Technical Report

CSIRO Mathematical and Information Sciences

Formalisation of MPEG-1 compressed domain audio features

By Silvia Pfeiffer and Thomas Vincent

18th December 2001

Report Number:
01/196

CSIRO Mathematical and Information Sciences
Locked Bag 17, North Ryde NSW 1670
Australia
Contact: Silvia.Pfeiffer@csiro.au

TABLE OF CONTENTS

1 Introduction

In this paper we give an overview of existing audio content analysis approaches in the compressed domain and incorporate them into a coherent formal structure. We first examine the kinds of information accessible in an MPEG-1 compressed audio stream and describe a coherent approach to determine features from these. These features are generic enough to be further processed with standard audio content analysis approaches. We report on a number of applications that have been presented making use of the compressed domain features. Most of them aim at creating an index to the audio stream by segmenting the stream into temporally coherent regions, which are often classified into a pre-specified set of classes. We also discuss recognition and identification applications.

2 MPEG-1 compressed data

To understand the kind of features that can be extracted from an MPEG-1 compressed audio stream, we have to understand the meaning of the encoded fields (see [HAS97, ISA93, NUL97, PAN95]). To that end, we first briefly explain the encoding steps and the resulting field structures and then explain which fields contain useful information for content analysis.

2.1 MPEG-1 audio encoding

MPEG-1 audio encoding comes in three different flavours called Layers. They increase in complexity from Layer 1 to 3 yet all follow the same processing steps:

1. The sampled sound data is broken up into analysis windows and transformed into the frequency domain. A polyphase filterbank calculates 32 frequency band magnitudes (called **subband values**) for each of the three Layers, which is further refined to 576 subbands for Layer 3 only.

2. The resulting subband values are manipulated according to psychoacoustic models and the desired bitrate. The aim is to filter out sounds that are masked by other sounds and to arrive at a perceptually lossless compression. The extent of compression achieved by this psychoacoustic filtering is encoder-specific and not standardised.

3. The remaining subband magnitudes are linearised into a bitstream according to the bitstream format standardised for the respective Layer. In this last step, further compression can be done (such as Huffman encoding) which is lossless and exploits redundancies of the data contained within several successive analysis windows. The data is then encoded in a so-called audio frames. The resulting files therefore consist of a sequence of (audio) frames containing Layer-specific fields.

Figure 1 displays the transformation that a sequence of 1152 PCM samples go through during encoding. Layers 1 and 2 stop after the first transform (a polyphase filterbank), while Layer 3 goes through an additional Modified Discrete Cosine Transform (MDCT) step. Access of

transformation coefficients in Layer 3 can therefore be either at the filterbank or the MDCT level.

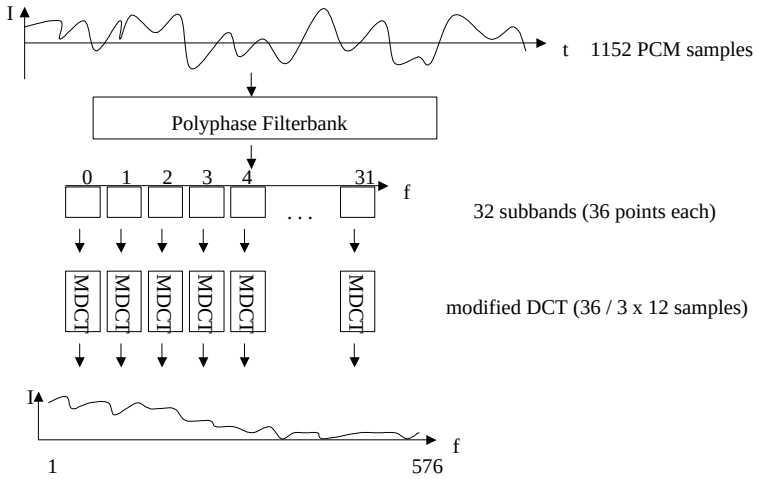

Figure 1: Frequency transformations of Layer 3

Figure 2 displays the frame formats of all three Layers. The 32 subband values are encoded in groups of 12 (Layer 1 & 2) or 18 (Layer 3) subband samples. We call these groups **granules**. There is only one such granule in a Layer 1 frame, whereas a Layer 2 frame contains three granules and a Layer 3 frame two granules to exploit further redundancies. A granule in Layer 3 can be viewed as either consisting of 18 values in each of 32 subbands or of one value in each of 576 subbands depending on whether one accesses the filterbank coefficients or the MDCT coefficients.

