based onset detection method, there are some cases where the algorithm works very well (e.g. signals 1, 6 and 9) while for other files the results are very mixed. The hop-size does not have a great influence on results, but using a small hop-size slightly improves detection results – using a hop-size of 256 samples leads to equal and, in most cases, better performance than using a hop-size of 512 samples. This is probably due to the better quantisation of the onset detection function when using smaller time-steps.

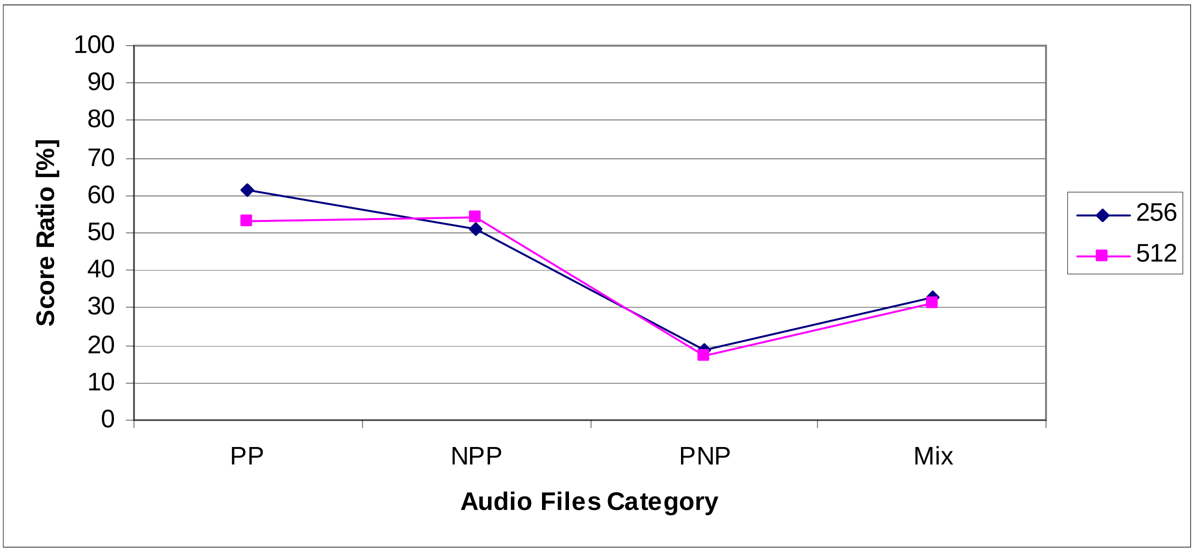


Figure 4.51 Mean complex frequency domain onset detection performance for pitched percussive, non-pitched percussive, pitched non-percussive and mixed audio files with hop-sizes of 256 and 512 samples

Figure 4.51 shows the mean performance of the onset detection approach in the complex frequency domain. As expected, the algorithm works well for percussive signals but not as well for non-percussive sounds. The best results are obtained when the signals are percussive – pitched and non-pitched. When faced with complex, polyphonic sounds, the algorithm does not work as well, but still significantly better than when faced with non-percussive sounds.

The figure also shows that a short hop-size leads to better overall results (about 5%) than a larger hop-size, due to the fact that the onset detection function is sampled at a higher rate.

4.3.3 MFCC-based Onset Detection

The onset detection based on the tracking of MFCCs over time is explained in chapter 4.2.5. As with the chroma-based methods, two different hop-sizes (256 and 512 samples) as well as different detection function creation methods, one using simple differentiation and half-wave rectification, the other using the compression method proposed in [42], are investigated.

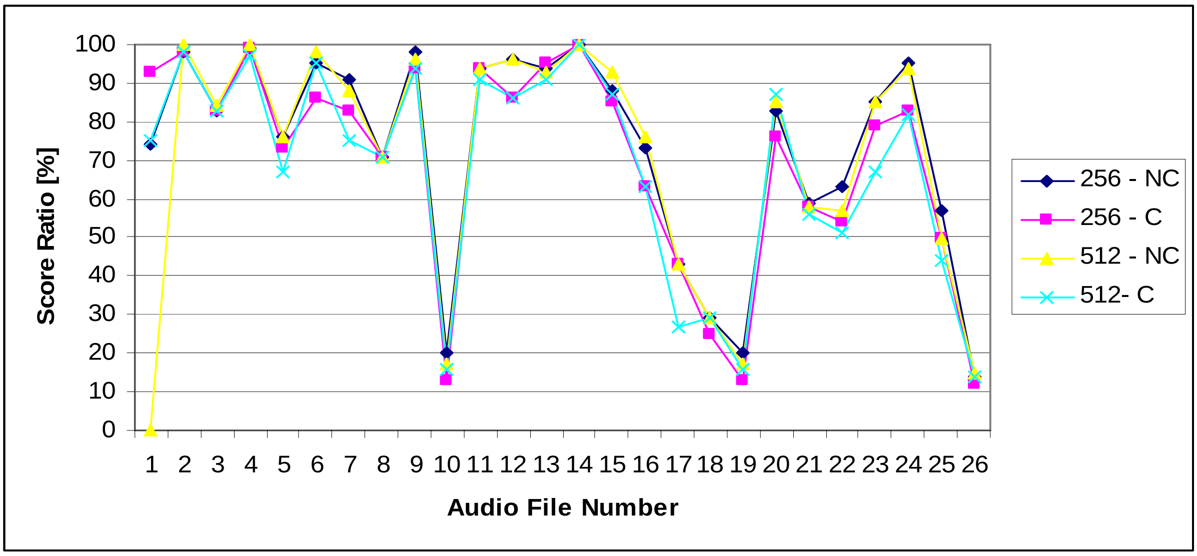


Figure 4.52 MFCC-based onset detection performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

As Figure 4.52 shows, this approach works well for a number of different audio signals – many detection results lie in the 80%-100% range. This means a high number of correct detections and a low number of false detections. However, there are some signals where the algorithm does not perform nearly as well.

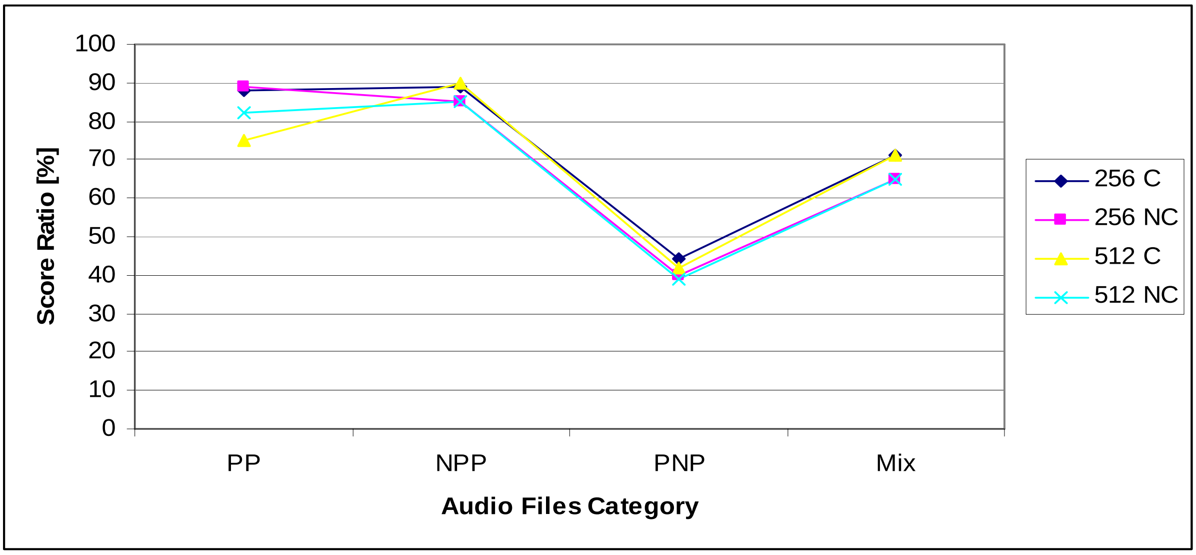


Figure 4.53 Mean MFCC-based onset detection performance for pitched percussive, non-pitched percussive, pitched non-percussive and mixed audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

As expected, the algorithm works well with percussive (pitched and non-pitched) signals and not very well with non-percussive signals. When faced with complex signals, the approach still works very well (a score of about 70% when using simple differentiation and half-wave rectification). This approach yielded the best results of all evaluated onset detection methods. This may be due to the fact that, as described in chapter 5.4, MFCCs in general and the first MFC coefficient in particular (which is also used to create the detection function) are heavily correlated with onset times. While the curves for the different hop-sizes and detection function creation modes have a similar shape, their deviation is evaluated below.

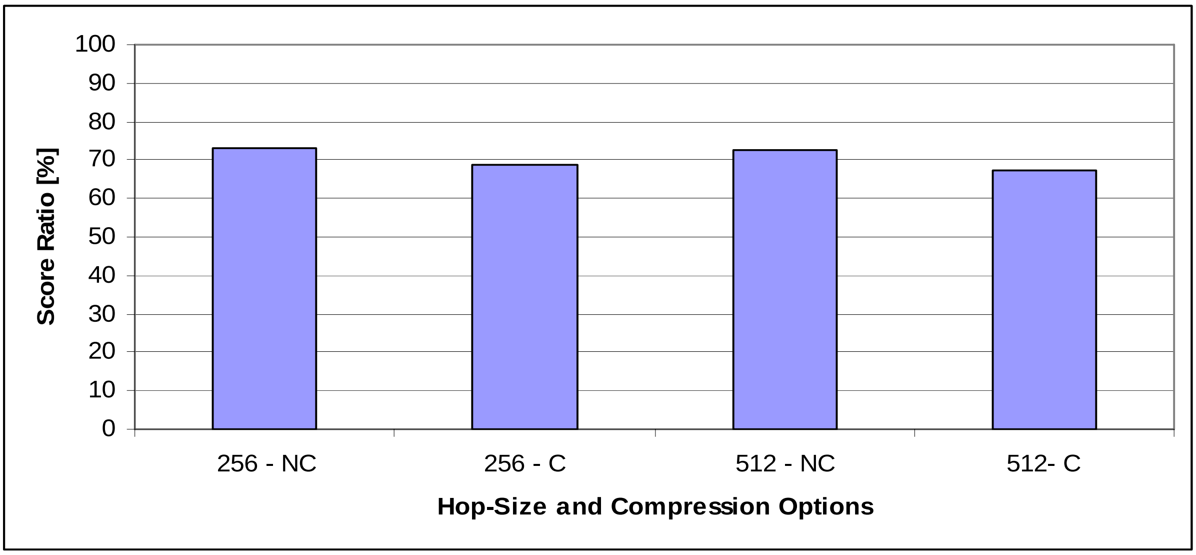


Figure 4.54 Mean MFCC-based onset detection performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C)

Figure 4.54 shows the mean onset detection performance of the MFCC-based method for different hop-sizes and detection function creation methods. The best results are achieved using the smaller hop-size and simple half-wave rectification and differentiation. The advantage of using a small hop-size is that the detection function is sampled at a higher rate, while the slightly worse performance when using compression may be due to the fact that the information about the signal spectrum is already highly compressed by the transformation into the cepstral domain.

4.3.4 Onset Detection based on Modulation Spectra

The two approaches to onset detection using modulation spectra that are implemented in this thesis are discussed in detail in section 4.2.6. Two different hop-sizes (256 and 512 samples) as well as different detection function creation methods (using simple half-wave rectification and differentiation, or on the other hand using the compression method proposed in [42]), are investigated. Also, two different modulation spectrum frame lengths are evaluated separately. They are set at 18 and 24 original STFT frames, i.e. 420 milliseconds and 560 milliseconds, respectively.

4.3.4.1 Sub-Band Energy Trajectories

As with the other onset detection methods, this approach is evaluated using a number of different options.

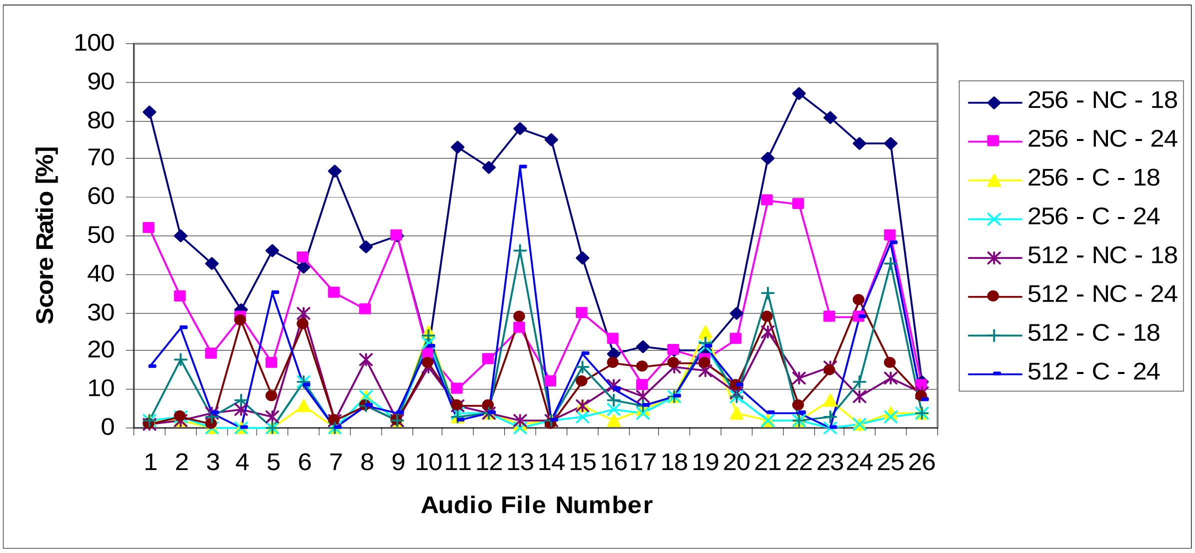


Figure 4.55 Onset detection based on sub-band trajectories performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C) and modulation spectrum frame lengths of 18 and 24 STFT frames

As can be seen in Figure 4.55, the results vary greatly according to the chosen options and the analysed audio files. The performance of the algorithm sorted according to the signal category is shown below.

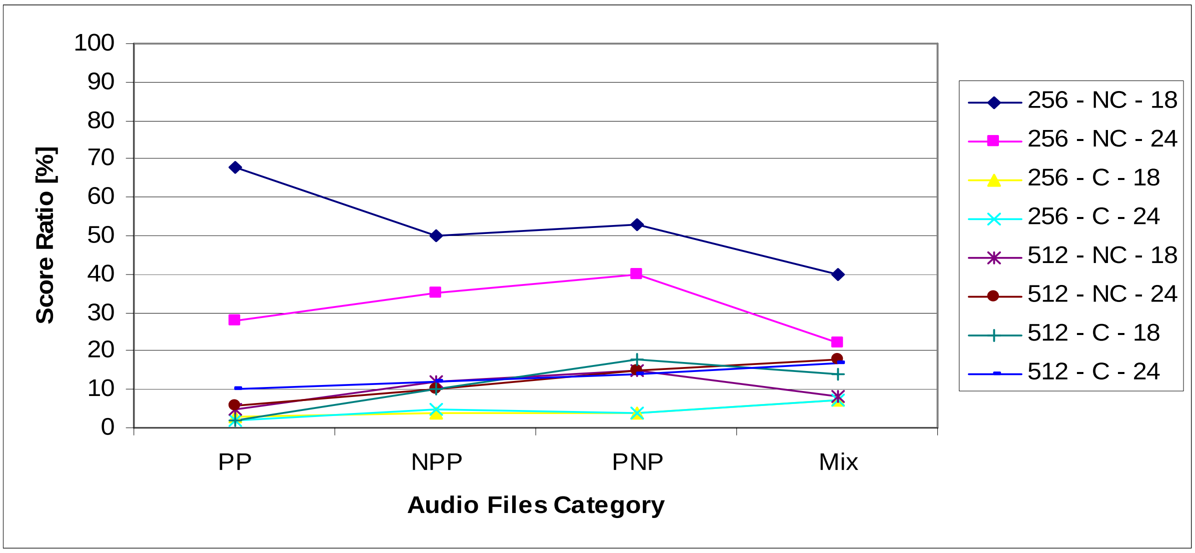


Figure 4.56 Mean onset detection using sub-band trajectories performance for pitched percussive, non-pitched percussive, pitched non-percussive and mixed audio files with hop-sizes of 256 and 512 samples and simple half-wave rectification and differentiation (NC) and the compression proposed in [42] (C) as well as modulation spectrum frame lengths of 18 and 24 STFT frames

Figure 4.56 shows that, using the options mentioned above, the best results are obtained for pitched percussive sounds, which means there is a broad-band increase across the modulation spectrum from the percussive parts and significant increase in the lower bands of the modulation spectrum stemming from the pitched parts. This onset detection method works quite well for pitched non-percussive sounds, probably due to the lower-band changes in the modulation spectrum mentioned earlier. However, when faced with complex signals, the results deteriorate, which may be due to the excess of information in the modulation spectrum.

