Classifying motion picture audio

Eirik Gustavsen

Master's Thesis
Master of Science in Media Technology
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College, 2007

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Faculty of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Abstract

Classification of the audio track, of a motion picture, into traditional audio classes is a challenging task. The reason for this is the amount of mixed content. Speech has often background music or environmental sounds, and music has often background environmental sounds or speech. Traditional methods for separating clear audio classes have limited performance on mixed audio content. New methods and tools for automatic classification of this type of audio are therefore needed. This project investigates combinations of low level descriptors, dimensionality reduction by PCA and classification by KNN. A feature set consisting of Audio Power (AP), Audio Wave Form (AWF), Root-Mean-Square (RMS), Short Time Energy (STE), Low Short-Time Energy Ratio, Zero-Crossing Rate (ZCR), High Zero-Crossing Rate Ratio (HZCRR) in the time domain and Audio Spectrum Centroid (ASC), Fundamental Frequency (FuF), Mel, Frequency Cepstral Coefficients (MFCC), Spectrum Flux (SF) in the frequency domain is extracted on 30ms windows and integrated over a 1.2 second frame to yield a 23-dimmensional feature vector. The most suitable combination for separating speech from background music were; AP, ASC, AWF, STE, RMS, SF, fourth MFCC, AP(min scalar), ZCR(min scalar) and STE(min scalar). The combination has a majority of descriptors from the time domain. Most suitable combination of LLDs to separate speech with background environmental sounds from clear environmental sounds was found to be the 1_{th}, 4_{th}, 5_{th}, 6_{th}, 8_{th} and 9_{th} Mel Frequency cepstran coefficients. MFCC is in the frequency domain. Best results were achieved when the PCA returned 3 dimensions, and when the KNN classified the samples based on the 4 closest neighbors. The results from testing different mixtures of speech and music, to find the boundary where speech with background music no longer is categorized as speech, showed that the music signal had to be minimum 8 dB below the speech signal to be classified as speech. Some of the low level descriptors which traditionally performs well when separating clear classes, performed poorly in the experiments with mixed classes. Especially ZCR and FuF failed to separate speech with background music from clear music. A final experiment classifies the audio track from the motion picture 'Groundhog Day'. First 80 percent of the movie is used for training of the KNN, and the remaining 20 percent was classified. After post processing the result was 76.9 percent correctly classified. A table of content was then based on this classification.

Preface

This Master's thesis was carried out at Gjovik University College, Department of Computer Science and Media Technology, in the period from January 2007 to july 2007, under the supervision of Dr. Tech. Faouzi Alaya Cheikh. I would like to thank Faouzi Alaya Cheikh for guidance, expertise and encouragement during the thesis work.

Eirik Gustavsen,
June 30, 2007

Contents

List of Figures

List of Tables

1 Introduction

These days there is an enormous amount of video material available to everyone. The challenge for most people is to find out something about the content just by reading the tag lines, plot of the movie, ratings and reviews. The plot outline of a movie tells something about the story, not how the story is presented. The ratings are often based on the presence of different features, like violence, bad language, nudity and drugs. And reviews are based on one individual's opinion of the movie. It is seldom possible to find out how much violence, dialog, music and silence there is in a movie. There is also little information about what genere of music, what type of dialog and if it is a noisy or quiet sound setting for the different scenes. Much of this could be helpful to know for deciding to watch a movie or not. Or perhaps, decide if it is suitable for a child to watch. Automatic content analysis, and categorizations, based entirely on video have sometimes failed just because they are based on the visual content. In such systems dialogs may pass undetected because the speakers are not visible, nudity filters may block documentaries about Indians in the rain forest and humor may be mistaken for violence. In these instances, audio analysis could have contributed to categorize the content more precisely. Because the different audio classes often are mixed together in motion picture audio, traditionally methods for separating audio classes will in most cases perform poorly. New methods will therefore be needed to classify these mixed classes correctly.

1.1 Thesis

It is possible to automatically create a table of contents of a video, based on its audio track only, to describe its audiovisual content.

1.2 State of the Art

Audio is sometimes more difficult to categorize than video. This is because two audio samples, that do not sound like at all, can be in the same category (e.g. a large jumbo jet sounds different from a small single-engine piston, but both are categorized as an airplane. And, a male and female voice sound different, but both are categorized as human voice). It is therefore a challenge to make algorithms that simulates the categorization done by the humans. In the last decades, automatic content analysis of motion picture has mostly been focusing on image and video research. But, there have been a few attempts to analyse motion picture and television audio [30] [31] too. Audio can have equal amount of semantic information as video, and some times more [24] [25] [26] [27]. Examples of this are for instance a sequence or a scene of a car passing by in the dark. In the video we can see the headlights and the movement, and then assume that it is a car. In the audio track we can hear that it is a big V8 engine, that the exhaust pipe is broken and that the wheel bearings are dry. But, of course, some times it is the other way around.

The first step in categorization of audio is to separate music, speech and background noise. Music is probably the easiest to detect since music contains features that other sound usually do not have [23]. The second step is to separate speech from the background sounds. One problem with analysing audio from motion picture is that the different audio categories often are mixed together. This causes problems for the algorithms designed to classify the features extracted from audio with little noise. Some extra features have to be added to get good classification under these conditions [36]. There are many examples of clear speech analysis [28] [29], and also speech/music classification [15] [22] [32] [33] [34] [42].

Other feature that is important to detect is the sound objects [35]. These objects are short segments of sounds that can give the listener vital information. This is sounds like a shot, a dog barking, a tyre skidding on asphalt etc. Much research has been directed towards classifying a short sound clip into one of a pre-specified set of categories [37] [38] [39] [40].

1.2.1 Content-based segmentation

Before any retrieval of audio content can be done, we have to structure the audio. The first classification of the audio should distinguish between speech, music, silence and other sound sequences. This is due to the fundamental differences between these classes. The next step could be to determine syllables, words or sentences in speech and notes, bars or themes boundaries for music [22].

1.2.2 Silence

The way humans determine silence is relative. Complete silence (0 dB) is very seldom in a natural environment, but can be found in digitalized audio. The silence level must therefore be calculated for every different sound sample and be adjusted by an adaptive threshold along the timeline.

1.2.3 Music

Music can be recognized by the frequency spectrum that it covers. Most of the time music will contain a wide range of frequencies, ranging from 16Hz to 20kHz, and a pitch over a span of six octaves. Because environmental sounds (noise) also often have the same range, the frequency range alone cannot distinguish music form other sounds. One way to separate music from other sounds is to analyze the spectrum for orderliness [1]. Tones and their characteristic overtone pattern do not appear in environmental sounds.

1.2.4 Speech

Speech has a more limited frequency range than music, usually from 100Hz to 8kHz and a pitch that spans three octaves. Speech alters between sound and silence in a syllabic rhythm. The vowels duration is some what very regular. Tone duration could therefore be a good discriminator [24].

1.3 Project Description

The primary goal for this thesis is to combine algorithms in such way that the following questions can be answered with a satisfying certainty:

How best to classify audio segments into for instance: Silence, Music, Speech and Noise.

How to create a table of contents of the video, based on the audio track only, to describe the video content.

1.4 Project Overview

Chapter 2 - Low Level Discriptors

This chapter describes the state of the art regarding the features used I this project. The low level descriptors in the time and frequency domain is explained together with a introduction to the scalable series.

Chapter 3 - Audio Classification

Presents the classification method KNN and the dimensional reduction by the PCA are described.

Chapter 4 - Experimental Setup

Presents the most suitable combinations of low level descriptors to classify audio samples, containing mixtures of audio classes.

Chapter 5 - Experimental Result

Presents the result from the classification of the audio track from the movie 'Groundhog Day'

Chapter 6 - Conclusion and Future Work

Gives the conclusion of the project and points out areas of future work

2 Low Level Descriptors

2.1 Introduction

2.1.1 Scalable Series

Scalable series description is a way of representing a series of LLD features extracted from sound frames at regular time interval [7]. By decomposing each serie of original sample into consecutive sub-sequences, and summarizing them into a single *scaled sample*. Figure 1 shows the scaling process and the resulting scalable series description.

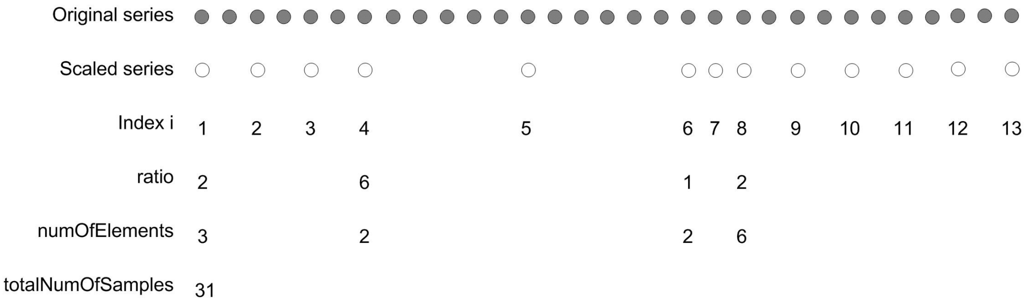

Figure 1: Illustration of the scaling process and the resulting scalable series description

The *i* is the index of the scaled series, the filled circles are the original series and the open circles are the summarizer samples of the scale series. The *scale ratio* indicates the number of original samples it describes. The *numOfElements* indicates the number of consecutive elements in a sequence of scaled samples that has the same *scale ratio*. The last attribute is the *totalNumOfSamples* which indicates the total number of samples in the original series.

MPEG-7

The MPEG-7 Framework has two distinct types of scalable series defined in it's standard. These represent series of scalars and series of vectors, both types are inherited from the scalable series description. In this thesis the same notation and attributes will be used as in the MPEG-7 standard regarding the scalable series. In the following sections, these two types of scalable series will be presented in detail [7].

Series of Scalars

Any temporal series of scalar LLDs can be represented by the MPEG-7 *SeriesOfScalar* descriptor. The attributes of a *SeriesOfScalar* description are:

- *Raw:* Contains usually the original series of scalars whitout any scaling operation applied. Only used if the *Scaling* flag is absent.
- *Weight:* This is an optional serie of weights. Each weight corresponds to a sample in the original series, if this attribute is pressent. A way to use these parameters, is to control scaling.

- *Min, Max, Mean:* These coefficients represent a series of samples from the original series. *Min* is the minimum value, *Max* is the maximum and *Mean* is the mean sample value in the original sample series. The original samples are average by aritmetic mean, taking the sample weights into account if the *Weight* attribute is present.
- *Variance:* The elements in this real-valued vector correspond to the variance computed within the corresponding groups of the original samples. If the *weight* attribute is present, the computation may take the sample weights into account. This attribute is absent if the *Raw* element is present.
- *Random:* This vector contains samples taken randomly within each group of original samples. This attribute is absent if the *Raw* element is present.
- *First:* This vector contains the first sample within each group of original samples. This attribute is absent if the *Raw* element is present.
- *Last:* This vector contains the last sample within each group of original samples. This attribute is absent if the *Raw* element is present.

These different attributes allow us to summarize any series of scalar features. This type of description allows scalability, so that a scaled series can be derived indifferently from an original series or from a previously scaled *SeriesOfScalar*. Initially, a series of scalar LLD features is stored in the Raw vector. Each element $Raw(l)(0 \leq l \leq L-1)$ contains the value of the scalar feature extracted from the lth frame of the signal. Optionally, the $Weight$ series may contain the weight $W(l)$ associated to each $Raw(l)$ feature.

A new *SeriesOfScalar* is generated by grouping the original samples (see Figure ref) and calculating the abovementioned attributes when a scaling operation is performed. The Raw attribute is absent in the scaled series descriptor. We can assume that the ith scaled sample stands for the samples $Raw(l)$ contained between $l = lLo(i)$ and $l = lHi(i)$. The corresponding Min and Max values are then defined as:

$$Min(i) = min_{l=lLo(i)}^{lHi(i)} Raw(l) \;\; and \;\; Max(i) = max_{l=lLo(i)}^{lHi(i)} Raw(l) \tag{2.1}$$

The $Mean$ value is given by:

$$Mean(i) = \frac{1}{ratio} \sum_{l=LLo(i)}^{lHi(i)} Raw(l) \tag{2.2}$$

if no sample weight $W(l)$ are specified in $Weight$. If weights are present, the $Mean$ value is computed as:

$$Mean(i) = \sum_{l=LLo(i)}^{lHi(i)} W(l)Raw(l) / \sum_{l=LLo(i)}^{lHi(i)} W(l) \tag{2.3}$$

In the same way, there are two computational methods for $Variance$ depending on wether the original sample weights are absent:

$$Variance(i) = \frac{1}{ratio} \sum_{l=LLo(i)}^{lHi(i)} [Raw(l) - Mean(i)]^2 \tag{2.4}$$

or present:

$$\mathrm{Variance}(i) = \sum_{l=LLo(i)}^{lHi(i)} W(l)\,[\mathrm{Raw}(l) - \mathrm{Mean}(i)]^2 \,/ \sum_{l=LLo(i)}^{lHi(i)} W(l) \tag{2.5}$$

Finally, the weights $W(i)$ of the new scaled samples are computed, if necessary, as:

$$W(i) = \frac{1}{\mathrm{ratio}} \sum_{l=LLo(i)}^{lHi(i)} W(l) \tag{2.6}$$

Series of Vectors

When the LLDs consist of multi-dimentional vectors, the MPEG-7 *SeriesOfVectors* descriptor represents the temporal series of feature vectors. The attributes of a *SeriesOfVectors* description are:

- *Weight* This is an optional series of weights. Each weight corresponds to a sample in the original series, if this attribute is pressent. A way to use these parameters, is to control scaling in the same way as for the *SeriesOfScalars*.
- *Min, Max, Mean:* are three real-valued matrices where the number of rows is equal to the sum of *numOfElements* over the scaled series. The number of columns is equal to *vectorSize* and each row characterizes a scaled vector. For a given scaled vector, a *Min, Max and Mean* row vector is extracted from the corresponding group of vectors in the original series.The row vector *Min, Max and Mean* contains the minimum, maximum and mean coeffecients observed among the original vectors. These attributes are absent if the *Raw* element is present.
- *Variance:* The variance vectors series size is set to *vectorSize*. Each vector corresponds to a scaled vector, and its coefficients are equal to the variance computed within the corresponding group of original vectors. The computation may take the sample of weights into account if the *Weight* attribute is present. This attribute is absent if the *Raw* element is present.
- *Covariance:* This is a series of covariance matrices which is represented as a three dimentional matrix. The number of rows is equal to the sum of *numOfElements* parameters over the scaled series; the number of columns and the number of pages are both equal to *vectorSize*. Each row is a covariance matrix describing a given scaled vector. It is estimated from the corresponding group of original vectors. This attribute is absent if the *Raw* element is present.
- *VarianceSummed* This is an series of summed variance coeffecients and each coefficient corresponds to a scaled vector. For a given scaled vector, it is obtaind by summing the elements of the corresponding *Variance* vector. This attribute is absent if the *Raw* element is present.
- *MaxSqDist:* This is a series of maximum squared distance coefficients. For each scaled vector, an MSD coefficient is estimated, representing an upper bound of the distance between the corresponding group of original vectors and their mean. This attribute is absent if the *Raw* element is present.
- *Random:* This vector contains samples taken randomly from each group of original samples. This attribute is absent if the *Raw* element is present.

- *First:* This vector contains the first sample within each group of original samples. This attribute is absent if the *Raw* element is present.
- *Last:* This vector contains the last sample within each group of original samples. This attribute is absent if the *Raw* element is present.

Binary Series

The MPEG-7 standard also defines a binary form of the *SeriesOfScalar* and the *SeriesOfVectors* descriptors. These are called *SeriesOfScalarBinary* and *SeriesOfVectorsBinary*, and are used to instantiate series of scalars or vectors with a uniform power-of-2 *ratio*.

2.2 Time Domain

When a signal is analysed with respect to time, the term Time Domain is used. The signal will have a value for each various discrete time point.

2.2.1 Basic Parameters

Notation used for input audio signal:

- n is the index of time samples
- $s(n)$ is the input digital audio signal
- F_s is the sampling rate of $s(n)$

Notation used for the frames:

- l is the index of the time frames
- $hopSize$ is the time interval between two successive time frames
- N_{hop} denotes the integer number of time samples corresponding to $hopSize$
- L_w is the length of a time frame
- N_w denotes the integer number of time samples corresponding to L_w
- L is the total number of time frames in s(n)

In figure 2 the different notations are portrayed. The size of *hopSize* and Lw that is chosen depends on what kind of descriptors to extract. The *hopSize* is usually selected to be a integer multiple or divider of 10ms (its default value).

2.2.2 Audio Power (AP)

To calculate the Audio Power (AP) the audio signal instantaneous power has to be temporally smoothed. The coefficients are the average square of waveform values $s(n)$ within successive non-overlapping frames ($L_w = hopSize$). The lth frame of the signal can be described like this.

$$AP(l) = \frac{1}{N_{hop}} \sum_{n=0}^{N_{hop}-1} |s(n) + lN_{hop}|^2 \quad (0 \leq l \leq L-1) \tag{2.7}$$

L is the total number of frames. The AP gives a fast representation of the spectrogram

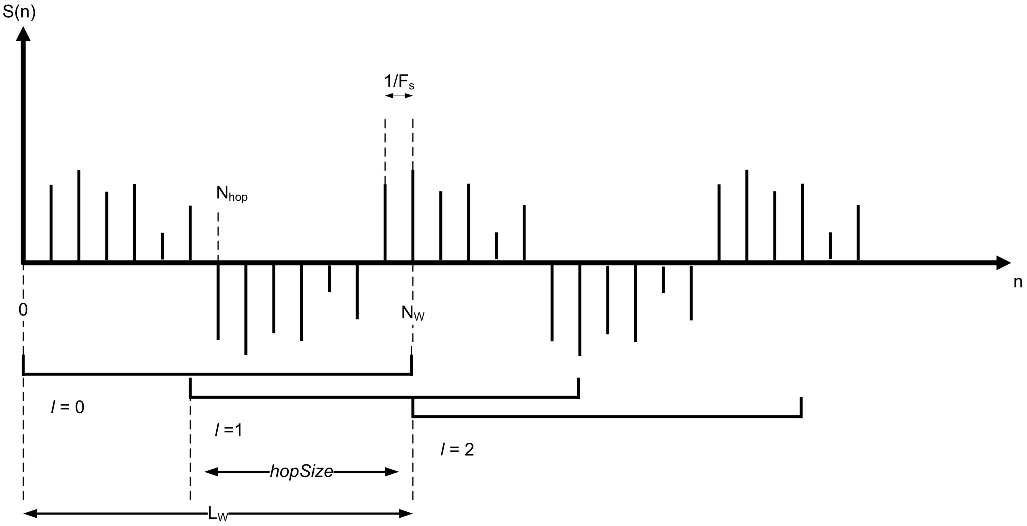

Figure 2: Illustration of the notations used in Time Domain

of a signal, and makes it possible to measure the evolution of the amplitude as a function of time. An example of a wave signal described by AP is given in figure 3. First half is speech and second is music.