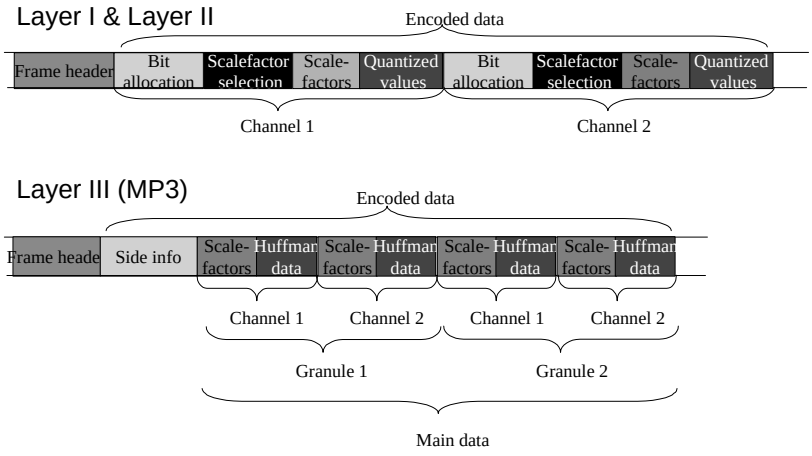

Figure 2: Frame formats for all three Layers

Except for the number of granules, Layer 1 and 2 encodings are the same. Their subband values are encoded in the quantised values field after having being normalised with scalefactors. The number of bits required for the quantised values is encoded in the bit allocation field. Additionally, the scalefactor selection field used by Layer 2 stores how many of the three scalefactors of each granule were different and thus had to be encoded.

Layer 3 is different. As mentioned above, it results in *576* subband values. These are further compressed mainly by use of a Huffman compression scheme after careful grouping of subbands (see Figure 3).

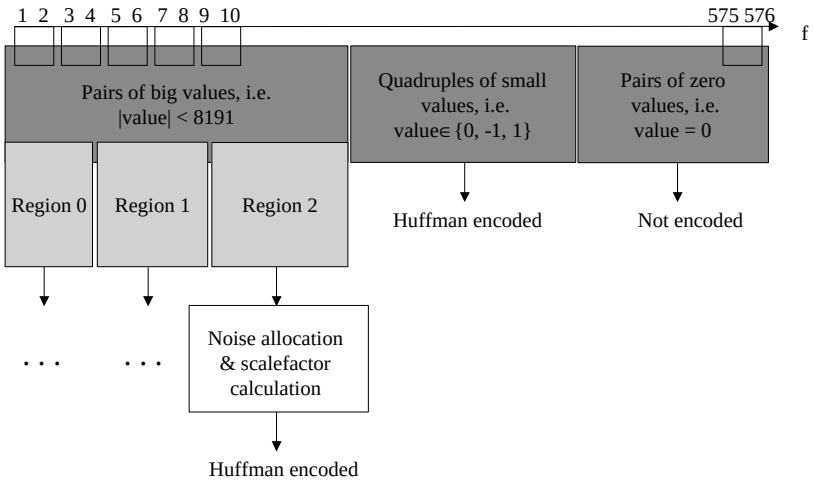

Figure 3: Layer 3 encoding of frequency bands

2.2 Field information

Without going back to decoding an MPEG audio file to PCM samples, there are two types of information that can be used as features on which to base audio content analysis approaches: the information encoded in the header-like fields (header, bit allocation, scalefactor selection, scalefactors, side information) and the encoded subband values.