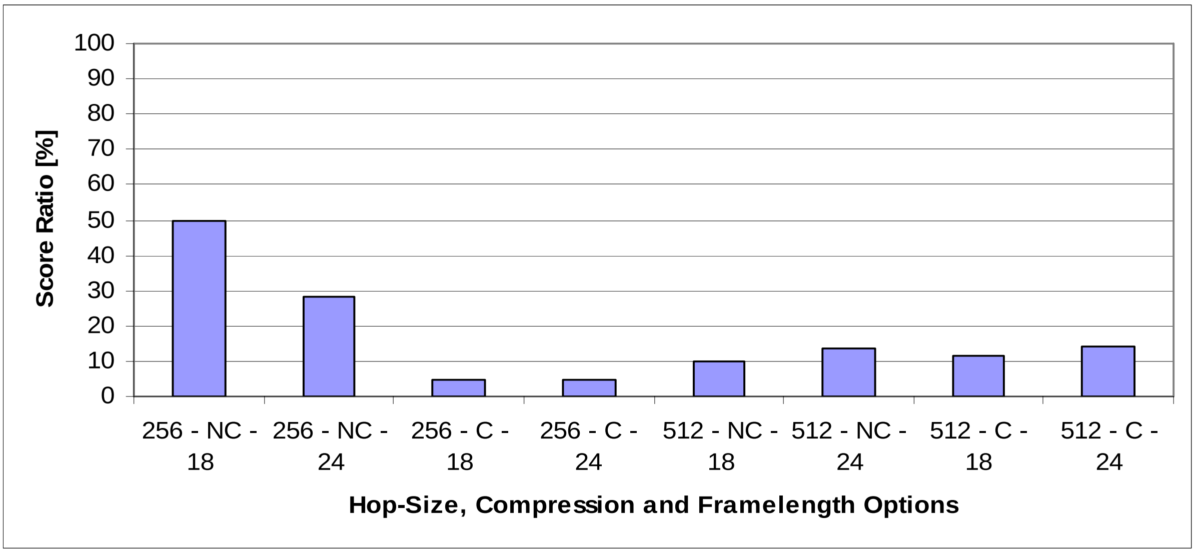


Figure 4.57 Mean onset detection using sub-band trajectories performance for all signals with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C) as well as modulation spectrum frame lengths of 18 and 24 STFT frames

The best onset detection results by far are obtained using a STFT hop-size of 256 samples, simple differentiation and half-wave rectification and a modulation spectrum frame length of 18 STFT frames. This is due to the better detection function quantisation when using smaller time-steps. Using the compression method proposed in [42] has a positive impact when using bigger modulation spectrum frame lengths because the larger amount of information is compressed.

4.3.4.2 Weighted Sub-Band Energy Trajectories

This detection method performed very badly over a wide range of audio files.

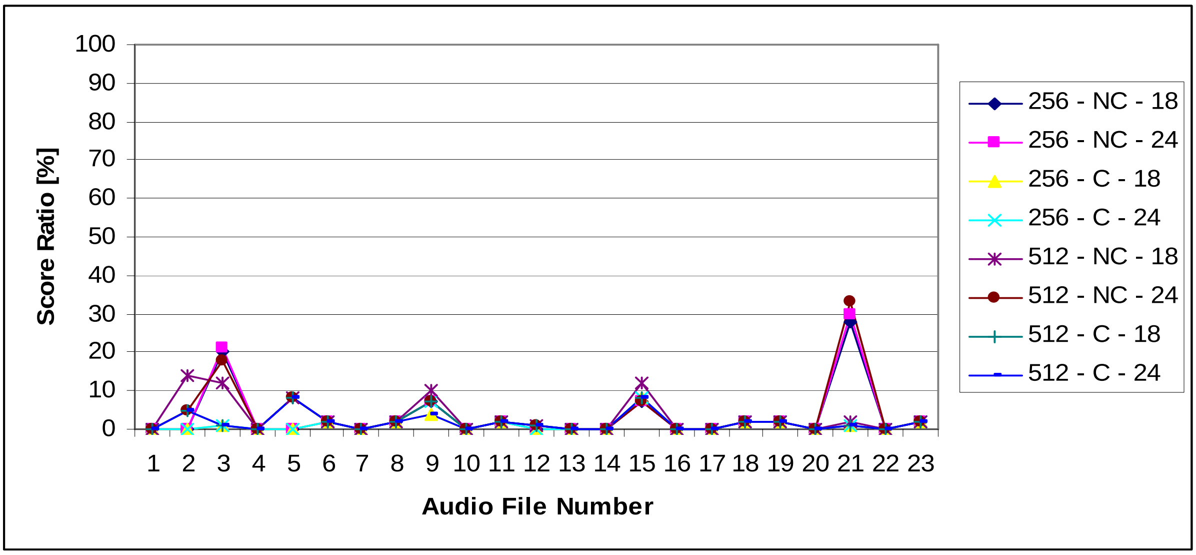


Figure 4.58 Onset detection based on weighted sub-band trajectories performance for all audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C) and modulation spectrum frame lengths of 18 and 24 STFT frames

As can be seen in Figure 4.58, the algorithm did not perform well for any audio signal. The best results are worse than the worst results of the other detection functions. This is true for all different signal categories, as shown below.

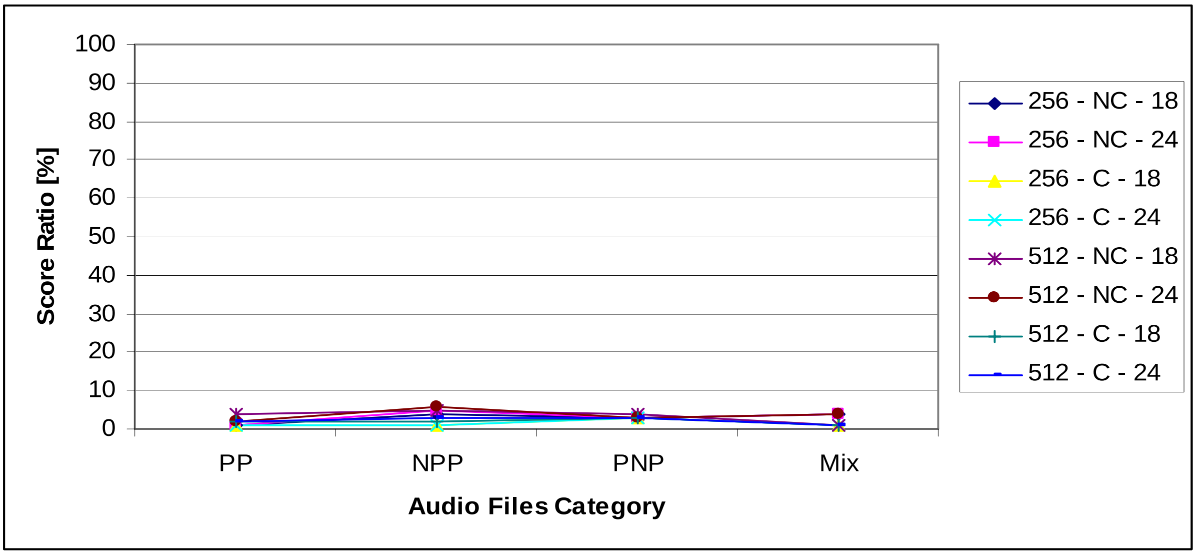


Figure 4.59 Mean onset detection using weighted sub-band trajectories performance for pitched percussive, non-pitched percussive, pitched non-percussive and mixed audio files with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C) as well as modulation spectrum frame lengths of 18 and 24 STFT frames

While the algorithm performs best when faced with percussive signals, as is the case with the other detection methods, results in all categories are well below a 10% score (Figure 4.59). This is unacceptable for any application.

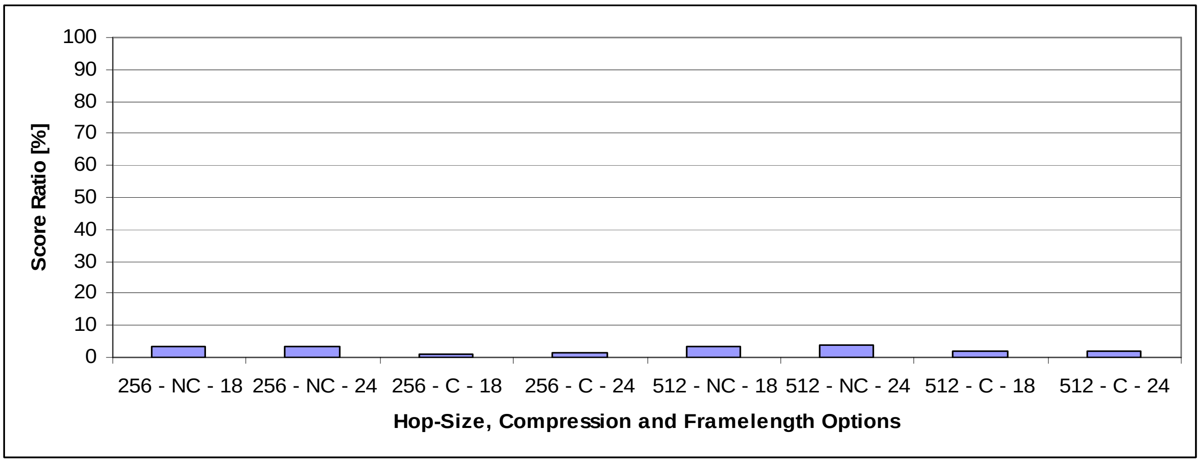


Figure 4.60 Mean onset detection using weighted sub-band trajectories performance for all signals with hop-sizes of 256 and 512 samples and simple differentiation and half-wave rectification (NC) and the compression proposed in [42] (C) as well as modulation spectrum frame lengths of 18 and 24 STFT frames

Figure 4.60 shows the influences of different STFT hop-sizes, detection function creation methods and modulation spectrum frame lengths. The alternative using a 512-sample STFT hop-size, simple half-wave rectification and differentiation and a modulation spectrum frame length of 24 STFT frames works best. However, it is difficult to gather information about the algorithm performance from this data as all values are so small.

4.3.5 Comparison

In order to find the detection function that works best over a wide range of signals, the results obtained with the different onset detection methods presented above are compared directly. For every method, the combination of options which leads to the best mean results over all audio files is used.

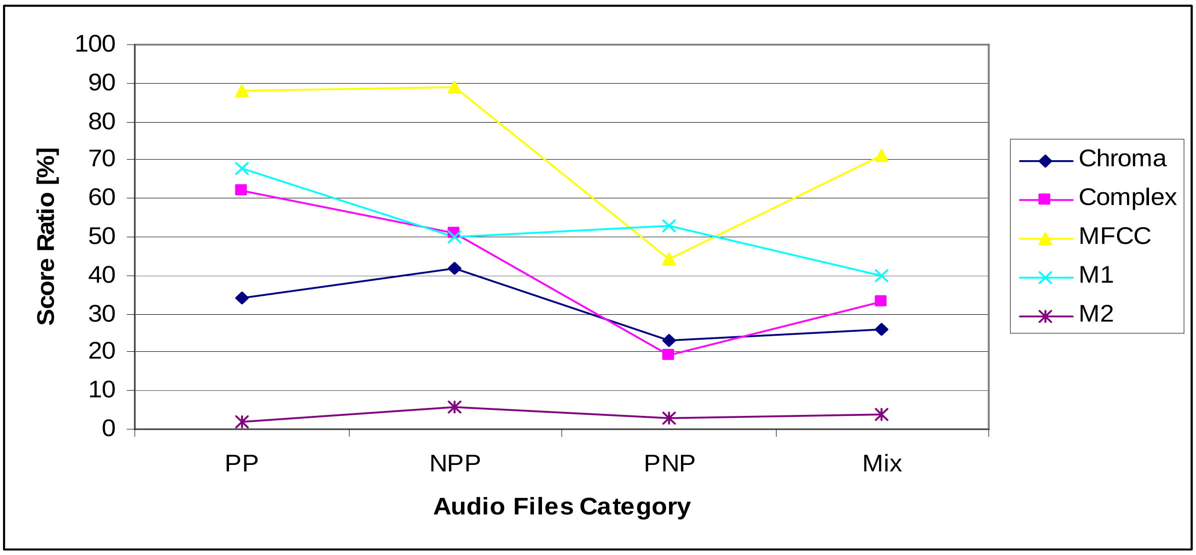


Figure 4.61 Mean performance of onset detection methods („Chroma“ for chroma-based, „Complex“ for complex frequency domain, „MFCC“ for MFCC-based, „M1“ for sub-band trajectory based and “M2” for weighted sub-band trajectory based approaches) for different audio signal categories (pitched percussive, non-pitched percussive, pitched non-percussive and mixed)

As seen in Figure 4.61, most detection methods work best when faced with percussive sounds. This is explained by the fact that all methods are in principle spectrum-based, and percussive sounds lead to broad-band energy increases.

The detection method based on following the first MFC coefficient over time works best for three out of four categories, which makes it the ideal candidate for onset detection tasks that have to deal with different signal types. The good performance can be explained by the fact that the MFCC structure over time is heavily correlated with onset times, as explained in chapter 5.4. The versatility of this approach is demonstrated in the next figure, where the mean performance of the different onset detection algorithms over all audio files is pictured.

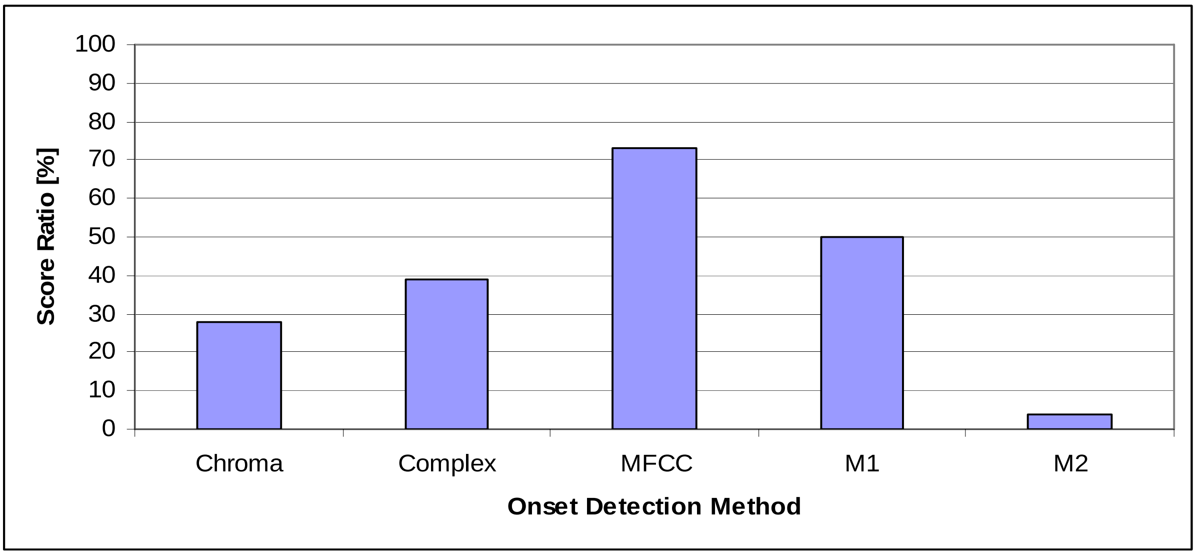


Figure 4.62 Mean performance of onset detections („Chroma“ for chroma-based, „Complex“ for complex frequency domain, „MFCC“ for MFCC-based, „M1“ for sub-band trajectory based and “M2” for weighted sub-band trajectory based approaches) for all analysed audio signals

The success rate of the MFCC-based approach can be described as very good over a variety of different signals, as can be seen in Figure 4.62. For a complete overview, the results of all onset detection methods with all different options for all evaluated audio signals are given in Table 4.1 and Table 4.2.