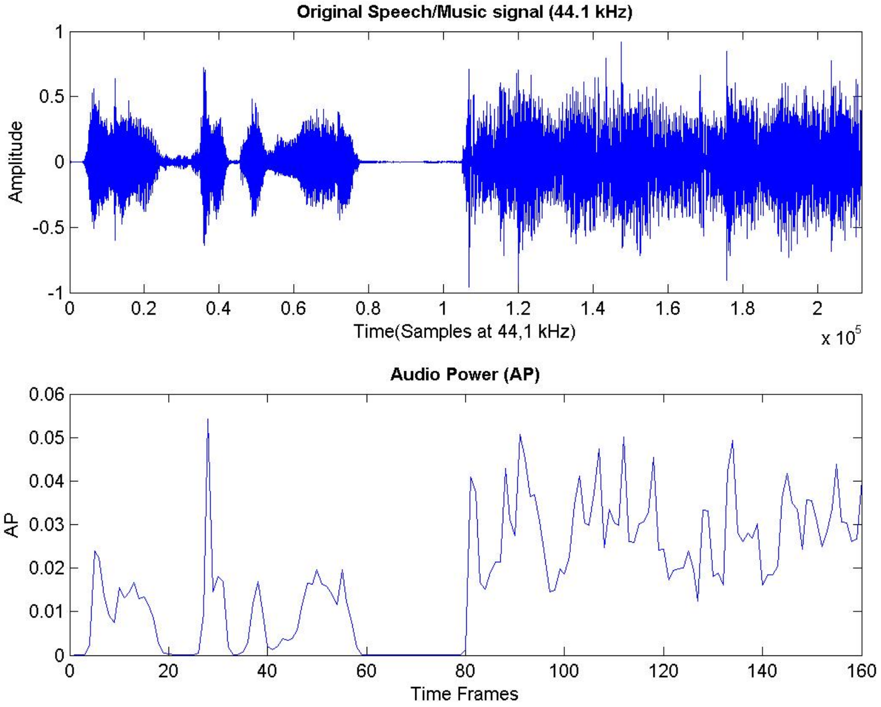

Figure 3: Audio Power description of a speech/music signal (first half speech, second music, 30ms hopSize)

2.2.3 Audio Wave Form (AWF)

AWF descriptor consists of the resulting temporal series of the lower limit and the upper limit of the amplitude in a frame. The AWFs temporal resolution is given by the *hopSize* parameters. AWF has a low storage cost and provides a simple way to display or compare waveforms. Figure 4 shows the waveform in it's original form and in the form described by the AWF.

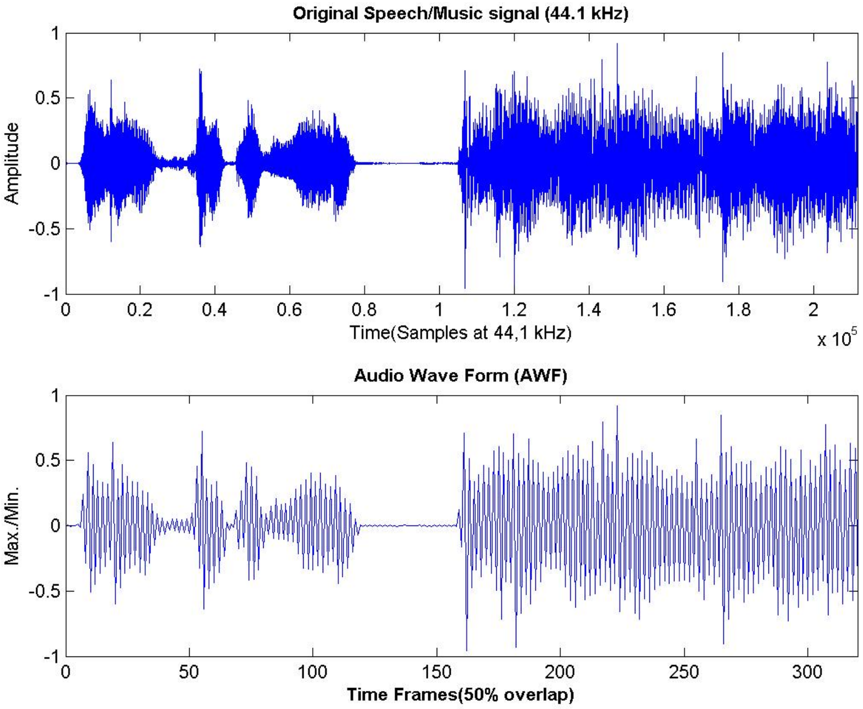

Figure 4: Audio Wave Form description of a speech/music signal (first half speech, second music, 30ms hopSize)

2.2.4 Root-Mean-Square (RMS)

RMS is often used as a measure of loudness and is a unique feature to segmentation, even if it is closely related to short time energy. RMS is computationally inexpensive, easy to implement and used in most audio analysis and genre classification approches [9]. It is mostly used as a part of a low-level descriptor set [10], but has also been used to analyze different musical aspects. RMS values can be used for estimation of tempo and beat, which approximate the time envelope [11]. Because RMS is linked to the perceived intensity, it can be used for mood detection. But this relation is not captured by the RMS without first splitting the signal into several frequency bands [12]. The coefficients are the square root of the mean of the squares of waveform values $s(n)$ within successive non-overlapping frames ($L_w = hopSize$). The lth frame of the signal can be described like this.

$$\mathrm{RMS}(l) = 10\log_{10}\sqrt{\frac{1}{N_{hop}}\sum_{n=0}^{N_{hop}-1} s^2(n)} \quad (0 \leq l \leq L-1) \tag{2.8}$$

where L is the total number of time frames. Figure 5 describes RMS for a audio signal, where the first half is speech and the second half is music. The RMS of speech tend to have larger variation than of music.

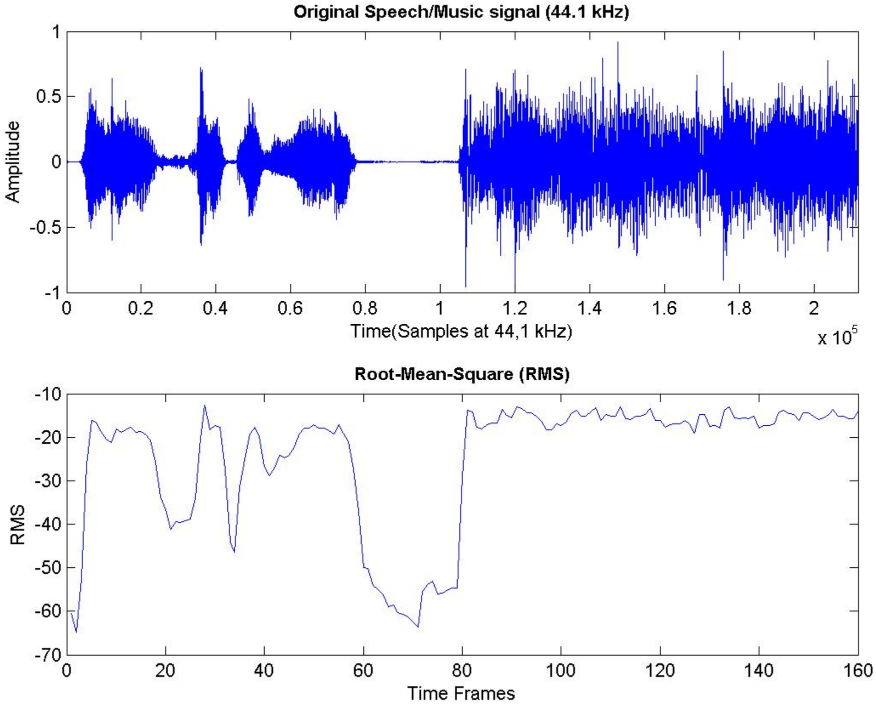

Figure 5: Root-Mean-Square description of a speech/music signal (first half speech, second music, 30ms hopSize)

2.2.5 Short Time Energy (STE)

Short Time Energy (STE) is an often used and very simple low level descriptor. STE is defined as the total squared energy in a signal $s(n)$ within successive non-overlapping frames ($L_w = hopSize$). The lth frame of the signal can be described like this.

$$\mathrm{STE}(l) = \sum_{n=0}^{N_{hop}-1} s^2(n) \quad (0 \leq l \leq L-1) \tag{2.9}$$

With N_{hop} as the length of the frame and L is the total number of time frames. Speech is composed of syllables and pauses between them. Relatively strong STE-values indicate parts in the signal where voice is present. Unvoiced and silence parts will give much lower STE-values. Because of this, a speech audio sample will give large variations in the STE-values. Music has much shorter pauses, or no pauses, and will therefore have a more constant STE-level. This can be seen in figure 6 which is a sample of 2.4 seconds of speech followed by 2.4 seconds of music. Music also often carries more total energy in its signal, so its STE-level is usually higher than the speech STE-level. Because the variation in the STE-values, when speech is present, is relatively easy to distinguish from both steady high valued STE in music and steady low valued STE in silence, STE is usually a good indicator for speech presence in an audio signal.

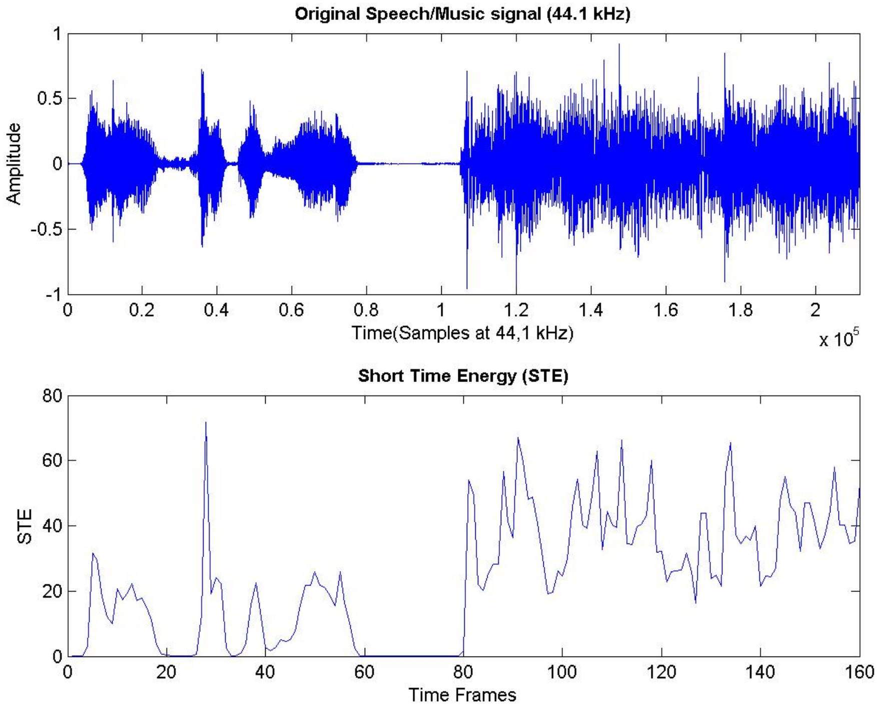

Figure 6: Short Time Energy description of a speech/music signal (first half speech, second music, 30ms hopSize)

2.2.6 Low Short-Time Energy Ratio

The Low Short-time Energy Ratio (LSTER) is defined as the ratio of the number of frames whose STE are less than 0.5 the average short-time energy in the window [2].

$$LSTER = \frac{1}{2N} \sum_{n=0}^{N-1} [sgn(0.5 * avSTE - STE(l)) + 1] \tag{2.10}$$

$$avSTE = \frac{1}{N} \sum_{n=0}^{N-1} STE(l) \tag{2.11}$$

Where N is the total number of frames, $STE(l)$ is the short time energy at the lth frame and avSTE is the average in the window. Speech usually has more silence frames than music, LSTER values for speech are therefore usually higher than the values for music. Because of this, LSTER is a good feature for discriminating speech and music signals.

2.2.7 Zero-Crossing Rate (ZCR)

Zero-Crossing Rate is a much used low level descriptor in a wide range of audio applications. The definition of ZCR is the number of sign changes in signal $s(n)$ within successive non-overlapping frames ($L_w = hopSize$) devided by the length of the frame.

$$ZCR(l) = \frac{1}{N_{hop}} \sum_{n=0}^{N_{hop}-1} |sgn\,[s(n)] - sgn\,[s(n-1)]| \quad (0 \leq l \leq L-1) \tag{2.12}$$

where L is the total number of time frames.

ZCR gives a good indication of the dominant frequency, and therefore correlates well with the spectrum centroide of a signal. Speech is mostly constructed of voiced parts and pauses in between. A pitch can be found in the voiced periodes. The silent parts can be seen as noice. Voiced parts will therefore give relatively small values and unvoiced larger values. The ZCR signal will show large variations for speech and smaller variations for music. This can be seen in figure 7. The first half of the signal is speech, and the second half is music.

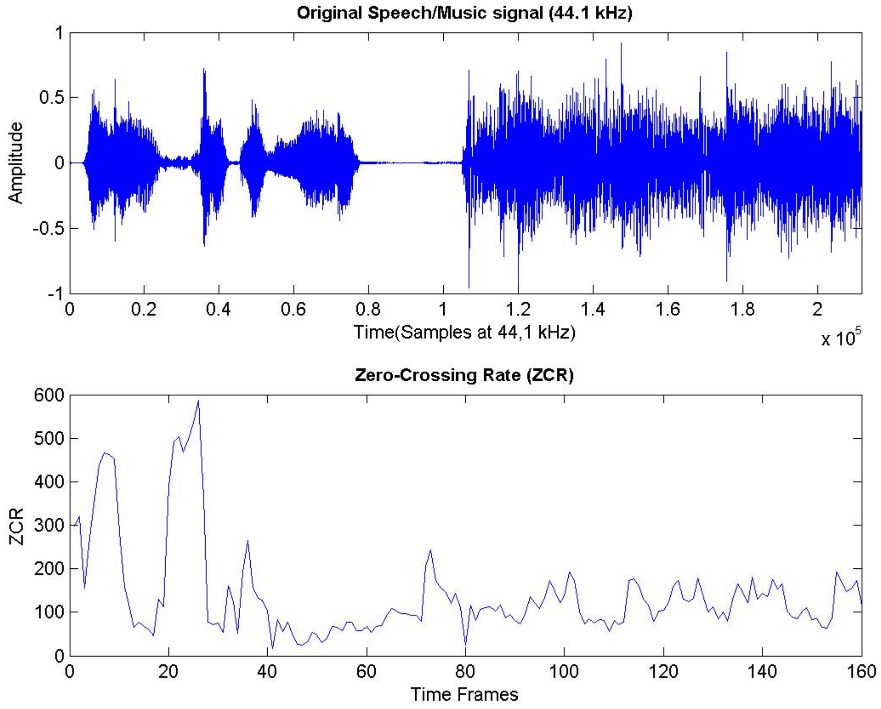

Figure 7: Zero-Crossing Rate description of a speech and music signal (first half speech, second music, 30ms hopSize)

2.2.8 High Zero-Crossing Rate Ratio (HZCRR)

Research has shown that the variation of the ZCR is a better discriminator than the actual ZCR value [23]. They developed the HZCRR (High Zero-Crossing Rate Ratio). The HZCRR is the ratio of the number of frames whos ZCR are above 1.5-folds the average ZCR in a 1-s window.

$$HZCRR = \frac{1}{2L} \sum_{l=0}^{L-1} [sgn(ZCR(l) - 1.5avZCR) + 1] \tag{2.13}$$

$$avZCR = \frac{1}{L}\sum_{l=0}^{L-1} ZCR(l) \tag{2.14}$$

Speech is constructed out of syllables and silence. Therefore, most of the time, speech will have higher HZCRR than Music [2].

2.3 Frequency Domain

When a signal is analysed with respect to frequency, the term Frequency Domain is used. The values extracted from the frequency domain consist information of which frequencies is present in the signal analysed.