2.2.1 Header-type information

Wang and Vilermo [WAN01] have used the **window type information** of Layer 3 to detect beats. Layer 3 uses four different kinds of analysis windows: long, long-to-short, short, and short-to-long. The short windows are used for short but intensive sounds for which the long window would introduce too much pre-echo. They found that the window-switching pattern of pop-music beats for their specific encoder at bitrates of 64-96 kbps gives (long, long-to-short, short, short, short-to-long, long) window sequences in 99% of the beats.
Our own research has also examined the header-type fields [BAR97]. We have used the following fields of Layers 1 & 2 with the given feature interpretation:

- **Bit allocation**: stores the dynamic range of a sequence of subband values.
- **Scalefactors**: stores information on the maximum loudness of a sequence of subband values.
- **Scalefactor selection information** (Layer 2 only): stores how the loudness changes on three subsequent groups; a value of 0 indicates no change, so the loudness is stable, a value of 2

indicates a transient change, and the values 1 and 3 indicate an unstable change.

2.2.2 Subband values

Basically all other published research on compressed -domain audio analysis has used the **subband values** as a starting point for feature calculation. Thus it is required to decode the MPEG audio stream enough to access the subband values. For all three Layers, the subband values are **not** available directly in the linearised file/stream but have to be reconstructed from the encoded fields implying some processing cost. The most time consuming step for decoding an MPEG audio stream is however the resynthesis of PCM samples and this is avoided as the subband values are still in the compressed domain.

In Layers 1 and 2 the subband values may be approximated by directly using the quantised values in an encoded frame (which can only be extracted from the file with help of the bitallocation information). This however ignores the fact that the values are normalised by the scalefactors in each of the *32* subbands. So, to arrive at the subband values encoded in the file, one has to use the quantised values and denormalise them.

In Layer 3 there are *576* subband values. To extract the frequency band magnitudes from the file, it is necessary to decode the quantised samples with a Huffman decoder. Then, scalefactors have to be readjusted, which served to colour the quantisation noise, and quantisation has to be reversed. After this, one reaches the alias reduced MDCT coefficients, which we call the *576* subband values. To achieve features which are comparable independent of the encoded Layer, it is possible to further decode the 576 MDCT coefficients to the original 32 Polyphase Filterbank coefficients, but there is a processing cost associated and a loss of frequency resolution. One however gains on temporal resolution, which might be more appropriate in certain application areas.

Subband values are in the following denoted by $s_i(n)$, i being the subband number, $0 \le i \le I-1$, ($I=32$ in Layers 1 and 2, and possibly *576* in Layer 3) and n the time index. In the following, all index values will start with 0. The time index n is viewed from a whole file perspective. As an example, if we take a Layer 2 encoded audio file, its 5th frame, its 2nd granule, and the 5th subband value (out of the 12) of the 10th subband, we access the value $s_9(160)$ (see Figure 4).

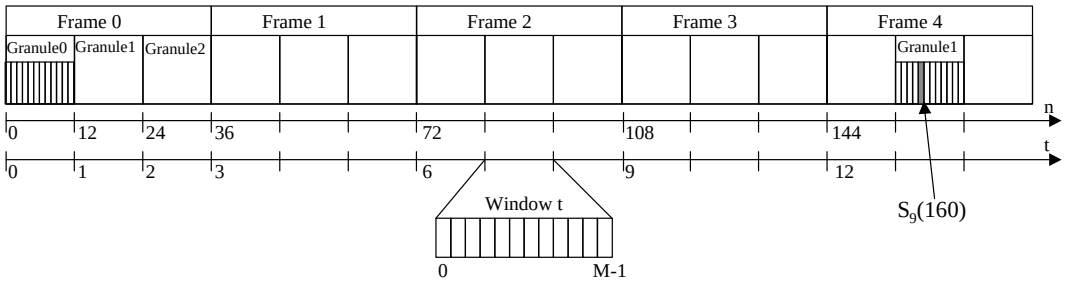

Figure 4: Explanation of subband numbering scheme.

Depending on the required resolution of an analysis, one may choose to calculate features on the subband value resolution level, on the granule resolution level, or on the frame resolution level. A statistical analysis on a larger time window may thus be calculated on a multiplicity of any of these resolution levels. In our own research we have chosen the subband value or granule resolution, while many others prefer the frame resolution.