	Chroma				**Complex**		**MFCC**			
Hop-size	256	256	512	512	256	512	256	256	512	512
Compression	No	Yes	No	Yes	-	-	No	Yes	No	Yes
Bass	80	32	42	38	71	55	74	93	0	75
Boulevards	52	59	38	50	63	61	98	98	100	98
Clansman	30	30	30	31	36	29	83	83	84	83
Dingdong	36	41	30	34	47	44	99	99	100	97
Distguit	28	13	14	12	61	44	76	73	76	67
Drums	37	38	39	41	88	88	95	86	98	95
Drumduet	12	12	15	7	47	45	91	83	88	75
Drum_bands	54	62	67	75	4	27	71	71	71	71
FX	4	2	4	3	96	93	98	94	96	94
Godzilla	6	6	4	4	15	21	20	13	17	16
Guitar	31	38	27	35	40	40	94	94	94	91
Hiphop	37	43	30	35	31	32	96	86	96	86
Musclemuseum	30	31	28	28	34	34	94	95	93	91
Piano	25	14	16	19	40	38	100	100	100	100
Pingpong	57	51	48	39	52	47	88	85	93	87
Pop1	33	31	29	31	43	39	73	63	76	63
Pop2	9	12	12	8	14	17	43	43	43	27
Readyornot	29	19	22	14	10	7	29	25	29	29
Rock	6	6	4	6	16	21	20	13	17	16
Sunshine	34	30	25	27	46	31	83	76	85	87
Synth1	5	2	9	3	6	10	59	58	58	56
Synth2	9	10	11	19	13	13	63	54	57	51
Synthbass	37	30	11	12	61	50	85	79	85	67
Übermensch	37	38	36	36	45	42	95	83	94	82
Violin	61	61	71	62	52	44	57	50	50	44
Vox	13	11	13	13	13	13	14	12	15	14

Table 4.1 Onset detection results for chroma-based, complex frequency domain and MFCC-based onset detection methods with hop-size and compression options

	M1								**M2**							
Hopsize	256	25 6	256	256	51 2	51 2	512	512	25 6	25 6	256	256	51 2	51 2	512	512
Compression	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes
New Frame Length	18	24	18	24	18	24	18	24	18	24	18	24	18	24	18	24
Bass	82	52	2	2	1	1	2	16	2	2	2	2	2	2	2	2
Boulevards	50	34	2	3	2	3	18	26	2	2	2	2	5	2	2	3
Clansman	43	19	0	0	4	1	3	4	0	0	0	0	0	0	0	0
Dingdong	31	29	0	0	5	28	7	0	0	0	0	0	0	0	0	0
Distguit	46	17	0	0	3	8	0	35	0	0	0	0	14	5	5	5
Drums	42	44	6	12	30	27	12	11	20	21	1	1	12	18	1	1
Drumduet	67	35	0	0	2	2	0	0	0	0	0	0	0	0	0	0
Drum_bands	47	31	8	8	18	6	6	6	0	0	0	0	8	8	8	8
FX	50	50	2	2	2	2	2	4	2	2	2	2	2	2	2	2
Godzilla	20	19	25	22	16	17	24	21	0	0	0	0	0	0	0	0
Guitar	73	10	3	4	6	6	3	2	2	2	2	2	2	2	2	2
Hiphop	68	18	4	4	4	6	4	4	7	7	4	7	10	7	7	4
Musclemuseum	78	26	1	0	2	29	46	68	0	0	0	0	0	0	0	0
Piano	75	12	2	2	2	1	2	2	2	2	2	2	2	2	2	2
Pingpong	44	30	6	3	6	12	16	19	0	0	0	0	1	1	1	1
Pop1	19	23	2	5	11	17	7	10	0	0	0	0	0	0	0	0
Pop2	21	11	5	4	8	16	6	6	0	0	0	0	0	0	0	0
Readyornot	20	20	8	8	16	17	8	8	7	7	8	8	12	7	8	8
Rock	20	18	25	21	15	17	22	21	0	0	0	0	0	0	0	0
Sunshine	30	23	4	8	9	11	8	11	0	0	0	0	0	0	0	0
Synth1	70	59	2	2	25	29	35	4	2	2	2	2	2	2	2	2
Synth2	87	58	2	2	13	6	2	4	2	2	2	2	2	2	2	2
Synthbass	81	29	7	0	16	15	3	0	0	0	0	0	0	0	0	0
Übermensch	74	29	1	1	8	33	12	29	28	30	1	1	2	33	1	1
Violin	74	50	4	3	13	17	43	48	0	0	0	0	0	0	0	0
Vox	12	11	4	4	9	8	4	7	2	2	2	2	2	2	2	2

Table 4.2 Onset detection results for sub-band trajectories (M1) and weighted sub-band trajectories (M2) onset detection methods with hop-size, compression and modulation spectrum frame length options

5 Features

While human listeners intuitively judge the similarity or dissimilarity between audio signals, the task of comparing signals with a computer algorithm is quite difficult. As the temporal and the spectral evolutions of different signals cannot be compared directly, any implementation of similarity rating has to rely on a number of quantised signal characteristics called *audio features*. The general procedure is to divide the signal into short-time frames in which the signal is assumed to be stationary. For these reduced segments, analysis is performed that leads to a number of *audio descriptors* or *low-level features*. These can be combined to produce a *high-level feature* representation, in most cases a vector, which describes the signal as fully as possible.

Since some of the features are heavily correlated (for example, the spectral roll-off and brightness measures presented in chapter 5.2 essentially describe the same audio characteristic), the results of a listening test concerning subjective sound similarities are evaluated to find a combination of features that characterises the audio signal while relying on as few features as possible. This is explained in detail in chapter 6.2.2.

This chapter presents the features that are used in this thesis. First, features which describe signal characteristics in the time domain are explained, then features that reflect the spectral structure of the sound and features extracted from statistical properties of the signal spectrum are detailed. The last part of the chapter is dedicated to the correlation between the change in low-level features and the positions of onsets in audio signals.

The temporal and most of the spectral features are calculated “by hand”, the statistical properties and some spectral features are computed using the MIR Toolbox [16]. The audio signals are analysed after decimation to 11.025 kHz and low-pass filtering using a Chebyshev low-pass filter. A STFT frame size of 11.6 milliseconds (128 samples at 11.025 kHz) and a hop-size of 5.8 milliseconds (64 samples at 11.025 kHz) are used. To minimise spectral leakage, the signal is windowed using a Hanning window. For better spectral resolution, the frame is expanded to 23.2 milliseconds (256 samples at 11.025 kHz) by simple zero-padding.

5.1 Temporal Low-Level Features

Temporal low-level features are used to describe the behaviour of signals in the time domain. Features concerning the energy (*RMS*) and the noisiness of the signal (*Zero-Crossing Rate*) are evaluated.

5.1.1 RMS

The course of the global energy of a signal can be described by using the *root-mean square* (RMS) measure.

$$\overline{x}_{RMS} = \sqrt{\frac{1}{N}\sum_{n=0}^{N} x^2[n]} \tag{5.1}$$

This represents the mean of the squared signal amplitude in such a way that larger values have a greater influence on the mean than smaller values. The RMS value is closely related to the perceived loudness of audio signals.

5.1.2 Zero-Crossing Rate

The *zero-crossing rate* describes the number of times the signal amplitude crosses zero per time unit, i.e. the number of sign changes in the time domain. This can be interpreted as a measure of noisiness or tonality – it is also correlated with the signal pitch, since a noisy signal will tend to change signs more often.

$$zc = \frac{1}{2}\sum_{n=1}^{N}\left|\operatorname{sgn}(x[n]) - \operatorname{sgn}(x[n-1])\right| \qquad \text{with } \operatorname{sgn}(x) = \begin{cases} -1 & \mathit{for} \quad x < 0 \\ 1 & \mathit{for} \quad x \geq 0 \end{cases}$$

5.2 Spectral Low-Level Features

Spectral low-level features are used to describe the spectral structure of audio signals. Features evaluated in this thesis include the Mel-Frequency Cepstral Coefficients (MFCC) which are also used to extract onset information from signals (see chapter 4.2.5), features describing the amount of signal energy in high frequency bands (*Spectral Roll-Off, Brightness*), the relationships of partial tones (*Roughness*, *Irregularity*) and the tonal quality and center of the spectrum (*Spectral Spread, Spectral Flatness, Pitch*).

5.2.1 Mel-Frequency Cepstral Coefficients

Mel-Frequency Cepstral Coefficients represent the spectral characteristics of a signal in a very compact way. They are used in speech processing and coding algorithms because the *cepstral domain*, defined as the inverse transform of the logarithmic signal spectrum, is very useful to extract and manipulate the spectral envelope of a signal [44]. Recently, MFCCs have been discovered to work well as audio features ([13], [53]). The number of used MFC coefficients determines the resolution of the spectral envelope.

The MFCCs are computed as follows: first, the signal magnitude spectrum is determined, in this case by computing the FFT. The magnitude spectrum is then filtered by a Mel filter bank, a filter bank of triangular filters that groups together frequency components according to the Mel scale[1]. The resulting groups are summed and the logarithm is computed, mirroring the behaviour of the human cochlea, where neuronal impulses are evaluated in frequency groups, resulting in an integration of the impulses and compression of the signal dynamic. In the last stage of the calculation, the values obtained from the filter bank are transformed into the cepstral domain using the Discrete Cosine Transform (DCT) [45].

[1] a frequency scale based on the human perception of pitch

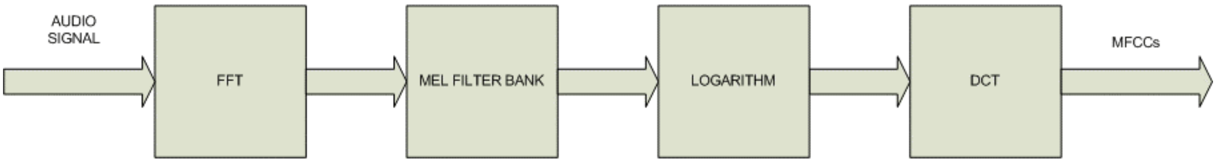


Figure 5.1 Mel-Frequency Cepstral Coefficient Calculation

There are also other methods of calculating the MFCCs – some authors compute the logarithm before filtering [53], others use different filter bank scales such as the *Bark* or *ERB* (*Equivalent Rectangular Bandwidth*) scales [13]; there are also other methods which differ from those presented mainly in the filter bank design and the compression method, the bandwidth of the evaluated spectrum and the number of computed MFCCs [45].

The first stage of the MFC calculation algorithm implemented in this thesis consists of dividing the signal into frames of equal length. These frames are expanded with *overlap segments* to ensure correct calculation near the frame borders. The overlap segments are faded in and out by multiplication of the segment with sin²- and cos²- functions. This is to make sure that there is no energy loss and no clipping.

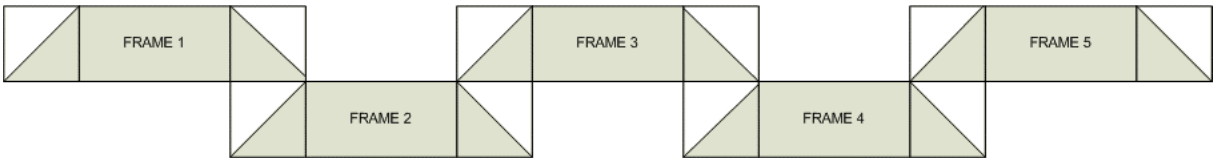


Figure 5.2 Overlapping frames

The frames are 256 samples (23.2 milliseconds at 11.025 kHz sampling rate) long, the overlap segments have a quarter of that length, i.e. 64 samples (about 6 milliseconds at 11.025 kHz). This means that the overall time resolution of the algorithm amounts to 320 samples, which at 11.025 kHz sampling rate corresponds to approximately 29 milliseconds, a length of time where the signal is assumed to be stationary.

The linear frequency scale is set by default to values from 20 Hz to half the sampling frequency, which in our case equals 5512 Hz (rounded). This scale is then mapped to the non-linear Mel scale as follows:

$$f_{mel} = 2595 \cdot \log_{10}\left(1 + \frac{f_{lin}}{700}\right) \tag{5.2}$$

with f_{mel} as the frequency on the Mel scale and f_{lin} as the linear frequency [54]. This closely approximates the Mel scale for frequencies below 1 kHz.

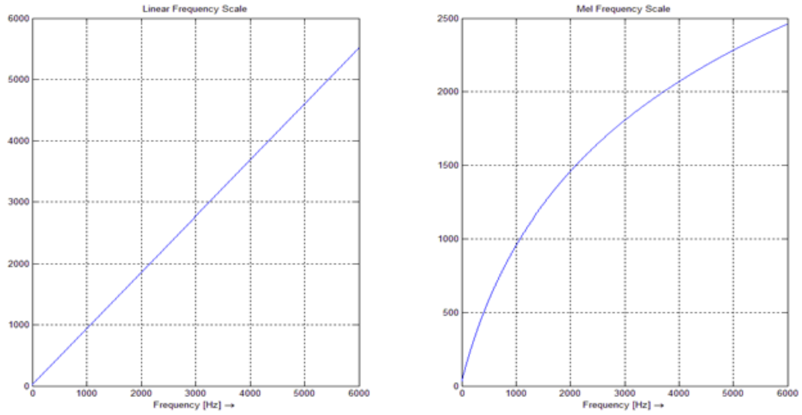


Figure 5.3 Linear and logarithmic (Mel) frequency scales

The Mel frequency scale is roughly linear up to 1 kHz and has a logarithmic slope above 1 kHz. Across this frequency scale triangular filters are placed, the number of which is dictated by the number of MFC coefficients to be calculated, which in our case is 20. The filters are uniformly distributed along the frequency axis and the area under the triangle always stays the same. The setup of the filter bank follows Slaney's Auditory Toolbox definition mentioned in [54]: This method was chosen because it performed well in evaluation tests [54].

$$H_i(k) = \begin{cases} 0 & \text{for} \quad k < f_{b_{i-1}} \\ \dfrac{2\left(k - f_{b_{i-1}}\right)}{\left(f_{b_i} - f_{b_{i-1}}\right)\left(f_{b_{i+1}} - f_{b_{i-1}}\right)} & \text{for} \quad f_{b_{i-1}} \le k \le f_{b_i} \\ \dfrac{2\left(f_{b_{i+1}} - k\right)}{\left(f_{b_{i+1}} - f_{b_i}\right)\left(f_{b_{i+1}} - f_{b_{i-1}}\right)} & \text{for} \quad f_{b_i} \le k \le f_{b_{i+1}} \\ 0 & \text{for} \quad k > f_{b_{i+1}} \end{cases} \tag{5.3}$$

with $i = 1,2,...,M$ as the filter number, f_b as the triangle center frequency and $k = 1,2,...,N$ as the spectral bins. The result is a filter matrix with M rows and N columns containing the DFT coefficients, with the row sum staying constant.

To model the human perception more closely, the filter bank is modified using an A-weighting filter. This mostly affects the lower frequencies, which are suppressed in accordance to the fact that the human hearing system is not very sensitive in low frequency regions, whereas the highest frequencies that appear in this approach lie in the range of 5 kHz, where the human ear is at its best. The filter weights are set to

$$A_{dB(A)}(f)=\frac{12200^2\cdot f^4}{\left(f^2+20.6^2\right)\cdot\left(f^2+12200^2\right)\cdot\sqrt{f^2+107.7^2}\cdot\sqrt{f^2+737.9^2}} \qquad (5.4)$$

The triangular filter bank is weighted with the normalised weights of the dB(A) filter by multiplication.

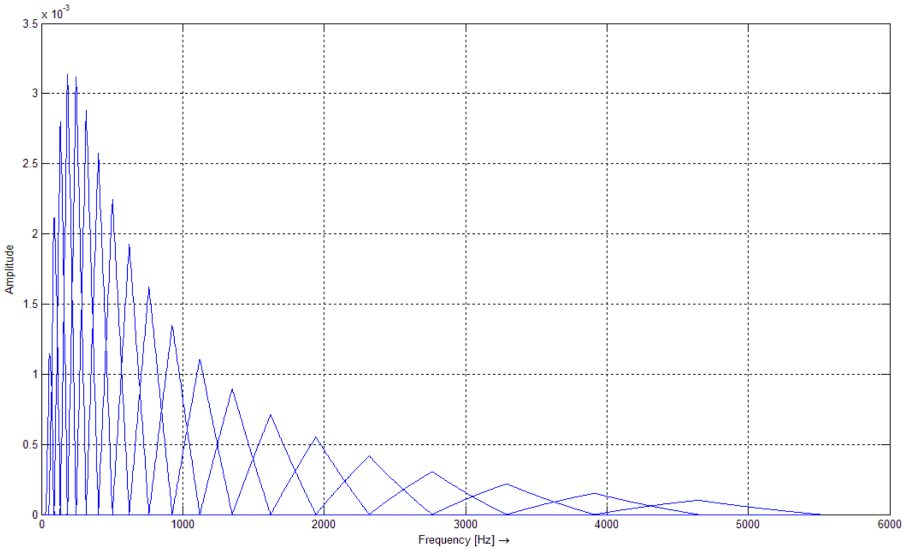


Figure 5.4 Triangular filter bank weighted with dB(A) filter

After filtering the amplitude spectrum of the signal and taking the logarithm, the transformed signal is given by

$$X'[m]=\log\left(\sum_{k=0}^{N-1}\left|X[k]\right|\cdot H[k,m]\right) \qquad (5.5)$$

where *m* is the frame number. In the last processing stage, the DCT is used to transform the signal into the cepstral domain. The DCT is a transform that is used mostly in speech and image processing. An important property thereof called *energy concentration* is that, in contrast to the DFT, a major part of the signal energy is concentrated in the first few coefficients, which makes the DCT attractive for data compression tasks [55].

