Music Information Retrieval

Alexander Schindler

May 12, 2016

Contents

Music Information Retrieval
http://www.ifs.tuwien.ac.at/mir

Alexander Schindler
Institute of Software Technology and Interactive SystemsVienna University of Technology-http://www.ifs.tuwien.ac.at/~schindler

Chapter 1

Music Information Retrieval

Music Information Retrieval (MIR) is a multidisciplinary research field drawing upon expertise from information retrieval, library science, musicology, audio engineering, digital signal processing, psychology, law and business [1]. This relative new discipline gained on momentum in the early 2000s and is supported by an eager research community, the International Society for Music Information Retrieval (ISMIR), with its homonymous annual conference.

A general concern of the MIR research field is the extraction and interpretation of meaningful information from music. That this is a challenging endeavor can be seen by the different music representations such as audio recordings, music scores, lyrics, playlists, music videos, album cover arts, music blogs, etc. Music sheets contain formal descriptions of music compositions. Audio recordings capture the information about the actual interpretation of a composition. Lyrics and music videos add further layers of artistic and semantic expression to a music track. Playlists set different tracks into relation and blog posts contain opinions about music, artists and performances. All this relevant information comes in different modalities. Recording are provided as digital audio requiring digital signal processing to extract the information. Lyrics and blog posts require text and natural language analysis as well as music videos and album cover art analysis require image processing and computer vision techniques. Mining playlists and social media data requires knowledge of collaborative filtering methods. Finally, it has proven to be advantageous to combine two or more modalities to harness the information provided by the different layers. Given that wealth of music information one of the main tasks is to reduce as much information and to retain the most relevant notion of it to provide fast and accurate methods of indexing music and to provide new and intuitive ways to search and retrieve music.

1.1 What is Music?

The Oxford Dictionary defines music as "the art of combining vocal or instrumental sounds (or both) to produce beauty of form, harmony, and expression of emotion" [2]. The discussion about a concise definition of music is still ongoing and the remarks by [3] denote its complexity: *'Explications of the concept of music usually begin with the idea that music is organized sound. They go on to note that this characterization is too broad, since there are many examples of organized sound that are not music, such as human speech, and the sounds non-human animals and machines make'*

Recorded Audio

This might be the most palpable form of music. Referring to the cited definition of the Oxford Dictionary, music is a combination of vocal or instrumental sounds. In a more general form this is referred to as a performance. Thus, when we speak of music we generally refer to recorded performances whose copies are distributed by resellers or played on radio stations. The history of recorded music started in 1896 when Thomas Alva Edison established the National Phonograph Company to sell music pressed on wax cylinders for the phonograph. In 1901 the Victor Talking Machine Company introduced the phonographs whose 78-rpm flat disc defined the standard for the future development of music distribution. This further lead to the development of the Long Play (LP) album which was replaced by the cassette in the early 80s of the last century. With the introduction of the Compact Disc (CD) a major shift to digital music consumption was performed. Digital audio is the main source for content based audio analysis which is one of the major sub-fields of MIR. We will introduce this in more detail in the following chapters.

Symbolic Music

Symbolic music is a representation of music in a logical structure based on symbolic elements representing audio events and the relationships among those events. The most commonly known form of symbolic music is the music score which is also referred to as 'Sheet Music'. Music scores are available as printed sheets or in various digital representations such as MIDI, MusicXML, ABC and GUIDO. The most powerful among them is MusicXML which was released first in 2004 and describes among the composition also the layout of the score, lyrics and textual annotations and further features. The main difference between symbolic music and recorded or perceived music is, that scores only describe how a piece of music is intended to be performed. In other words: It is a guideline that is used by a performer to play a certain piece of music. For MIR, symbolic music has some advantageous properties compared to recorded music. The most obvious is the exact knowledge about played notes and their durations. Such information needs to be estimated from the audio spectrum of a recorded music track which is a complex and not yet matured task.

Text

Text based music related information is presented in form of music lyrics, music metadata such as title, album and artist names, textual information provided on artist web pages, blog posts, music reviews, including social tagging (e.g. last.fm, MusicStrands, etc.) and Wikipedia. To harness such information, classical information retrieval is applied to the text extracted from different sources.

Visual Media

Visual based music related information is provided by album cover arts, music videos and music advertising. In the last century the visual representation has become a vital part of music. Album covers grew out of their basic role of packaging to become visual mnemonics to the music enclosed. Stylistic elements emerged into prototypical visual descriptions of genre specific music properties. Initially intended to aide or sway customers in their decision of buying a record, these artworks became an influential part of modern pop culture. The 'look of music' became an important factor in people's appreciation of music. The rise of music videos in the early 80s provided further momentum to this development. This emphasis on visual aspects of music transcended from topics directly connected with music production into aspects of our daily life. The relationship to fashion tightened, yielding to different styles that discriminated the various genres. How much influence visual music had on us over half a century is hard to evaluate, but we grew accustomed to a visual vocabulary that is specific for a music style in a way that the genre of a music video can often be predicted despite the absence of sound.

1.2 MIR Tasks and Challenges

Genre classification

Classification of music into different categories such as music genres, styles, etc. This is often also referred to as automatic labelling or tagging. Musical genre is probably the most popular metadata for the description of music content. Music industry promotes the use of genres and home users like to organize their audio collections by this annotation. Consequently, the need of automatic classification of audio data into genres increased substantially, as did the number of researchers addressing this problem. Besides recent advances in genre classification there is still the question, what exactly defines a genre, or whether it is mainly dependent on a user's experience and taste. Though the concept of musical genre might be illdefined, recent approaches that use audio feature extraction combined with machine learning techniques achieve promising results. Genre classifiers typically work well with clearly described, well-distinguishable genres.

Mood classification

Categorizing music into different moods such as *angry*, *sad*, *happy*, *calm*, etc.

Music Recommendation

Recommending new music to listeners.

Artist identification

Identifying the performing artist.

Artist similarity

Identifying similarities between artists.

Cover song detection

Identify that a given song is a cover version of another original version.

Rhythm and beat detection

Identify the beat and rhythm of a track.

Score following

Score following relates to the process of aligning the notes of a score to its interpretation during a specific performance. This might be during a live performance, or to an audio recording. Research on automatic score following through computers distinguishes between different scenarios: * Symbolic to MIDI * Symbolic to Audio

Their main differences are based on the digital representation of music. Symbolic music is usually represented in a machine processable form, such as MIDI or MusicXML. Recorded music is represented as a time-series of sampled audio. More generally speaking, sampled audio is an exhaustive sequence of numbers representing the measured auditive energy at a certain time. Information provided in this form is not suitable for automatic processing at first hand and has to be reduced and transformed into an appropriate representation. This process is the focus of the research domain music information retrieval (MIR). While, in the symbolic representation of music, it is clear which note is being played at a certain time, in digital audio this information has to be deduced from the spectral properties of the sampled audio through digital signal processing.

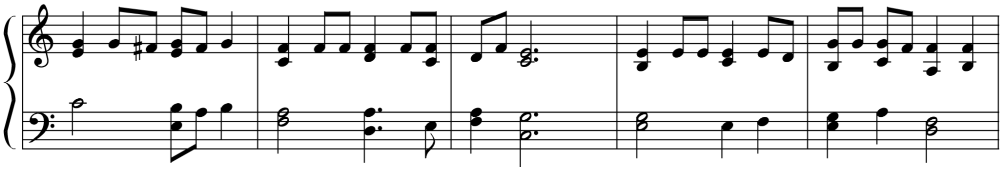
Symbolic to MIDI alignment: This scenario is based on the prerequisite that the performed music is available in the same symbolic format of the score. This is the case for set ups where a musical instrument is connected to the computer through a Musical Instrument Digital Interface (MIDI). Music events such as hitting a key on an electronic piano equipped with a MIDI interface, are immediately communicated to the connected computer and are available in an interpretable from. Having both sources - scores and performance - in a comparable format makes it more convenient for further processing (e.g. score following, automatically assisted training, etc.).

Symbolic to Audio alignment: While symbolic music unambiguously describes which note is played at which time of the track, this does not apply to recorded music. The main challenge with sampled audio is that it is a mix of frequencies, originating usually from a multitude of individual instruments and voices, which sources currently cannot fully be separated again, after they have been fixed in an audio mix. This topic is subject to slowly progressing research. Musical notes refer to audio frequencies (e.g. concert pitch = 440Hz). Thus, it seems obvious, that sampled audio can be transcribed into symbolic music by assigning note values to audio frequencies. In a simplified approach this works for monophonic tunes played by a single instrument. Having multiple instruments playing polyphonic tunes (like chords and harmonies) creates overlapping frequencies, partial frequencies caused by the instrument's timbre and other influences in the overall audio mix, which cause complex distributions of the sound energy over the frequency spectrum of the recording. Thus, it is not computationally distinguishable anymore which notes have been played by the distinct instruments. To align symbolic music to sampled audio, both types have to be transformed into a representation that can be compared directly. This is the prevalent case for the score-following prototype developed during the Europeana-Sounds project and will be described in the remainder of this document.

Optical music recognition

Optical Music Recognition (OMR) is similar to Optical Character Recognition (OCR) and intends to convert music scores into a machine interpretable form. Actually, OMR is a descendant of OCR. Most systems are based on top of OCR engines and extend their functionality by adding layers that recognize and interpret music notations. Contrary to the advanced solutions provided for optical character recognition, OMR is still a challenging research topic and far from being solved. In other words: The results of OMR technology are currently less reliable than those of the OCR technology it builds upon.

The aim of optical music recognition is to get a score from the paper into a machine interpretable format, to have it processed or played. This requires converting it from its analog into a digital form, which is usually accomplished through scanning a page. The result is a digital image of the score which can be displayed on a screen or printed. Yet, it provides no more information to the computer than its analogue form (it's a digital image file). The visual information of the image has to be extracted and converted into a processable format such as MIDI or MusicXML. Similar to OCR, document image analysis is applied to detect the page layout and to recognize where the semantically relevant information is located.



Example music score

The advantage of traditional text recognition over OMR is that once the locations of columns and lines are known, characters can be recognized one by one. There is no further dimension to keep in mind that influences the recognition of the consecutive character. Music scores on the other hand rely on conditions set at the start of a line or even further behind. Different clefs result in different note assignments. This information is given at the left side of a staff and has to be remembered for the entire line. Yet, this is just one of many challenges OMR has to face. There are many ambiguities that are comprehended by musicians, but are really difficult to be translated into a general model or template for music notation. The following table, taken from [8] lists some of these problems:

Chord detection

detect and transcribe the chords of music segments

Audio Fingerprinting

Audio fingerprinting is a technique that is frequently used for music identification. This algorithm is also used by the well-known Shazam music identification service.

Audio segmentation

Detect different segments within a track such as verse, chorus, bridge, etc.

Instrument detection

detect the different instruments that were used in the performance.

Automatic source separation

Identify and separate the different sources such as instruments that were originally used to perform the track.

Onset detection

Detect the temporal onsets of music events, such as beats, chord changes, etc.

Melody transcription

convert recorded music into symbolic music.

Similarity Retrieval

Finding similar songs based on certain criteria.

Rather than searching for songs that sound similar to a given query song, users often are more interested in songs that cover similar topics, such as ‘love songs’, or ’Christmas carols’, which are not acoustic genres per se. Songs about these particular topics might cover a broad range of musical styles. Similarly, the language of a song’s lyrics often plays a decisive role in perceived similarity of two songs as well as their inclusion in a given playlist. Even advances in audio feature extraction will not be able to overcome the fundamental limitations of this kind. Song lyrics therefore play an important role in music similarity. This textual information thus offers a wealth of additional information to be included in music retrieval tasks that may be used to complement both acoustic as well as metadata information for pieces of music.