Most analysis algorithms work on a window of samples. Independent of the choice of resolution, we denote the window size by M and the time position within a window by m, $0 \le m \le M-1$. t will denote the window number while goin g over a file, which is closely related to the time position within a file. A subband value at window position m is accessed depending on the choice of resolution for analysis. For example, if we select a granule as window size, have consecutive non-overlapping windows only, and work on a subband value resolution level, the above subband value $s_9(160)$ will be in window number t=13 at position m=4 (M=12 is the implied window size for the Layer 2 granule here) (see Figure 4).

Synopsis of used terms:

n	Time index $0 \le n \le N-1$
i	Subband index $0 \le i \le I-1$
$s_i(n)$	Subband value at time index n for subband I
m	Time position within window $0 \le m \le M-1$
M	Analysis window size
t	Analysis window number

3 Low-level audio features

Going from the subband values to high-level analysis such as segmentation, classification, recognition and identification, requires firstly the calculation of low-level audio features on which the further analysis will be based. Most often information on the subband energies is used as a starting point for the analysis of the features.

3.1 Pre-processed subband information

Instead of using the subband values themselves to access subband energies, some researchers preprocess them to achieve different goals.

- Tzanetakis et al. [TZA00] use the **root mean squared (RMS) subband vector** on a frame resolution because this is a better measure of signal energy than the subband values themselves. A generalised formula for their approach, independent of granule, frame or any larger window, is given by:

  $$\overline{s_i}(t) = \sqrt{\frac{1}{M}\sum_{m=0}^{M-1} s_i^{\,2}(Mt+m)}.$$

  This enables one to arrive at *32* or *576* subband values by averaging the *M* subband values in the window.

- Boccignone et al. [BOC99] use the **subband mean** value in a window of size *M* as feature as a way to go to a frame resolution. A generalised formula for their approach, independent of granule, frame or any larger window, is given by:

$$\mu_i(t) = \frac{1}{M}\sum_{m=0}^{M-1} s_i(Mt+m).$$

- Nakajima et al. [NAK99] calculate the **normalised subband energy** from the subband samples of a frame to absorb sound level dependency on audio source. The following formula normalises a single subband value on the maximum of all subband values at the same time index *n*:

$$\varsigma_i(n) = 10\log_{10}\left(\frac{s_i^2(n)}{\max(s_j^2(n): 0 \le j \le I-1)}\right).$$

  Going to a lower resolution for a window of size *M*, we have also used the subband mean a s a basis for calculation of a normalised subband energy:

$$\varsigma_i(t) = 10\log_{10}\left(\frac{\mu_i^2(t)}{\max(\mu_j^2(t): 0 \le j \le I-1)}\right).$$

- In our own work we have made use of the scalefactors in Layer 1 and 2 for a granule resolution subband energy measure [BAR97]. The **scalefactors** are the maximum value of the sequence of subband values within a granule:

  $scf_i(t) = \max(|\, s_i(Mt+m)\,|: 0 \le m \le M-1)$, with $0 \le i \le I-1$.

The two values $s_i(n)$ and $\varsigma_i(n)$ work at the subband value resolution, while $\overline{s_i}(t)$, $\mu_I(t)$,$\varsigma_i(t)$ and $scf_i(t)$ work on a granule, frame or any larger window resolution. In the following subsections, any of these values may be used in the formulas interchangeably. The choice depends on the goals of the analysis. As a placeholder we will use $s_i(t)$.

3.2 Cepstral features

For speech and speaker recognition approaches it is standard to use a representation of the audio signal in the frequ ency domain as feature. Cepstral coefficients have proven to be particularly successful. They are calculated by performing another frequency transform on the logarithm of spectral coefficients. Both the *32* subband values of Layer 1 and 2, and the *576* subband values of Layer 3 are linearly spaced spectral coefficients and serve well as a basis for calculation of cepstral coefficients.

- Venugopal et al. [VEN99] calculate **linear frequency cepstral coefficients** from the subband values and use them for speaker identification.