5.2.2 Pitch

The pitch of a sound is defined as the fundamental tone of a sound and is calculated following the approach presented in [35]. A local maximum in the magnitude spectrum of a signal at the bin k_0 by itself leads to a pitch frequency estimation

$$\hat{f}_0 = k_0 \cdot \frac{f_s}{N} \tag{5.6}$$

The frequency resolution is then improved by evaluating the phase difference between the maximum in successive frames. The conclusive frequency estimate is the given by

$$\hat{f}_0 = \frac{\varphi_{n+1} - \varphi_n}{2\pi h} \cdot f_s \tag{5.7}$$

with *h* as the analysis frame hop-size, φ_{n+1} as the combination of the expected phase and the phase difference between expected and actual phase of the second frame and φ_n as the known phase of the first frame. This correction stage that considers the phase difference leads to clearly improved pitch detection results [35].

5.2.3 Spectral Centroid

The *spectral centroid* is an indicator of the center of gravity of the spectral energy distribution. It correlates with the fullness of the sound – the higher the centroid, the more present are higher partial tones and the more “brilliant” the sound.

$$CN = \frac{\sum_{k=1}^{N/2} k \cdot |X[k]|^2}{\sum_{k=1}^{N/2} |X[k]|^2} \cdot \frac{f_s}{N} \tag{5.8}$$

with *k* as the frequency bin number and *N* the total of frequency bins.

5.2.4 Spectral Spread

The *spectral spread* describes the spread of the signal spectrum in relation to the spectral centroid introduced above. It can be interpreted as a measure of tonality, where noisy signals that display a broad-band spectrum will have a higher spread than tonal sounds that are confined to narrow-band peaks.

$$SP = \sqrt{\frac{\sum_{k=1}^{N/2}\left(k\cdot\frac{f_s}{N}-CN\right)^2\cdot\left|X[k]\right|^2}{\sum_{k=1}^{N/2}\left|X[k]\right|^2}} \tag{5.9}$$

with k as the FFT bin number, N as the total of frequency bins and CN as the spectral centroid.

5.2.5 Spectral Roll-Off

The *spectral roll-off* loosely describes the shape of the signal spectrum. It is a measure of the frequency range where the major part of the signal energy is found. The roll-off frequency value is found by summing the signal energy across the frequency spectrum and finding the frequency where a certain percentage of the signal energy lies (the percentage is mostly defined as 85% [15]). It is defined as the frequency bin number *ro* that satisfies

$$\sum_{k=1}^{ro}\left|X[k]\right|^2 \geq 0.85\cdot\sum_{k=1}^{N/2}\left|X[k]\right|^2 \tag{5.10}$$

with N as the number of FFT bins. The roll-off frequency itself is then computed by

$$RO = ro\cdot\frac{f_s}{N} \tag{5.11}$$

5.2.6 Spectral Brightness

The *spectral brightness* is a measure of high-frequency energy content similar to the spectral roll-off described above. The approach differs from the roll-off calculation in that a threshold frequency is fixed, in this case at 1.5 kHz and the percentage of energy above that cut-off frequency is computed.

$$BN = \frac{\sum_{k=N\cdot\frac{f_c}{f_s}}^{N/2} \left|X[k]\right|^2}{\sum_{k=1}^{N/2} \left|X[k]\right|^2} \tag{5.12}$$

with f_c as the threshold frequency and N as the FFT bin number.

5.2.7 Roughness

Roughness is a measure of the *sensory dissonance* that is produced by two sinusoidal signals with regard to the frequency ratio between them and is derived from Plomp and Levelt's concept of *tonal consonance* [56]. The two signals can be separate notes or partial tones of a harmonic sound with the constraint that both have to belong to the same critical band [57].

The roughness is calculated using the MIR Toolbox [16]. It is computed pairwise between all local maxima in the spectrum. The final roughness value is gained by calculating the average over all roughness values.

5.2.8 Irregularity

The *irregularity* measure describes the amount of variation of the distance between successive partial tones in a harmonic sound [57]. This can be interpreted as an indicator for the tonality of a signal, where the irregularity will be lower when the partials are harmonically related.

The irregularity feature is given by

$$IR = \frac{\sum_{k=1}^{N} \left(a_k - a_{k+1}\right)^2}{\sum_{k=1}^{N} {a_k}^2} \tag{5.13}$$

where a_k denotes the k-th partial tone. This amounts to measuring the distances between consecutive partials.

5.3 Statistical Features

In addition to evaluating audio signals regarding their temporal and spectral characteristics, some statistical features are calculated and analysed in regard to their usefulness for describing the signals. This is done by computing the 3rd and 4th *central moments* as well as the shape of the probability distribution of the signal spectra. Central moments are defined as the moments around the mean of the analysed function. The relationship between the moments of a function, often also called moments about zero, and its central moments is given by

$$\gamma^{(m)} = \sum_{k=0}^{m} \binom{m}{k} (-1)^k \mu^k r^{(m-k)} \tag{5.14}$$

with *m* as the moment number, *r* as the function moment and *γ* as the central moment [58].

5.3.1 Skewness

The third central moment of a function is called *skewness* and describes the asymmetry of the probability distribution. This means that a symmetric uniform distribution will have a skewness of zero while distributions that have few values much larger than the mean and many values smaller than the mean[1] will have a negative skewness value. Distributions displaying a large number of values around and above the mean will have a positive skewness value.

Skewness is defined as the normalised third central moment of a function and is dimensionless.

$$SK = \frac{\gamma^{(3)}}{\sigma^3} \tag{5.15}$$

[1] In this case, the distribution will have a long “tail” to the left

with σ as the standard deviation.

5.3.2 Kurtosis

The fourth central moment or *kurtosis* contains information about the general shape of the probability distribution of a function. It describes the breadth and height of a distribution compared to a normal distribution that is defined with the same variance.

The kurtosis is defined as

$$KU = \frac{\gamma^{(4)}}{\sigma^4} - 3 \tag{5.16}$$

where the subtraction of 3 is common in order to set the kurtosis coefficient value of normal distributions to zero. A high kurtosis value indicates a broad distribution while a low kurtosis value indicates a highly bounded distribution [58].

5.3.3 Flatness

Another feature that describes the shape of probability distributions is the *flatness*. This is a measure of how flat or narrow and "spiky" a distribution is. It is defined as the ratio of the geometric mean to the arithmetic mean.

$$FL = \frac{\overline{x}_g}{\overline{x}_a} = \frac{\sqrt[N]{\prod_{n=1}^{N} x[n]}}{\frac{1}{N} \cdot \sum_{n=1}^{N} x[n]} \tag{5.17}$$

Since the geometric mean will always be smaller than the arithmetic mean, except for the case that the data set values are all equal (in this case the means are equivalent), a flat distribution will result in a flatness value close to one, while spikier distributions will result in values closer to zero.

5.4 Correlation of Low-Level Features with Onsets

To evaluate the relationship between changes in feature values over time and onset events, the behaviour of 16 different audio signal features over time is correlated with the onsets present in 15 different audio signals.

The evaluated features, which are described in detail in the previous chapters, include the RMS energy (RMS), the zero-crossing rate (ZC), the spectral roll-off factor (RO), the spectral brightness (BN), roughness (RN), irregularity (IR), spectral centroid (CT) and spread (SP) as well as the statistical features flatness (FN), skewness (SK), kurtosis (KU) and the first five MFC coefficients (M1…M5).

A pulse train is created by setting pulses at onset times and windowing those using Hanning windows of 50 milliseconds to account for potential labelling errors or uncertainties. These modified pulse trains are then correlated with the feature changes over time and normalised by the signal energy. The same approach is used to determine the covariance[1] between the modified pulse train and the signal features. With μ and v as the expectation values of *x[36]* and *y[36]* respectively [58], the normalised cross-correlation is given by

$$\tilde{r}_{xy}[m] = \frac{r_{xy}[m]}{\sqrt{\left|x[n]\right|^2 \cdot \left|y[n]\right|^2}} \tag{5.18}$$

with $r_{xy}[m]$ as the cross-correlation while the normalised covariance is given by

$$\tilde{\gamma}_{xy}[m] = \frac{\gamma_{xy}[m]}{\sqrt{\left|x[n]\right|^2 \cdot \left|y[n]\right|^2}} \tag{5.19}$$

with $\gamma_{xy}[m]$ as the covariance.

This leads to correlation and covariance curves describing the relationship between features and onset times.

[1] correlation with the mean removed

An important indicator of the similarity between two signals is the maximum of the cross-correlation or cross-covariance at lag zero[1]. This value is less an indicator of structural similarity[2] than of overall similarity.

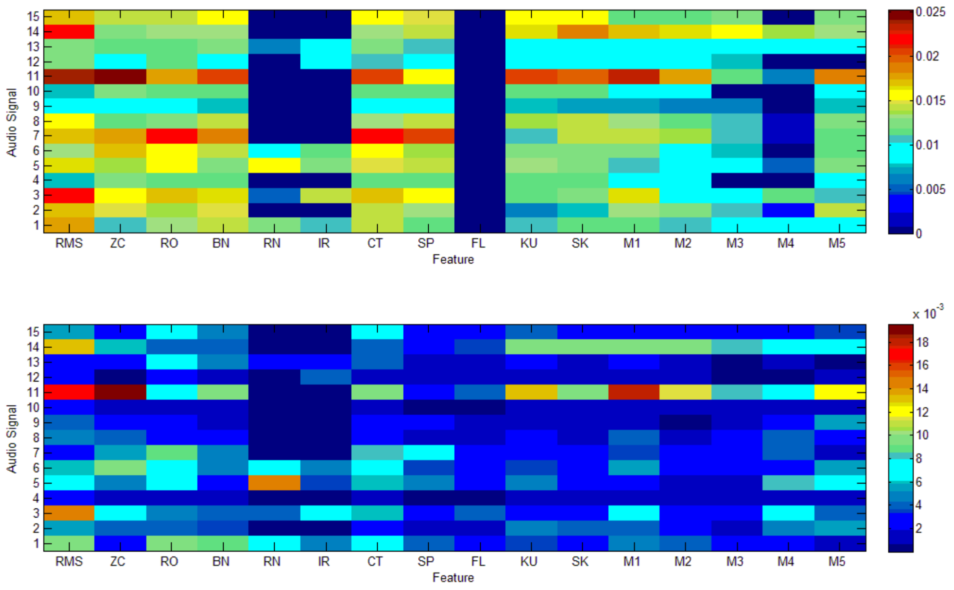


Figure 5.5 Correlation (top) and covariance (bottom) maxima between features and onset times for 15 different audio signals

Figure 5.5 shows the correlation and covariance between the analysed features and the onset times present in different audio signals. It is obvious that some features seem to be more correlated to onsets and are therefore more useful for onset detection purposes – for example, the RMS energy, the zero-crossing rate and the first MFC coefficient seem to be more correlated with onset times than features such as flatness, roughness or irregularity. In order to further investigate this assumption, the mean correlation and covariance between features and onset times is calculated over the number of used audio signals.

[1] when the signals are compared directly without time delay

[2] for example, the correlation of two pulse trains shows a number of high peaks at lags corresponding to the pulse time spans with very low values in between

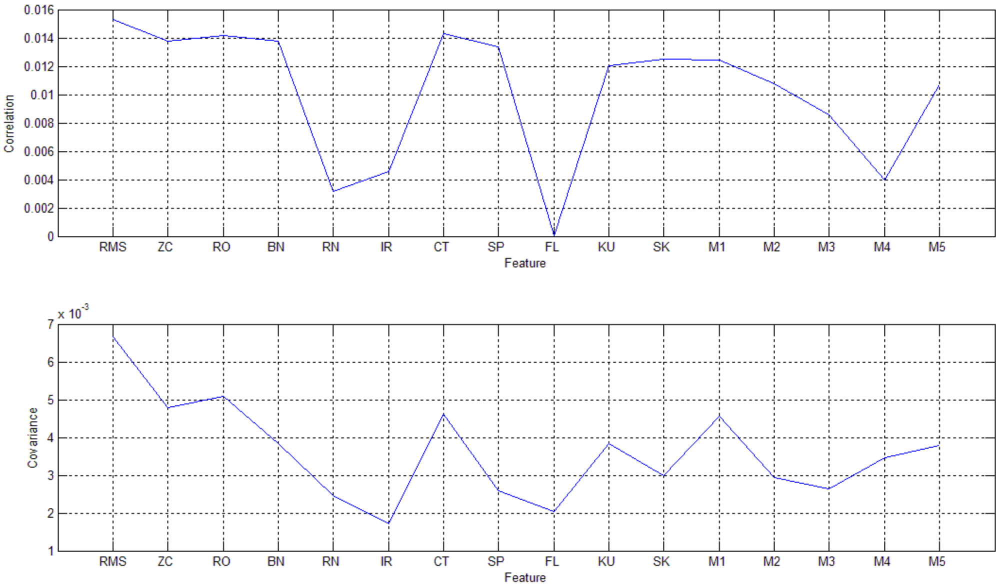


Figure 5.6 Mean correlation (top) and mean covariance (bottom) maxima between features and onset times over 15 different audio signals

As mentioned above and shown in Figure 5.6, spectral signal characteristics like the roughness or irregularity measure as well as statistical features like the flatness seem unsuitable for the task of onset detection, while features that are related to the energy content or the tonal quality of the signal such as the RMS energy, zero-crossing rate, spectral roll-off, spectral centroid or the first MFC coefficients will be better suited to the task. It is also interesting to note that features concerning the tonal quality of the sound are correlated more closely to the onset structure of the signal than features concerning the spectral or harmonic shape of the signal such as the roughness, irregularity or spectral spread measures.

6 Subjective Similarity Evaluation

A crucial task in any implementation of concatenative music synthesis is to find the audio segments that closely match the composer's vision of the overall sound. This is far from easy since the composer's concept of a sound may not match the actual physical parameters of the sound. This necessitates the characterisation of sound not by abstract physical parameters but by perceptually meaningful feature parameters extracted from the audio data. To gather more information about how listeners evaluate subjective similarity between different audio signals and to try to find a feature or a combination of features that best describes this perceived similarity, a listening test was carried out. From this test, the perception space that listeners use to evaluate similarities is analysed by using *Multi-Dimensional Scaling* (MDS), leading to a graphical representation of the perceived similarity distances. The results of the listening test are evaluated using the statistical software package SPSS. The design and the realisation of the listening test are described in detail in the next chapter while the evaluation and interpretation of the results are detailed in chapter 6.2.

6.1 Listening Test

For concatenative music synthesis implementations, a very important part of the algorithm is concerned with the selection of musical fragments which are consecutively concatenated to produce a new signal. The difficulty herein is that as a rule the composer does not work with physical parameters of the sound but has a concept of how the music piece should sound. This complicates the task of finding the database sound fragments that match the user's vision of the overall sound. Therefore, a psychoacoustic description of the stored data is desirable. By describing sound fragments with perceptually relevant parameters, the database can also be restructured, clustering perceptually similar segments. This leads to a more intuitive way of database organisation and has the added advantage of speeding up the search process. In order to learn more about how similarity between signals is perceived by human listeners, a listening test was carried out which is described in detail in this chapter. Another purpose of this test was to determine the most meaningful way of database organisation. Three organisation methods were evaluated by the subjects: the first uses an approach where the audio elements are sorted into clusters (see section 3.1.1) according to their length and into sub-clusters according to their spectral centroid, the second method sorts the elements according to length and chroma value and the third used length and pitch of the signal to determine clusters and sub-clusters, respectively.

The listening test was designed as a simple A-B comparison test, meaning the audio samples were played pair-wise. The task of the test subject was to define the subjective similarity between the two samples with a set of discrete values.