Chapter 2

Introduction to Audio Feature Extraction

Feature Extraction is the core of content-based description of audio files. With feature extraction from audio, a computer is able to recognize the content of a piece of music without the need of annotated labels such as artist, song title or genre. This is the essential basis for music information retrieval tasks, such as similarity based searches (query-by-example, query-by-humming, etc.), automatic classification into categories, or automatic organization and clustering of music archives. Features extracted from the audio signal are intended to describe the stylistic content of the music, e.g. beat, presence of voice, timbre, etc.

Information Reduction

A typical CD quality mainstream radio track has an average length of three minutes. This means, that the song is digitally described in Pulse-code Modulation (PCM) by almost 16 million numbers (3 [minutes] x 60 [seconds] x 2 [stereo channels] x 44100 [sampling rate]). This information requires 30MB of memory and a considerable amount of time to process. Processing the small number of 100 tracks, which relates to about 10 audio CDs, would require about 3GB of memory, which is currently about the average size of memory provided in personal computers. Processing 100000 songs would require 3TB of memory, which requires vast resources (e.g. acquisition, hosting, energy consumption, etc.) and is only suitable for academic or industrial settings. Consequently, there is a strong desire to reduce the information provided in an audio track and distill it into a smaller set of representative numbers that capture higher level information about the underlying track.

2.1 Music Files

A set of music files will be used to demonstrate different aspects of music feature etraction. The audio tracks used in this article were downloaded from the FreeMusicArchive and are redistributable licensed under the Creative Commons license. To visualize the expressivenmess of music features and their ability to discriminate different types of music, the songs of this article originate from different music genres.

In the code-block below the sound files used in this tutorial will be specified. Please change the paths of the files according your local settings. Because MP3-decoding is not constistently implemented across all platforms, it is required that to manually convert the audio files into wave format as a prerequiste.

The following code embeds the audio player from the FMA Web page into this notebook. Thus, it is possible to pre-listen the audio samples online.

```
[3]: list_audio_samples(sound_files)
```

```
Out[3]: <IPython.core.display.HTML object>
```

2.2 Audio Representations

Basic knowledge of the production process of digital audio is essential to understand how to extract music features and what they express.

2.3 Sampled Audio

Audio signals as perceived by our ears have a continuous form. Analog storage media were able to preserve this continuous nature of sound (e.g. vinyl records, music cassttes, etc.). Digital logic circuits on the other hand rely on electronic oscillators that sequentially trigger the unit to process a specific task on a discrete set of data units (e.g. loading data, multiplying registers, etc.). Thus, an audio signal has to be fed in small pieces to the processing unit. The process of reducing a continuous signal to a discrete signal is called sampling. The audio signal is converted into a sequences of discrete numbers that are evenly spaced in time.

As an example one could monitor the temperature in an office by measuring every minute the current degree of Celsius. We fruther simplify this example by accepting only integer values. In this case the continous change of temparature in the office is sampled at a rate of 60 samples per minute. Since Celsius values in offices seldom rise above 128 or drop below -128 degree, it is sufficient to use 8 Bits to store the sampled data. The process of turing continous values (e.g. temperature, sound pressure, etc.) into discrete values is called quantization.

For digitizing audio especially music in CD quality, typically a sampling rate of 44100 Herz at a bit depth of 16 is used. This means, that each second of audio data is represented by 44100 16bit values.

- The time domain
- The frequency domain
- The Fourier Transform
- The Short-Time Fourier Transform

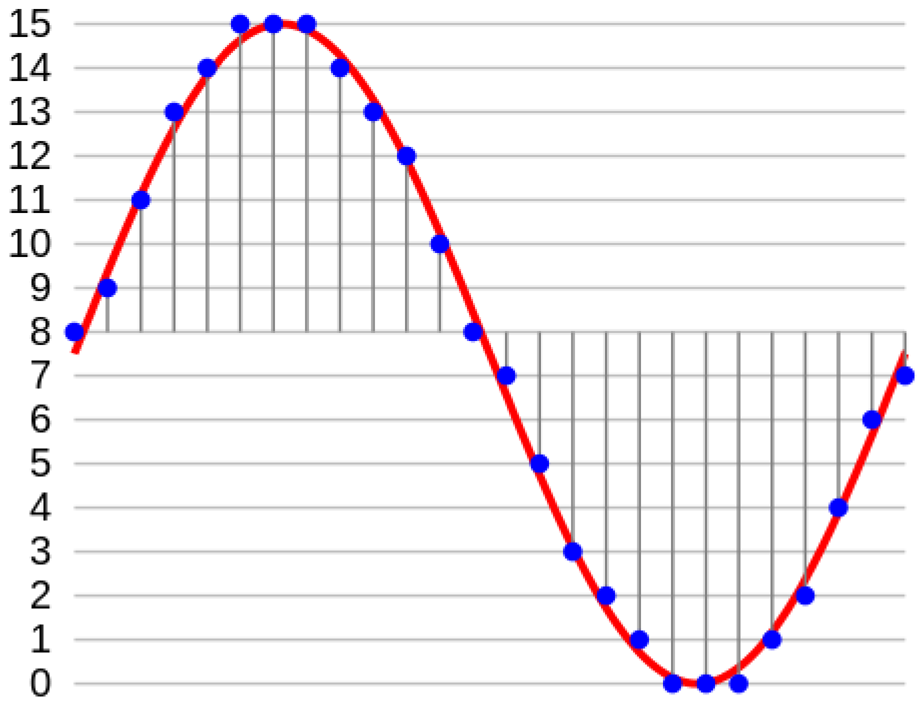
To start the feature extraction process, the audio files have to be opened and loaded. Usually audio files are opened as streams and processed sequentially, but for this tutorial it is more convenient to fully keep them in memory. After the audio data has been loaded two essential data blocks are known: the actual audio data and the rate the source has been sampled with. From this information it is easy to derive the first audio feature: the length of the track. Since the samplerate is defined as number of samples per second, the length is simply calculated by dividing the sample cound by the samplerate.

2.4 The Time Domain

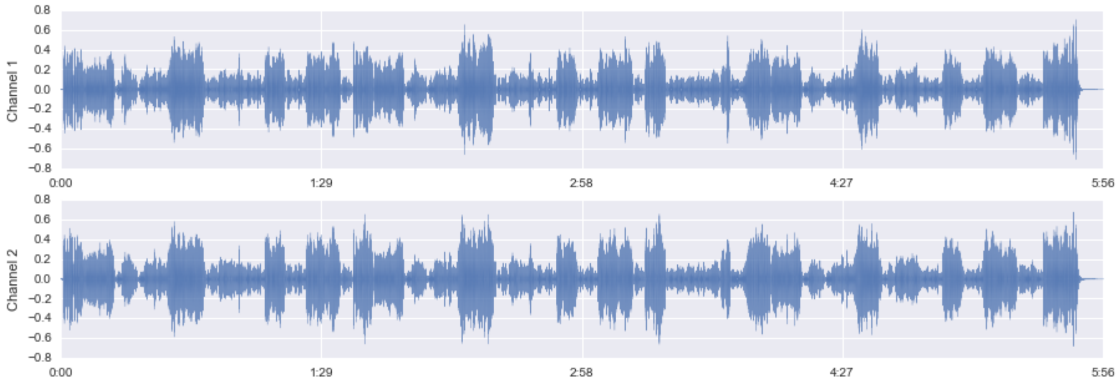
Time domain analysis is analyzing the data over a time period. Functions such as electronic signals, market behaviors, and biological systems are some of the functions that are analyzed using time domain analysis. For an electronic signal, the time domain analysis is mainly based on the voltage – time plot or the current – time plot. In a time domain analysis, the variable is always measured against time. There are several devices used to analyze data on a time domain basis. The cathode ray oscilloscope (CRO) is the most common device when analyzing electrical signals on a time domain.

2.4.1 Waveform

A waveform is the shape and form of a signal such as a wave moving in a physical medium or an abstract representation. In many cases the medium in which the wave is being propagated does not permit a direct visual image of the form. In these cases, the term 'waveform' refers to the shape of a graph of the varying quantity against time or distance. An instrument called an oscilloscope can be used to pictorially represent a wave as a repeating image on a screen. By extension, the term 'waveform' also describes the shape of the graph of any varying quantity against time



An example of 4-bit pulse code modulation (16 different binary-coded possibilities) showing quantization and sampling of a sine-wave signal



```
<matplotlib.figure.Figure at 0xb5f8630>
```

2.5 The Frequency Domain

2.5.1 Fourier Transform

- Essential part of any audio feature extraction algorithm
- Audio waves contain a spectrum of many different frequencies, each with its own amplitude and phase.

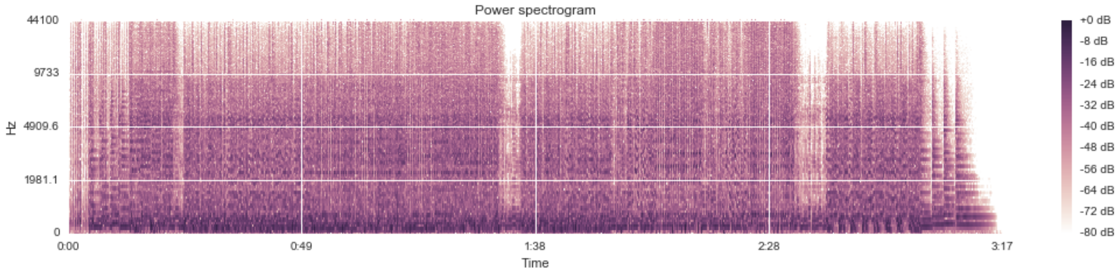
- Idea is that every complex continuous signal can be explained by decompose a wave into its component frequencies and phases.
- Inverse Fourier Transform Transform back from frequency into time domain
- No loss of data when transformation is applied
- Results of the Fourier Transform are
  - Phase histogram (rarely used)
  - Frequency histogram
  - Sets of bins
  - Each corresponding to a different range of frequencies

Magnitude Spectrum vs. Power Spectrum

- the power spectrum is the magnitude spectrum squared (calculated for each bin, by summing the square of the imaginary output of the FFT with the square of the real value)
- magnitude spectrum and power spectrum rarely used directly as features (too much raw information)
- many spectral features are derived from either the power spectrum or the magnitude spectrum

Fast Fourier Transform

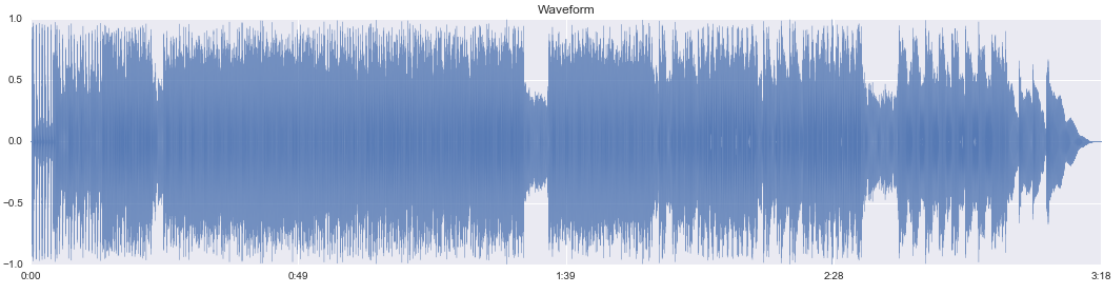
- efficient algorithm to compute the discrete Fourier transform (DFT)
- divide and conquer algorithm
- $O(NlogN)$ instead of $O(N2)$
- N must be a power of 2



2.5.2 Audio Pre-processing

Combine separate channels.

Below an example waveform of a mono channel after combining the stereo channels by arithmetic mean:



Chapter 3

Audio Features

3.1 Time Domain Features

3.1.1 Zero Crossing Rate

The Zero-crossing rate (ZCR) is a simple, straightforward and inexpensive feature. It measures how often the amplitude of a signal crosses the zero-line. This feature is directly related to the requency of an audio sample. Higher frequencies have shorter wavelengths and thus cross the zero-line more often. Consequently, the ZCR describes a spectral property of the recorded audio track. Because its values are not weighted by other attributes such as amplitude, the ZCR is easily biased by high pitched noise. Thus, the ZCR is often used to describe the amount of noise which is present in a recording.