- Yapp et al. [YAP97] use the quantised values of the first nine subbands for their speech recognition system. To reach a higher frequency resolution they apply the Fast Fourier

Transform (FFT) on subband windows of size *25 ms* using a Hamming window and padding the number of samples to a power of *2*. They take the magnitude of the FFT values and assemble them over the subbands into one single frequency vector. This feature vector is now mel-warped and used to calculate **cepstral, delta cepstral, and acceleration coefficients** as features.

3.3 Energy features

Signal energy features are closely related to the human loudness perception. When calculating energy features in the compressed domain rather than from uncompressed PCM samples, the results are closer approximations of perceptual loudness because the subband values have been filtered by the psychoacoustic model and thus the influence of non-hearable frequencies is reduced. The disadvantage however is that signal energy is distributed over the frequency bands and thus has to be added up requiring higher computational complexity than on uncompressed signals. Signal energy measures are often used for segmentation of an audio stream. A signal's start and end times are then usually determined by thresholding.

- Patel et al. [PAT96] calculate **signal energy** for a window of size *M* from the subband values. Nakajima et al. [NAK99] restrict their loudness measurement to the energy in the lowest subband as this is the one where most energy is concentrated and this restriction provides a considerable efficiency increase. A generalised formula for signal energy is given by:

  $$E(t) = \frac{1}{I \cdot M} \sum_{i=0}^{I-1} \sum_{m=0}^{M-1} s_i^{\,2}(Mt+m) \cdot h(M-1-m).$$

  The window function *h(m)* may be e.g. a Rectangular, Hamming, Hanning, Welch or Bartlett window depending on the required narrowness or peakness of spectral leakage.

- Tzanetakis et al. [TZA00] prefer to use the **RMS of the signal energy** for loudness approximation which achieves a better separation for low level values.

- Another loudness approximation is the **signal magnitude**, which is less sensitive to noise than signal energy. It can be calculated analogously to signal energy [PAT96] via:

  $$M(t) = \frac{1}{I \cdot M} \sum_{i=0}^{I-1} \sum_{m=0}^{M-1} |\, s_i(Mt+m) \,| \cdot h(M-1-m).$$

- In our own work we have used the **sum of scalefactors** on a granule resolution for a fast approximation of the signal magnitude on Layer 1 & 2 frames [BAR97, PFE99].

- Wang and Vilermo [WAN01] calculate the **band energy** of several subbands and use a threshold on them to determine a confidence for a pop-music beat in the granule:

  $$E_{(J1,J2)}(t) = \sum_{i=J1}^{J2} \sum_{m=0}^{M-1} s_i^{\,2}(Mt+m) \cdot h(M-1-m).$$

3.4 Silence statistics

Silence statistics are often used as indicators for classification of audio segments into different signal classes. Speech segments for example generally contain a lot more silence than music segments.

- Therefore, Patel et al. [PAT96] propose **pause rate** as an indicator to separate speech from non-speech signals:
  $$P(t) = \frac{1}{M}\sum_{m=0}^{M-1}(E(Mt+m) > T_s) \wedge (E(Mt+m+1) > T_s)$$
  with T_s as the silence energy threshold. *P* counts the number of silent segments on a time interval of size *M*.

- Similarly, Tzanetakis et al. [TZA00] use a **low energy** feature to separate speech from music. On a window of about *1 sec* (*M=40* frames) they calculate the percentage of frames that have less than the average energy $\overline{E(t)}$:
  $$L(t) = \frac{1}{M}\sum_{m=0}^{M-1}(E(Mt+m) < \overline{E(t)}).$$

- Nakajima et al. [NAK99] call their silence statistic **energy density**. It is also defined on a window of *1 sec* and basically calculated as the log value of the variance of *L(t)*.

3.5 Spectral energy statistics

Spectral energy statistics capture subband energy distribution features, which are indicative for specific types of sounds.