2.3.1 Basic Parameters

The following notation will be used in the frequency domain:

- k is the frequency bin index
- $S_i(k)$ is the spectrum extracted from the lth frame of $s(n)$
- $P_i(k)$ is the power spectrum extracted from the lth frame of $s(n)$

The following is based on squared magnitudes of discrete Fourier transform (DTF) coefficients. After multiplying the frames with a windowing function $w(n)$ (Hamming window), the DFT is applied as:

$$S_l(k) = \sum_{n=0}^{N_{FT}-1} s(n + lN_{hop})w(n)exp^{(-j\frac{2\pi nk}{N_{FT}})} \quad (0 \le l \le L; 0 \le k \le N_{FT} - 1) \tag{2.15}$$

where N_{FT} is the size of the DFT ($N_{FT} \geq N_w$). In general, a fast Fourier transform (FFT) algorithm is used and N_{FT} is the power of 2 just larger than N_w (the enlarged frame is then padded with zeros). According to Parseval's theorem, the average power of the signal in the lth analysis window can be written as:

$$\bar{P_l} = \frac{1}{E_w}\sum_{n=0}^{N_w-1} |s(n + lN_{hop})w(n)|^2 = \frac{1}{N_{FT}E_w}\sum_{n=0}^{N_w-1} |S_l(k)|^2 \tag{2.16}$$

where the window normalization sactor E_w is defined as the energy of $w(n)$:

$$E_w = \sum_{n=0}^{N_w-1} |w(n)|^2 \tag{2.17}$$

The power spectrum $P_l(k)$ of the lth frame is defined as the squared magnitude of the DFT spectrum $S_l(k)$. Since the signal spectrum is symmetric around the Nyquist frequency $F_s/2$, it is possible to consider the first half of the power spectrum only ($0 \le k \le N_{FT}/2$) without losing any information. In order to ensure that the sum of all power coefficients equates to the average power defined in Equation 2.16, each coefficient can be normalized in the following way:

$$P_l(k) = \frac{1}{N_{FT}E_w} |S_l(k)|^2 \quad (for \ k = 0 \ and \ k = \frac{N_{FT}}{2}) \tag{2.18}$$

$$P_l(k) = 2\frac{1}{N_{FT}E_w} |S_l(k)|^2 \quad (for \ 0 < k < \frac{N_{FT}}{2}) \tag{2.19}$$

In a FFT spectrum, the discrete frequencies corresponding to bin indexes k are:

$$f(k) = k\Delta F \quad (0 \leq k \leq N_{FT}/2) \tag{2.20}$$

where $\Delta F = F_s/N_{FT}$ is the frequency interval between to successive FFT bins, Inverting the preceding equation, we can map any frequency in the range $[0, F_s/2]$ to a discrete bin in $\{0, 1....N_{FT}/2\}$:

$$k = \mathrm{round}(f/\Delta F) \quad (0 \leq f \leq F_s/2) \tag{2.21}$$

2.3.2 Audio Spectrum Centroid (ASC)

The ASC gives the centre of gravity of a log-frequency power spectrum. The ASC is used as a measure for sound sharpness or brightness. The high frequency part is is primarily measured because the cofficients for low frequencies are small. This makes it vulnerable to the presence of white noise in the signal. All power coefficients below 62.5 Hz are summed and represented and/or very low-frequency components from having dispropotionate weight. On the discrete frequency bin scale, this corresponds to every power coefficient falling below the index [7]:

$$K_{low} = \mathrm{floor}(62.5/\Delta F) \tag{2.22}$$

where floor(x) gives the largest integer less than or equal to x, and $\Delta F = F_s/N_{FT}$ is the frequency interval between two FFT bins. The result in a new power spectrum P'(k') is given by:

$$P'(k') = \sum_{k=0}^{K_{low}} P(k) \quad \text{for } k' = 0 \tag{2.23}$$

$$P'(k') = P(k' + K_{low}) \quad \text{for } 1 \leq k' \leq \frac{N_{FT}}{2} - K_{low} \tag{2.24}$$

The frequencies f'(k') corresponding to the new bin k' are given by:

$$f'(k') = 31.25 \quad \text{for } k' = 0 \tag{2.25}$$

$$f'(k') = f(k' + K_{low}) \quad \text{for } 1 \leq k' \leq \frac{N_{FT}}{2} - K_{low} \tag{2.26}$$

where f(k) is defined as in Equation 2.20. The nominal frequency of the low-frequency coefficients is chosen at the middle of the low-frequency band: $f'(0) = 31.25$Hz. Finally, for a given frame, the ASC is defined from the modified power coefficients P'(k') and their corresponding frequencies f'(k') as:

$$ASC = \frac{\sum_{k'=0}^{(N_{FT}/2)-K_{low}} \log_2(\frac{f'(k')}{1000} P'(k'))}{\sum_{k'=0}^{(N_{FT}/2)-K_{low}} P'(k')} \tag{2.27}$$

Each frequency $f'(k')$ of the modified power spectrum is weighted by the corresponding power coefficient $P'(k')$. Several other implementations and definitions of the spectrum centroid can be found in the literature [43]. The log-frequency scaling approximates the perceptual of frequencies in the human hearing system.

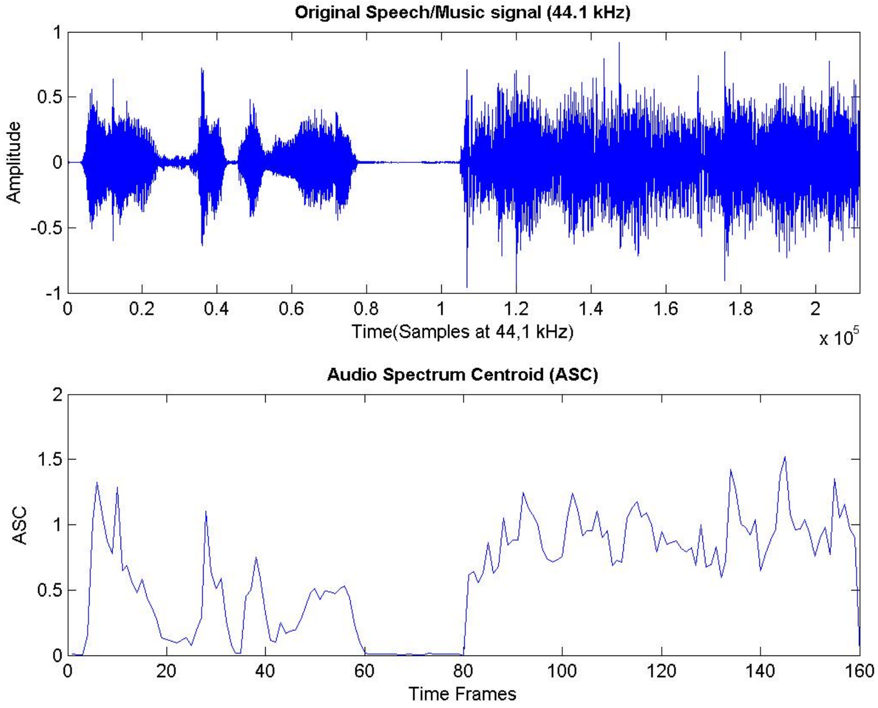

Figure 8: Audio Spectrum Centroid description of a speech/music signal (first half speech, second music, 30ms hopSize)

2.3.3 Fundamental Frequency (FuF)

Typical music consists of a series of chords which changes frequently. These chords can be seen as groups of frequencies in a spectrum, and are present for a longer time. Music can then be segmented into entities. On these entities, it is now possible to perform a fundamental frequency determination [22]. The fuf results from overlying the higher frequencies. If two frequencies f1 and f2 are played, and they are a fifth apart from each other, the fundamental frequency, $f0, is(I + 1/I)f1$. (Where I is between 2 and 5) In Figure 9 the different frequencies are illustrated and Figure 10 illustrates a speech/music signal.

- Determine the lowest frequency in the signal, called $f1$
- Check for a frequency a fift, fourth, major or minor above $f1$
- Set $f0 = 1/I * f1$ if Yes
- In not, choose $f1$ as the fuf

2.3.4 Mel Frequency Cepstral Coefficients (MFCC)

The quest for better speech parameterization led to various speech features, which were reported to provide advantage in specific conditions and applications [17]. Moreover, for some speech features, such as the well-known and widely-used MFCC, multiple implementations were developed [18] [19] [20]. These implementations differ mainly in the number of filters, the shape of the filters, the way the filters are spaced, the bandwidth of

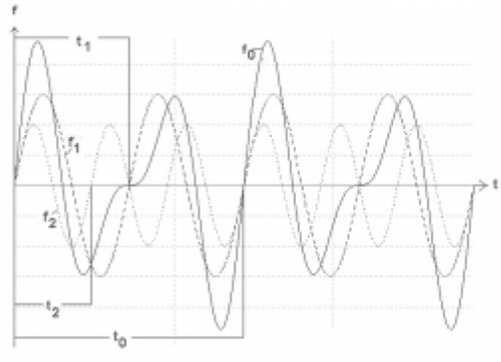

Figure 9: Overlying frequencies f1 and f2

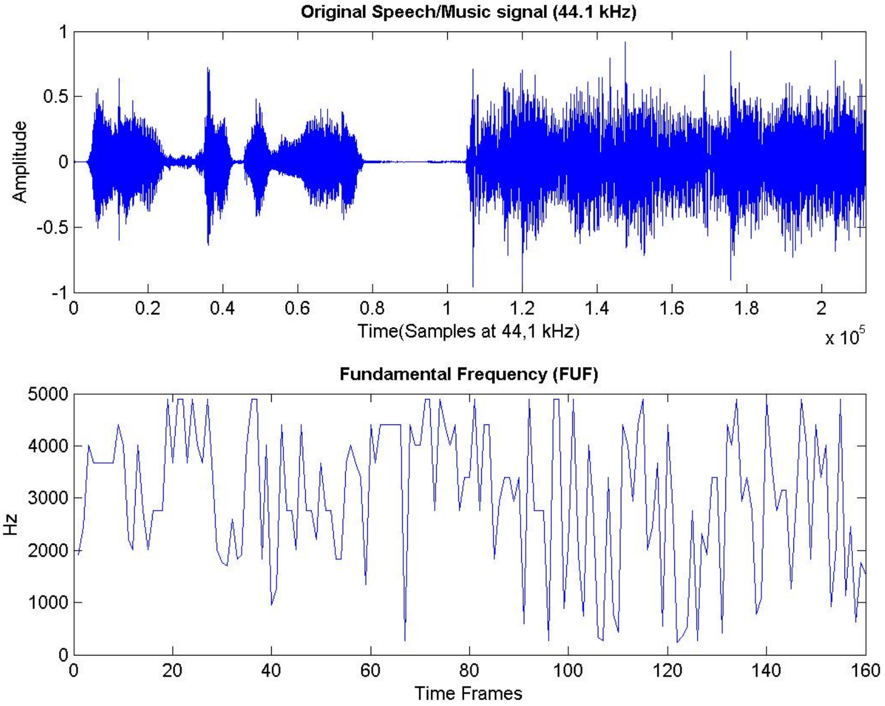

Figure 10: Fundamental Frequency description of a speech/music signal (first half speech, second music, 30ms hopSize)

the filters, and the manner in which the spectrum is warped. How these filters are used to extract Mel cepstrum is illustrated in Figure 11. In addition, the frequency range of interest, the selection of actual subset and the number of MFCC coefficients employed in the classification [13].

An example of a Mel spaced filter bank is illustrated in Figure 12.

The mel scale, which is divided into mel units, is based on an empirical study of the human perceived pitch or frequency. The scale of pitches are judged by listeners to be equal in distance from another. A test was carried out where test persons were presented with a tone at exactly 1000 Hz (which was labeled 1000 mels for reference) at 40 dB

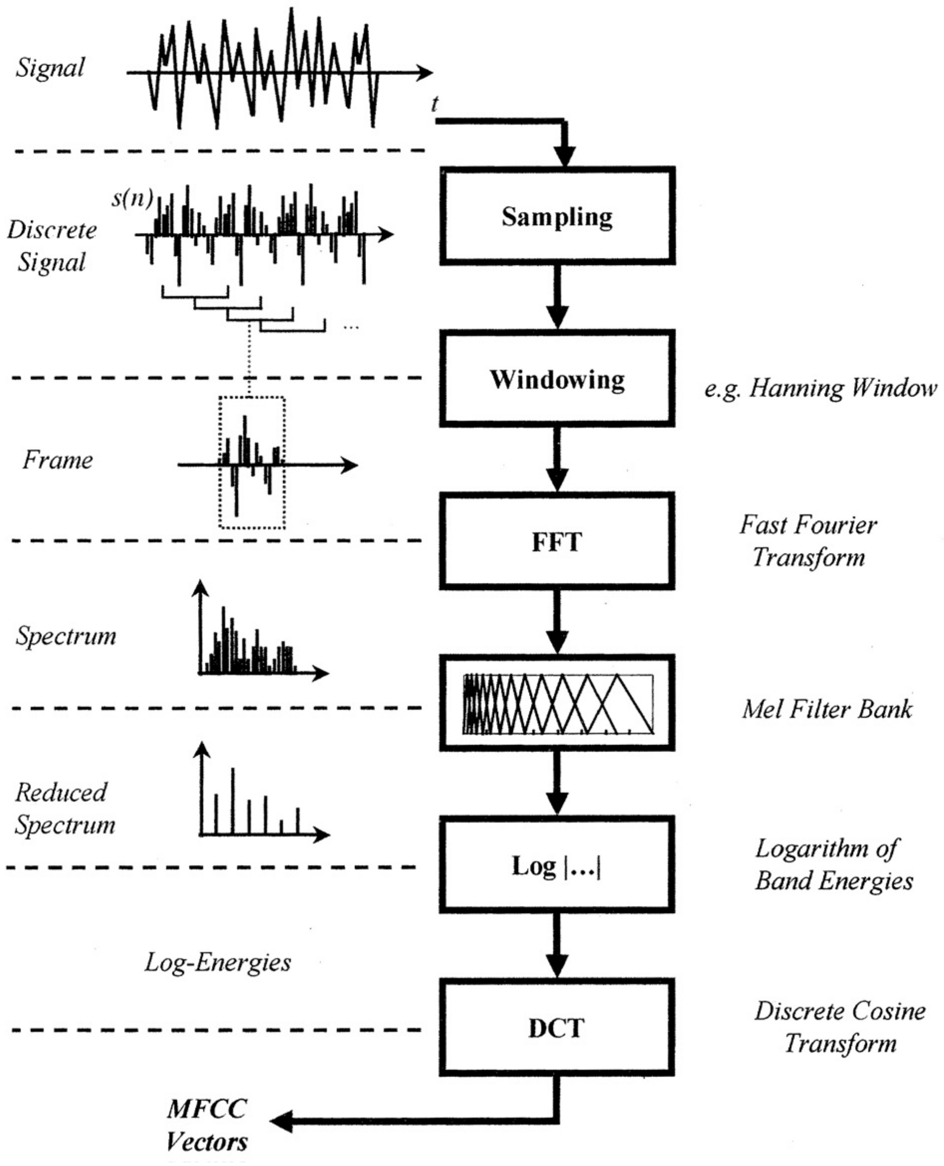

Figure 11: MFCC Extraction of MFCC vectors ([7])

above the listeners threshold, then they were asked to change the frequency until they perceived the frequency to be twise the reference. This frequency was the labeled 2000 mels. Below 500 Hz the mel and Hertz scale coincide; above that, larger and larger intervals are judged by the listeners to produce equal pitch increments [21].

The mapping between the mel frequency scale f_{mel} and the linear scale f is usually done using an approximation:

$$f_{mel} = 2595 * \log_{10}\left(1 + \frac{f_{lin}}{700}\right) \tag{2.28}$$

To extract the MFCC vectors the input signal $s(n)$ is first devided into overlapping frames of N_w samples. Typically, frame duration is between 20 and 40 ms, with 50 percent overlap between adjacent frames. In order to minimize the signal discontinuities

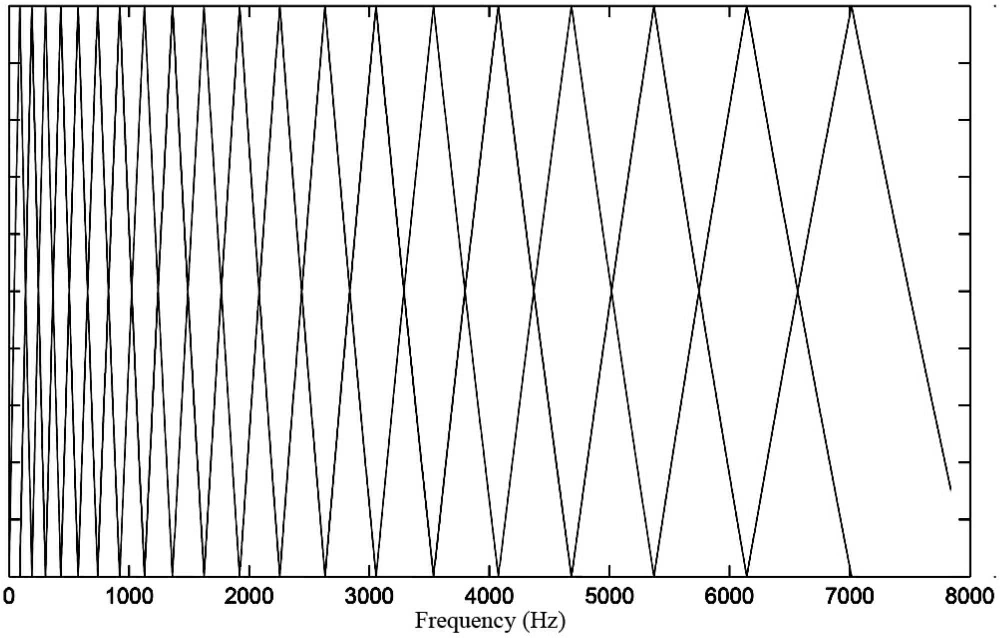

Figure 12: Mel spaced filter bank

at the borders of each frame a windowing function is used, such as the Hanning function defined as:

$$w(n) = \frac{1}{2}\left\{1 - \cos\left[\frac{2\pi}{N_w}\left(n + \frac{1}{2}\right)\right]\right\} \quad (0 \leq n \leq N_w - 1) \tag{2.29}$$

To obtain the magnitude spectrum, an FFT is applied to each frame and the absolute value is taken. The spectrum is then processed by a mel-filter bank. The log-energy of the spectrum is measured within the pass-band of each filter, resulting in a reduced representation of the spectrum. The cepstral coefficients are finally obtained through a Discrete Cosine Transform (DCT) of the reduced log-energy spectrum:

$$c(i) = \sum\left\{\log(E_j)\cos\left[i\left(j - \frac{1}{2}\right)\frac{\pi}{N_f}\right]\right\} \quad (1 \leq i \leq N_f) \tag{2.30}$$

where c_i is the ith-order MFCC, E_j is the spectral energy measured in the critical band of the jth mel filter and N_f is the total of mel filters (typically $N_f = 24$). N_c is the number of cepstral coefficients c_i extracted from each frame (typically $N_c = 12$). The global log-energy measured on the whole frame spectrum - or, equivalently, the c_0 MFCC calculated according to the formula of Equation refeq:fmelApp with $i = 0$ - is generally added to the initial MFCC vector. The extraction of an MFCC vector from the reduced log-energy spectrum is illustrated in Figure reffig:eightOrderMFCC.