The Zero Crossing Rate is defined by:

$$zcr = \frac{1}{N-1} \sum_{i=1}^{N-1} |sign|x(i)| - sign|x(i-1)||$$

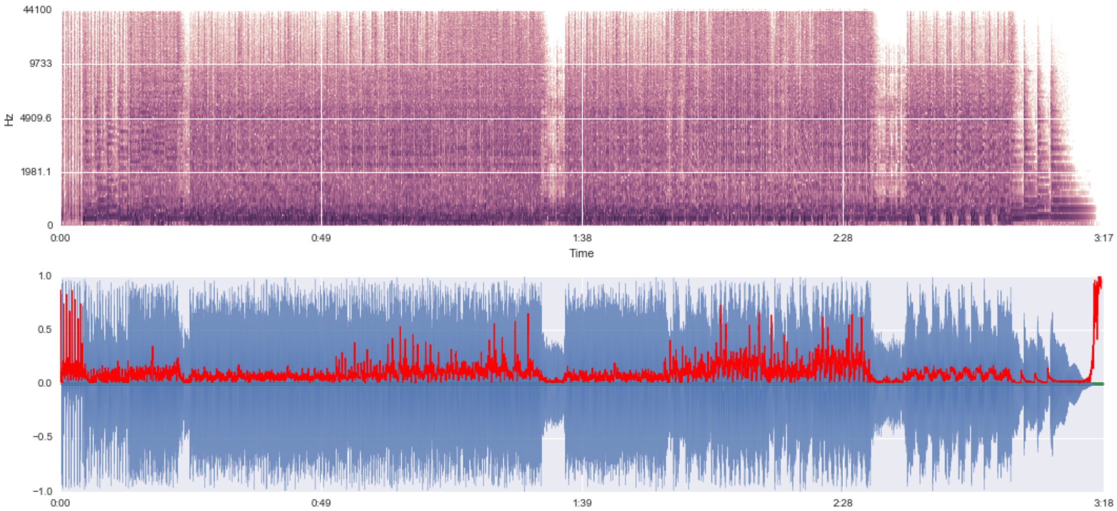
where N is the number of samples of an audio file and the signum function is defined by:

$$sign|x(i)| = \begin{cases} 1, & \text{if } x(i) > 0 \\ 0, & \text{if } x(i) = 0 \\ -1, & \text{if } x(i) < 0 \end{cases}$$

average number of times the audio signal crosses the zero amplitude line per time unit.

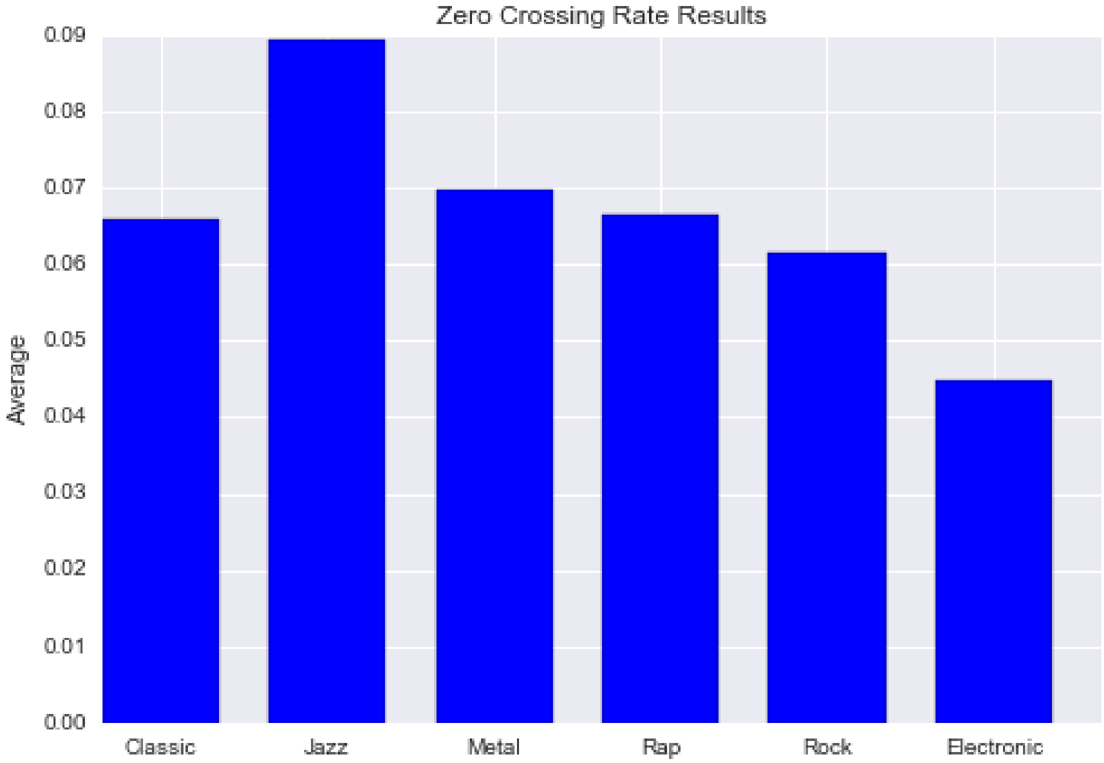
A major advantage of the ZCR is that it is very simple to compute. It has been applied to speech processing to distinguish voiced sections from noise and it has also been applied to MIR tasks such as classifying percussion sounds and genres.

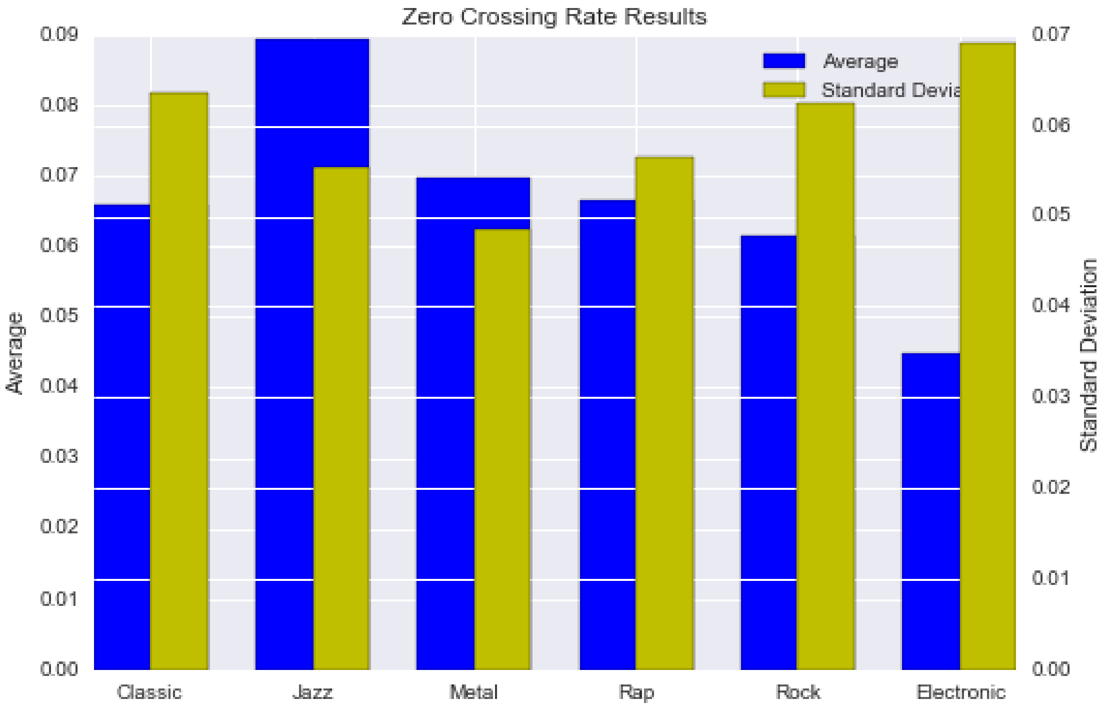
The following charts visualize how the Zero Crossing Rate correlates with the audio signal. The upper chart shows the spectrogram of the track. The lower chart superimposes the Zero Crossing Rate over the waveform of the track. It can be observed that sewuences with increased energy in high frequency reagions have a higher zero crossing rate.



```
<matplotlib.figure.Figure at 0x2cac6e80>
```

Use ZCR to compare music





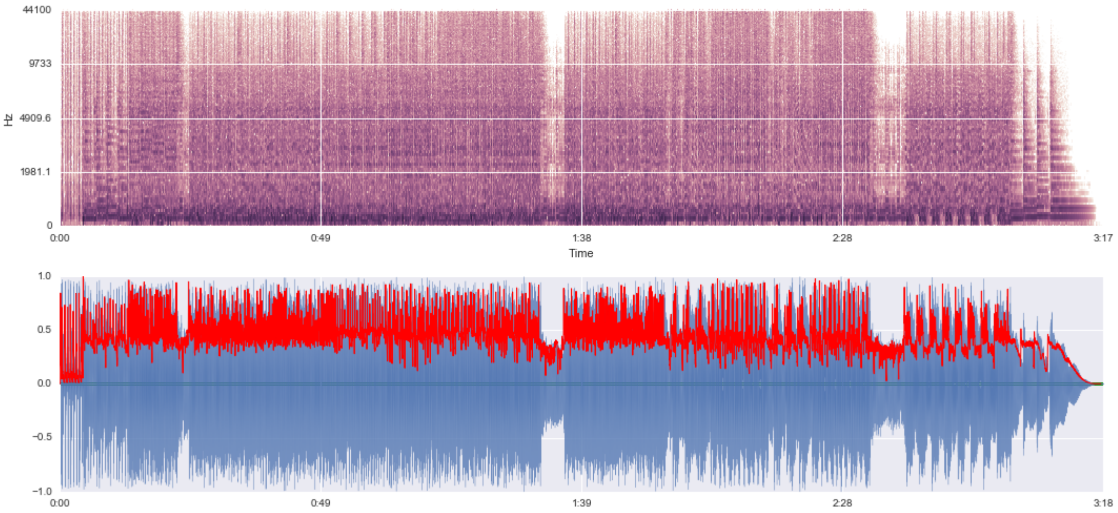
3.1.2 Root Mean Square

Root Mean Square is a way of comparing arbitrary waveforms based upon their equivalent energy. RMS voltage is the contant (DC) voltage that would be required to produce the same heat in a resistive load, indicating equivalent ability to do work. You can't use a simple "average voltage": Consider that a sine wave has positive and negative phases that would average to zero, yet it still generates heat regardless of the polarity of the voltage.