- The **spectral centroid** is the balancing point of the subband energy distribution [BAR97, TZA00]. It is thus calculated as the first moment of the subband energy distribution:
  $$C(t) = \frac{\sum_{i=0}^{I-1}(i+1)s_i(t)}{\sum_{i=0}^{I-1}s_i(t)}.$$
  It determines the frequency area around which most of the signal energy concentrates and is thus closely related to the time-domain zero crossing rate (ZCR) feature often used in speech recognition systems to determine exact start- and endpoints of talkspurts. It is also frequently used as an approximation for a perceptual brightness measure [BAR97]. Nakajima et al. [NAK99] use the squared subband samples in their spectral centroid calculation to better spread out the centroid values.

- The s**pectral rolloff point** R is determined where *85%* of the window's energy is achieved:
  $$\sum_{i=0}^{R}s_i(t) = 0.85 \cdot \sum_{i=0}^{I-1}s_i(t).$$

It is used to distinguish voiced speech from unvoiced speech and music [TZA00], which have a higher rolloff point because their power is better distributed over the subband range.

- Patel et al. [PAT96] propose a feature called **band energy ratio**, which sets the energy of the low frequencies (subbands *0* to *J-1*) in relation to the high frequencies (subbands *J* to *I-1*):

$$B(t) = \frac{\sum_{i=0}^{J-1}\sum_{m=0}^{M-1} s_i^{\,2}(Mt+m)\cdot h(M-1-m)}{\sum_{i=J}^{I}\sum_{m=0}^{M-1} s_i^{\,2}(Mt+m)\cdot h(M-1-m)}.$$

This is indicative of the voicedness of a sound. They claim that *J=2* is a good choice because voiced signal energy concentrates below *1.5kHz* while unvoiced signal energy is distributed over all subbands.

- The **spectral flux** of two successive windows *t* and *t+1* is calculated as the 2-norm of the difference between normalized subband value vectors at *t* and *t+1* [TZA00]:

$$\Delta(t,t+1) = \sqrt{\sum_{i=0}^{I-1} \left| \frac{s_i(t)}{\max(s_j(t): 0 \le j \le I-1)} - \frac{s_i(t+1)}{\max(s_j(t+1): 0 \le j \le I-1)} \right|^2}.$$

While the first three statistics calculated spectral energy distribution features on one window, the spectral flux determines changes of spectral energy distribution of two successive windows.

- The **subband central moments** calculated by Boccignone et al. [BOC99] on the contrary calculate statistics within subbands over several frames. They capture how much a subband's energy is dispersed from its mean:

$$D_i^k(t) = \frac{1}{M}\sum_{m=0}^{M-1} (s_i(Mt+m) - \mu_i(t))^k \quad \text{with } k=2,...,5.$$

3.6 Bandwidth features

- The **bandwidth** covered by a window is calculated from all subbands with sufficient energy:

$$BW(t) = \max(i : (0 \le i \le I-1) \wedge (s_i(t) > T_s)) - \min(i : (0 \le i \le I-1) \wedge (s_i(t) > T_s)).$$

It is stipulated that the bandwidth of speech is usually narrower than that of music [NAK99, VEN99].

- Nakajima et al. [NAK99] propose to determine bandwidth information by counting the **number of subbands with significant level**:

$$SB(t) = \sum_{i=0}^{I-1} (s_i(t) > T_s).$$

If a window's content covers a lot of subbands such as music, *SB(t)* becomes large. In addition to measuring the subband range that a window contains, this also takes into account how strong the subbands in between are represented.

3.7 Pitch features

Pitch is indicative of a speaker and thus an important property of a sound.

- Patel et al. [PAT96] calculate the **pitch** of a sound signal in the compressed domain by using the autocorrelation function of the values of the first subband on *30%* overlapping windows. With an overlap of *o* samples, the related generic formula can be given via:
  $$A(t,k) = \frac{1}{M}\sum_{m=0}^{M-1}(s_0((M-o)t+m)\cdot s_0((M-o)t+m+k)).$$
  They calculate the pitch only on windows of sufficient energy to reduce processing time on silences. In addition they perform nonlinear clipping of small subband values to avoid confusion of the first and second formants. They choose the largest autocorrelation peak value as the pitch if it contains more than *30%* of the window's energy.