The estimation of the derivative and acceleration of the MFCC feature are usually added to the initial vector in order to take into account the temporal changes in the spectra. One way to capture this information is to use $deltacoefficients$ that measure a linear regression over a few adjacent frames. Typically, the two previous and the two following frames are used, for instance, as follows:

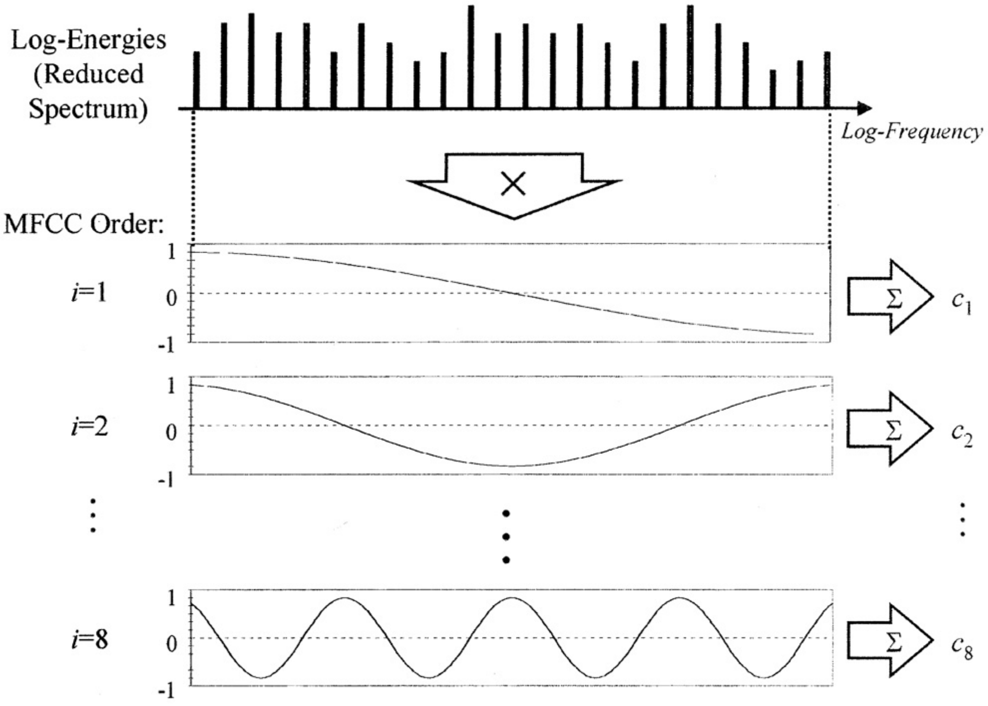

Figure 13: Extraction of an eight-order MFCC vector from a reduced log-energy spectrum (cite7)

$$\Delta c_i(l) = c_i(l-2) - \frac{1}{2}c_i(l-1) + \frac{1}{2}c_i(l+1) + c_i(l-2) \tag{2.31}$$

$$\Delta\Delta c_i(l) = c_i(l-2) - \frac{1}{2}c_i(l-1) - c_i(l) - \frac{1}{2}c_i(l+1) + c_i(l-2) \tag{2.32}$$

where $c_i(l)$ is the ith-order MFCC extracted from the lth frame of the signal. The $\Delta c_i(l)$ and $\Delta\Delta c_i(l)$ coefficients are the estimates of the derivative and acceleration of coefficient c_i at frame instant l, respectively. Together with the cepstral coefficients $c_i(l)$, the Δ and $\Delta\Delta$ coefficients form the final MFCC vector extracted from frame l.

In this thesis a version of the MSCC FB-24 HTK is used. This version has its origins from the Cambridge HMM Toolkit described in [14]. They used a filter bank of 24 filters for speech bandwidth [0, 8000] Hz with sampling rate greater than 16 kHz. The Mel cepstrum coefficients for a 24 second speech sample and 24 second music sample is illustraded in Figure 14 and Figure 15.For Mel Frequency the HTK is defined by equation 2.30.

2.3.5 Spectrum Flux (SF)

Spectrum Flux(SF) is defined as the average variation value of spectrum between the adjacent two frames. Experiments show that music has the lowest SF values, speech has higher values than music and environmental sounds have the highest SF values [23]. Figure 16 illustrates a audio signal containing speech and music. SF is computed as the average squared difference between tow successive spectral distributions [2]:

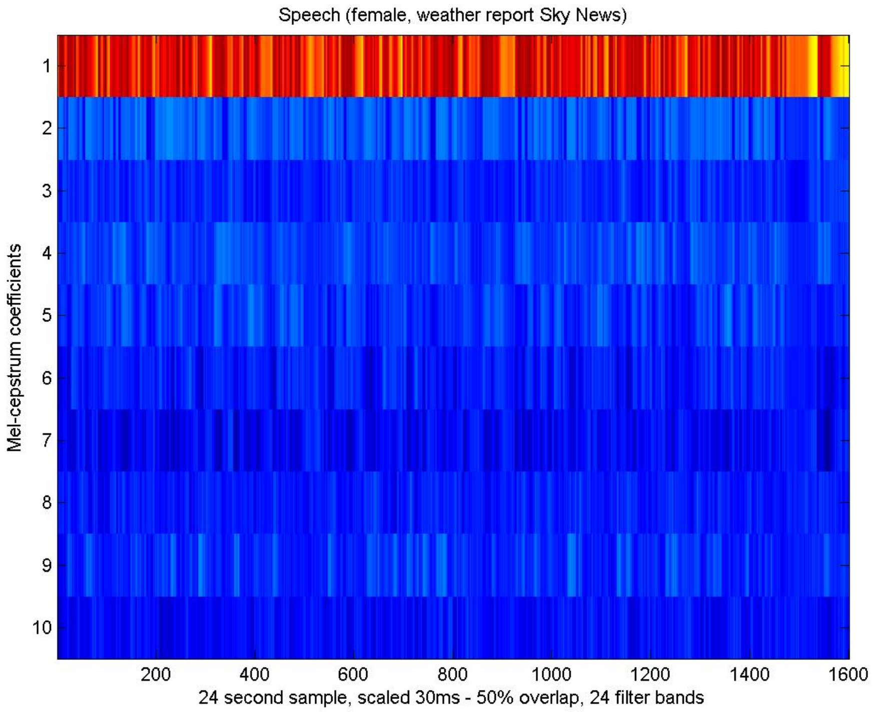

Figure 14: MFCC for 24 seconds of speech

$$SF = \frac{1}{LN_{FT}} \sum_{k=0}^{L-1} \sum_{k=0}^{N_{FT}^{-1}} [\log(|S_l(k)| + \delta) - \log(|S_{l-1}(k)| + \delta)]^2 \tag{2.33}$$

where $S_l(k)$ is the DFT of the lth frame, N_{FT} is the order of the DFT, L is the total number of frames in the signal and δ is a small parameter to avoid calculation overflow.

2.4 MPEG-7 Low-Level Descriptors

The MPEG-7 low-level descriptors consist of a collection of simple, low-complexity audio features, and form the foundation layer of the standard. These features can be used to characterize any type of sound. The MPEG-7 standard consists of 18 generic LLD's, and the temporal and spectral LLDs can be classified into the following groups:

- Basic descriptors: Audio Waveform (AFW) and Audio Power (AP)
- Basic spectral descriptors: Audio Spectrum Envelope (ASE), Audio Spectrum Centroid (ASC), Audio Spectrum Spreads (ASS) and Audio Spectrum Flatness (ASF).
- Basic Signal Parameters: Audio Harmonicity (AH) and Audio Fundamental Frequency (AFF)
- Temporal Timbral Descriptors: Log Attack Time (LAT) and Temporal Centroid (TC)
- Spectral Timbral Descriptors: Harmonic Spectral Centroid (HSC), Harmonic Spectral Deviation (HDS), Harmonic Spectral Spread (HSS), Harmonic Spectral Varia-

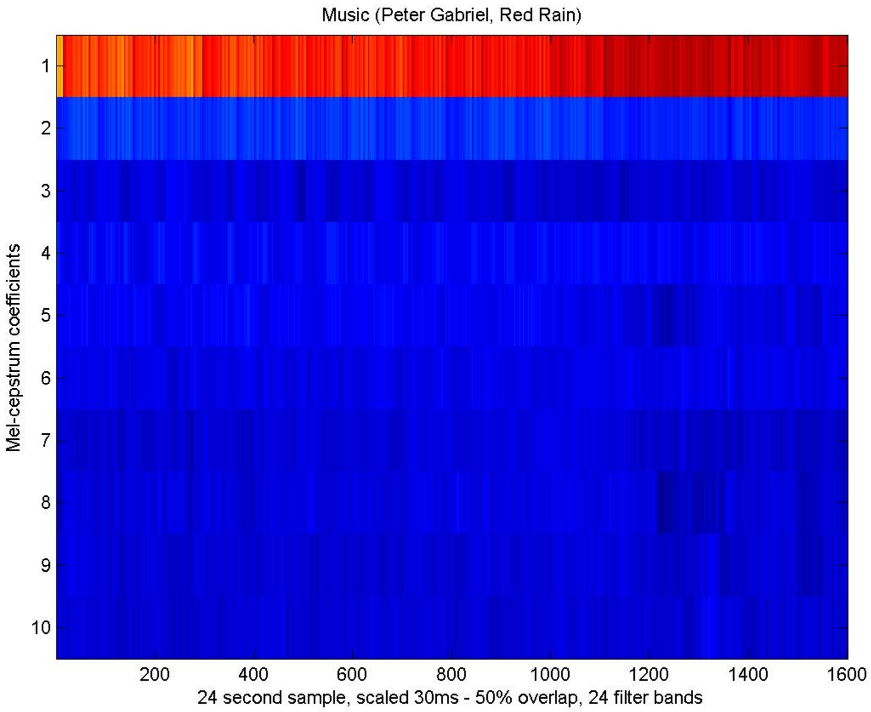

Figure 15: MFCC for 24 seconds of music

tion(HSV) and Spectral Centroid (SC)

- Spectral Basis Representation: Audio Spectrum Basis (ASB) and Audio Spectrum Projection (ASP)

In this thesis only AFW, AP, ASC and AFF will be used from the MPEG-7 standard.

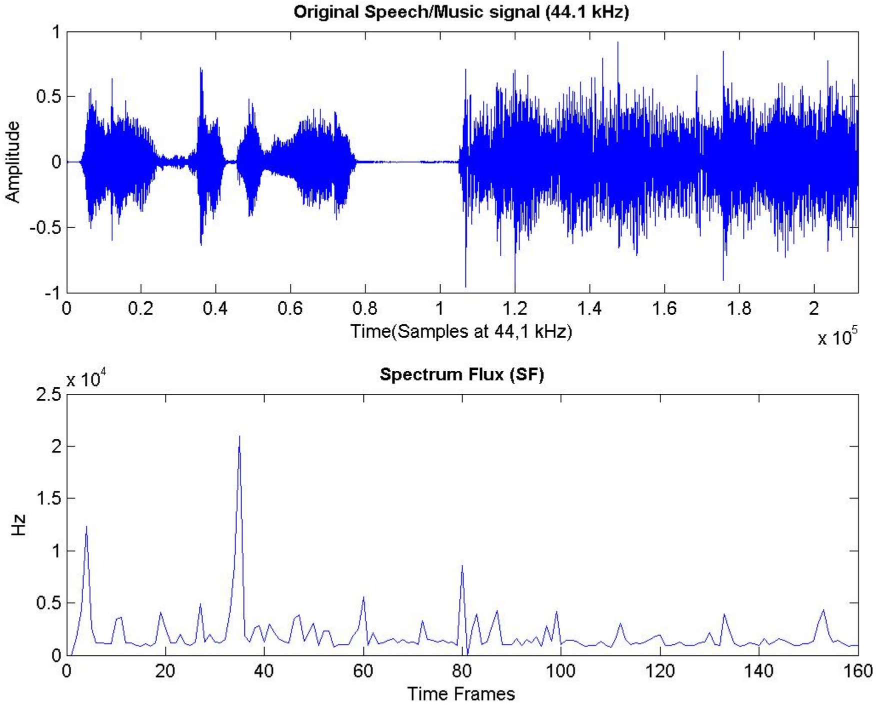

Figure 16: Spectrum Flux description of a speech/music signal (first half speech, second music, 30ms hopSize)

3 Audio Classification

Most common segmentation is between speech, music, environmental sound and silence segments. Speech can be further segmented into gender, age or identity, and music can be segmented into genres and instruments. Sound can be segmented into different events, such as explosions, applause, animal sounds, engines, etc. To get a meaningful indexing of an audio track, the different segments and events has to be identified, and stored in such way that a similarity test is possible. Spectral features of sounds are a good way to describe the content. The values extracted and the time variation of the feature will identify the event, and can be seen as the sound events fingerprint. By storing these fingerprints in a database we can perform similarity calculations on fingerprints extracted from an audio track. The purpose of sound classification is to create sound classes, and try to distinguish which class the different extracted fingerprints belong to.

- The first step is to isolate the relevant sound segments from the not relevant sound, such as noise, background music or environmental sounds.
- Then the properties that are useful for classification are extracted. It is important that these feature vectors are rich enough to describe the content of the sound sufficiently. One of the most used feature vectors is based on MFCC. The MPRG-7 standard uses audio spectrum projection (ASP).
- In the classification step, reduced dimension feature vector is used to classify the sound into classes. Statistical models are often used to do these classifications. K Nearest Neighbor(KNN), Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs), Neural Network (NNs) and Support Vector Machines (SVMs) are examples of such classifiers.

What kind of feature vectors and classifier to use in the sound classification system is critical. A combination of the different low-level feature vectors are often used to create a compound feature vector for classification. Usually it is necessary to train the classifier to. This is done by processing known sound material which are labeled with their correct class.

3.1 Dimensionality Reduction

To reduce the size of the feature vectors, but retain as much as possible of the perceptual information as possible, we remove the statistical dependencies of observations. There are several methods known for this: Singular Value Decomposition (SVD), Principal Component Analyses (PCA), Independent Component Analysis (ICA) and Non-Negative Matrix Factorization (NMF) [7]. In this project the PCA will be used.

Principal Component Analysis (PCA)

To obtain a basis for an N-dimensional data set, such that variation of the data is maximized along the coordinate axes, PCA can be used. By ignoring the axes of lowest variation, thereby projecting the original data into a lower dimensional space with minimal loss of accuracy, data reduction is achieved. In figure 17, a set of data points is located in 2D space. The data exhibits equal variation along the x and y axis, however in the primed coordinate system, there is significant variation along the x' axis and little along y'. Thus, the points can be represented in a one-dimensional space (along the x' axis) while still retaining much of the information present in the two-dimensional representation.

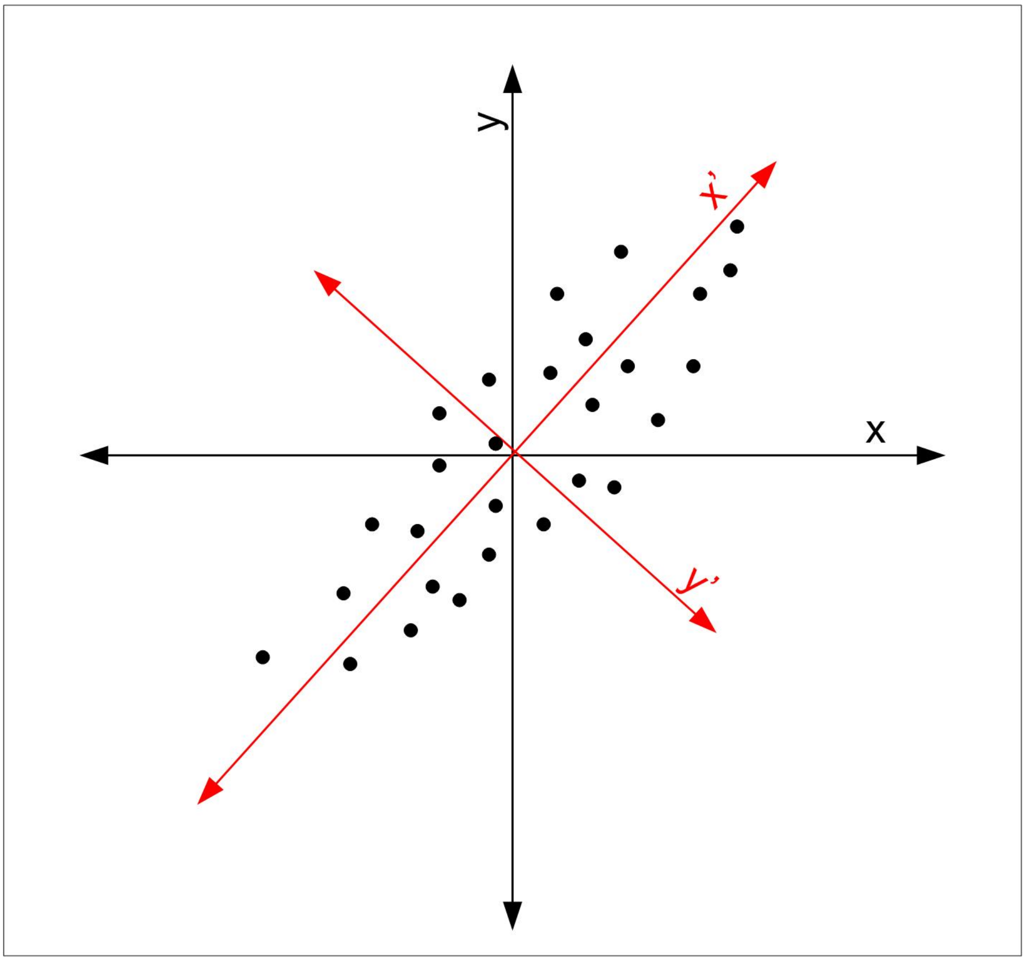

Figure 17: Variation of the data is greatest along the orthogonal axes of the primed coordinate system

We compute of the mean, T_{mean} of the N_a basis appearance vectors and form matrix A' whose columns $\tilde{T}^k$ are obtained by subtracting the mean from each of the T^k's ($T^k = T_{mean} + \tilde{T}^k$). Principal component analysis is performed via Singular Value Decomposition (SVD) on the matrix A'. SVD factors A' into an orthonormal $48s - by - N_a$ matrix Y whose columns span the range of A', a diagonal $N_a by N_a$ matrix S, and a $N_a by N_a$ matrix V^T.

$$A' = YSV^T \tag{3.1}$$

The diagonal elements σ_{kk} of S are the singular values of A', and σ_{kk} gives the variation of the data along the axis specified by $\tilde{T}^k$. We normalize the σ_{kk}'s such that the largest singular value is 1, and form matrix $\tilde{A}$ whose columns consist of $\tilde{T}^k$ whose corresponding singular value is greater than ϵ. The resulting columns of $\tilde{A}$ form a basis for a low rank approximation to the range of A' that is accurate to ϵ percent least squares error.

$$span(\tilde{A}) \approx span(A') \tag{3.2}$$

Let $\tilde{N}_a$ be the rank of $\tilde{A}$. Now, the appearance of an arbitrary model deformation is represented by reduced space coordinate $\tilde{q}_a^j$. $\tilde{q}_a^j$ consists of $\tilde{N}_a$ components of $SV^T\tilde{q}_a^j$ corresponding to basis vectors retained in $\tilde{A}$. The appearance vector $\tilde{T}^j$ is then computed by:

$$\tilde{T}^j = T_{mean} + \tilde{A}_r\tilde{q}_a^j \tag{3.3}$$

Runtime computation is reduced to the linear combination of $\tilde{N}_a$ basis vectors added to the mean model appearance. In practice, only a few basis vectors are required to approximate the original appearance space with reasonable accuracy ($\tilde{N}_a << N_a$), resulting in a significant reduction in the size of the required runtime data set.