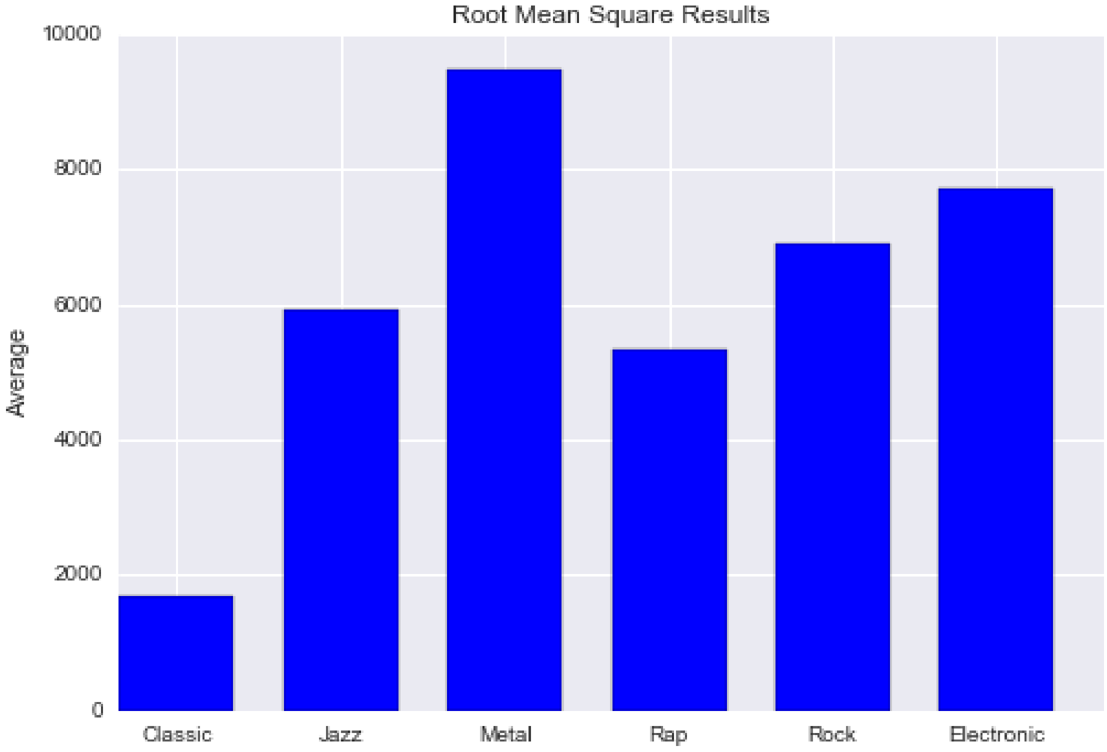
The RMS method takes the square of the instantaneous voltage before averaging, then takes the square root of the average. This solves the polarity problem, since the square of a negative value is the same as the square of a positive value. For a sine wave, the RMS value thus computed is the same as the amplitude (zero-to-peak value) divided by the square root of two, or about 0.7071 of the amplitude. For a repetitive waveform like this, an accurate calculation can be done by averaging over a single cycle of the wave (or an integer number of cycles), but for random noise sources the averaging time must be long enough to get a good representation of the characteristics of the source.

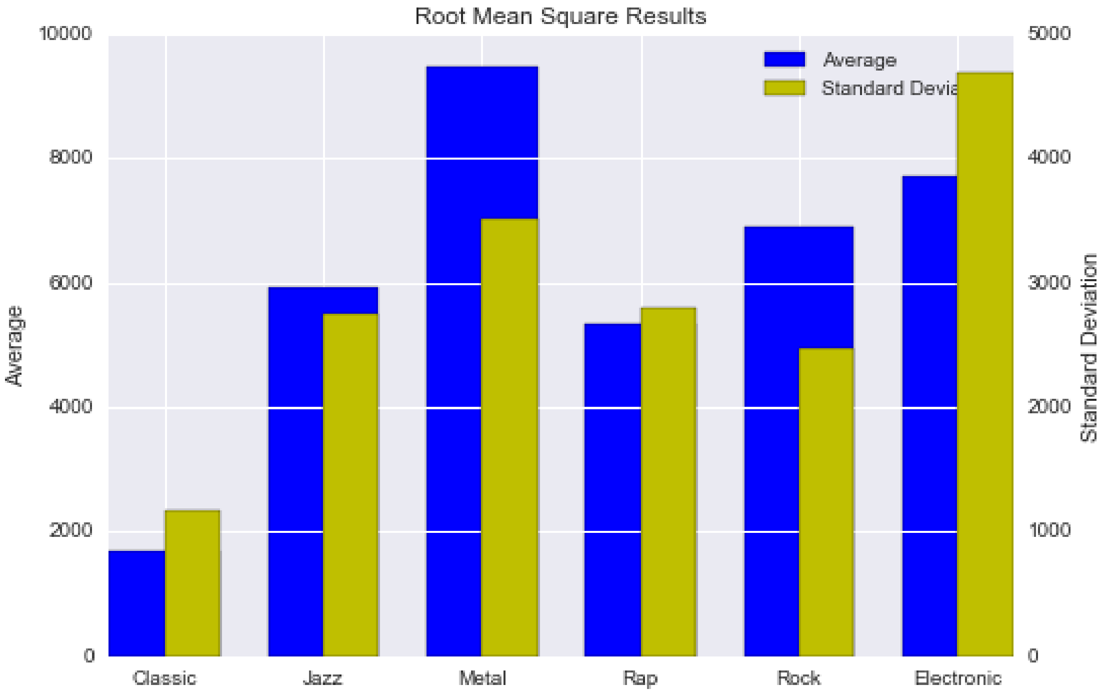
For noise or signal bursts, the RMS value is still the effective heating value, but of course it is reduced because the signal is not always present. If you know the RMS value of the continuous signal, the true RMS of the burst will be the continuous RMS times the square root of the fraction of the time the signal is on.

$$r_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$


44100
9733
Hz
4909.6
1981.1
0
0:00
0:49
1:38
2:28
3:17
Time
1.0
0.5
0.0
−0.5
−1.0
0:00
0:49
1:39
2:28
3:18

```
<matplotlib.figure.Figure at 0x2cc806d8>
```


Root Mean Square Results
10000
8000
6000
4000
2000
0
Average
Classic
Jazz
Metal
Rap
Rock
Electronic



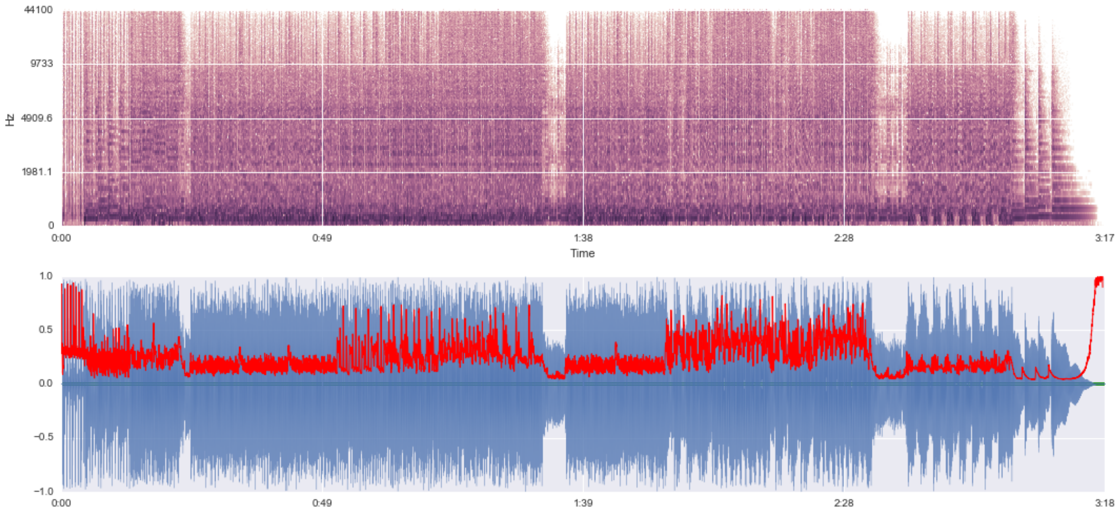
3.2 Spectral Features

3.2.1 Spectral Centroid

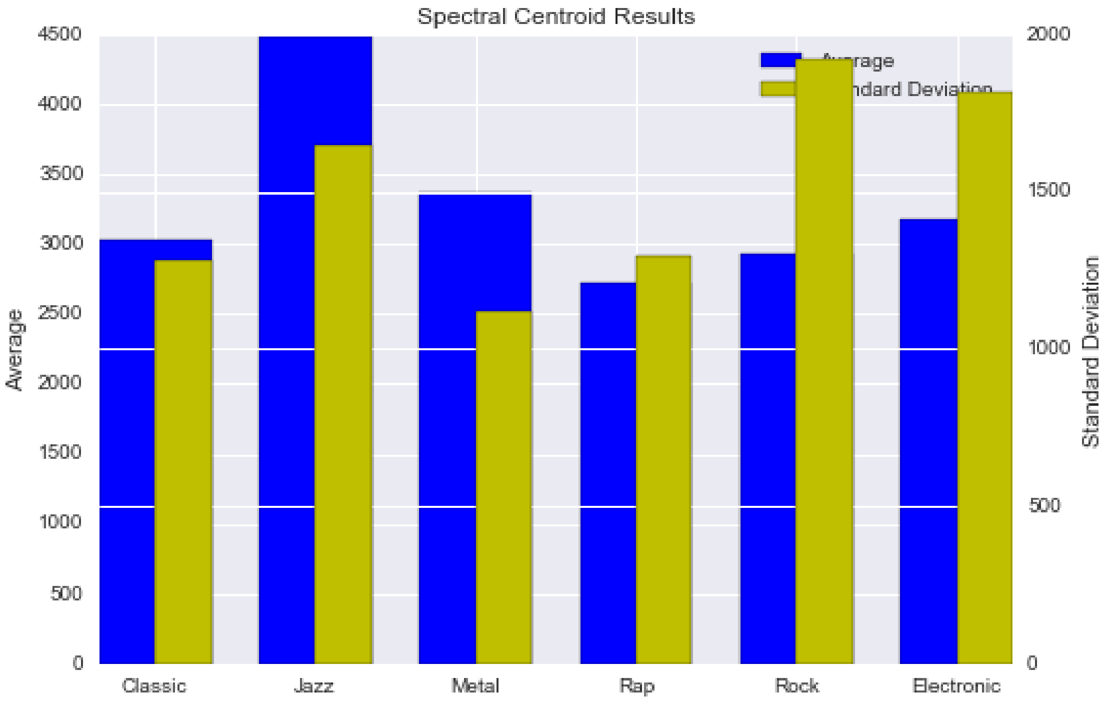
The Spectral Centroid (SC) is on of a set of features which are also known as the Spectral Shape Features, because they distinctively describe different aspects of the acoustic texture of a sound. The SC is the frequency-weighted sum of the power spectrum (=squared Magnitude spectrum M) normalized by its unweighted sum. It could be described as the center of gravity or the balancing point of the spectrum. It determines the frequency area around which most of the signal energy concentrates and gives an indication of how dark or bright a sound is:

$$SC(t) = \frac{\sum_{n=1}^{N} |M(t,n)|^2 * n}{\sum_{n=1}^{N} |M(t,n)|^2}$$

where t represents the current time frame and n is the index for one of N frequency bins calculated by the FFT.