A total of 21 audio samples was used in the test. The samples were taken from three databases containing the same audio material but organised in the different ways mentioned above (according to length/centroid, length/chroma value and length/pitch of the segment), which means 7 samples from each database.

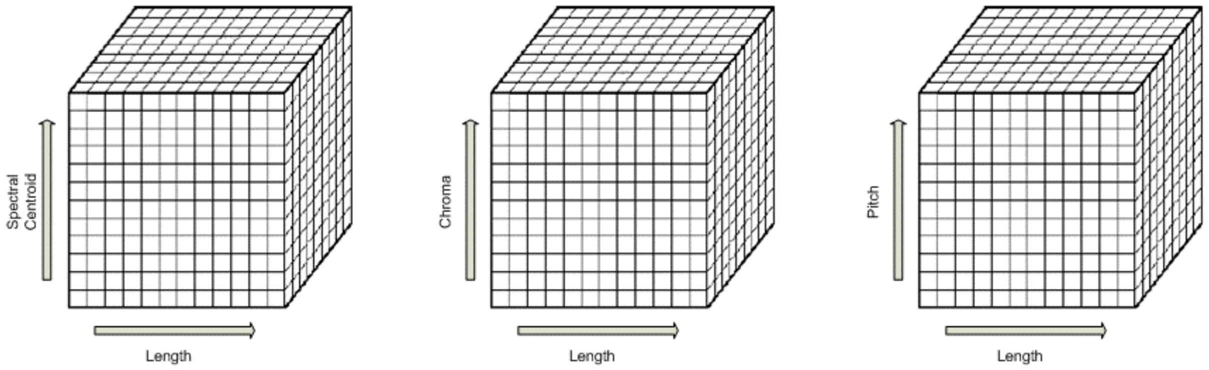


Figure 6.1 Evaluated database organisation methods

The 7 samples from the respective databases were selected according to the following principle: for a meaningful comparison, the samples had to be sufficiently long so that the subjects could concentrate on the perceptual attributes but not too long so as not to confuse the subjects. A length of about 0.5 seconds was found to be suitable. Another constraint was that the samples should be of approximately equal length to ensure convenient comparison, which limited the database area from which samples could be chosen. In the end, the samples were taken from one cluster or two neighbouring clusters at the end of the length spectrum.

From the selected clusters, three samples were taken from one of the first few sub-clusters, one from a sub-cluster near the middle and another three samples from one of the last sub-clusters.

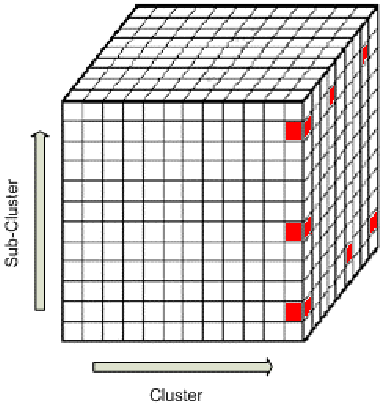


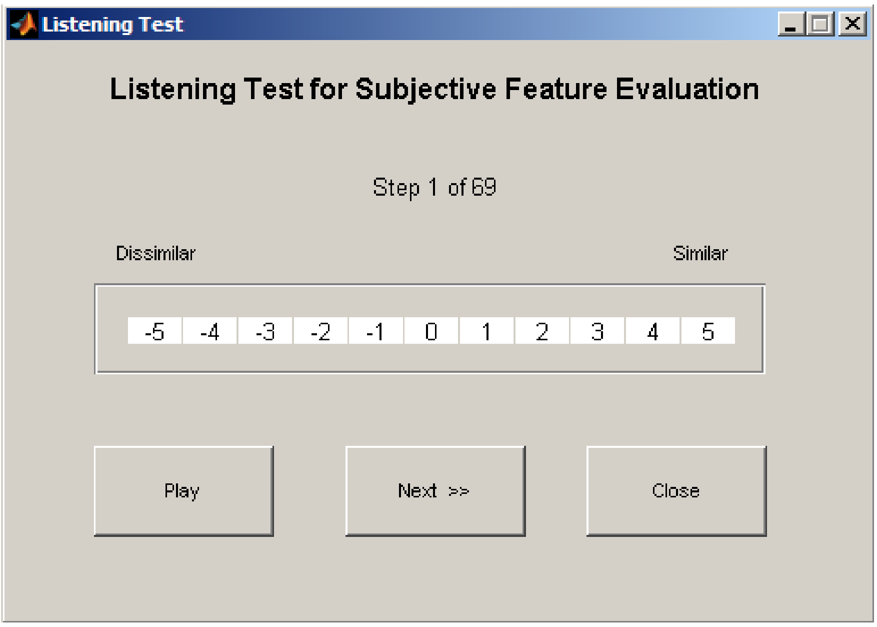
Figure 6.2 Sample selection

In order to monitor the reliability of the answers of the test subjects, one sample pair was repeated twice at arbitrary points for every database organisation method. In order to evaluate all possible combinations between the samples, the number of comparisons is $n \cdot \left[(n-1)/2\right]$, assuming that the dissimilarity is symmetrical[1]. This led to a total of 69 sample pairs the subjects had to evaluate (seven samples were taken from each of the three databases, which leads to 3 times 21 sample pairs, with 2 control pairs for every database).

The test took place in a sound-proof studio at the Institute of Electronic Music and Acoustics in Graz using a PC as input and display device. The samples were played back using a RME Hammerfall MultiFace audio interface and AKG K-240 earphones. The subjects - 11 in all - were for the most part senior students of the audio engineering curriculum and can therefore be classified as experienced listeners after completing a number of courses on ear training and music knowledge.

A graphical interface was implemented in Matlab where the subjects could listen as often as they wished to the particular sample pair and where they charted their similarity impression on a defined value scale.

[1] this means that the order in which the samples are played back is of no relevance to the subjective perception



Listening Test

Listening Test for Subjective Feature Evaluation

Step 1 of 69

Figure 6.3 Listening test interface

The subjects were asked to rate the similarity of the sample pair on a scale of -5 (dissimilar) up to +5 (similar). The samples were played with a one-second interval between them and could be repeated as often as the subject wished. After listening to a sample pair and rating their similarity, the subject could go on to the next sample pair. He or she could not, however, go back to change a rating already given. The results were stored for later evaluation.

6.2 Evaluation

The results obtained from the listening test were evaluated using SPSS after hand-checking the reliability of the results in Matlab. SPSS is a program that is widely used for statistic transformations and data analysis. This chapter will describe the results of analysing the listening test data using the Multi-Dimensional Scaling (MDS) technique, which attempts to visualise the correlation between subjective parameters. A distinction is made between *metric MDS* methods and *non-metric MDS* methods. The difference between these two approaches lies in the assumptions about the relationships of evaluated data and both are explained in this chapter. For the evaluation of the listening test that was carried out in the course of this thesis, the non-metric version was used for reasons explained later in the chapter.

6.2.1 Multi-Dimensional Scaling

An important psychoacoustic research field is the evaluation of audio signal attributes in relation to the human subjective perception of the signal. A perceptual space is stipulated where the signal is placed by a person according to its perceived attributes. This space is assumed to have as many dimensions as the number of weights the person uses to form an opinion about the signal [59].

In order to visualise this multi-dimensional perceptual space, Multi-Dimensional Scaling (MDS) is used to graphically model the space. The idea of this approach is to find the parameters that subjects use to form an evaluation of a given set of stimuli, in the process inferring the dimensionality of the used perceptual space [17]. The data points calculated by the MDS analysis tool are represented as points in an *n*-dimensional space with the distances between the points corresponding to the similarity between the evaluated stimuli.

MDS operates on data derived from the difference measure or distances called *proximities* amongst a data set. A suitable representation of all respective proximities is the *proximity matrix*, a symmetric matrix with zero along the main diagonal. To arrive at the proximities of a given data set, either direct or indirect methods can be used. Direct methods let the subject assign a subjective similarity value to a pair of stimuli or sort the stimuli according to their similarity, while indirect methods derive the proximities from measures other than the similarity, for example from so-called “confusion data”. The advantage of using a direct rating method is that the resulting proximity data can be analysed without having to take additional processing steps, while the disadvantage lies in the fact that the number of sample pairs grows rapidly when the number of data objects increases. Indirect rating methods can be implemented more efficiently, but additional measurements are required to extract the proximity data.

Metric MDS (which is also referred to as *classical MDS* [17]) assumes that the distances between the data values that are analysed can be described on a metric scale. This holds true, for example, when analysing the distances between cities or the height of test subjects. When using metric MDS, the distances between the data objects are preserved as broadly as possible in the graphical representation of the MDS space. This simplifies the calculation since there are no iterations to be performed. The solution is found by using linear algebra.

On the other hand, the assumption that the distances between data values can be expressed in metric terms probably does not apply to data derived from human subjective similarity evaluations. For this type of problem, a variant of MDS called non-metric MDS is suitable. Here, it is assumed that the order of the proximities and not the proximities themselves is meaningful. This means switching from an interval scale to an ordinal scale. In the graphical representation, only the order of the distances between data points is reflected while the distances themselves are not relevant.

The non-metric MDS representation is extracted from data by monotonic transformation of the object proximities. The points in the *n*-dimensional space are placed in such a way as to minimise the squared deviations between the scaled proximities and the distances between the points themselves. Mathematically, this requirement is described by the *stress factor*

$$s = \sqrt{\frac{\sum\left(f(\vec{p})-\vec{d}\right)^2}{\sum \vec{d}^2}} \tag{6.1}$$

where *s* is the stress factor that has to be minimised, $\vec{p}$ is the proximity matrix, $f(\vec{p})$ stands for the monotonic transformation of $\vec{p}$ and $\vec{d}$ is the vector containing the distances between the points in the MDS representation. While there are different definitions of the stress factor available, SPSS uses the version defined by Kruskal [17] described above. The stress decreases when the number of dimensions is increased. Stress above 0.2 is considered an indication of a poorly-fitting solution while any number below 0.025 is judged to provide an excellent fit.

6.2.2 Evaluation of obtained results

The previous section describes the method used to evaluate the subjective similarity ratings the subjects assigned to the respective sample pairs. This chapter presents the results of this evaluation and interprets the findings of the listening test. In a first step, the subjective ratings are transformed and normalised to ensure comparability between the respective results. The next step consists of checking the reliability of the subjects' answers by looking at the rating of the control pairs repeated during the test. Then, the corrected results from the reliable test subjects are evaluated using MDS and represented graphically in a two-dimensional space.

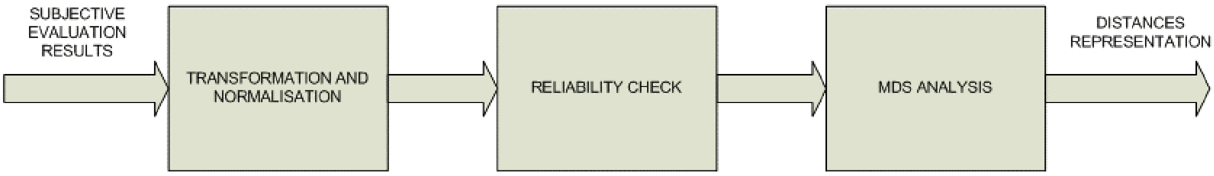

SUBJECTIVE EVALUATION RESULTS
TRANSFORMATION AND NORMALISATION
RELIABILITY CHECK
MDS ANALYSIS
DISTANCES REPRESENTATION

Figure 6.4 Listening test results evaluation

As said previously, the subjects were asked to rate the similarity or dissimilarity of two audio signals on a scale ranging from -5 (dissimilar) to +5 (similar). However, as MDS operates on dissimilarity values only, the first step was to transform the assigned values so that small values signify similar objects, while large values correspond to a high degree of dissimilarity. This was done by using a scale between 0 (similar) and 10 (dissimilar) using the following operation:

$$v_n[n] = -v[n] + 5 \qquad (6.2)$$

with *v[n]* as the original value and $v_n[n]$ as the transformed value. The results were further processed to ensure comparability between the subjects' answers by subtracting the mean value of answers and dividing the result by the variance of the answers. This led to a new similarity scale that ranged from -1 to +1. To ensure correct analysis of the data by the SPSS tool, the values were again shifted by one, leading to a value representation between zero (similar) and 2 (dissimilar).

$$v_c[n] = \frac{v_n[n] - \overline{v}_n}{\sigma_v^2} + 1 \qquad (6.3)$$

with $v_c[n]$ as the final similarity value.

To check the reliability of the subjects' answers, for each of the three databases, one of the sample pairs was randomly chosen as control pair. This pair was repeated twice during the course of the listening test. By comparing the values assigned to the similarity between the two samples, the constancy of a subject's rating criteria was evaluated. To be considered reliable, the subject's answers to the control pairs should remain roughly constant with each repetition of the sample pair. A jump of more than 2 scale values indicated that the answers of a particular subject were not reliable over the whole data set.

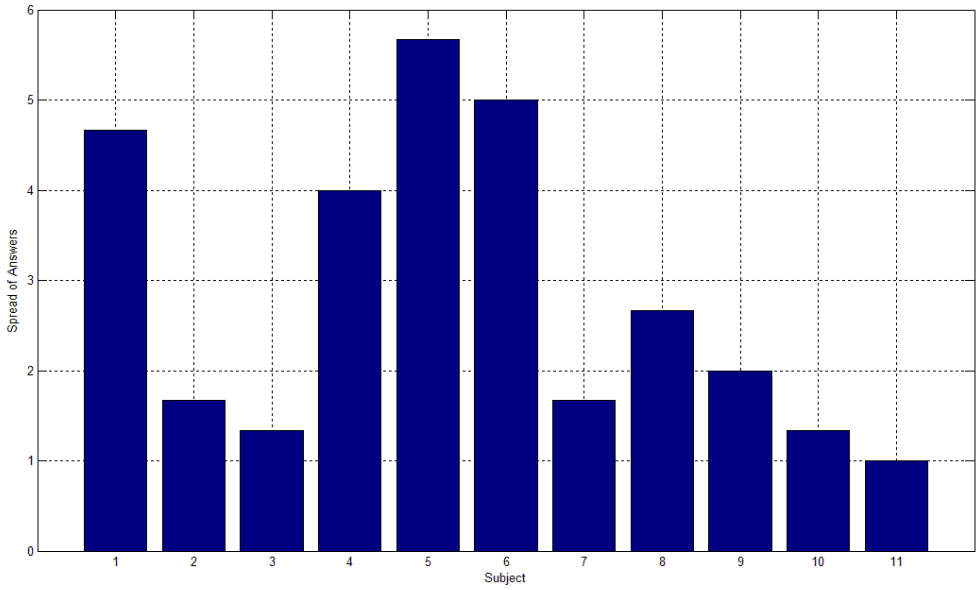


Figure 6.5 Spread of values assigned to the control pairs by listening test subjects

Figure 6.5 shows that subjects 1, 4, 5, 6 and 8 do not meet the above-mentioned criterion for reliability. However, detailed analysis showed that the spread of the answers of subject 8 was skewed by one value which was significantly higher than the others at the beginning of the test. This was attributed to the fact that subjective evaluation principles slightly change after hearing a number of samples. Since the later rating values assigned by subject 8 to the control pairs were very similar, the subject was added to the group of reliable subjects. This meant that after the exclusion of subjects 1, 4, 5 and 6 there were still 7 subjects left that showed reliable answering behaviour over the course of the listening test. Only the results of these seven reliable subjects were used in the further evaluation process.

The next task was to create proximity matrices from the similarity data obtained by the listening test. This was done by creating proximity matrices $\vec{P}_b$ containing the mean of the subjective similarities of the sample pairs over the 7 remaining subjects for every database, where b is the database number. The matrix is symmetric, contains zeros along the main diagonal and has the size 7x7. The matrix elements $p_{i,j}$ contain the subjective similarity between the sample $i = 1,2,...,7$ and the sample $j = 1,2,...,7$.