- Venugopal et al. [VEN99] use the analysis-by-synthesis approach of the **Multiband Excitation Vocoder** for pitch estimation. In it, speech is synthesised and an unbiased error measure is calculated by comparison to the original speech. The pitch period is the period used when the error is minimum.

4 Segmentation

When talking about segmentation of an audio stream, temporal segmentation is usually the subject. The identification of the sound components that belong to one specific sound event could be regarded as a spatio-temporal segmentation. This is a hard task and being researched in the field of "computational auditory scene analysis (CASA)". We are not aware of any approaches toward CASA in the MPEG-1 compressed domain. So, here we concentrate on temporal segmentation in which specific temporal fragments of an audio stream are identified for their homogenous content according to some criteria. Existing segmentation approaches determine fragment boundaries based on e.g. strong changes of a specific feature or relative pauses.

For such segmentation approaches, the presented features are often not used directly – instead their mean and variances are calculated on larger windows of about *1-4 sec*. Additionally, log-transforms of the results can be used to reduce the dynamic range and make the clusters in classification more compact [TZA00].

4.1 General segmentation

- Tzanetakis et al. [TZA00] perform generic audio segmentation at **"texture" change** instants. They use the features low energy, and mean and variances of spectral centroid, spectral rolloff point, spectral flux, and RMS energy in a feature vector. They calculate the Mahalanobis distance between successive feature vectors, and differentiate it. Peaks are then picked as segment boundaries via an adaptive thresholding algorithm, which includes a

minimum duration condition to avoid small regions. They achieve up to *75%* consistency with human segmentation results.

- Our own approach to segmentation uses the signal magnitude as feature [PFE01] to calculate **relative pauses**. The segmentation algorithm also follows an adaptive thresholding approach on *2 sec* intervals. Windows are determined as silence if their signal magnitude stays under the threshold. Segmentation uses a minimum duration and a maximum tolerated interruption parameter. A sequence of silence windows gets clustered into a pause segment if it covers at least the minimum duration and is not interrupted by non-silence windows longer than the tolerated interruption. We achieve hit rates between *46%* and *97%* when comparing to human segmentation results depending upon the SNR of the material.

4.2 Scene change

- Barrass [BAR97] calculates a running average of the spectral centroid called "brightness history". Sudden changes in brightness are used for scene change detection.

- Boccignone et al. [BOC99] calculate video scene changes based on audio and video breaks. Video analysis provides shot boundaries, which are scene change candidates. Audio analysis validates the candidates. They calculate the subband mean energy and four subband central moments on the first 8 subbands and accumulate these into one feature vector. Then, an Artificial Neural Network is trained to partition the feature space into silence, speech, music, and noise resulting in transition points between different sound classes. These are used to validate the shot boundaries. They achieve a hit rate between *62%* and *93%* for audio breaks in comparison to human results.

5 Classification

Although temporal segmentation is an important first step in determining the structure of an audio stream, automatic determination of more information on the actual content of the fragments is of higher value. Thus the next step is to classify the fragment content into a given set of sound classes. Generic classes are silence, music, speech, and noise. According to the requirements of the application, more specific classifications may be required, such as the determiniation of the type of sound effect or the separation of speakers.

5.1 Silence determination

- Patel et al. [PAT96] use a standard threshold approach on the average signal energy of a video clip to determine whether the clip contains silence.

- Nakajima et al. [NAK99] use the variance of subband *0* energy to distinguish between silence and non-silence:

$$\sigma^2(n) = \frac{1}{M}\sum_{m=0}^{M-1}(s_0^2(m) - \mu_0^2(m))^2.$$

They choose *M = 1 sec* and use only one subband sample per frame. This sub-sampling increases calculation speed enormously. A silent segment is declared where $\sigma^2(n) < T_s$ with T_s as the silence energy threshold. They achieve a hit rate of *91%* with *13%* false hits. The false hits are mainly attributed to mixed signal *1 sec* windows.

5.2 Music/speech determination

- Patel et al. [PAT96] classify audio segments determined by video shot boundary detection. If the band energy ratio lies above *0.8* or the pause rate is below *0.2* or there is no pitch found, the segment is classified as a musical clip, else as speech clip.