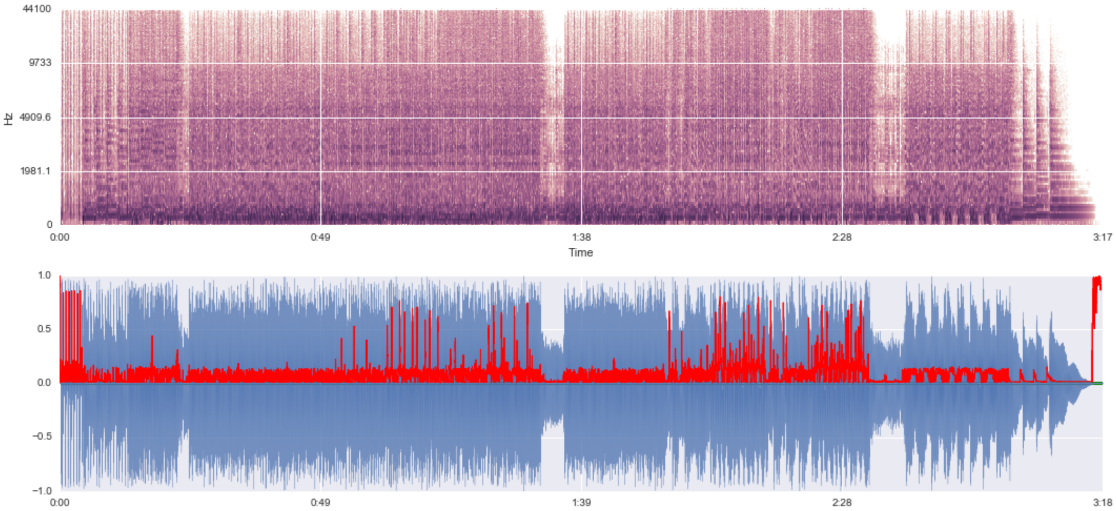
```
<matplotlib.figure.Figure at 0x55f710b8>
```



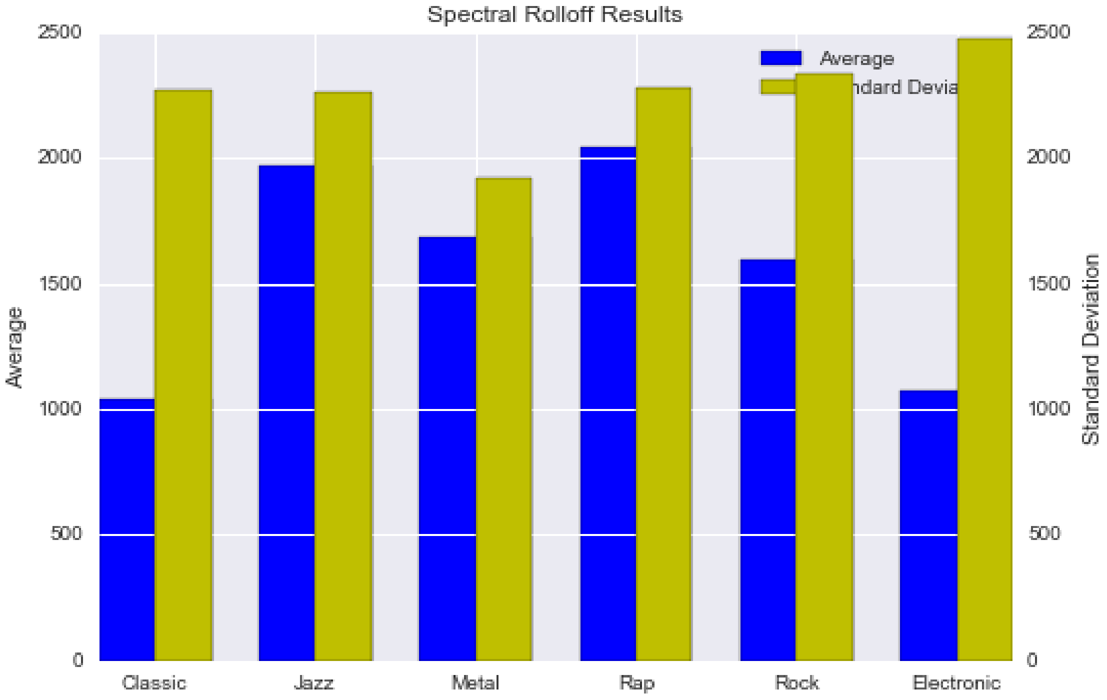
3.2.2 Spectral Rolloff

The Spectral Rolloff (SRO) is the frequency below which some fraction k (typically 0.85, 0.9 or 0.95 percentile) of the cumulative spectral power resides. It is a measure of the skewness of the spectral shape and an indication of how much energy is in the lower frequencies. It is often used to distinguish voiced from unvoiced speech or music.

$$\sum_{n=1}^{SR(t)} |M(t,n)|^2 = n * \sum_{n=1}^{N} |M(t,n)|^2$$



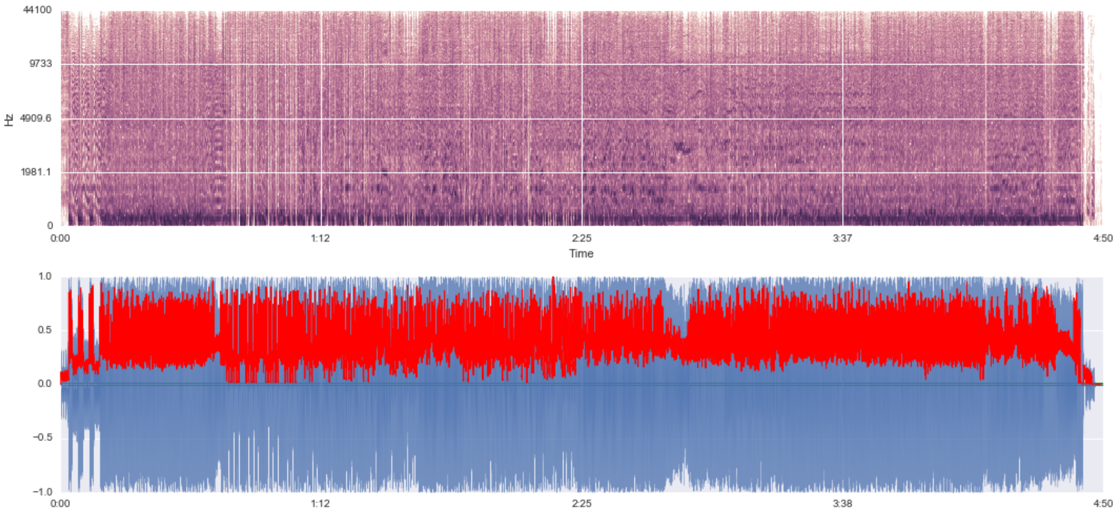
```
<matplotlib.figure.Figure at 0x2d6fdef0>
```



3.2.3 Spectral Flux

- squared differences in frequency distribution of two successive time frames
- measures the rate of local change in the spectrum

$$SF(t) = \frac{\sqrt{\sum_{n=1}^{N} (|M(t,n)| - |M(t-1,n)|)^2}}{N}$$



```
<matplotlib.figure.Figure at 0xb5295c0>
```

3.2.4 Spectral Variability

- standard deviation of the bin values of the magnitude spectrum
- provides an indication of how flat the spectrum is and if some frequency regions are much more prominent than others

3.2.5 Strongest Partial

- center frequency of the bin of the magnitude or power spectrum with the greatest strength
- can provide a primitive form of pitch tracking

3.2.6 MPEG7 Features

MPEG-7 is a multimedia content description standard defined by the Moving Picture Expert Group aka MPEG.

- **Multimedia Content Description Interface**
- ISO/IEC standard by MPEG (Moving Picture Experts Group)
- Providing meta-data for multimedia
- MPEG-1, -2, -4: make content available

- MPEG-7: makes content accessible, retrievable, filterable, manageable (via device / computer).
- was Adopted in 2002
- Details:
  - ISO/IEC JTC1/SC29/WG11N6828; editor:José M. Martínez Palma de Mallorca, Oct. 2004, MPEG-7 Overview (version 10)
  - http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
- Specifies
  - Set of audio descriptors
  - a set of Reference Software

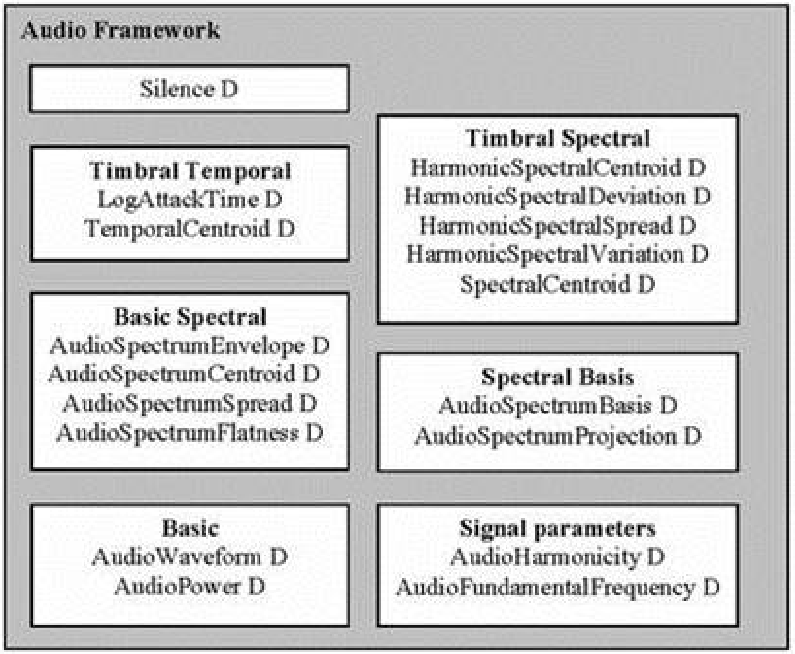
Low-level descriptors:

- spectral, parametric, and temporal features of a signal

High-level description tools:

- specific to a set of applications
- general sound recognition and indexing
- instrumental timbre
- spoken content
- audio signature description scheme
- melodic description tools to facilitate query-by-humming

MPEG 7 Audio Framework Stack



Audio Framework

Silence D

Timbral Temporal
LogAttackTime D
TemporalCentroid D

Basic Spectral
AudioSpectrumEnvelope D
AudioSpectrumCentroid D
AudioSpectrumSpread D
AudioSpectrumFlatness D

Basic
AudioWaveform D
AudioPower D

Timbral Spectral
HarmonicSpectralCentroid D
HarmonicSpectralDeviation D
HarmonicSpectralSpread D
HarmonicSpectralVariation D
SpectralCentroid D

Spectral Basis
AudioSpectrumBasis D
AudioSpectrumProjection D

Signal parameters
AudioHarmonicity D
AudioFundamentalFrequency D

The MPEG 7 Audio Framework Stack

Basic: temporally sampled scalar values for general use. * AudioWaveform Descriptor: waveform envelope for display purposes. * AudioPower Descriptor: temporally-smoothed instantaneous power quick summary of a signal. Applicable to all kinds of signals.

Basic Spectral: single time-frequency analysis of signal * AudioSpectrumEnvelope: Base class. the short-term power spectrum: display, synthesize, general-purpose search * AudioSpectrumCentroid: is the spectrum dominated by high or low frequencies? * AudioSpectrumSpread: the power spectrum centered near the spectral centroid, or spread out over the spectrum? pure-tone and noise-like sounds * AudioSpectrumFlatness: the presence of tonal components

Signal Parameters: periodic or quasi-periodic signals * AudioFundamentalFrequency: "confidence measure", replacing "pitch-tracking" * AudioHarmonicity: distinction between sounds with a harmonic / inharmonic / non-harmonic spectrum

Timbral Temporal: temporal characteristics of segments of sounds, musical timbre * LogAttackTime * TemporalCentroid: where in time the energy of a signal is focused. Useful when attack times are identical

Timbral Spectral: spectral features in a linear-frequency space * SpectralCentroid: power-weighted average of the frequency of the bins in the linear power spectrum. distinguishing musical instrument timbres. * 4 Ds for harmonic regularly-spaced components of signals: * HarmonicSpectralCentroid * HarmonicSpectralDeviation * HarmonicSpectralSpread * HarmonicSpectralVariation

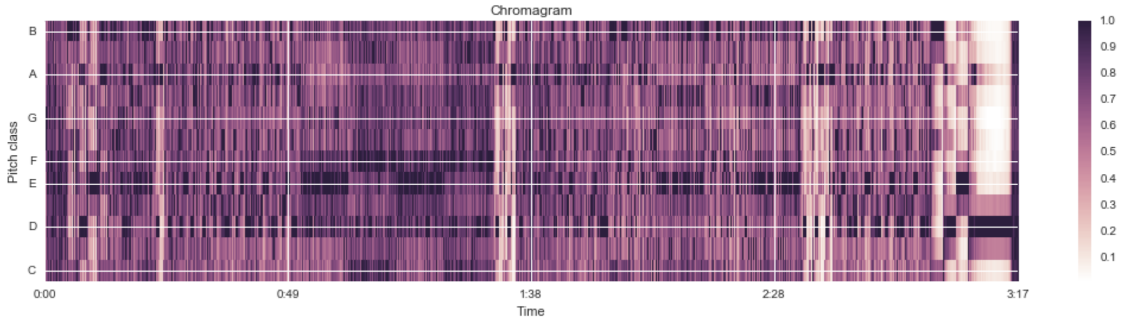
Spectral Basis: low-dimensional projections of a spectral space to aid compactness and recognition * AudioSpectrumBasis: a series of (time-varying / statistically independent) basis functions derived from the singular value decomposition of a normalized power spectrum. * AudioSpectrumProjection: low-d features of a spectrum after projection upon a reduced rank basis. independent subspaces of a spectra correlate strongly with different sound sources. Provide more salience using less space. With Sound Classification and Indexing Description Tools.