The proximity matrices were then entered into the SPSS interface and analysed using the integrated MDS algorithm ALSCAL developed by Forrest Young [60]. This led to a two-dimensional graphical representation of the audio sample relationships (Figure 6.6). The stress factors that were obtained iteratively amounted to the following values, indicating a very good quality of fit:

Database Organisation		Stress
Length / Spectral Centroid		0,00427
Length / Pitch		0,00482
Length / Chroma		0,00506

Table 6.1 Stress factors for 2-dimensional MDS analysis for the three database organisation methods

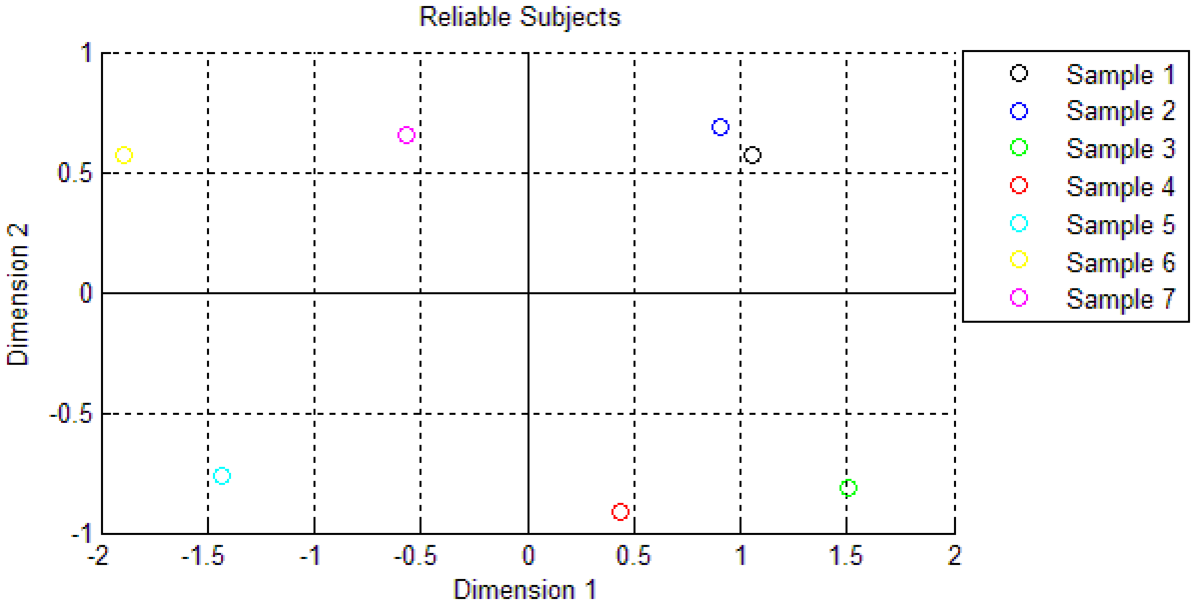


Figure 6.6 MDS representation of sample distances for reliable listening test subjects (database organised by length and spectral centroid)

For comparison purposes, the MDS representation obtained by evaluating all subjects' answers is shown in Figure 6.7. While the position of most samples does not change significantly, the differences in the positions of others are evident. This reinforces the assumption that when the similarity or dissimilarity between audio signals is not obvious, the judgment of subjects that have well-defined evaluation criteria is preferable to the judgment of more people who are not as thorough in their evaluation.

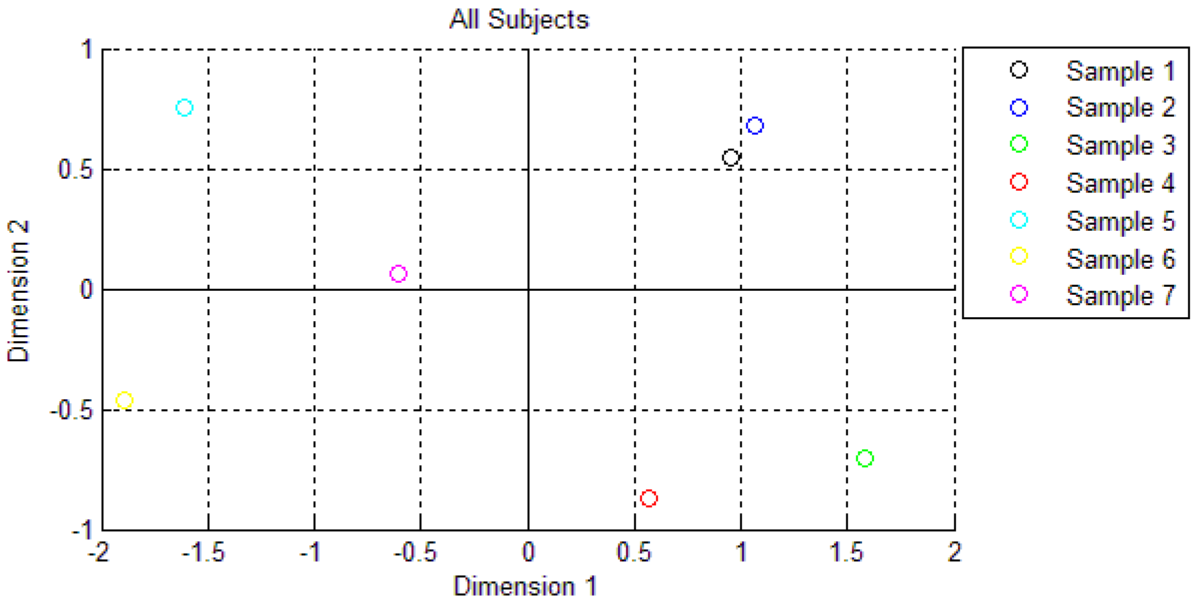


Figure 6.7 MDS representation of sample distances for all listening test subjects (database organised by length and spectral centroid)

In order to better describe the relationships between the audio samples in the graphical MDS representation, meaningful descriptions for the two axes have to be found.

The most important terms in which any audio signal can be described are *pitch*, *loudness*, *duration* and an aspect that is not well defined, *timbre*, which is an aggregate of a number of spectral and temporal features [34]. If we assume that the MDS representation in some way models the space of attributes that listeners use to judge the similarity of signals, we can arrive at a useful denomination of the MDS axes. Descriptions based on qualitative signal aspects (for example "brightness" or "attack") have been proposed as well as descriptions based on quantitative attributes ("Log-Attack-Time", "Harmonic Spectral Centroid) [34]. The correlation between similarity maps and subjective sound characterisations has also been investigated in [1] by using attribute pairs like "thin-thick" or "sharp-dull".

After subjective evaluation of the differences between the samples in the respective dimensions, the first dimension seemed best explained by a measure describing the change in the dynamic of the signal[1], similar to the ADSR (Attack-Sustain-Decay-Release) description. This dynamic change is best visible in the temporal envelope of the signal and is loosely related to the *Log-Attack Time* (the logarithm of the duration between the signal start and the signal maximum) measure used as dimension description in [34] which can be viewed as the "attack" phase. This assumption is supported by Figure 6.8, which shows the temporal envelopes of audio samples 3 and 6, which are the samples with the largest distance between them in the MDS representation (Figure 6.6). While sample 3 shows a significant increase in magnitude in the first part of the signal with a slow decay afterwards, there is no sharp change in the magnitude of sample 6. The dynamic changes in the other signals conform to this assumption in a similar way. So the first dimension of the MDS depiction can be defined as representing the change in the temporal envelope of the signal, ranging from high dynamic change (as is the case with sample 3) to little or no dynamic change (sample 6). The positions of the other samples are correlated with their time envelopes in a similar way.

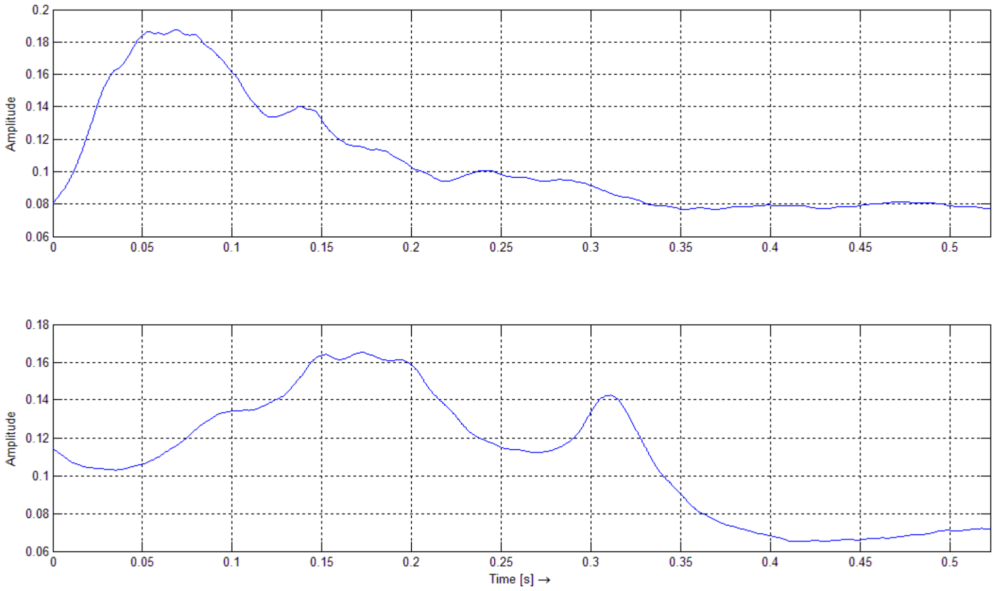


Figure 6.8 Temporal envelopes of samples 3 (top) and 6 (bottom) taken from the database organised by length and spectral centroid

The same procedure as described above is followed in determining the second dimension of the MDS distance representation. After listening closely to the sample pairs, a good description of the second dimension seemed to be the "fullness" or "breadth" of the signal spectrum. This subjective description can be set in relation to objective signal parameters like

[1] Not to be confused with the percussiveness of a signal

the spectral spread[1] or measures describing the harmonic fullness, i.e. the presence of partial tones in the spectrum. A similar description is also found in literature [34], where dimensions are labelled as *Harmonic Spectral Centroid* or *Harmonic Spectral Spread*. A comparison between the distances between points in the MDS space and the shape of the spectral envelope of the matching signals validates the dimension description mentioned above. The spectral envelopes of the signals whose data points are farthest from each other in respect to the second MDS dimension are pictured in Figure 6.9. The first spectral envelope (sample 2) is very narrow compared to the second one (sample 4). This suggests that the second dimension of the MDS space can be defined as describing the breadth and sharpness of flanks of the spectral envelope of the analysed signals, ranging from narrow-band spectra with sharply falling flanks to broad-band spectra. The other sample signal spectral envelopes conform to this assumption in a similar way.

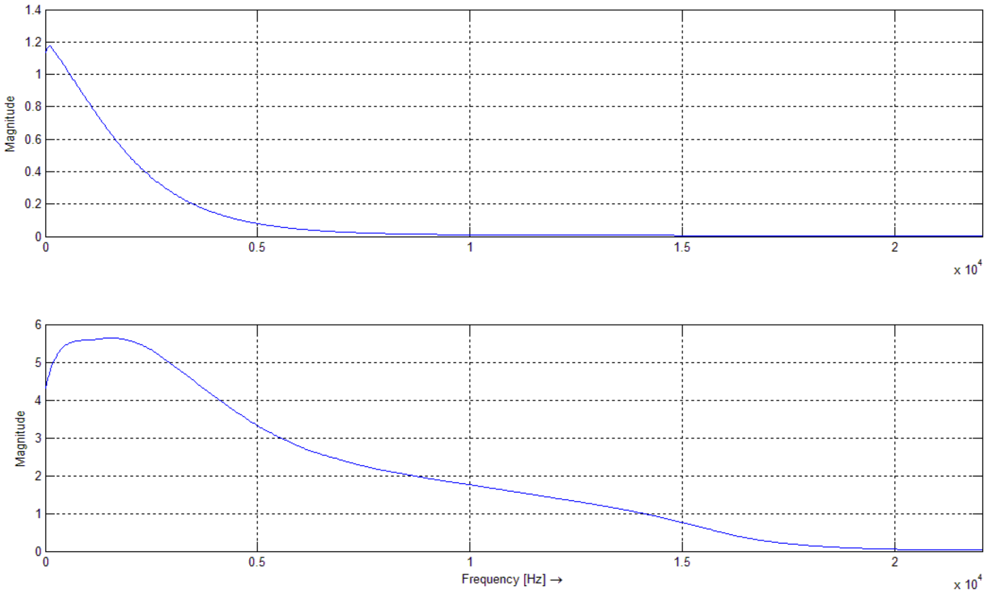


Figure 6.9 Spectral envelopes of samples 2 (top) and 4 (bottom) taken from the database organised by length and spectral centroid

The MDS representation is now characterised by the distribution of sample data points in a two-dimensional space that is defined by spectral and temporal attributes of the sample signals. An example is shown in Figure 6.10, which shows the same thing as Figure 6.6, with

[1] for detailed description of the spectral spread, see chapter 5.2

the horizontal axis now labelled and describing the manner and height of the change in the temporal structure of the signal (*dynamic change*) and the new vertical axis describing the signal change in the spectral domain (*spectral change*).

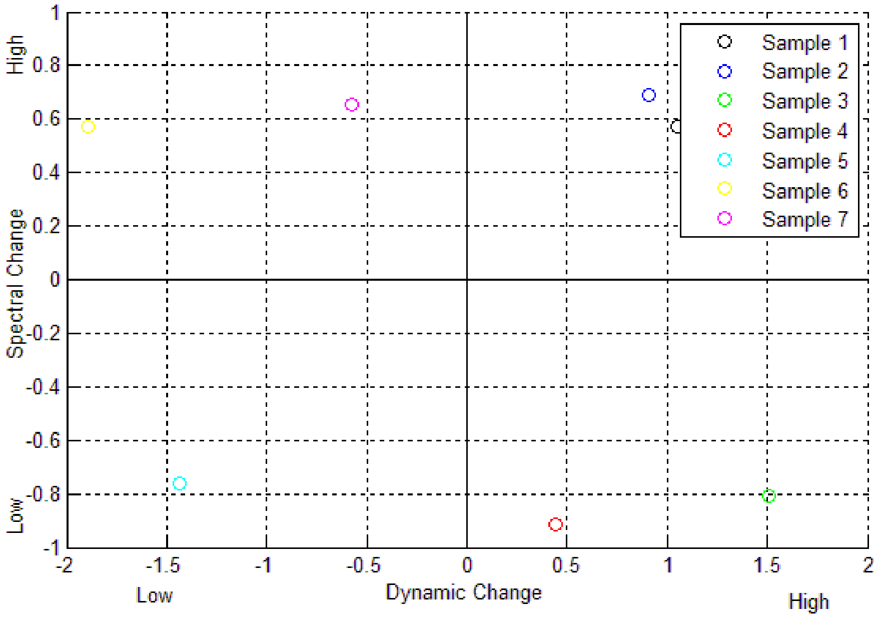


Figure 6.10 MDS representation of audio sample attributes and distances (database organised by length and pitch)

Since the perception of spectral characteristics seems to be of great influence when evaluating the similarity of sounds, it was decided to organise the synthesis database by length and pitch. While the chroma value of a sound can be likened to the pitch value, the fact that there are 12 fixed chroma classes makes a distribution of samples across the database that is close to uniform very unlikely.

From the data obtained by MDS analysis of the listening test results, a meaningful distance measure describing the similarity of audio samples is obtained. This is done by combining a number of low-level features into a feature vector. To determine the features best suited to this task and their optimal weighting, linear regression is used.

To this end, over 40 different combinations of spectral. temporal and statistical features are evaluated. A matrix representation of an equation system is used to find the least-squares solution to the problem of finding the feature combination that yields the minimal residual between estimated and actual distances.

$$\vec{y} = \vec{x} \cdot \vec{c} + \vec{\varepsilon} \tag{6.4}$$

where $\vec{y}$ is a vector containing the distances between sample points obtained by the MDS analysis, $\vec{x}$ is a matrix composed of elements $x_{i,j}$ that describe the squared difference of the *i*-th low-level feature for the *j*-th audio sample pair, $\vec{c}$ stands for the vector of unknown weighting coefficients and $\vec{\varepsilon}$ for the residual between the estimated and actual distances. The residual describing the goodness of fit of the estimated coefficients to the model is calculated as follows:

$$\vec{\varepsilon} = \vec{y} - \hat{\vec{y}} \tag{6.5}$$

with $\hat{\vec{y}} = \vec{x} \cdot \vec{c}$, i.e. the estimated solution. This calculation is executed for a number of different low-level feature combinations. By checking the size of the residual for every solution, the combination of features that comes closest to describing the sample point distances with a minimal error is found.