3.2 Classification Methodes

The classification method used in this project, is the K Nearest Neighbor (KNN).

Nearest Neighbour classifier Model (KNN)

The K nearest neighbour classifier is an example of a non parametric classifier. The basic algorithm in such classifiers is simple. For each input feature vector to be classified, a search is made to find the location of the K nearest training examples, and then assign the input to the class having the most members in this location. This is illustrated in Figure 18. Euclidean distance is commonly used as the metric to measure neighbourhood. For the special case of K=1 we will obtain the nearest neighbour classifier, which simply assigns the input feature vector to the same class as that of the nearest training vector. The Euclidean distance between feature vectors $X = (x1, x2, ..., xn)$ and $Y = (y1, y2, ..., yn)$ is given by:

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)} \tag{3.4}$$

The KNN algorithm, as mentioned earlier, is very simple yet rather powerful, and used in many applications. However, there are things that need to be considered when KNN classifiers are used. The Euclidean distance measure is typically used in the KNN algorithm. In some cases, use of this metric might result in an undesirable outcome. For instance, in cases where several feature sets (where one feature set has relatively large values) are used as a combined input to a KNN classifier, the KNN will be biased by the larger values. This leads to a very poor performance. A possible method for avoiding this problem would be to normalise the feature sets. The aim is to use the KNN classifier for finding the class of an unknown feature X.

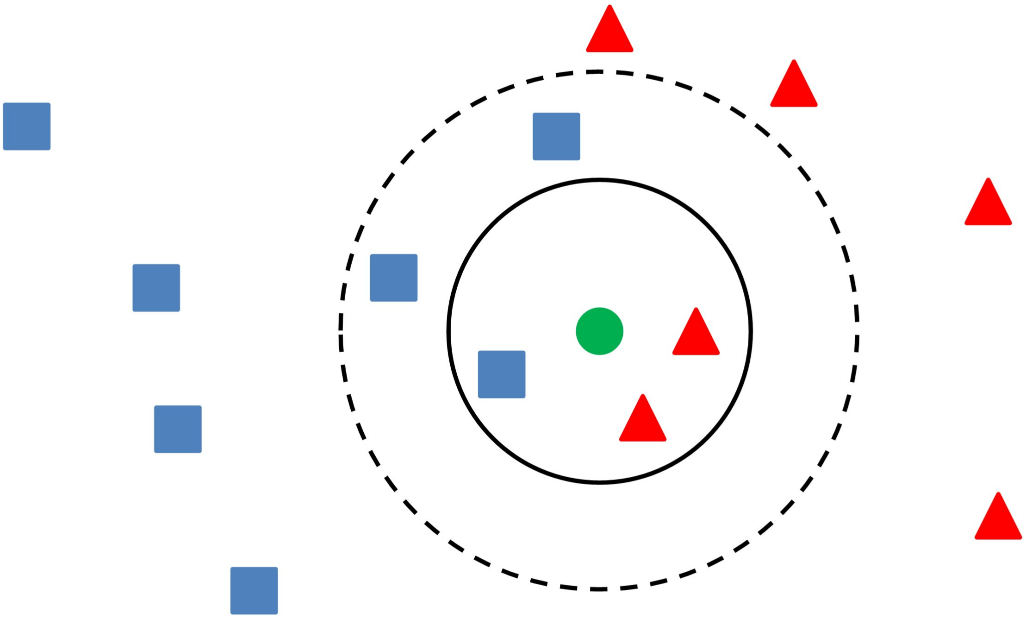

Figure 18: Illustration of the KNN classification method. The green circle is the sample which is to be classified, blue squares and red triangles illustrates samples from two different classes in the training set. If K is 3, then the 3 nearest neighbors (illustrated by the black circle with the solid line) will decide which class the analyzed sample will be assigned. If K is 5, the 5 nearest neighbors will be considered (illustrated by the black circle with the dotted line).

4 Experimental Setup

4.1 Audio Database

To test the algorithms and methods we are proposing, a database of test data was needed in addition to the audio tracks from different movies. Therefore, several samples of speech were collected from different television channels, listed in Table 1. Talk shows and news broadcasts were recorded to get audio samples for clear voice in quiet environment.

Table 1: Speech Samples

Sample	Gender	Type	Duration	Source
speech1	female	talkshow	24 s	TV2
speech2	male	talkshow	24 s	TV2
speech3	male	talkshow	24 s	TV2
speech4	male	talkshow	24 s	TV2
speech5	female	weather	24 s	Sky News
speech6	female	news	24 s	BBC World
speech7	female	news	24 s	CNBC
speech8	female	news	24 s	Sky News
speech9	female	news	24 s	Sky News
speech10	male	news	24 s	CNBC

Music was sampled from different CD's and MP3's. Several different categories of music are present, both voiced and instrumental. Table 2 lists the different music samples.

Table 2: Music Samples

Sample	Type	Voice	Duration
music1	pop	Yes	24 s
music2	metal	Yes	24 s
music3	mood	No	24 s
music4	pop	Yes	24 s
music5	pop	No	24 s
music6	mood	No	24 s
music7	mood	Yes	24 s
music8	rock	No	24 s
music9	rock	Yes	24 s
music10	metal	No	24 s
music11	metal	Yes	24 s

The environmental sound samples are constructed out of segments from videos, examples of background sounds digitized from radio and Internet and nature sounds sampled from nature sound CDs. Table 3 lists the different environmental samples.

Beside the above mentioned samples of clear sound classes, Segments from the motion picture ”The Godfather” are used for testing and training of the KNN classification. ”Groundhog Day” is used in the final experiment to test the settings purposed and chosen in this chapter.

Table 3: Environmental Sounds

Sample	Location	Type	Duration
envir1	City	Downtown	24 s
envir2	City	Highway	24 s
envir3	City	Engines and Car horns	24 s
envir4	City	Trucks and Car horns	24 s
envir5	Nature	Waterfall	24 s
envir6	Nature	Thunder and rain	24 s
envir7	Nature	Heavy rain	24 s
envir8	Nature	Jet plane pasing	24 s
envir9	Nature	Florida swamp	24 s
envir10	Nature	Waves, surf	24 s

4.2 Ground Truth

Manually categorizing of every 1.2 second of the movie ”Groundhog Day” was needed. A Matlab function was therefore created for categorizing each 1.2 second of audio. The function played a 1.2 second sample and then asked for a category. Three different ground truth sets were created out of this category; main classes, detailed classes and RGB-colored for use in scatter diagram. The answer was stored in a vector according to the sample number. Table 4 lists the main classes. Silence in the audio track of a motion picture is more to be considered as environmental content than no content. This is because the silence in the movie describes an environment setting, and not sections with non information.

Table 4: Main classes in the Ground Truth

Class Id	Class
1	Music
2	Speech
3	Environmental sound

The more detailed sub classes of the ground truth set describes the variations of the main classes. When separating between speech with background music and music with vocals it is necessary to have more than one music category. Table 5 lists the different subclasses of music.

Table 5: Sub classes in the Ground Truth, Music

Class Id	Class	Sub-class
11	Music	Clear
12	Music	Voiced
13	Music	Environmental sounds in background
14	Music	Voice and environmental sounds in background
15	Music	Strong (dominant) voice
16	Music	Background music

The LLDs extracted from speech samples have very different characteristics when background audio is present or not. Recognizing speech with different sounds in the background is important when analyzing the audio track from a motion picture video. Several sub classes of speech are therefore created to be able to recognize different types of speech. Table 6 lists the different subclasses of speech.

Table 6: Sub classes in the Ground Truth, Speech

Class Id	Class	Sub-class
21	Speech	Clear female
22	Speech	Clear male
23	Speech	Female speech with music in background
24	Speech	Male speech with music in background
25	Speech	Female speech with environmental sounds in background
26	Speech	Male speech with environmental sounds in background

In this ground truth, silence is regarded as an environmental setting, in the same way as city, industrial and nature sounds. Noise is also included in this category. The environmental category is essential to describe semantic content in a motion picture. Table 7 lists the different subclasses of environmental sounds.

Table 7: Sub classes in the Ground Truth, Environmental/noise sounds

Class Id	Class	Detailed
31	Environmental	Silence
32	Environmental	City
33	Environmental	Industrial
34	Environmental	Nature
35	Environmental	White noise
36	Environmental	Brownian noise

4.3 Low-Level Descriptors

To find the most suitable LLDs for correctly classifying the different audio samples into the predefined classes, some preliminary tests had to be carried out. First step was to find good samples of speech, music and environmental sounds. A set of different samples of speech and music was sampled from various TV-channels and CD's. The environmental sounds were sampled from various motion picture and documentary videos. See Table 1, Table 2 and Table 3 for details. The speech and music samples were used to construct a sample file containing alternating 24 second music and speech sample, se Figure 19.

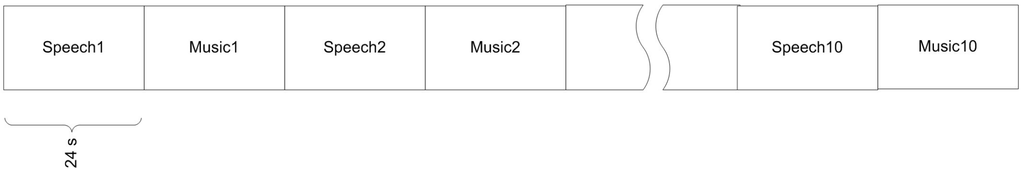

Figure 19: Alternating speech and music samples of 24 seconds

All the 23 LLD features were extracted from this sample file on a 30ms window and grouped into 1.2 second frames. We applied PCA analysis on these features vectors to reduce their dimensionality to only 6. The scatter diagram of the first three elements of these PCA vectors are plotted in Figure 20.

Speech sample 9 and music sample 9 were found to be the two samples that are the farthest apart from each other, and should therefore be the easiest samples to separate

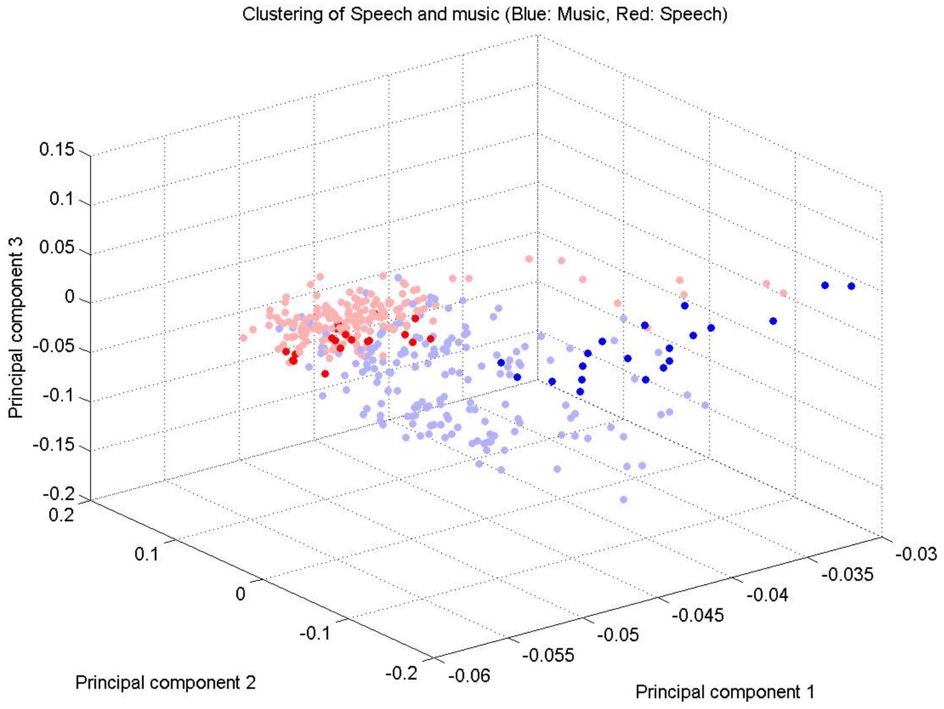

Figure 20: Bright red and blue labels the speech and music sample which is farthest apart from each other

with a classifier. These are marked with bright red and blue labels in Figure 20. It is worth mentioning that speech sample one and two are much closer to music than any other of the speech samples. The only difference noticed when listening to these samples is that they are in Norwegian. Sample one and two are plotted in green in Figure 21. The most likely explanation to this is that the pauses between the syllables in the Norwegian language are shorter than in English. The variation of the power in the signal, in the time domain, is therefore closer to music, where the variation is usually very small. Because the speech1 sample is of a female, the average frequency level is higher than speech2 sample which are of a male speaker. This places speech1 even closer to the music samples. Sample speech3 and speech4 are also in Norwegian, but the speaker is talking in a dialect.

The second step was to find the boundary after which it is no longer possible to separate speech from background music. To find this boundary, a new set of samples were constructed out of the speech, music and environmental sounds samples. Figure 22 illustrates how these new samples are constructed. Every possible combination of the speech and music, and the speech and environmental sounds, samples was made, creating a total of 200 constructed samples. This step will be described in subsection 4.3.1.

All different low level features were then extracted from these audio samples, and ground truth sets were constructed. Then PCA was applied to all different combinations of the LLDs and the output vectors used to classify the audio samples into two groups by the KNN. Finally, the results were checked against the ground truth. This procedure was

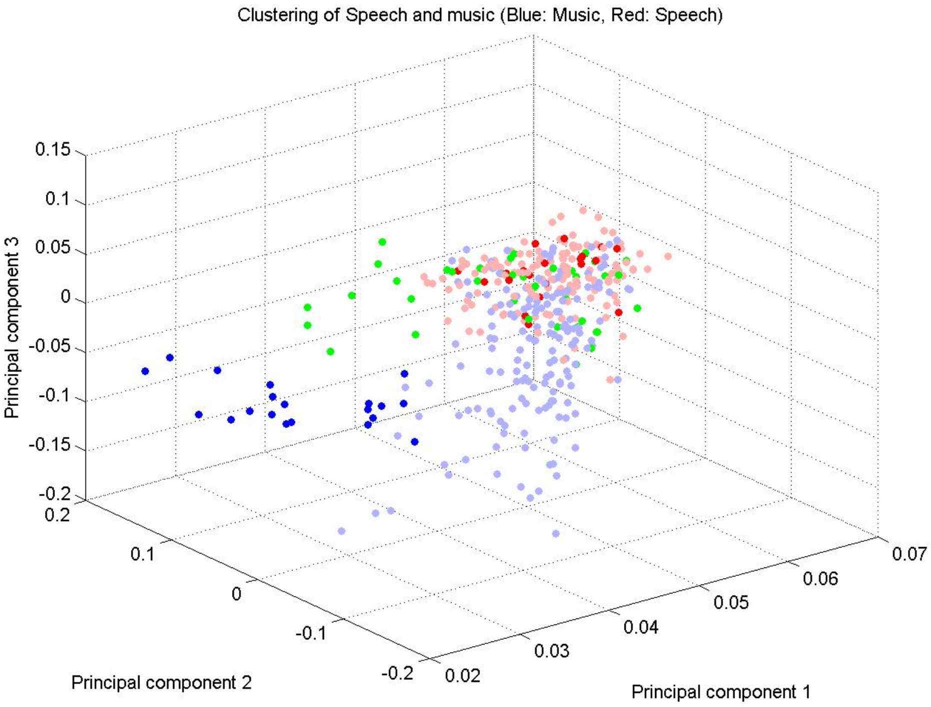

Figure 21: Green labels clear Norwegian speech, one female and one male

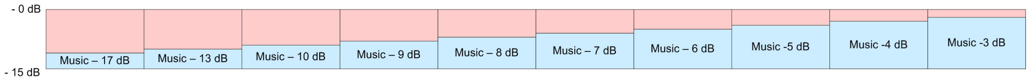

Figure 22: Sample constructed for finding speech/music bounarie in dB

done in three different experiments: First the best combination of LLDs was determined, second the optimal number of dimensions is estimated using returned PCA, and the number of neighbors considered in the KNN. Finally the different mixtures of speech and music, and speech and environmental sounds, were used to find the boundary where speech with background music, and speech with background environmental sounds, is no longer categorized as speech.

4.3.1 Finding the best combinations of LLDs

Every possible combination of the 23 LLDs was tested. The combinations of the LLDs was tested both with dimensional reduction by the PCA and without, to see if there were any differences in the performance. In most cases, when testing on mixed classes, the results were better when PCA was applied to the extracted LLDs. The results from the tests when PCA was not applied had fewer LLDs in the combination, but overall results were lower than when PCA was applied on a larger set of LLDs. This is illustrated in Figure 23 and Figure 24.

There is no obvious good way to illustrate the result graphically. Because there is only one result per combination, only two dimensions are available. One interesting part about this figure is that there is a repeating pattern in the figure. In the same way that

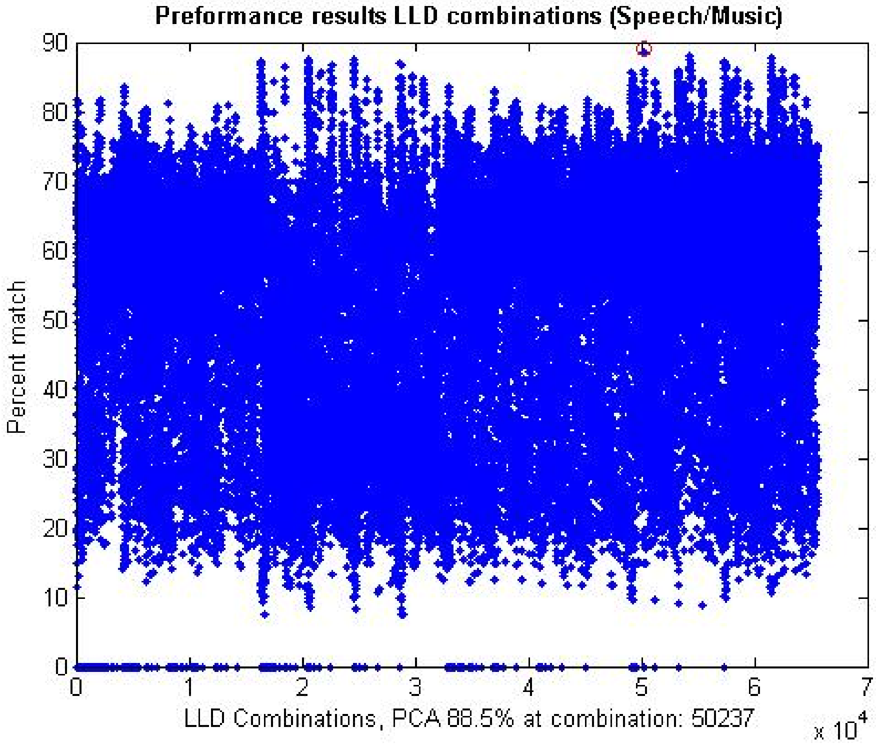

Figure 23: Most suitable LLD combination when PCA was applied. (The blue dots on the zero line is the combinations with less than 3 descriptors, these is not calculated since the PCA needs a minimum of 3 dimensions as input in this experiment.)

some LLD combinations creates high matching rates between categorized frames and the ground truth, while some LLD combinations create a low match. Because the test is done in a binary way, (1000, 0100, 1100, 0010, 1010, 0110, 1110, ..., 1111), all combinations of the previous LLDs will be repeated every time a new LLD is added.