Silence segment: no significant sound * aid further segmentation of the audio stream, or as a hint not to process a segment

3.3 Music Theoretical Features

3.3.1 Chroma Features

Chroma Features represent the 12 distinct semitones (or chroma) of the musical octave. This results in one or a sequence of twelve dimensional vectors where - for example - the bin that corresponds to the pitch class A captures the spectral energy of A0 and all its corresponding subband pitches A1, A2, etc.



3.4 Psychoaccoustical Features

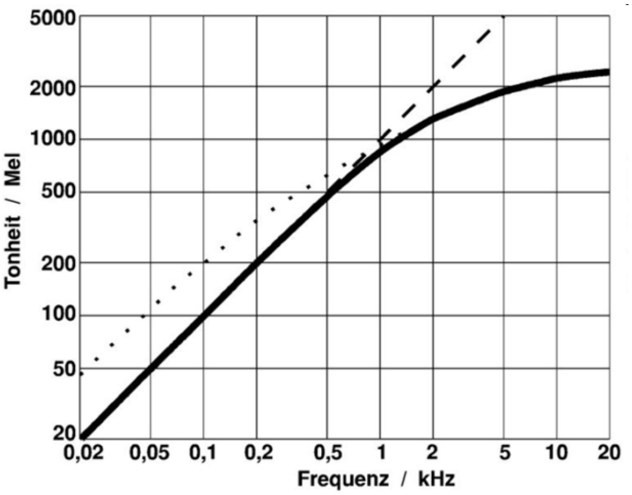
3.4.1 Psychoaccoustics

- psychological (subjective) correlations of the physical parameters of acoustics
  - Humans are most sensitive to sounds in the 1-5 kHz range

3.4.2 Mel-Frequency Cepstral Coefficients (MFCC)

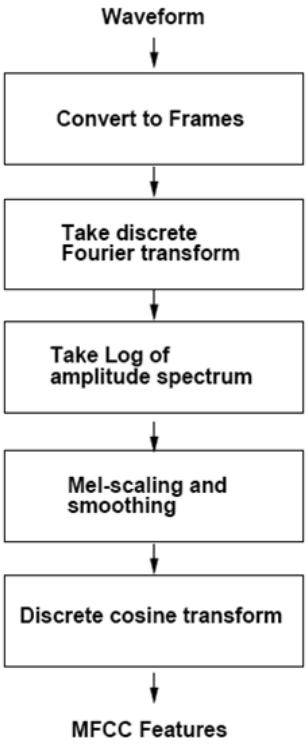
This feature set was used previously in speech recognition and intends to model human auditory response by transforming it to the Mel scale. The 'cepstrum' ('s-p-e-c' reversed) results of taking the Fast Fourier transform (FFT) of the decibel spectrum as if it were a signal. The result shows the rate of change in the different spectrum bands. It is a dominant feature in speech recognition, because of it's ability to represent the speech amplitude spectrum in a compact form. It also has proved to be highly efficient in music retrieval. Represent the rate of change in the different spectrum bands it is a good timbre descriptor. The MFCCs are the most commonly used features in music processing. They represent the rate of change in the different spectrum bands and are generally known to be good descriptors of music timbre.

The Mel-Scale



Mel-scale

- perceptually motivated scale
- human auditory system does not perceive pitch in linear manner
- Mel comes from the word melody to indicate that the scale is based on pitch comparisons
- maps between actual frequencies and perceived pitch
- was obtained empirically by listening experiments
- reference point:
  - 1000 Hz tone, 40 dB above listener's threshold = 1000 Mels
  - This is the range where humans are most sensitive to variations in sound
- Mapping is approximately
  - linear below 1kHz and
  - logarhitmic above

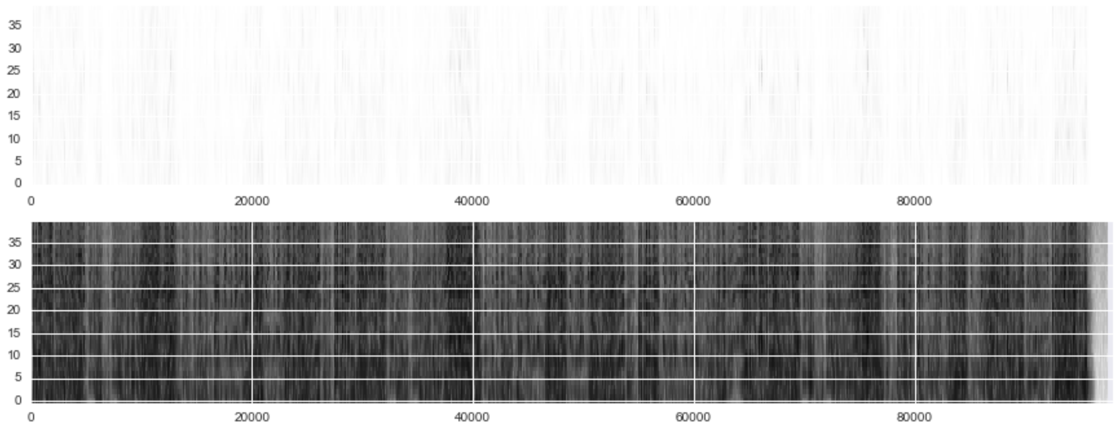

Waveform
Convert to Frames
Take discrete Fourier transform
Take Log of amplitude spectrum
Mel-scaling and smoothing
Discrete cosine transform
MFCC Features

Mel-scale

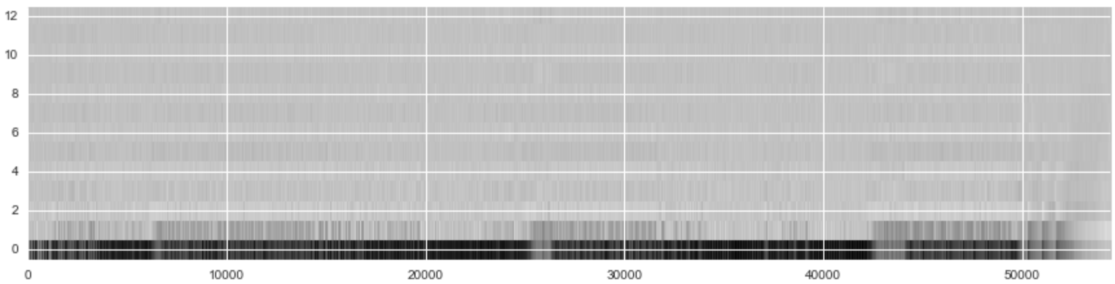
MFCC Computiation

1. Pre-processing in time-domain (pre-emphasizing)
2. Compute the spectrum amplitude by windowing with a Hamming window
3. Filter the signal in the spectral domain with a triangular filter-bank, whose filters are approximatively linearly spaced on the mel scale, and have equal bandwith in the mel scale
4. Compute the DCT of the log-spectrum

Comparing the original spectrogram against the Mel-scale transformed Spectrogram



Compute the MFCC by computing the DCT of the log-spectrum

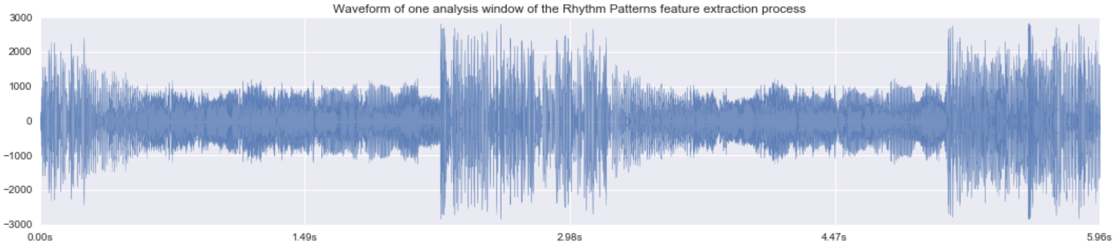


3.4.3 Rhythm Patterns

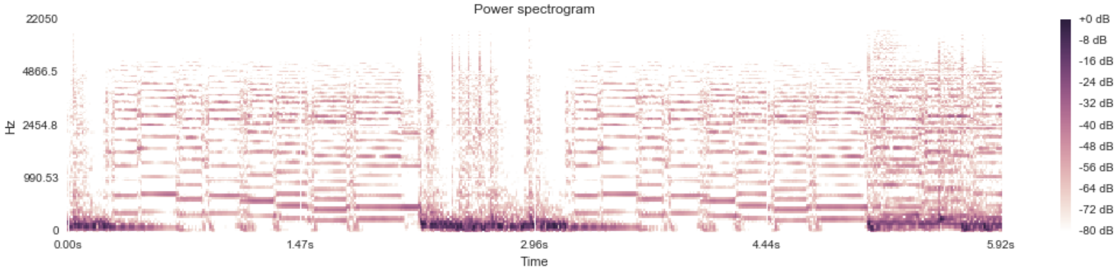
Rhythm Patterns (also called Fluctuation Patterns) describe modulation amplitudes for a range of modulation frequencies on 'critical bands' of the human auditory range, i.e. fluctuations (or rhythm) on a number of frequency bands. The feature extraction process for the Rhythm Patterns is composed of two stages:

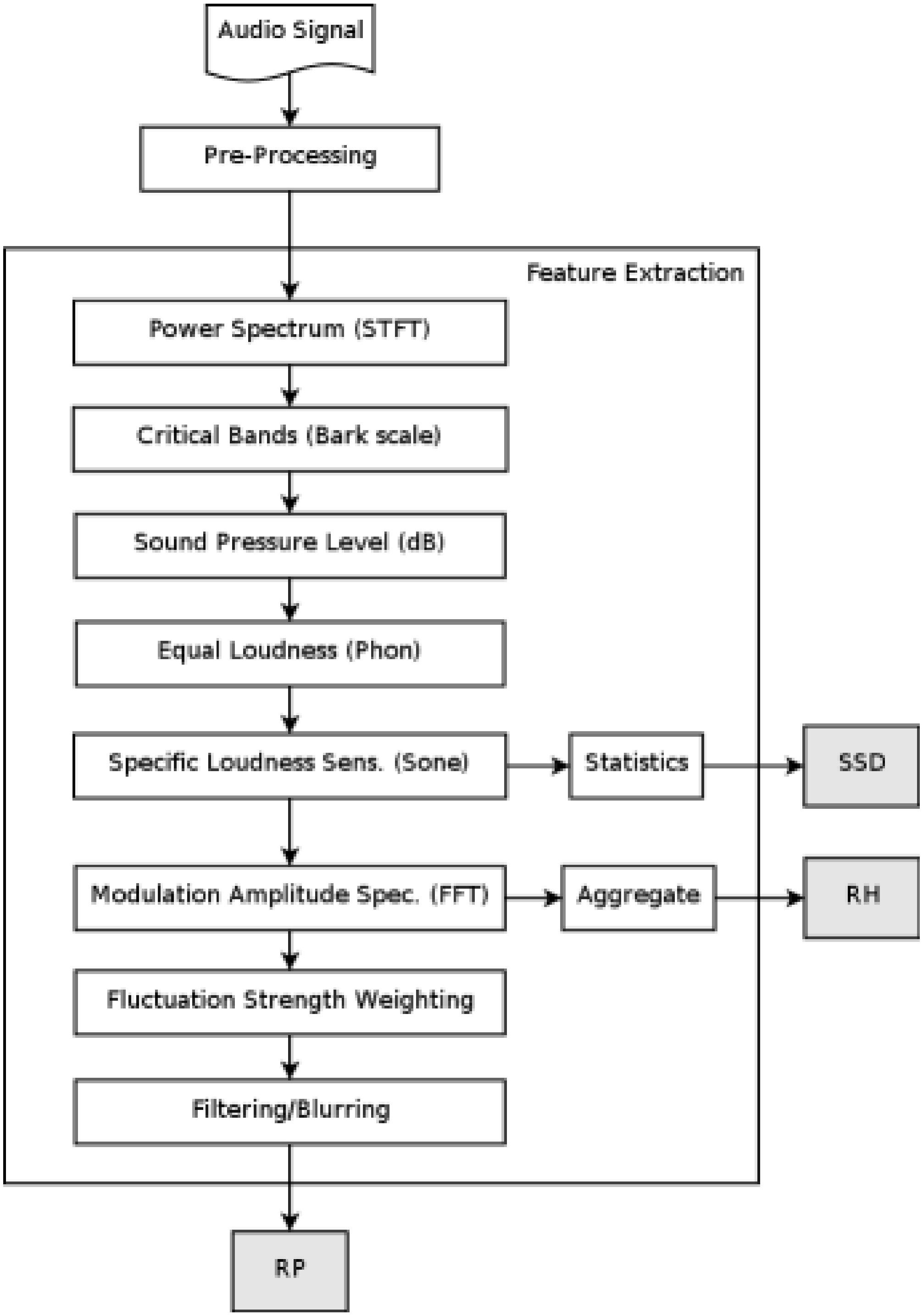
First, the specific loudness sensation in different frequency bands is computed, by using a Short Time FFT, grouping the resulting frequency bands to psycho-acoustically motivated critical-bands, applying spreading functions to account for masking effects and successive transformation into the decibel, Phon and Sone scales. This results in a power spectrum that reflects human loudness sensation (Sonogram).