Feature Combination Name	No. of Features	Features	Residual
All	15	RMS, ZC, PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP, SK, KU, FL	0,006264923
Temporal / Spectral	12	RMS, ZC, PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP	0,045699485
Spectral / Stats	13	PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP, SK, KU, FL	0,092906633
Spectral	10	PT, MFCC1, MFCC2, MFCC3, RO, CT, BN, RN, IR, SP	0,11985442
MFCCs / Energy / Stats	11	PT, MFCC1, MFCC2, *MFCC3*, RO, CT, BN, IR, SP, SK, KU, FL	0,166007352
Tonality / Energy / Stats	13	PT, MFCC1, ZC, RO, CT, RN, RMS, BN, IR, SP, SK, KU, FL	0,178389407
MFCC1 / Centroid / Zerocross / Stats	6	MFCC1, CT, ZC, SK, KU, FL	0,187361823
MFCC1 / Pitch / Zerocross / Stats	***6***	***MFCC1, PT, ZC, SK, KU, FL***	***0,189502726***
MFCCs / Energy	8	RMS, MFCC1, MFCC2, MFCC3, RO, CT, BN, SP	0,22390577
Pitch	6	PT, MFCC1, RO, CT, BN, SP	0,243138767
Tonality	5	ZC, MFCC1, IR, SP, RN	0,264080275
MFCC1 / Pitch / Zerocross	3	MFCC1, PT, ZC	0,279085262
MFCC1 / Centroid / Zerocross	3	MFCC1, CT, ZC	0,298354604
Temporal / Stats	5	RMS, ZC, SK, KU, FL	0,30137455
Pitch / Zerocross / Stats	5	PT, ZC, SK, KU, FL	0,304335861
Pitch / RMS / Stats	5	PT, RMS, SK, KU, FL	0,308709821
MFCC1 / Pitch / RMS / Stats	6	PT, RMS, MFCC1, SK, KU, FL	0,322857438
Temporal / Spectral Energy	5	RMS, SP, RO, CT, BN	0,378814348
MFCC1 / Centroid / RMS / Stats	6	MFCC1, CT, RMS, SK, KU, FL	0,401318649
Tonality / Energy 2	4	IR, RN, RMS, BN	0,45468815
MFCCs	3	MFCC1, MFCC2, MFCC3	0,478421498
Temporal	2	RMS, ZC	0,491990676
Pitch / Zerocross	2	PT, ZC	0,497078406
MFCC1 / RMS / Stats	5	MFCC1, RMS, SK, KU, FL	0,500295704
MFCC1 / Pitch / RMS	3	MFCC1, PT, RMS	0,518645457
Pitch / RMS	2	PT, RMS	0,526578748
MFCC1 / Stats	4	MFCC1, SK, KU, FL	0,550317002
RMS / Harmonicity	3	RMS, BN, RN	0,578099999
MFCC1 / Centroid / RMS	3	MFCC1, CT, RMS	0,680020088
Stats	3	SK, KU, FL	0,857694604
Tonality / Energy 1	2	RN, RMS	0,869715207
MFCC1	1	MFCC1	0,891612578

Table 6.2 Feature vectors and the respective mean residual between estimated and actual distances

Table 6.2 shows the feature vectors that result in low mean residual errors between the estimated and the actual distance vectors. It shows that the combination of all used features leads to the smallest least-squares error, which is to be expected. The following vectors are composed of a combination of spectral low-level features with temporal features and statistical features. Using only spectral features works quite well and a combination of temporal and statistical features also leads to acceptable results. The meaning of the feature abbreviations is listed in Table 6.3. For a detailed description of low-level features used, see section 5.

RMS	Root Mean Square Energy		BN	Brightness
ZC	Zero-Crossing Rate		RN	Roughness
PT	Pitch		IR	Irregularity
MFCC1	First MFC Coefficient		SP	Spectral Spread
MFCC2	Second MFC Coefficient		SK	Skewness
MFCC3	Third MFC Coefficient		KU	Kurtosis
RO	Spectral Roll-Off		FL	Flatness
CT	Spectral Centroid			

Table 6.3 Used low-level features

As Table 6.2 shows, there is a trade-off between how well a combination of features represents the characteristics of a signal and the number of features needed to do so, which is directly related to the computation time necessary. The feature highlighted in Table 6.2 consisting of six elements – pitch, first MFC coefficient, zero-crossing rate and the statistical features skewness, kurtosis and flatness – was found to be suitable for designing a distance measure that represents the similarity or dissimilarity of audio signals. The residual between the actual distance and the distance achieved by linear regression is sufficiently small to suggest reasonable results while the number of features is sufficiently low to enable short calculation times. Another advantage of this feature set is that the calculation of the signal pitch is already implemented in the database organisation, which is arranged according to signal length and pitch.

7 Discussion of Results and Perspective

In this thesis, a software tool for concatenative music synthesis was implemented. To this end, a graphical interface was created to give the user control over different aspects of the synthesis algorithm.

The algorithm uses a pre-defined database of audio signals for the re-synthesis of target songs. These target songs are analysed in regard to their beat structure so as to achieve musically meaningful segmentation results and the resulting segments are replaced by database segments using a similarity distance measure.

While evaluation of the re-synthesis results “by ear” show that the overall algorithm is far from perfect, it can be viewed as an encouraging step into the direction of musically and artistically valid algorithmic sound synthesis. It could be developed as a powerful tool for electronic music composition, where similar approaches have been introduced in recent years (for example by [19], [30]). It could eliminate the time-consuming and tiresome process of searching for sound sample material that corresponds to the artist’s vision – he could enter an exemplary sample and have the algorithm search for a matching sound segment.

This approach could also be integrated into process-based music and performances, allowing the artist to focus on sound combination and processing issues while the sound material is chosen algorithmically.

Two topics that have been discussed at great length in this thesis, onset detection and beat tracking, could also be of value to different applications in audio or speech processing. It would be interesting to evaluate the detection methods based on MFC coefficients and on modulation spectra in the context of speaker identification and segmentation because both MFCCs and modulation spectra are well known in the speech processing field ([32], [53]).

However, there are still some aspects of the re-synthesis algorithm that could be improved. The next sub-sections will take a closer look at some key issues discussed in this thesis and will provide an insight into how well the implemented solutions work as well as how they can be improved.

7.1 Interface Platform and Implementation

The ConCat Music Synthesis interface is implemented in Matlab, a programming environment that is widely used in the academic as well as the industrial field, which is one of the reasons why the environment was chosen. Many people in the audio processing and signal processing fields are familiar with Matlab and it is used in similar sound synthesis applications [19]. Another advantage of Matlab is that many functions are available that would otherwise have to have been implemented “by hand”, which greatly reduces the programming time needed to create algorithms.

However, Matlab as a high-level programming language and environment is very slow compared to programming languages that are more hardware-oriented. Besides, in order for the interface to run, a working and up-to-date version (some issues concerning compatibility between different Matlab versions are known to occur) of Matlab is required.

These drawbacks could be eliminated by porting the re-synthesis algorithm to a language such as C or C#, which would most likely lead to significant reductions in computation time as well as widen the circle of possible users of the synthesis algorithm by eliminating the need for a running Matlab copy on the user’s system.

Another drawback of this system is that it works exclusively with uncompressed .wav audio files. In order to make the system more accessible, it will need to support audio files in different formats.

7.2 Database organisation

As described in chapter 3.1.1, the database where segments are stored for later re-synthesis is designed as a three-dimensional cubic structure where segments are placed according to their length and their pitch value.

While some re-synthesis results may not be optimal when the segments that are to be replaced have length or pitch values close to the boundaries between the cluster or sub-cluster elements, this organisation greatly reduces computation time and ensures optimal usage of disk space, which is an issue with large sound databases.

For example, a rather small database of 40422 different sound segments would, using the feature vector presented in section 6.2.2, have to compare two six-dimensional vectors 40422 times. In contrast, using the above-mentioned organisation, only the segments in a particular cluster and sub-cluster would have to be evaluated. In this thesis, the size of sub-clusters was limited to 25 elements, which means the computation time needed to find the most similar frame that matches the target song frame is reduced by approximately the factor 10^3.

The database is also organised in a way so as to optimally distribute the sound fragments between the clusters and sub-clusters. Again, this leads to a higher number of sub-optimal fragment match results (the more boundaries, the more yes/no-decisions are necessary and therefore errors are made) but greatly reduces the time needed for search and comparison operations.

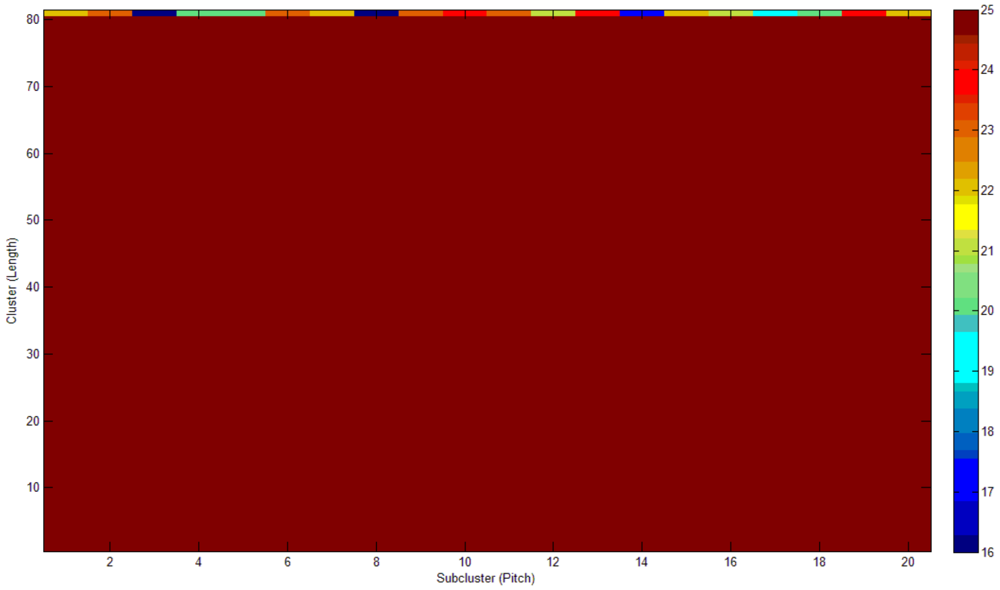


Figure 7.1 Distribution of sound fragments among clusters and sub-clusters

As Figure 7.1 shows, the database sound segments are evenly distributed among clusters and sub-clusters – only in the last cluster (cluster no. 81) the elements are not uniformly distributed.

To further improve the database organisation, some form of compression could be used where length and pitch regions that are more frequent than others are quantised using smaller steps. For example, a probability distribution could be approximated from the length and pitch values and the quantisation step-size[1] could be modelled after the distribution, which could lead to improved re-synthesis results at the expense of computation time.

7.3 Beat Tracking

Beat tracking and onset detection are a widely researched ([l], [2], [3], [6], [7], [9], [11], [27], [36], [42]) and discussed topic. As chapter 4.3 or [3] show, there are some onset detection approaches that work well in specific contexts and for specific signal types, however, a method that yields satisfying results across a wide number of musical genres has not been found yet. Also, complex polyphonic music as well as signals with a prevalence of “soft” attacks lead to a high number of false detections regardless of the detection method used. There is also an inevitable trade-off between detection accuracy and computation time – while there have been some approaches that claim real-time ability ([2], [7], [50]), a fast but still reliable beat tracking method has yet to be determined.

One way to achieve reliable onset detection results over a wide span of musical signals could be the usage of a “modular” approach, where the audio signal is first analysed in regard to certain properties and characteristics and the onset detection method is selected based on these properties. Such properties can include, for example, the overall harmonic structure, the percussiveness, the complexity or the genre of the audio signal. There have been attempts to classify the genre of musical signals [15], which can be used to determine the suitable onset detection method for audio signals – for example, rock songs usually display significant energy changes due to their percussive nature, which would suggest an energy-based onset detection method, while soul music will have a number of “soft” onsets suggesting the use of a phase-based detection method (see also chapter 4.1).

The inter-onset interval beat tracking system presented in this thesis relies on a quite simple beat structure model. The beat tracker works well in situations where there is a clear, stable beat structure present in the signal. It does not account for tempo changes or more complex rhythmic patterns. A possible approach to this problem could be to use a number of different metric levels ([42]).

Another issue in beat tracking problems is the amount of musical or psycho-acoustic knowledge used to design the tracking algorithm. Psycho-acoustic hearing models as well as rudimentary musical knowledge have been used in onset detection applications ([9], [42]). While the usage of assumptions based on musical knowledge (such as the location of certain beats in certain rhythmic structures) will improve results when dealing with certain types of music (for example, pop/rock songs), this approach will also lead to an increased number of false detection when the audio signal does not conform to the assumed rhythmic model.

[1] i.e. the location of the cluster and sub-cluster boundaries

7.4 Features and similarity

In this thesis, a number of temporal, spectral and statistical features are evaluated in regard to their relationship with the onset structure of audio signals (chapter 5.4) and with the perceived similarity of audio signals (chapter 6.2.2).

Especially in the context of database management and audio signal description[1], features have become an important issue in audio applications. Research topics in this field include instrument sound description [34], audio classification ([61], [62]) and the correlation between features [13].

This thesis presents a six-dimensional feature vector using temporal, spectral and statistical features to calculate the distance (which in this case corresponds to the dissimilarity) between audio signals. The features used in this vector stem from a listening test concerning subjective similarity between audio signals, implying that human perception is a crucial issue in audio similarity applications. Future tasks will include deeper research into the correlation between human perception and signal similarity. The relationship between subjective attributes such as “brightness”, “sharpness” or “compactness” [1] and objective signal parameters such as “attack time” or “spectral deviation” [34] has not yet been fully explained. The main issue will be finding the smallest possible feature set that best describes the character of an audio signal. Although this thesis has shown a possible solution for this problem, there is still room for improvement in performance.

7.5 Conclusion

Listening to the re-synthesis results shows that the principal structure of the target song is reproduced quite well, while melodic and rhythmic details and finer structures are not reproduced as well. While a larger database than the one used for evaluation purposes in this thesis will lead to better results, there are still many issues that need to be addressed so that the algorithm can produce good results. Some of those issues were addressed in the previous

[1] the MPEG-7 standard defines such a standard for audio signal description [31]

chapters, among them the need for improved onset detection and feature comparison, which in the author's opinion are the most important unsolved problems. In conclusion, it can be said that this diploma thesis represents a step towards musically acceptable concatenative re-synthesis of audio signals, but there is still a lot of work to be done.