Because of the very time consuming process of testing all the 16 777 215 possible combination of the 23 LLDs, some LLDs was excluded during the testing. Some of them because they performed poorly in several of the tests where LLDs where randomly selected, and some because it was logical to remove them. Examples of LLDs which performed poorly are Mel-cepstrum coefficients 3 to 7, and ZCR and FUF was logical to remove. This is further explained in the two following sub sections.

Time Domain

From the time domain AP, RMS and STE were found to perform well in combination, when separating speech and music. This is illustrated in Figure 25. RMS and the STE calculated in the min scalar is present in all the combinations. This correlates well with the fact that these LLDs describe the power in the signal and that music is known to have more power than speech. Although ZCR is known to be a good descriptor to separate speech and music [5], it is not true in this experiment. The reason for this is that speech is mixed together with music in the samples, and the characteristics for speech are seldom detected. HZCRR, which describes the variation in ZCR, is not present in any of the combination. This is most likely because the variation of the ZCR is very little influenced

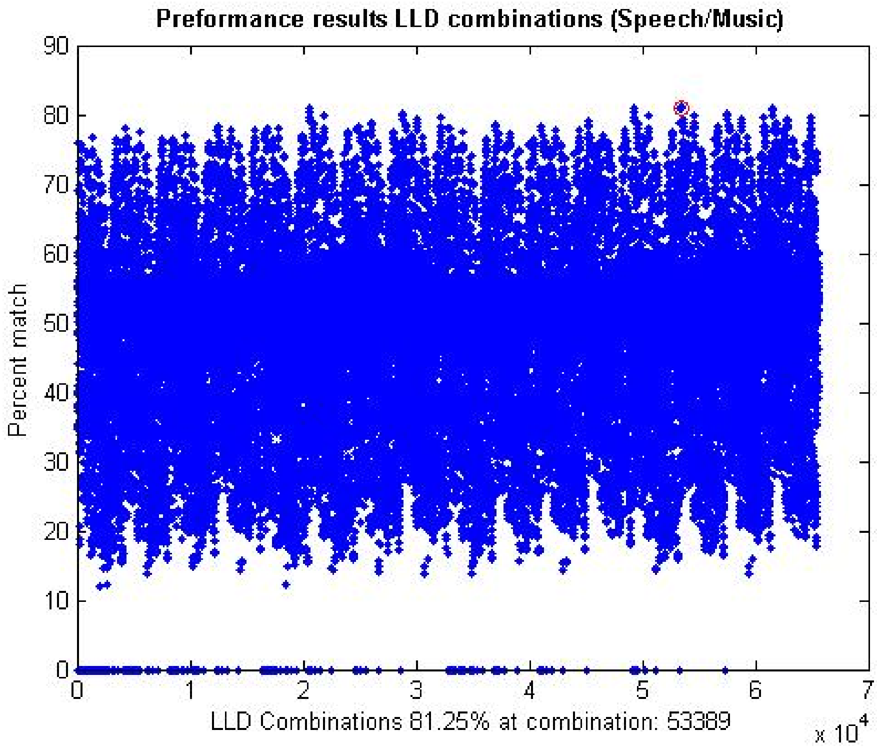

Figure 24: Most suitable LLD combination without dimensional reduction by the PCA. (The blue dots on the zero line is the combinations with less than 3 descriptors, these is not calculated since the PCA needs a minimum of 3 dimensions as input in this experiment.)

by the power variations. One exception is the ZCR(min) which performs well. This is probably because speech influences more in the low frequency area than in the high frequency area. And when we only look at the changes in the low frequency area, which ZCR(min) do, there is a detectable difference between speech with background music and clear music. This is also strengthened by the fact that HZCRR, which describes the high changes in ZCR, performs poorly when separating speech with background music from clear music .

Frequency Domain

In the frequency domain ASC and the MFCCs were present in most combinations. This is illustrated in Figure 26. ASC, which describes the gravity of the most dominant frequency, is known to be a good to separate speech and music because speech is most of the times in a lower frequency area than music. When speech and music are mixed together, speech will influence the gravity of the frequency spectrum so that clear music will differentiate from speech with background music. The MFCCs, which divides the frequency area into frequency bands perceptual equal in distance from each other, describes the frequency presence in the signal. Figure 26 illustrates that the first and the last cepstrums is most present in the combinations. This correlates well with the knowledge that speech generates most changes in the low frequency area and music generates changes in the high frequency area. FUF, which searches for fundamental frequencies in the signal, performs not so well in this experiment. Since speech and music is mixed

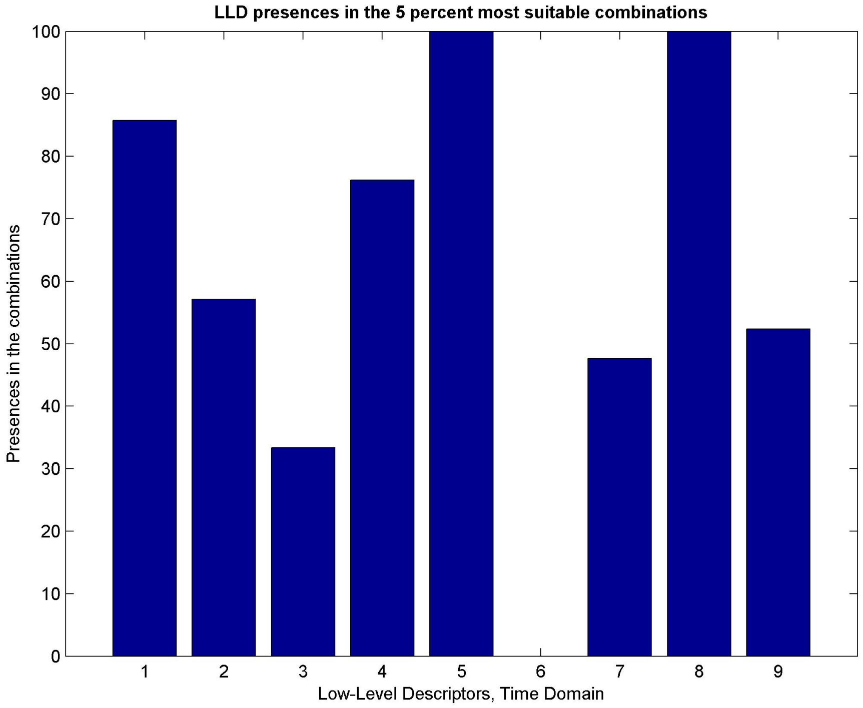

Figure 25: LLD presence in the 5 percent most suitable combinations of the LLDs in the time domain. 1: AP, 2: AFW, 3: ZCR, 4: STE, 5: RMS, 6: HZCRR, 7: AP(min), 8: ZCR(min), 9: STE(min).

together in these samples, a fundamental frequency is often detected whether speech is present or not.

Speech with background music

The most suitable combination for separating speech from background music were; AP, ASC, AWF, STE, RMS, SF, fourth MFCC, AP(min), ZCR(min) and STE(min). This is a mix of descriptors from both time and frequency domain, with a majority in time domain. Descriptors in the frequency domain do not perform so well due to the fact that the power in the signal does not influence the frequencies much.

Speech with background environmental sounds

The most suitable combination for separating speech from background environmental sounds were only from the MFCC features. These are the first, fourth, fifth, sixth, eighth and ninth Mel-cepstrum coefficient. MFCC is features in the frequency domain. This means that the frequencies present in speech and environmental sounds differentiates more than the power of the signals. This is somewhat logical. Environmental sounds vary a lot in rhythm and frequency, and the power carried by the signal is usually lower than music. The descriptors in the time domain will therefore not detect much difference between speech and environmental sounds.

4.3.2 Pure speech and background music discrimination

In this experiment RMS, STE and ZCR was extracted from each of the constructed music samples. How these samples are constructed is described above and is illustrated in

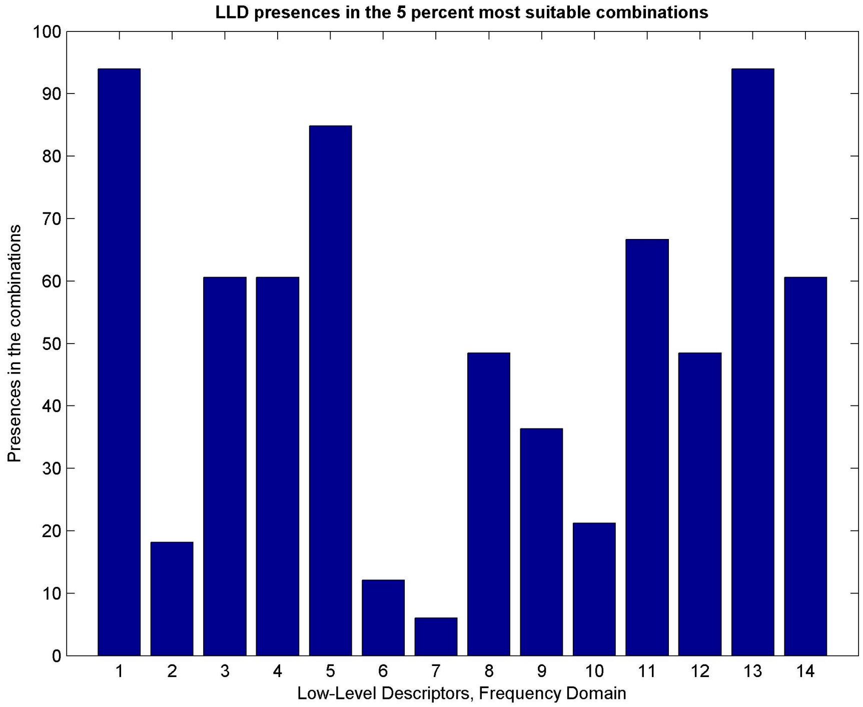

Figure 26: Most suitable LLD combination when PCA was applied 1: ASC, 2: FUF, 3: SF, 4-13: MFCC, 14: ASC(min).

Figure 22. The mean value of these LLD vectors is plotted together with the standard deviation and the mean LLD value for the speech samples. Two times the standard deviation is used as the confidence interval in this experiment because there are to few samples to calculate a 95 percent confidence interval with an acceptable error rate. The confidence interval for the speech signal is not plotted since the standard deviation were to small to be visible in the plot. Figure 27 illustrates the development of the RMS as the distance between the average speech dB level and the music dB level decreases. The lines crosses when the music signal is about 6 dB lower than the speech signal. If we consider the confidence interval, a distance of 8 db will separate all the test samples. This indicates that it is possible to separate speech with background music from clear music, as long as the distance between the powers of the music signal is 8 dB lower than the average power of the speech signal.

The development of the STE(Figure 28), as the distance between the average speech dB level and the music dB level decreases, is relatively similar to the RMS development. Main differences are that mean STE music line crosses the average STE speech line at -5 dB, and that the confidence interval is larger. When consider the confidence interval, a distance of 6 dB, between speech level and background music level, is necessary to separate speech with background music from clear music.

Figure 29 illustrates the development of the ZCR as the distance between the average speech dB level and the music dB level decreases. As visible in the figure, the ZCR line for the music signal does not change. This is because the ZCR feature is not influenced

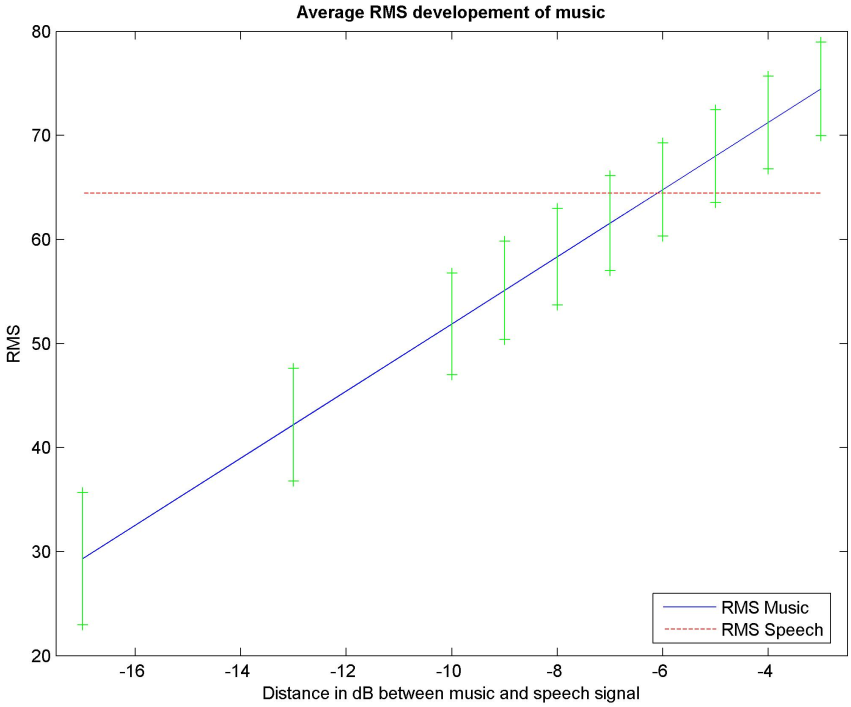

Figure 27: RMS developement of background music(Blue line) with standard deviation (Green) and average speech level (Red line)

by the signal power. One visible difference from the two other LLDs, is that the ZCR level for music is higher than speech. This is because the average frequency is higher for music than speech.

4.3.3 Pure speech and background environmental sounds discrimination

Figure 30 and Figure 31 illustrates the development of the RMS and STE as the distance between the average speech dB level and the environmental sounds dB level decreases. The results are very similar to the ones from the speech / music experiment. This is logical since these descriptors describe the power in the signal, and that it is well known that noisy environmental sounds is difficult to separate from music. This is because consistent environmental sounds and noise (especially white noise) carries much power and has a large frequency range. One difference is that STE for environmental sounds has a much larger standard deviation than for music. This can be explained by the fact that environmental sounds varies much more in beat, pattern and frequency than music.

The ZCR development, Illustrated in Figure 32, is very similar to the one for music. One noticeable difference is the level of the two ZCR lines. Environmental sounds have a higher level than music. The reason for this is probably the high frequency present in many of the environmental sound samples. The sound of rain falling has a very high frequency, and can be considered close to white noise.

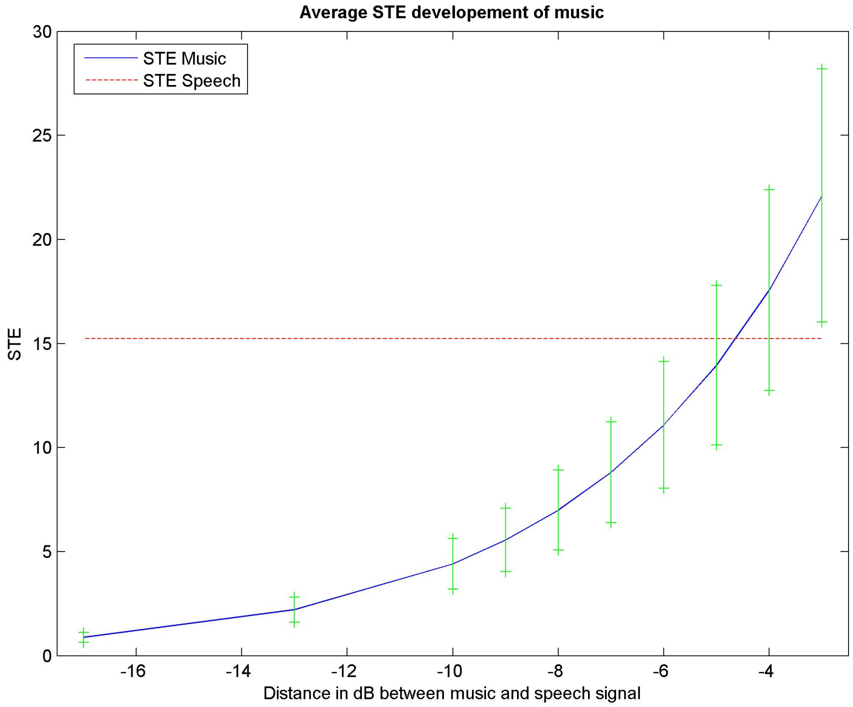

Figure 28: STE developement of background music(Blue line) with standard deviation (Green) and average speech level (Red line)

4.4 Classifier Evaluation

The KNN classifier was tested on the first 8 minutes of the movie Groundhog Day. The LLDs found to be the most suitable combination in the previous section, was extracted from the sample. Then PCA analyses were applied on these feature vectors, and reduced their dimensionality to 3 elements per vector. Finally the elements were grouped into two classes by the KNN. Figure 33 illustrates the result matched against the ground truth, which was a 88.25 percent match.

The first part of the sample (frame 1 - 38) is music that starts very slow and quiet and then picks up in power and pace. Since no training is applied to the KNN, some for the first frames are mistaken for speech (background music). Frame 39 - 163 (only interrupted by a jingle in frame 116 - 118) is clear speech, from four different speakers, with some environmental noise. Frame 164 - 212 is clear music. Frame 213 - 259 is speech from three different speakers, with background music. Frame 260 - 303 is clear music. Frame 304 - 356 is speech from three different speakers with background environmental sounds. Frame 357 - 372 is bad quality music (radio in background) with some environmental noises. Frame 373 - 400 is speech (radio hosts) in background.