In the second step, the spectrum is transformed into a time-invariant representation based on the modulation frequency, which is achieved by applying another discrete Fourier transform, resulting in amplitude modulations of the loudness in individual critical bands. These amplitude modulations have different effects on human hearing sensation depending on their frequency, the most significant of which, referred to as fluctuation strength, is most intense at 4 Hz and decreasing towards 15 Hz. From that data, reoccurring patterns in the individual critical bands, resembling rhythm, are extracted, which – after applying Gaussian smoothing to diminish small variations – result in a time-invariant, comparable representation of the rhythmic patterns in the individual critical bands.



Convert to frequency domain



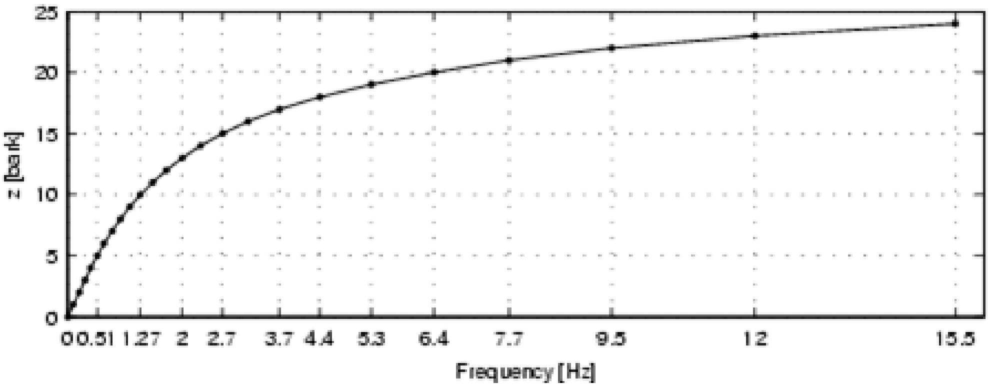

Audio Signal
Pre-Processing
Feature Extraction
Power Spectrum (STFT)
Critical Bands (Bark scale)
Sound Pressure Level (dB)
Equal Loudness (Phon)
Specific Loudness Sens. (Sone)
Statistics
SSD
Modulation Amplitude Spec. (FFT)
Aggregate
RH
Fluctuation Strength Weighting
Filtering/Blurring
RP

The Rhythm Patterns feature extraction process

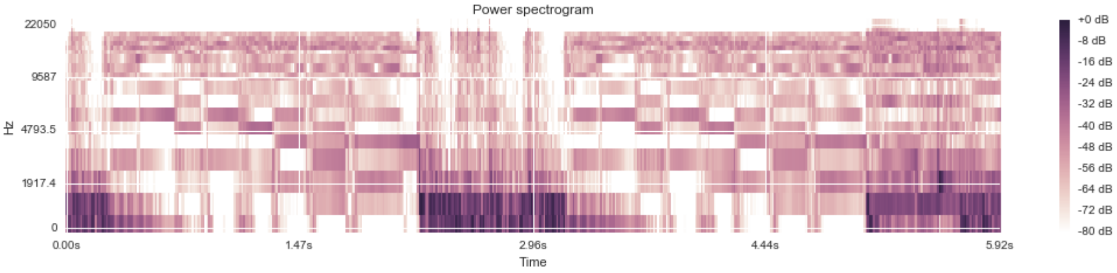
3.4.4 Map to Bark Scale

Bark Scale

- psychoacoustical scale (related to Mel scale)
- proposed by Eberhard Zwicker in 1961
- named after Heinrich Barkhausen who proposed the first subjective measurements of loudness
- 24 ‘critical bands’ of hearing (non linear)
- a frequency scale on which equal distances correspond with perceptually equal distances
- Associated to mel scale: 1 Bark corresponds to 100 mel



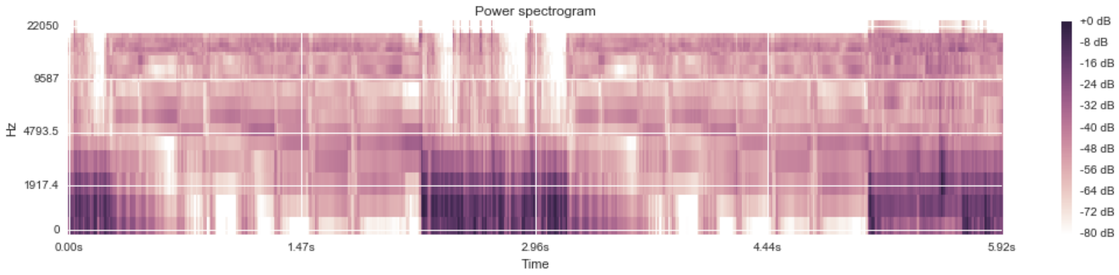
Bark Scale



3.4.5 Spectral Masking

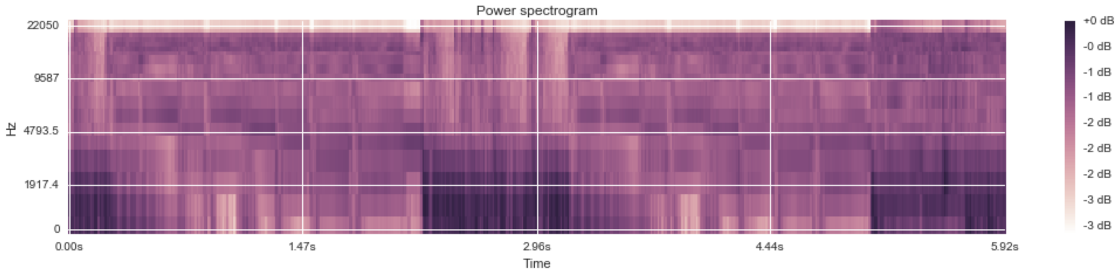
- Occlusion of a quiet sound by a louder sound when both sounds are present simultaneously and have similar frequencies
    - Simultaneous masking: two sounds active simultaneously
    - Post-masking: a sound closely following it (100-200 ms)
    - Pre-masking: a sound preceding it (usually neglected, only measured during about 20ms)
- Spreading function defining the influence of the j-th critical band on the i-th

For Example: * A quiet sound is masked by a loud sound that has the same frequency * if both are played simultaneously * if the are played very close to each other



3.4.6 Map to Decibel Scale

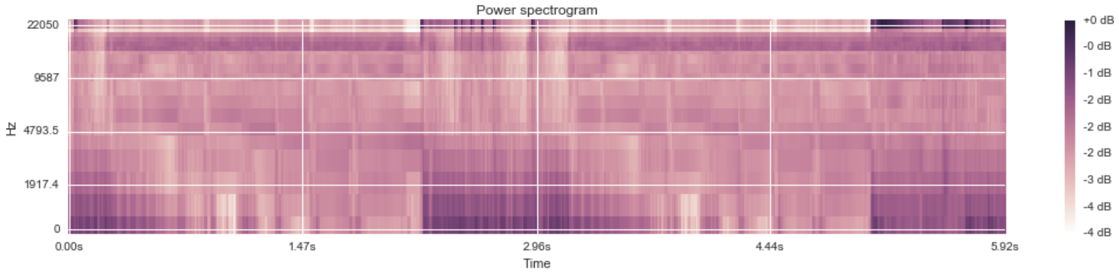
Transform the energy values on the critical bands into decibel scale

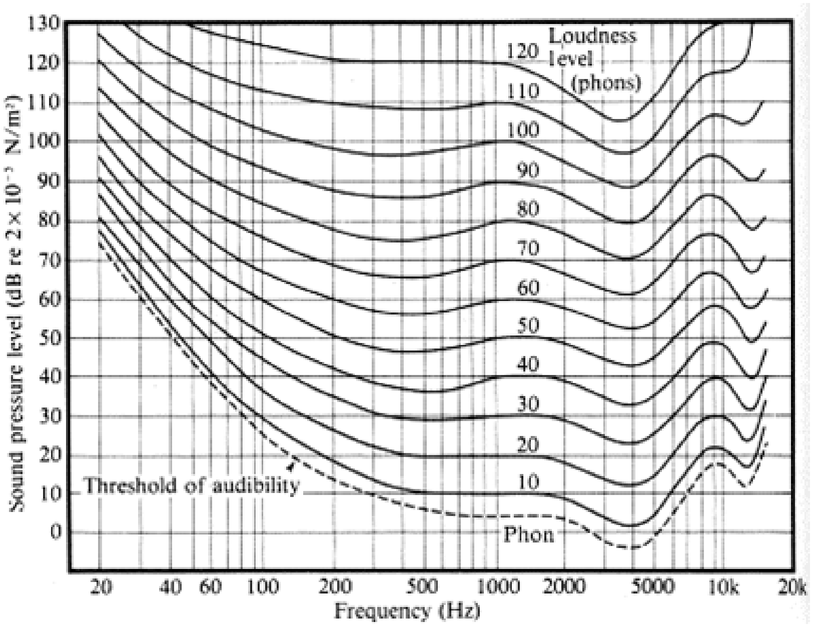


3.4.7 Transform to Phon Scale

Equal loudness curves (Phon)

- Represents the relationship between the sound pressure level in decibel and the perceived hearing sensation.
- This relationship is not linear and depends on the frequency of the tone
- Centered around 1khz
- equal loudness contours for 3, 20, 40, 60, 80, 100 phon
- To perceive a 40Hz tone with the same sensation sf loudness, 5 times more sound presure is required.