8 References

[1] Feiten Bernhard, Günzel Stefan, “A Sound-Retrieval Index Based on Two-Dimensional Similarity Maps”, Proceedings of the 94th AES Convention, Berlin 1993

[2] Scheirer Eric, “Tempo and Beat Analysis of Acoustic Musical Signals”, Journal of the Acoustical Society of America, Volume 103 Issue 1, 1998

[3] Davies Matthew, Plumbley Mark, “Context-Dependent Beat Tracking of Musical Audio”, IEEE Transactions on Speech and Audio Processing, Volume 15 Issue 3, 2007

[4] Bello Juan, Daudet Laurent, Abdallah Samer, Duxbury Chris, Davies Mark, Sandler Mark, “A Tutorial on Onset Detection in Music Signals”, IEEE Transactions on Speech and Audio Processing, Volume 13 Issue 5, 2005

[5] Bello Juan, Pickens Jeremy, “A Robust Mid-Level Representation for Harmonic Content in Music Signals”, Proceedings of the 6th International Symposium on Music Information Retrieval, London 2005

[6] Foote Jonathan, “Automatic Audio Segmentation Using a Measure of Audio Novelty”, IEEE International Conference on Multimedia and Expo, New York 2000

[7] Goto Masataka, Muraoka Yoichi, “An Audio-Based Real-Time Beat Tracking System and its Applications”, Proceedings of the International Computer Music Conference, San Francisco 1998

[8] Verfaille Vincent, Arfib Daniel, “A-DAFX: Adaptive Digital Audio Effects”, Proceedings of the Conference on Digital Audio Effects, Limerick 2001

[9] Klapuri Anssi, "Sound Onset Detection by Applying Psychoacoustic Knowledge", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Phoenix 1999

[10] Levine Scott, "Audio Representations for Data Compression and Compressed Domain Processing", Ph. D. Dissertation, University of Stanford 1998

[11] Bello Juan, Duxbury Chris, Davies Mike, Sandler Mark, "On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain", IEEE Signal Processing Letters, Volume 11 Issue 6, 2004

[12] Masri Paul, "Computer Modelling of Sound for Transformation and Synthesis of Musical Signals", Ph.D. Dissertation, University of Bristol 1996

[13] Mörchen Fabian, Ultsch Alfred, Thies Michael, Löhken Ingo, Nöcker Mario, Stamm Christian, Efthymiou Niko, Kümmerer Martin, "MusicMiner: Visualising Timbre Distances of Music as Topographical Maps", Technical Report No. 47, University of Marburg 2005

[14] Bartsch Mark, Wakefield Gregory, "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing", Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz 2001

[15] Tzanetakis George, Essl Georg, Cook Perry, "Automatic Musical Genre Classification of Audio Signals", Proceedings of the 2nd International Symposium on Music Information Retrieval, Indiana 2001

[16] Lartillot Olivier, "MIR Toolbox 1.0", downloaded from http://users.jyu.fi/~lartillo/mirtoolbox/ , accessed April 27, 2008

[17] Wickelmaier Florian, "An Introduction to MDS", Reports from the Sound Quality Research Unit, Aalborg University 2003

[18] Roads Curtis, "Microsound", MIT Press, Cambridge 2004

[19] Sturm Bob, “Adaptive Concatenative Sound Synthesis and Its Application to MicroMontage Composition”, Computer Music Journal Volume 30 Issue 4, 2006

[20] Schwarz Diemo, “Concatenative Sound Synthesis: The Early Years”, Journal of New Music Research, Volume 35 Number 1, 2006

[21] Smith Julius O. III, “Viewpoints on the History of Digital Synthesis”, Proceedings of the International Computer Music Conference, Montréal 1991

[22] Mathews Max, “The Technology of Computer Music”, MIT Press, Cambridge 1969

[23] Kostelanetz Richard, „John Cage“, M. DuMont Schauberg, Köln 1973

[24] Harley James, „Xenakis – His Life in Music“, Routledge, New York 2004

[25] Budon Osvaldo, “Composing with Objects, Networks and Time Scales: An Interview with Horacio Vaggione”, Computer Music Journal, Volume 24 Issue 3, 2000

[26] Simon Ian, Basu Sumit, Salesin David, Agrawala Maneesh, “Audio Analogies: Creating New Music from an Existing Performance by Concatenative Synthesis”, Proceedings of the International Computer Music Conference, Barcelona 2005

[27] Jehan Tristan, “Event-Synchronous Music Analysis / Synthesis”, Proceedings of the 7th International Conference on Digital Audio Effects Naples 2004

[28] Hazel Steven, Soundmosaic Website, http://awesame.org/soundmosaic/, accessed April 14, 2008

[29] Lazier Ari, Cook Perry, “MoSievius: Feature Driven Interactive Audio Mosaicing”, Proceedings of the Conference on Digital Audio Effects, London 2003

[30] Schwarz Diemo, “A System for Data-Driven Concatenative Sound Synthesis”, Proceedings of the Conference on Digital Audio Effects, Verona 2000

[31] MPEG-7 ISO Standard, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, accessed July 30, 2008

[32] Goodwin Michael, Avendano Carlos, "Frequency-Domain Algorithms for Audio Signal Enhancement Based on Transient Modification", AES Journal Volume 54 Issue 9, 2006

[33] Luig Johannes, "Resynthese von Audiosignalen mittels Feature Extraction", Project Thesis, Institute for Electronic Music and Acoustics, University of Music and Dramatic Arts Graz, 2007

[34] Peeters Geoffroy, McAdams Stephen, Herrera Perfecto, "Instrument Sound Description in the Context of MPEG-7", Proceedings of the International Computer Music Conference Berlin 2000

[35] Zölzer Udo, "DAFX: Digital Audio Effects", John Wiley & Sons, Chichester 2002

[36] Duxbury Chris, Sandler Mark, Davies Mike, "A Hybrid Approach to Musical Onset Note Detection", Proceedings of the Conference on Digital Audio Effects, Hamburg 2002

[37] Verma Tony, Levine Scott, Meng Teresa, "Transient Modeling Synthesis: A Flexible Analysis/Synthesis Tool for Transient Signals", Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Seattle 1998

[38] Verma Tony, Meng Teresa, "Extending Spectral Modeling Synthesis with Transient Modeling Synthesis", Computer Music Journal, Volume 24 Issue 2, 2000

[39] Daudet Laurent, "Transients Modelling By Pruned Wavelet Trees", Proceedings of the International Computer Music Conference, Havana 2001

[40] Abdallah Samer, Plumbley Mark, "Probability As Metadata: Event Detection in Music Using ICA As Conditional Density Model", Proceedings of the 4th

International Symposium on Independent Component Analysis and Blind Signal Separation, Nara 2003

[41] Harte Christopher, Sandler Mark, „Automatic Chord Identification Using a Quantised Chromagram“, Proceedings of the 118th AES Convention, Barcelona 2005

[42] Klapuri Anssi, Eronen Antti, Astola Jaakko, "Analysis of the Meter of Acoustic Musical Signals", IEEE Transactions on Speech, Audio and Signal Processing, Volume 14 Issue 1, 2006

[43] Leveau Pierre, "Sound Onset Labeliser", downloaded from www.lam.jussieu.fr/src/Members/Leveau/SOL/SOL.htm#DL, accessed November 7, 2007

[44] Vary Peter, Martin Rainer, „Digital Speech Transmission“, John Wiley & Sons, Chichester 2006

[45] Sigurdsson Sigurdur, Petersen Kare Brandt, Lehn-Schioler Tue, "Mel Frequency Cepstral Coefficients: An Evaluation of Robustness of MP3 Encoded Music", Proceedings of the 7th International Symposium on Music Information Retrieval, Victoria 2006

[46] Campbell Joseph, "Speaker Recognition: A Tutorial", Proceedings of the IEEE, Volume 85 Issue 9, 1997

[47] Kauppinen Ismo, "Methods for detecting impulsive noise in speech and audio signals", Proceedings of the 14th International Conference on Digital Signal Processing, Santorini 2002

[48] Dixon Simon, "Automatic Extraction of Tempo and Beat from Expressive Performances", Journal of New Music Research, Volume 30 Issue 1, 2001

[49] Goto Masataka, Muraoka Yoichi, "Music Understanding at the Beat Level – Real-Time Beat Tracking for Audio Signals", Working Notes of the IJCAI-95 Workshop on Computational Auditory Scene Analysis, Montreal 1995

[50] Goto Masataka, Muraoka Yoichi, “Real-Time Rhythm Tracking for Drumless Audio Signals – Chord Change Detection for Musical Decisions”, Working Notes of the IJCAI-97 Workshop on Computational Auditory Scene Analysis, Nagoya 1997

[51] Gouyon Fabien, Herrera Perfecto, “Determination of the Meter of Musical Audio Signals: Seeking Recurrences in Beat Segment Descriptors”, Proceedings of the 114th AES Convention, Amsterdam 2003

[52] Collins Nick, “A Comparison of Sound Onset Detection Algorithms with Emphasis on Psycho-acoustically Motivated Detection Functions”, Proceedings of the 118th AES Convention, Barcelona 2005

[53] Logan Beth, “Mel Frequency Cepstral Coefficients for Music Modeling”, Proceedings of the 1st International Symposium on Music Information Retrieval, Plymouth 2000

[54] Ganchev Todor, Fakotakis Nikos, Kokkinadis George, “Comparative Evaluation of Various MFCC Implementations on the Speaker Verification Task”, Proceedings of the 10th International Conference on Speech and Computer, Patras 2005

[55] Oppenheim Alan, Schafer Roland, Buck John, „Zeitdiskrete Signalverarbeitung“, Pearson Studium, München 2004

[56] Plomp R., Levelt W. J. M., “Tonal Consonance and Critical Bandwidth”, Journal of the Acoustical Society of America, Volume 38 Issue 4, 1965

[57] Jensen Kristoffer, “Timbre Models of Musical Sounds”, DIKU Rapport 99/7, Copenhagen University 1999

[58] Manolakis Dimitris, Ingle Vinay, Kogon Stephen, “Statistical and Adaptive Signal Processing”, Artech House, Norwood 2005

[59] Bühl Achim, Zöfel Peter, “SPSS 12 – Einführung in die moderne Datenanalyse unter Windows”, Pearson Studium, München 2005

[60] Young Forrest, ALSCAL Site, http://forrest.psych.unc.edu/research/alscal.html, accessed May 29, 2008

[61] West Kris, Cox Stephen, “Features and Classifiers for the Automatic Classification of Musical Audio Signals”, Proceedings of the 5th International Symposium on Music Information Retrieval, Barcelona 2004

[62] Bagci Ulas, Erzin Engin, “Automatic Classification of Musical Genres Using Inter-Genre Similarity”, IEEE Signal Processing Letters, Volume 14 Issue 8, 2007

9 Appendix

9.1 Appendix A - List of Figures

9.2 Appendix B - Audio Signals used for Onset Detection Evaluation

Number	Short Name	Category	Description
1	Bass	PP	Monophonic Bass Track
2	Pingpong	NPP	Excerpt from The Computer Jockeys, "Ping Pong"
3	Sunshine	Mix	Excerpt from Cream, "Sunshine Of Your Love"
4	Übermensch	Mix	Excerpt from Die Ärzte, "Rock'n'Roll Übermensch"
5	Distguit	PP	Monophonic Distorted Guitar Track from [43]
6	Drums1	NPP	Monophonic Drum Track
7	Drums2	NPP	Monophonic Drum Track
8	Dingdong	Mix	Excerpt from E.A.V., "Ding Dong"
9	Godzilla	Mix	Excerpt from Fu Manchu, "Godzilla"
10	Readyornot	PNP	Excerpt from The Fugees, "Ready Or Not"
11	FX	PP	Monophonic Synthesizer Track
12	Guitar	PP	Monophonic Clean Guitar Track from [43]
13	Hiphop	Mix	Excerpt from Xzibit, "X"
14	Clansman	Mix	Excerpt from Iron Maiden, "The Clansman"
15	Musclemuseum	Mix	Excerpt from Muse, "Muscle Museum"
16	Drumduet	NPP	Excerpt from a Phil Collins Live Recording
17	Piano	PP	Monophonic Piano Track
18	Pop1	Mix	Pop Track from [43]
19	Pop2	Mix	Pop Track from [43]
20	Rock	Mix	Rock Track from [43]
21	Boulevards	NPP	Excerpt from Sin, "On Boulevards"
22	Synth1	PNP	Monophonic Synthesizer Track
23	Synth2	PNP	Monophonic Synthesizer Track
24	Synthbass	PP	Bass Track from [43]
25	Violin	PNP	Monophonic Violin Track
26	Vox	PNP	Monophonic Choral Track

9.3 Appendix C – Overview of the Concatenative Music Synthesis Implementation in Matlab

The implementation of the algorithm that re-synthesizes music from database fragments is explained in detail in this chapter. An overview of the algorithm structure and short descriptions of the used functions are given.

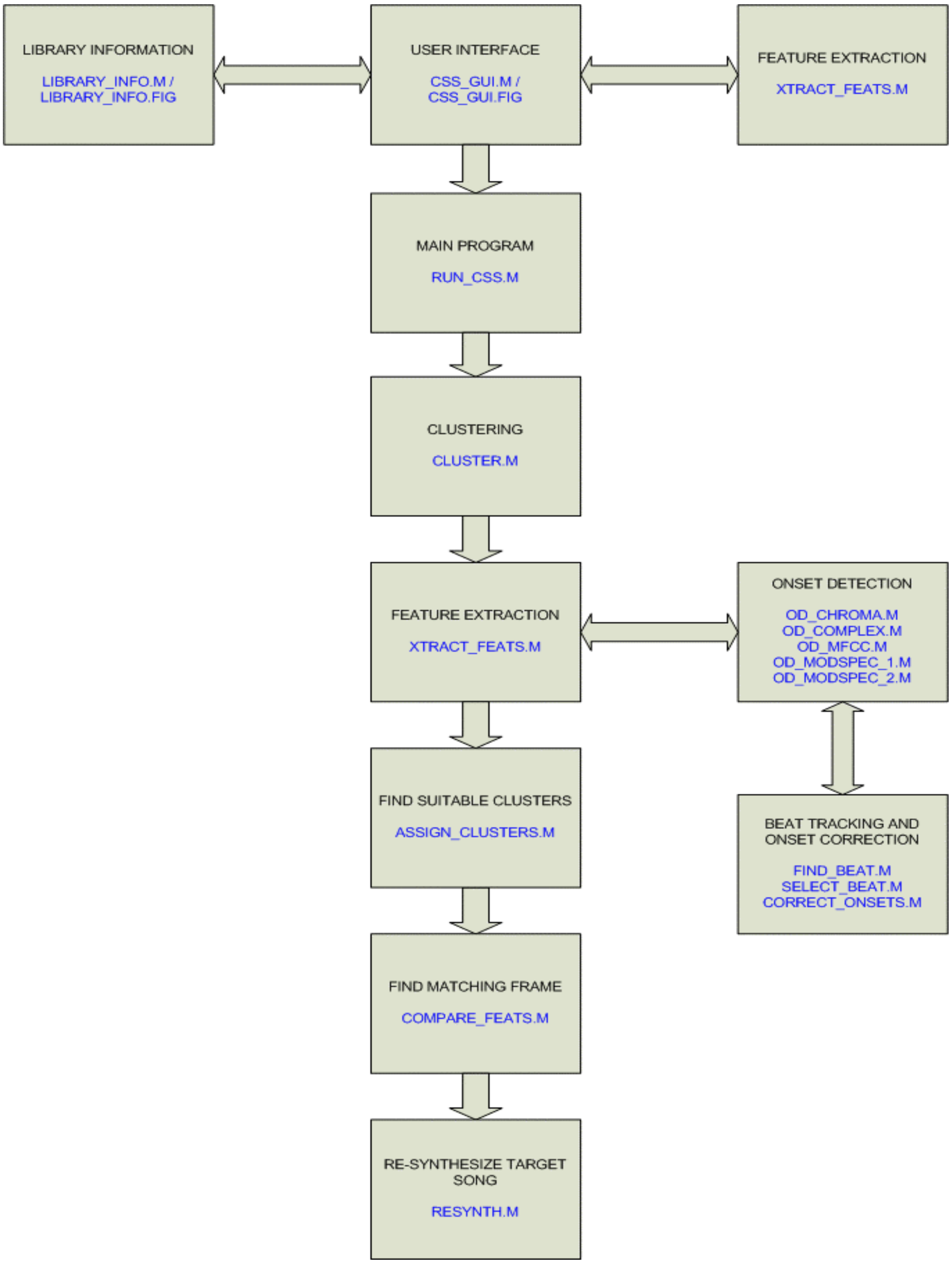


Figure 9.1 CSS algorithm overview with used functions (blue) and their purpose (black)

The algorithm starts with the Matlab GUI *CSS_GUI* that serves as an user interface. Databases (called “libraries”) can be created, loaded, manipulated and viewed (using the *Library_Info* GUI). Also, the user can choose between the onset detection methods presented in this thesis, and the implemented beat tracker can be turned on or off.

The *CSS_GUI* interface sends the chosen options and library data to the master program *run_css*. All other used functions are sub-functions used by this main script.

run_css starts by calling the sub-function *cluster*, which is used to organise the library frames into clusters and sub-clusters according to their length and pitch. The clustered library as well as information about cluster and sub-cluster boundaries is returned to *run_css*.

After a target song is selected, the next step consists of calling the sub-function *xtract_feats*, where onset detection is performed in order to be able to segment the signal into suitable frames. The onset detection method can be chosen among the methods presented earlier (*od_chroma* for the chroma-based method, *od_complex* for the complex frequency domain method, *od_mfcc* for the MFCC-based method, *od_modspec_1* for the sub-band trajectory method and *od_modspec_2* for the weighted sub-band trajectory method). The optional beat tracker then finds beat hypotheses (*find_Beat*), selects the most likely beat candidate (*select_Beat*) and corrects the onset detection results using the estimated beat (*correct_od*). For every audio segment, the low-level features needed for the six-dimensional feature vector describing the segment are calculated. This information is returned to *run_css*. As can be seen in Figure 9.1, this function is also used to analyse and segment the audio signals that are added to a library or when a new library is created – the only difference is that not a target song but the audio signal that is to be loaded into a library is analysed.

The library and target song information is then passed to *assign_clusters*, where every target song segment is assigned to a cluster and a sub-cluster so that the algorithm knows in which sub-cluster to look when comparing the target song segments to the database fragments. This is done by comparing the target song fragment length and pitch to the boundaries determined by the function *cluster*.

The next step consists of finding the library fragments that best match the target song segments. This is implemented in *compare_feats*. The six-dimensional feature vectors are compared to each other, and all available information about the best-fitting database fragment is passed back to *run_css*. The current library is then expanded with the target song data.

In the last algorithm step, the *resynth* sub-function cuts out the previously determined library fragments from the .wav audio signals that make up the database, performs minor signal manipulations like fade-in and fade-out as well as normalisation and concatenates the fragments to form a new .wav audio signal which can be saved.