To find the most suitable number of neighbors to consider, a classification was performed on K = 2 to K = 8. Figure 34 illustrates the result of this test. From the figure we can conclude that K = 4 gives the best result.

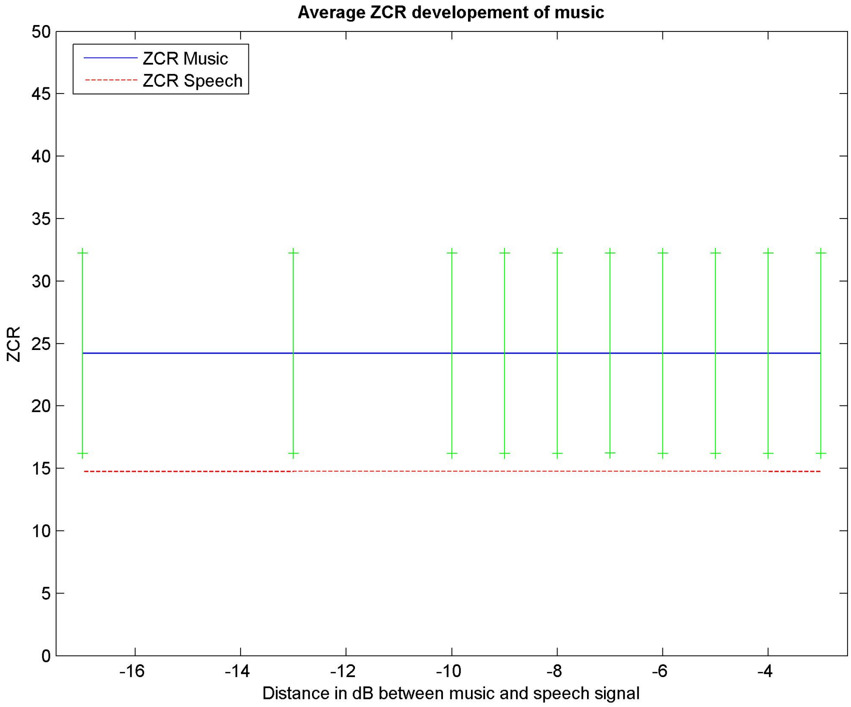

Figure 29: ZCR developement of background music(Blue line) with standard deviation (Green) and average speech level (Red line)

4.4.1 PCA dimensionality Reduction

The method for finding the number of dimensions returned from the PCA is straight forward. Since the most suitable combination of the LLDs contained 9 descriptors, the number of dimensions has to be between 1 and 9. Each number from 1 to 9 dimensions were returned from the PCA, classified by the KNN and checked against the ground truth. Figure 35 illustrates that 3 dimensions gives the best result.

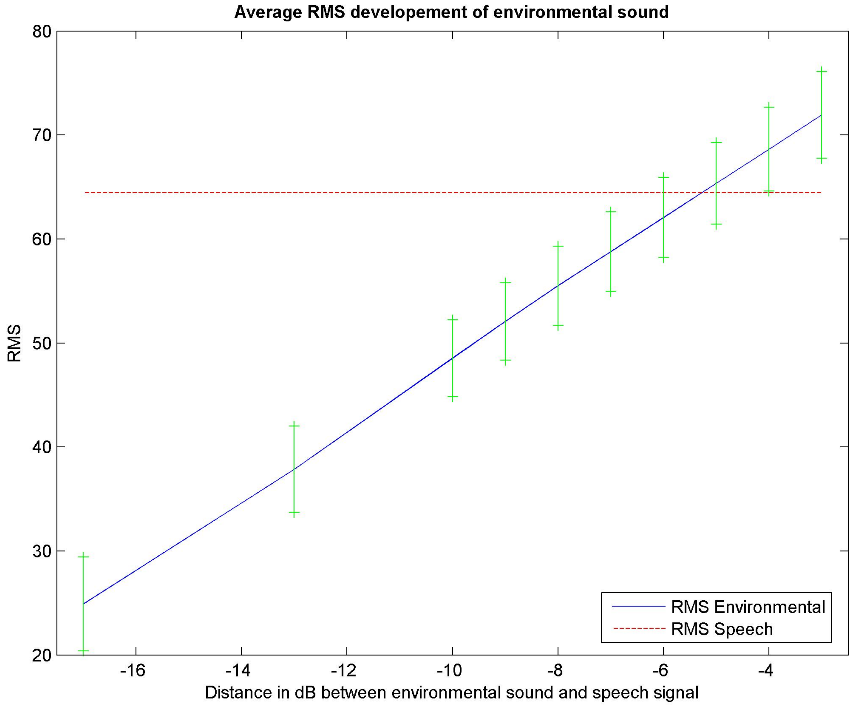

Figure 30: RMS developement of background environmental sounds(Blue line) with standard deviation (Green) and average speech level (Red line)

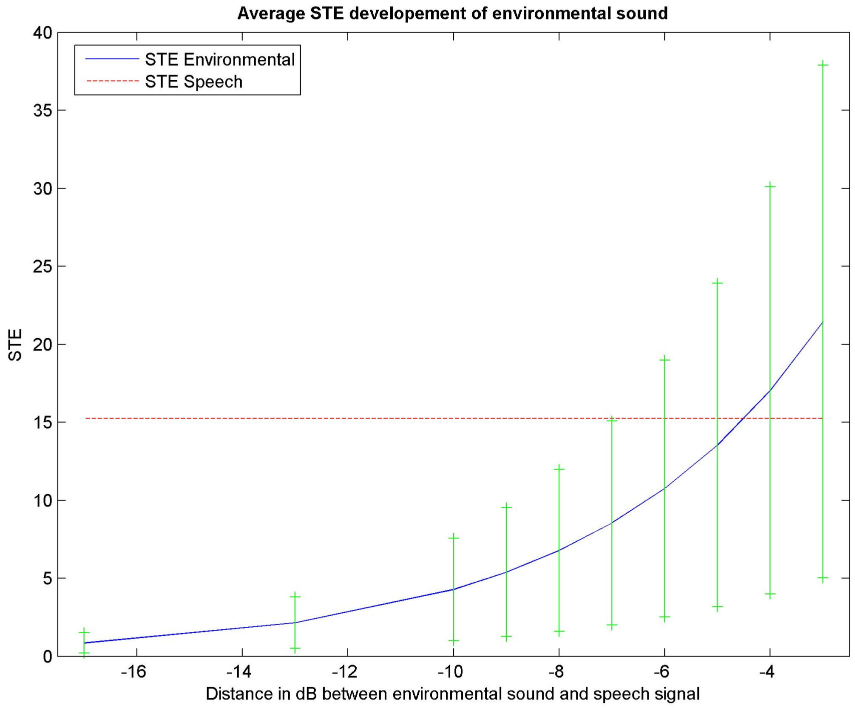

Figure 31: STE developement of background environmental sounds(Blue line) with standard deviation (Green) and average speech level (Red line)

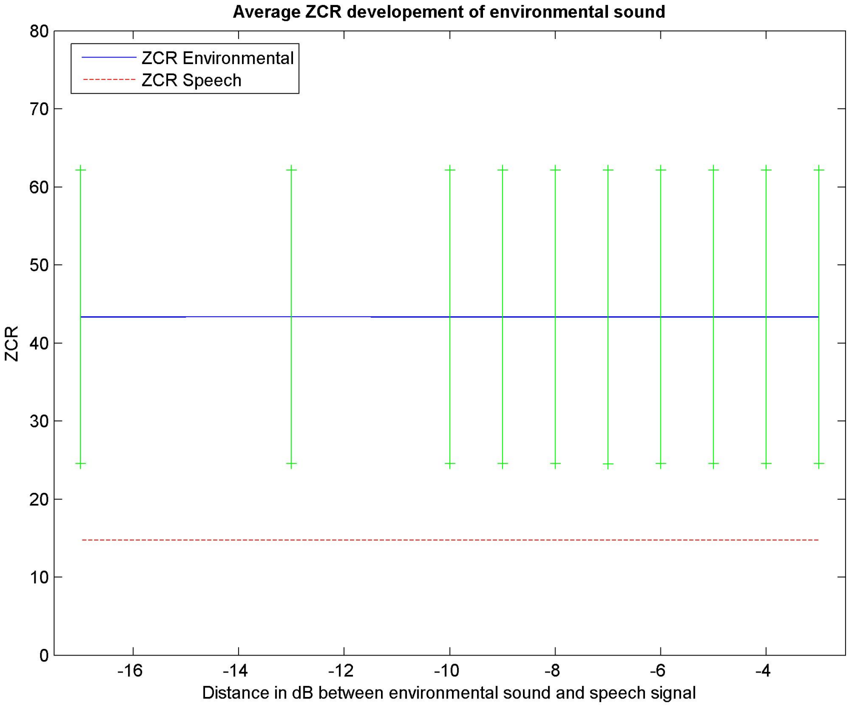

Figure 32: ZCR developement of background environmental sounds(Blue line) with standard deviation (Green) and average speech level (Red line)

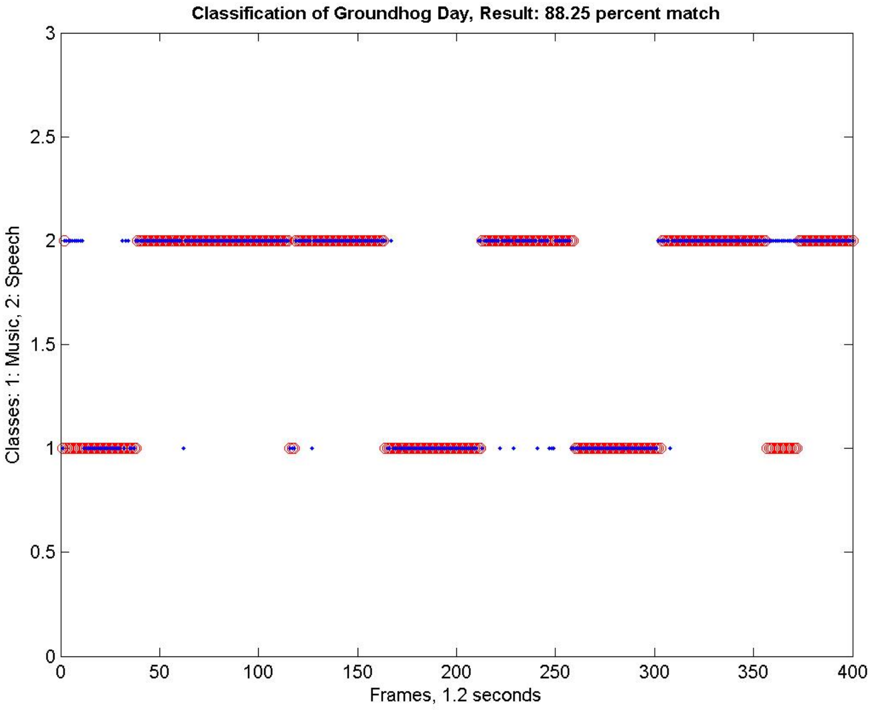

Figure 33: Results of KNN experiment on the first 8 minutes of 'Groundhog Day'. Red circles highlights the ground truth and blue dots highlights the classification don by the KNN

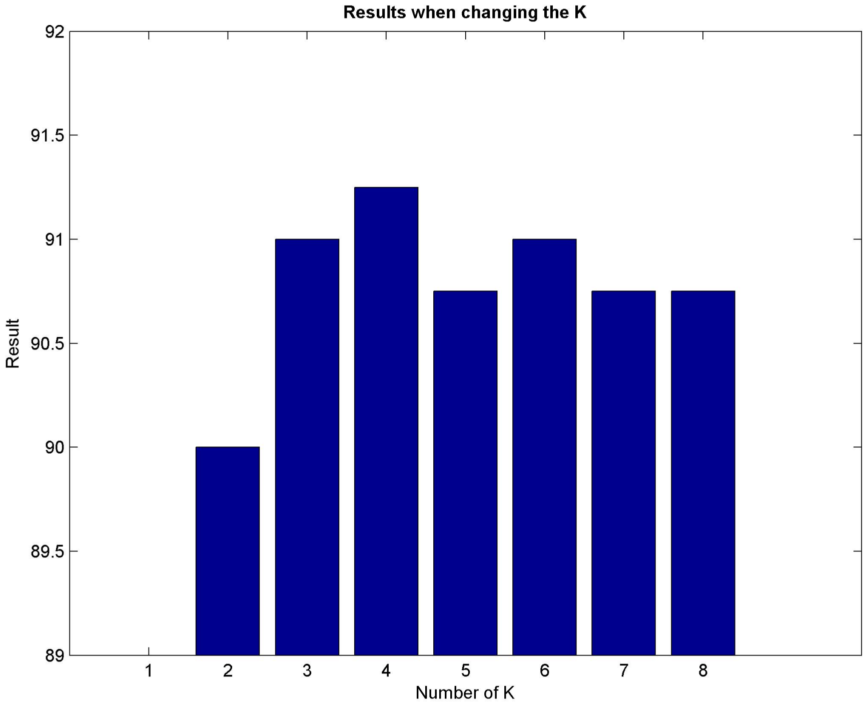

Figure 34: Results when changing the number of Ks in the KNN. Best result is achieved when 4 nearest neighbourer are used to classify the classes.

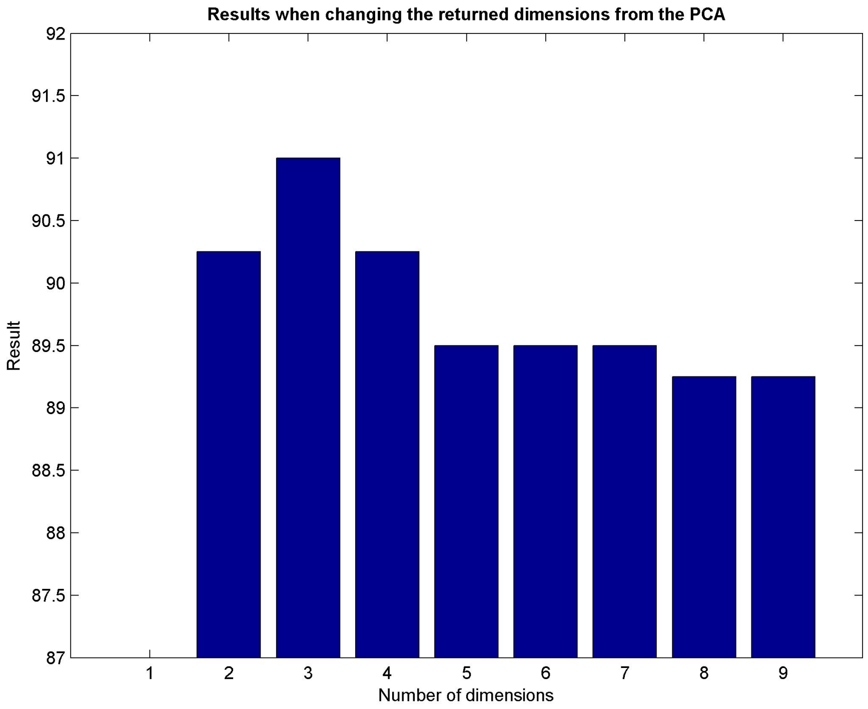

Figure 35: Results when changing the returned dimensions from the PCA. Best result is achieved when 3 dimensions is returned.

5 Experimental Result

In this final experiment all results from the previous experiments is used to classify the audio track from the ’Groundhog Day’.

5.1 System Setup

Figure 36 illustrates the system setup. First the signal is normalized in the preprocessing stage, and then all LLD features is extracted on a 30 ms basis and grouped into 1.2 seconds frames, as described in chapter 4, in the next stage. Then the frames are classified by the KNN in the third stage. In post processing stage is the ’silence frames’ classified by the RMS feature and single misclassified frames is re classified based on the four closest neighbors. In the final stage a table of content is created, based on the post processed result from the KNN.

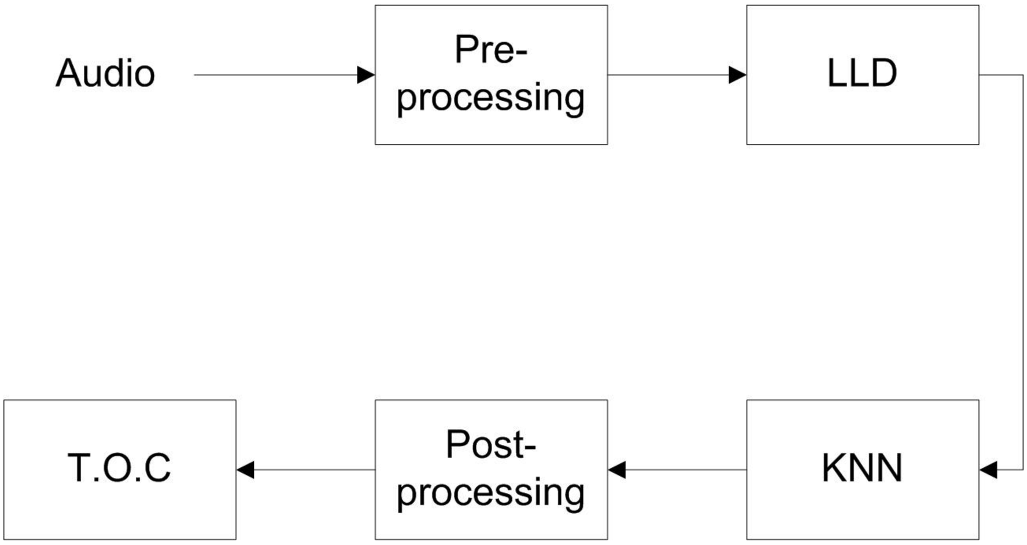

Figure 36: The five stages in the system setup

5.2 Audio Sample

The audio sample used in this final experiment is the audio track from the motion picture ’Groundhog Day’. The sample is 96 minutes long and includes both clear and mixed sections of speech, music, environmental sounds and silence. All 23 LLDs where extracted on 30 ms windows and grouped into a total of 4800 frames of 1.2 seconds. A Matlab function was created to generate the ground truth. The function played 1.2 seconds of the sample and then asked for the class, then next 1.2 seconds was played, and so on. The distribution, based on the ground truth, is 65.2 percent speech, 22.9 percent music and 11.9 percent environmental sounds.