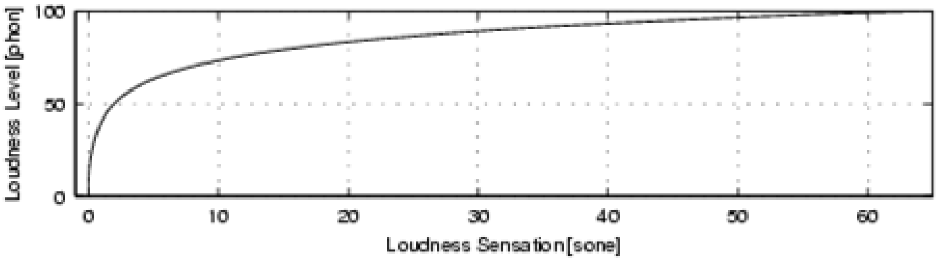
Equal loudness curves

3.4.8 Transform to Sone Scale

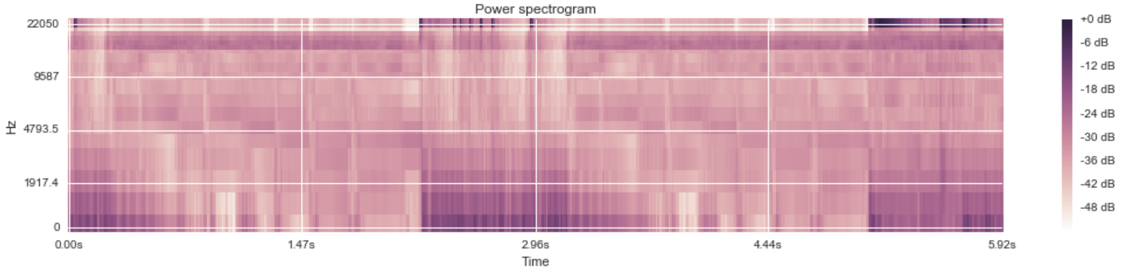
Sone Transformation

- Perceived loudness measured in Phon does not increase linearly
- Transformation into Sone
- Up to 40 phon slow increase in perceived loudness, then drastic increase
- Higher sensibility for certain loudness differences

Sone 1 2 4 8 16 32 64
Phon 40 50 60 70 80 90 100



Sone Scale

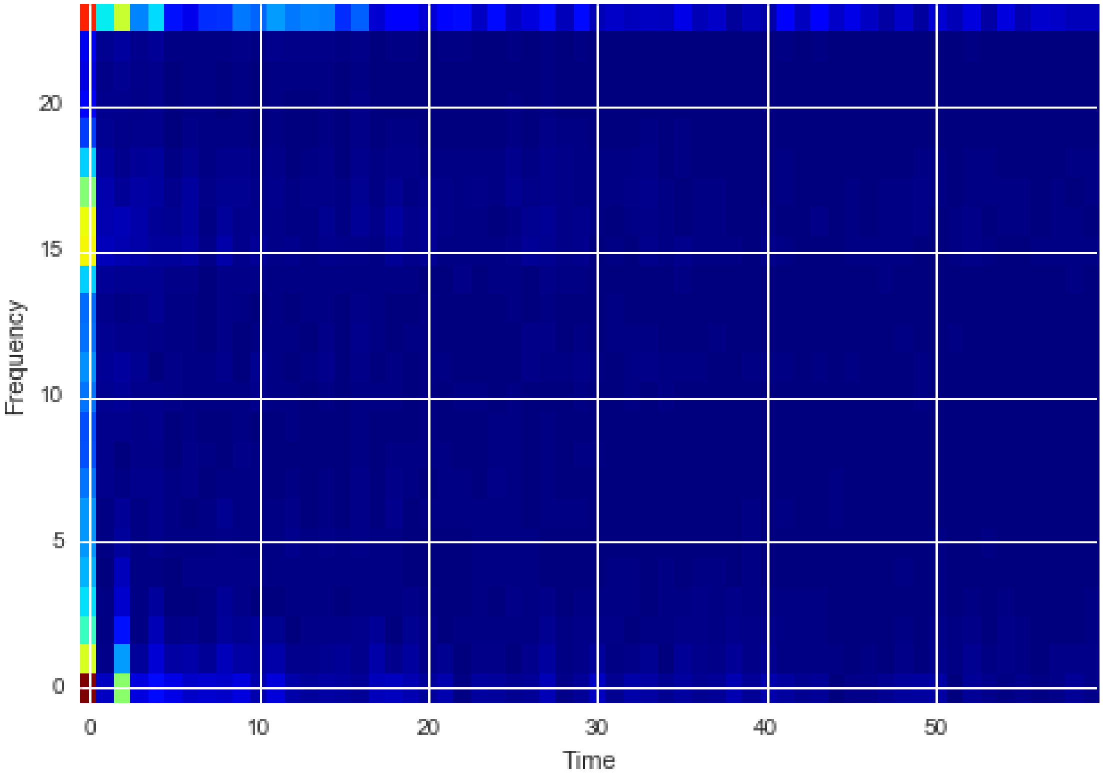


3.5 Statistical Spectrum Descriptors

The Sonogram is calculated as in the first part of the Rhythm Patterns calculation. According to the occurrence of beats or other rhythmic variation of energy on a specific critical band, statistical measures are able to describe the audio content. Our goal is to describe the rhythmic content of a piece of audio by computing the following statistical moments on the Sonogram values of each of the critical bands: mean, median, variance, skewness, kurtosis, min- and max-value

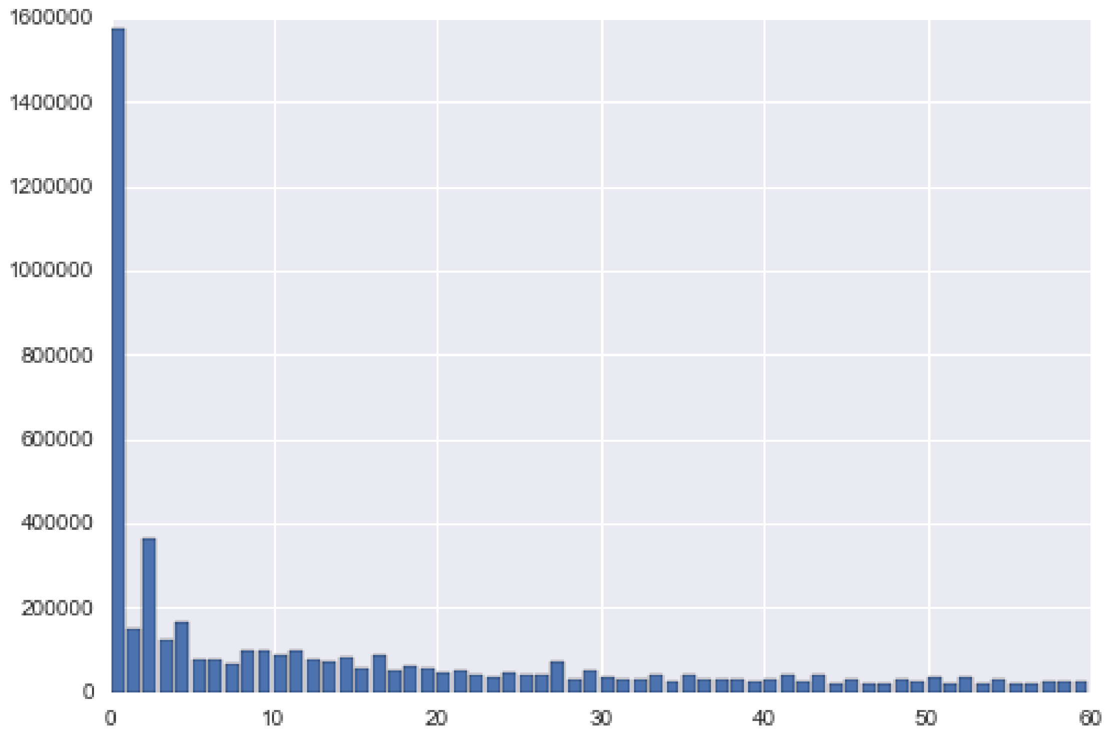
3.6 Rhythm Patterns

Calculate fluctuation patterns from scaled spectrum



3.7 Rhythm Histograms

The Rhythm Histogram features we use are a descriptor for general rhythmics in an audio document. Contrary to the Rhythm Patterns and the Statistical Spectrum Descriptor, information is not stored per critical band. Rather, the magnitudes of each modulation frequency bin of all critical bands are summed up, to form a histogram of 'rhythmic energy' per modulation frequency. The histogram contains 60 bins which reflect modulation frequency between 0 and 10 Hz. For a given piece of audio, the Rhythm Histogram feature set is calculated by taking the median of the histograms of every 6 second segment processed.



3.8 Modulation Variance Descriptors

This descriptor measures variations over the critical frequency bands for a specific modulation frequency (derived from a rhythm pattern).

Considering a rhythm pattern, i.e. a matrix representing the amplitudes of 60 modulation frequencies on 24 critical bands, an MVD vector is derived by computing statistical measures (mean, median, variance, skewness, kurtosis, min and max) for each modulation frequency over the 24 bands. A vector is computed for each of the 60 modulation frequencies. Then, an MVD descriptor for an audio file is computed by the mean of multiple MVDs from the audio file's segments, leading to a 420-dimensional vector.

Chapter 4

Music Similarity Retrieval

Load extracted features from prepared files

4.0.1 Feature space pre-processing

Normalization * to reduce the influence of outliers * Standard Score - Zero Mean and Unit Variance

$$z = \frac{x - \mu}{\sigma}$$

4.1 Nearest Neighbor Search

- Music Similarity based on vector similarity
- Requires a similarity metric or measure:
  - Manhattan Distance (L1)

    $$d(p, q) = \sum_{i=1}^{n} |q_i - p_i|$$

  - Euclidean Distance (L2)

    $$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

- for distance measures vectors with smaller distances are more similar
- rank results by ascending distance or descending similarity

```
Seed Song: jazz
========================================

Rank    L1                   L2
-----+--------------------+-------------
   1: classical            classical
   2: classical            classical
   3: classical            classical
   4: classical            classical
   5: classical            classical
   6: classical            classical
   7: classical            classical
   8: classical            classical
```

```
  9: classical            classical
 10: classical            classical
 20: classical            reggae
 50: disco                jazz
100: classical            rock
200: country              hiphop
500: jazz                 blues
```

Chapter 5

Music Classification

5.1 Genre Classification

Partition the data set. Create a subset that can be used to train the classifier and another to evaluate it.

Create the classifier. In this example we use a K-Nearest Neighbor classifier with k=1.

Train the classifier on the feature values with the previously created partition.

Predict genre label for all tracks of the test set based on the feature vector

```
blues      => blues
blues      => jazz
classical  => classical
classical  => disco
classical  => classical
classical  => country
country    => country
country    => rock
country    => rock
country    => blues
hiphop     => metal
hiphop     => reggae
jazz       => jazz
jazz       => jazz
metal      => metal
pop        => disco
reggae     => reggae
rock       => pop
rock       => metal
rock       => country
```

5.1.1 Evaluate Classification Experiments

Classification Metrics

Confusion Matrix

Cross Validation

Compare performance of all features

Run classification experiments

Display summary of results measured in mean classification accuracy.

Chapter 6

References

- Thomas Lidy, Andreas Rauber. Evaluation of Feature Extractors and Psycho-acoustic Transformations for Music Genre Classification. Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR 2005), pp. 34-41, London, UK, September 11-15, 2005. PDF
- A. Rauber, E. Pampalk, D. Merkl. The SOM-enhanced JukeBox: Organization and Visualization of Music Collections based on Perceptual Models. In: Journal of New Music Research (JNMR), 32(2):193-210, Swets and Zeitlinger, June 2003. Abstract
- A. Rauber, and M. Frühwirth. Automatically Analyzing and Organizing Music Archives. In: Proceedings of the 5. European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2001), Sept. 4-8 2001, Darmstadt, Germany, Springer Lecture Notes in Computer Science, Springer, 2001. PDF

.. [1] S.B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Trans. Acoustics. Speech, Signal Proc. ASSP-28 (4): 357-366, August 1980.“”"

Chapter 7

Download

Software for the extraction of Rhythm Patterns, Statistical Spectrum Descriptors, Rhythm Histograms, Modulation Frequency Variance Descriptor, Temporal Statistical Spectrum Descriptors and Temporal Rhythm Histograms is available from the download section.