5.3 Low Level Descriptors

The most suitable combinations of the LLDs found in section 4.3.1 were used on this experiment. In the earlier experiments the most suitable combination of the LLDs was very different when separating speech with background music and speech with background environmental sounds. But in this experiment the most suitable combination found for separating speech from background music was considerable better than the one for separating speech and environmental sounds. The reason for this is probably because both background music and environmental sounds were present in most cases in the 'Groundhog Day'. A separation based on the descriptors in the frequency domain was therefore not very suitable. Because of this, only the combination of the LLDs which was most suited for separating speech and background music was used.

5.4 Post processing

The post processing of the signal consists of two procedures. The first is for categorizing silence into the environmental sounds class. This is done by removing the 5 percent highest and lowest values from the RMS feature, and then normalize the vector. A threshold was then placed at 20 percent. Every frame with a value below the 20 percent where then considered to be silence. The other procedure was to correct the 'drop outs'. A median filter was first introduced, but this 'corrected' to many classifications. A function that looked at the two closest neighbors on both side where implemented. If both neighbors on both sides were of the same class, the investigated frame was moved to the same class.

5.5 Training of the KNN

The first 80 percent of the 'Groundhog Day' were used as training data for the KNN. Also the test samples of speech, music and environmental sounds were tested as training data for the KNN. These samples gave good classification where there were clear classes, or speech with background music, in the 'Groundhog Day' sample. But, these test samples failed dramatically when music and environmental sounds appeared simultaneously in the signal. A standard training set is theoretically possible, but it had to be a enormously big database of different music, speech and environmental sounds.

5.6 Classification of the 'Groundhog Day'

Much of the music in this movie is relatively quiet and without rhythmic instruments, such as drums and bass. Without these kinds of instruments, signal power will be significant lower. Because the combination of LLDs used in this experiment mostly are descriptors from the time domain, several music and environmental frames is misclassified as speech. Also environmental sounds are misclassified as speech. This is probably because of the same reason as for the speech and music separation, the lack of features from the frequency domain. But, if features from the frequency domain are included in the LLD combination, much of the speech with background music will be misclassified as music. Figure 37 illustrates the result of the classification.

5.7 ToC

The table of content will be described by the classification described in the previous section. Figure 38 illustrates a section from the classified sample, without the ground

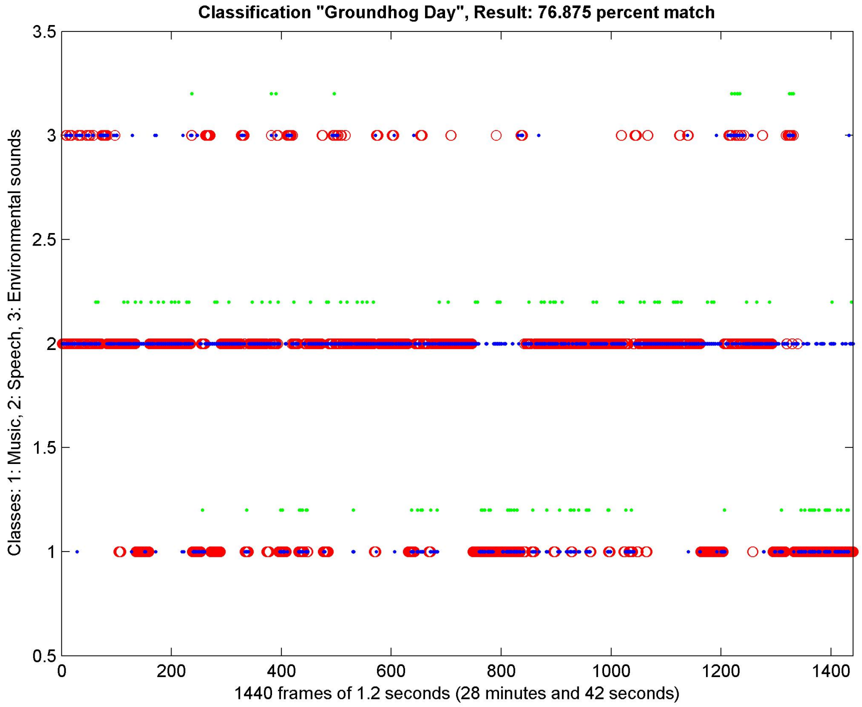

Figure 37: Experimental results, classification of the 'Groundhog Day'. Red circles is the ground truth, blue dots is the classification by the KNN and the green dots illustrated where the frames have been corrected by the post processing (The green dots is placed above the classification to them easier to observe).

truth and the corrections. The distribution, based on the result from the classification, is 72.2 percent speech, 20 percent music and 7.8 percent environmental sounds.

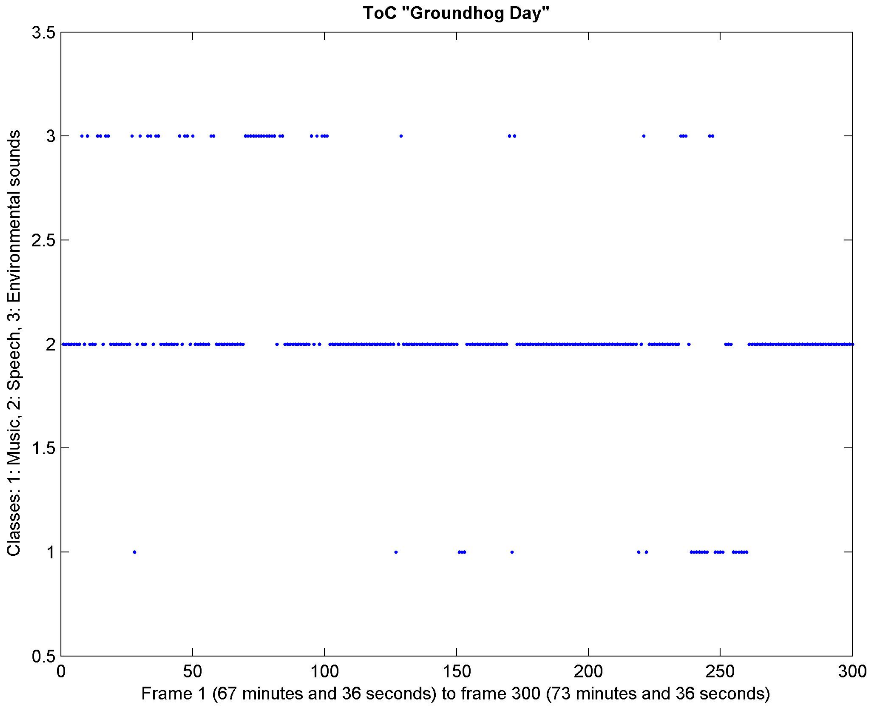

Figure 38: A section of the table of content generated by the automated classification of the ’Groundhog Day’

6 Conclusion

In this project we have investigated and implemented methods for content-based feature extraction and classification of motion picture audio. The system includes feature extraction, dimensional reduction and audio classification. Combination of these generates a table of content which describes the content of the speech, music and environmental sounds audio classes in the movie.

As expected, and mentioned in earlier chapters, the classification of motion picture audio needs a different approach from the one used in the separation of clear audio classes. The reason for this is that most of motion picture audio is mixed content. Music and environmental audio is used most of the time to create moods, and support the visual setting. This means that the frequency range and the power of the background music signal is added to the speech signal. In the Experimental Setup chapter, we found that there is still possible to separate speech with background music from clear music, if the background music signal is 8, or more, dB lower than the power of the speech signal. It is the features in the time domain that makes this possible. Tests showed that ZCR and FuF performed very poorly when trying to separate speech with background music from clear music. The reason for this is that the frequency present in the music signal is not particularly influenced by the power level of the signal. Because of this, most features in the frequency domain performs poorly when separating speech with background music from clear music.

When separating speech with background environmental sound and clear environmental sound, the results were opposite. In this case the features in the frequency domain performed well, and the features in the time domain performed poorly. The reason for this is that the average power carried in the environmental sounds signal, is lower than the music signal and is closer to speech. The power of the environmental sounds also tend to vary much more than music, this also resembles speech. But, in some cases the environmental sounds is closer to noise, and this tend to be misclassified as music, because of the wide frequency area (white noise is a well known example of this).

In the final experiment, when testing the audio track from the ’Groundhog Day’, the combination of LLDs that was most suitable for separating speech with background music from clear music performed much better than the combination for separating speech with background environmental sound from clear environmental sounds. One very likely explanation for this is that, based on the ground truth, only 11.9 percent of the audio track contained clear environmental sounds. The percentage of environmental sounds was even lower in the 20 percent of the audio track that were classified, only 7.8 percent. Another explanation is that the background environmental sounds are relatively constant in power regardless if speech is present or not. Background music tend to be faded down when speech is present, and then faded up again when speech is no longer present. Clear environmental sounds signals carry much power, and are therefore misclassified as music because of the majority of features from the time domain. To compensate for the many misclassifications of the environmental sounds a threshold on the RMS feature was introduced. After removing the upper and lower 5 percent of the RMS feature vector, the vector was normalized. Every value below 20 percent in this vector was then considered as silence. As explained earlier in this report a quiet environment is regarded as silence. Finally a variation of the median filter was used to correct misclassifications of single frames. This project has mainly focused on separating speech from other audio classes, and since 76.9 percent of the analysed test sample was correctly classified, the result has to be considered relatively good when 72.2 percent are judged to be speech by the ground truth.

To sum up the above, we can conclude that this project has proven that automatic classification is possible when analyzing audio with mixed content, if the distance between the dB levels of the speech signal is more than 8 dB from background music or environmental sounds. We have also proven that several of the low level descriptors that traditionally separates clear audio classes well, performs poorly on mixed audio classes.

6.1 Future Work

Experiments in this project has proven that LLDs that traditionally performs well when separating clear audio classes, performs poorly when separating audio classes with mixed content. Based on this, selecting LLDs that is more suited to separate mixed audio classes is probably the most relevant future task generated by this project. Results from the experiments shows that some LLDs gives good results in some variations, and performs poorly in other. Finding these variations could be a step in the right direction.

One of the obvious future tasks based on this project is to perform the experiments in a larger scale. The test samples used is too few to generate accurate results. Repeating the experiments with different window and frame sizes when extracting the low level descriptors may give different results. Another task could be to analyze further the classified classes. Speech can be processed to separate speech with background music from speech with background environmental sound. In the experiment where most suitable combinations of the LLDs were found, the result showed that there are large differences in samples of background music and background environmental sounds.

Also known methods for detecting sound objects should be possible to implement. When sound objects is present in a motion picture, the power of this signal is usually much higher than other sounds present.

Bibliography

[1] S. Pfeiffer, S. Fischer, andW. Effelsberg, "Automatic audio content analysis", in Proc. ACM Int. Conf. Multimedia, Boston, MA, Nov. 1996

[2] Lie Lu, Hong-Jiang Zhang, Senior Member, IEEE, and Hao Jiang, "Content analysis for audio classification and segmentation ", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 10, NO. 7, OCTOBER 2002

[3] J. Saunders, "Real-Time Discrimination of Broadcast Speech/Music ", Proc. ICASSP96, vol.11, pp.993-996, Atlanta, May, 1996

[4] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/Music Discrimination for Multimedia Applications", Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (Istanbul), pp. 2445-2448, June 2000.

[5] L. Lu, H. Jiang, and H. J. Zhang, "A Robust Audio Classification and Segmentation Method ", in Proc. 9th ACM Int. Conf. Multimedia, 2001, pp. 203-211.

[6] J. P. Campbell, Jr., "Speaker recognition: A tutorial", Proc. IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[7] Hyoung-Gook Kim, Nicolas Moreau, and Thomas Sikora, "MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval", Wiley and Sons, October 2005 - ISBN 0-470-09334-X

[8] Dan Ellis, "RASTA/PLP/MFCC feature calculation and inversion", http://www.ee.columbia.edu/ dpwe/resources/matlab/rastamat/

[9] Tim Pohle, "Extraction of Audio Descriptors and Their Evaluation in Music Classification Tasks", Diplomarbeit am Fachbereich Informatik der Technischen Universitat Kaiserslautern, January 2005

[10] M. F.McKinney and J. Breebaart. Features for audio and music classification. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR 03), Baltimore, MD, USA, October 26-30 2003.

[11] S. Dixon, E. Pampalk, and G. Widmer. Classification of dance music by periodicity patterns. In Proceedings of the Fourth International Conference on Music Information Retrieval (ISMIR 03), pages 159-166, Baltimore, MD, USA, October 26-30 2003. John Hopkins University

[12] D. Liu, L. Lu, and H.-J. Zhang. Automatic mood detection from acoustic music data. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR 03), Baltimore, MD, USA, October 26-30 2003.

[13] T. Ganchev, N. Fakotakis, and G. Kokkinakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in 10th International Conference on Speech and Computer (SPECOM 2005), vol. 1, 2005, pp. 191-194.

[14] Young, S.J., Odell, J., Ollason, D., Valtchev, V., Woodland, P., 'The HTK Book. Version 2.1', Department of Engineering, Cambridge University, UK, 1995.

[15] E. Scheirer and M. Slaney, 'Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator,' in Proc. ICASSP'97, Munich, Vol. II, pp. 1331-1334, 1997.

[16] Rosenberg A. et al.: 'Cepstral channel normalization techniques for HMM-based speaker verification', Proc. ICSLP -94, pp. 1835-1838.

[17] S.B. Davis and P. Mermelstein (1980), 'Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences', IEEE Trans. on ASSP 28, 357-366.

[18] Zheng F., Zhang, G., Song, Z., 'Comparison of different implementations of MFCC', J. Computer Science and Technology, 16(6):582-589, Sept. 2001.

[19] Shannon B.J., Paliwal K.K., 'A comparative study of filter bank spacing for speech recognition', Proc. of Microelectronic engi-neering research conference, Brisbane, Australia, Nov. 2003.

[20] Skowronski, M.D., Harris, J.G., 'Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition', Journal of the Acoustical Society of America, 116(3):1774-1780, Sept. 2004.

[21] S. S. Stevens and J. Volkmann, and E. B. Newman, 'A Scale for the Measurement of the Psychological Magnitude Pitch', The Journal of the Acoustical Society of America, pp. 185-190 ,January 1937

[22] Liu, Z., Y. Wang, and T. Chen. 'Audio feature extraction and analysis for scene segmentation and classification,' J. VLSI Signal Processing Syst. Signal, Image, Video Technol., vol. 20, pp. 61-79, Oct. 1998.

[23] Brian Whitman. 'SEMANTIC RANK REDUCTION OF MUSIC AUDIO',2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, October 19-22, 2003, New Paltz, NY

[24] A. Ghias, J. Logan, D. Chamberlain, and B.C. Smith. Query by humming: Musical information retrieval in an audio database. In Proceedings of Third ACM International Conference on Multimedia, pages 231-236, Anaheim, CA, November 1995.

[25] Guy J Brown and Martin Cooke. Computational auditory scene analysis. Computer Speech and Language, (8):297-336, August 1994.

[26] Alon Fishbach. Primary segmentation of auditory scenes. In Intl. Conf. on Pattern Recognition ICPR, pages 113-117, 1994

[27] Stephen W. Smoliar. In search of musical events. In Intl. Conf. on Pattern Recognition, pages 118-122, 1994.

[28] J. Nam, A. Cetin, and A. Tewfik, ’Speaker identification and video analysis for hierarchical video shot classification,’ in Proceedings of IEEE International Conference on Image Processing, vol. 2, Santa Barbara, CA, Oct. 1997, pp. 550-555.

[29] S. Srinivasan, D. Ponceleon, A. Amir, and D. Petkovic, ’What is that video anyway? In search of better browsing,’ in Proceedings of IEEE International Conference on Multi- media and Expo, July 2000, pp. 388-392.

[30] M. R. Naphade and T. S. Huang, ’Stochastic modeling of soundtrack for eficient segmentation and indexing of video,’ in Proceedings of SPIE Storage and Retrieval for Multimedia Databases, vol. 3972, Jan. 2000, pp. 168-176.

[31] M. Akutsu, A. Hamada, and Y. Tonomura, ’Video handling with music and speechdetection,’ IEEE Multimedia, vol. 5, no. 3, pp. 17-25, 1998.

[32] D. Ellis, ’Prediction-driven computational auditory scene analysis,’ Ph.D. dissertation, MIT, Cambridge, MA, 1996.

[33] P. Jang and A. Hauptmann, ’Learning to recognize speech by watching television,’ IEEE Intelligent Systems Magazine, vol. 14, no. 5, pp. 51-58, 1999.

[34] T. Zhang and C. Kuo, ’An integrated approach to multimodal media content analysis,’ in Proceedings of SPIE, IS and T Storage and Retrieval for Media Databases, vol. 3972, Jan. 2000, pp. 506

[35] Derek Hoiem, Yan Ke, Rahul Sukthankar,’SOLAR: Sound Object Localization and Retrieval in Complex Audio Environments’ School of Computer Science, Carnegie Mellon University; Intel Research Pittsburgh

[36] A. Dufaux, L. Besacier, et al., ’Automatic Sound Detection and Recognition for Noisy Environment,’ Proc. of the X Euro-pean Signal Processing Conference, 2000.

[37] Marina Bosi, Richard E. Goldberg and Leonardo Chiariglione, ’Introduction to Digital Audio Coding and Standards’, Kluwer Academic Publishers, 01 December, 2002.

[38] M. Casey, ’Reduced-Rank Spectra and Minimum-Entropy Priors as Consistent and Reliable Cues for Generalized Sound Recognition,’ Workshop for Consistent and Reliable Acoustic Cues, 2001.

[39] G. Guo and S. Li, ’Content-Based Audio Classification and Retrieval by Support Vector Machines,’ IEEE Trans. on Neural Networks, 2003.

[40] G. Li and A. Khokhar, ’Content-based Indexing and Re-trieval of Audio Data using Wavelets,’ IEEE Int. Conf. on Mul-timedia and Expo, 2000.

[41] S. Li, ’Content-Based Classification and Retrieval of Audio Using the Nearest Feature Line Method,’ IEEE Trans. on Speech and Audio Processing, 2000.

[42] E. Wold, T. Blum, et al., ’Content-Based Classification, Search, and Retrieval of Audio,’ IEEE Multimedia, 1996.

[43] Wang E.,Liu Z. and Huang J.-C, ’Multimedia Content Analysis Using Both Audio and Visual Cues’, IEEE Signal Processing Magazine, vol 17, no. 6. pp 12-36, 2000