- Nakajima et al. [NAK99] use the energy density and the average number of subbands with significant level on *1 sec* windows to distinguish between music and speech. Music has a higher energy density than speech. Music also usually has significant subbands up to subband number *20*, whereas speech rarely goes beyond subband *10*. They use a multivariate Gaussian distribution to model the classes and achieve a hit rate of *93%* for music (with *4%* false hits) and of *88%* for speech (with *16%* false hits). The false hits for speech stem from intermittent sounds such as drum solo.

- Tzanetakis et al. [TZA00] use the features low energy, and the mean and variances of the spectral centroid, spectral rolloff point, and spectral flux in a feature vector. They compare a multivariate Gaussian distribution classifier to a K-Nearest Neighbour classifier to distinguish speech from music. They evaluate them on about 2 hours of audio data and compare results on a frame basis achieving about *82%* accuracy for the Gaussian distribution and about *85%* accuracy for the K-NN. In comparison to the classification of PCM data, the results only degrade by about *2%*.

- Barrass [BAR97] determines music on Layer 2 files as a signal that exhibits long-term wide-band stability. This stability is calculated from the scalefactor selection information. A signal is determined as long-term stable if more than 60% of the non-zero subbands of a frame have a repeated scalefactor. Coverage of the subbands must be at least 24 out of 30 subbands. In contrast, speech is determined as a signal with low- or mid-range brightness and stability. The brightness is calculated via the spectral centroid of the scalefactors. Low-range brightness indicates a male voice, mid-range brightness a female one and high-range brightness again signifies music.

5.3 Sound effects

Nakajima et al. [NAK99] use the average and variance of the spectral centroid on 1 sec windows to determine **applause** on their TV program sound tracks. Applause has a continuous self-similarity and stable centre frequency. They achieve a hit rate of *74%* with *15%* false hits, which

occur mainly on mixed signal *1 sec* windows.

5.4 Noise

Barrass [BAR97] determines noise on Layer 2 files as a signal that exhibits long-term wide-band transience. This transience is calculated from the scalefactor selection information. A signal is determined as long-term transient if more than 30% of the non-zero subbands of a frame have non-repeated scalefactors. Coverage of the subbands must be at least 1 out of 4 subbands. In addition, short, loud and bright signals are determined as a "clang" and also classified as a noise.

5.5 Speaker

To distinguish between six different speakers, Venugopal et al. [VEN99] use normalised linear frequency cepstral coefficients and estimate Gaussian Mixture Model parameters using the Expectation Maximisation algorithm.

6 Recognition and Identification

On speech segments, recognition and identification of more specific sound content is possible such as the gender of a speaker segment, the speaker itself, and the content of his speech.

On music segments, recognition of beats and identification of rhythms can be performed. We report on one beat recognition approach based on MPEG-1 compressed domain features.

6.1 Gender

- Venugopal et al. [VEN99] use the pitch estimation of the Multiband Excitation Vocoder and declare the speaker as male if the pitch is between *60* and *120 Hz* and female between *120* and *200 Hz*. They achieve a hit rate of about *80%*.

- As mentioned above, Barrass [BAR97] determines the gender of a speech frame via the brightness of the signal, which is calculated from the spectral centroid of the scalefactors in Layer 2. If the spectral centroid lies below the second subband, it is determined as male, and between the third and sixth subband as female.

6.2 Speech recognition

Yapp and Zick [YAP97] implemented a speaker-dependent, small vocabulary speech recognition system that uses compressed domain features. They calculate cepstral, delta cepstral, and acceleration coefficients as described in Section 2.2.1. Then they train Hidden Markov Models

(HMMs) on 17 words. Training and recognition were both performed with continuously spoken sentences. A word-level accuracy of *99%* was obtained on Layer 2 encoded data at *32 kbit/s*. Their system works on Layer 1 and Layer 2 and interlayer training and recognition is possible.

6.3 Beat recognition

Wang and Vilermo [WAN01] presented a compressed-domain beat detector for pop-music with the aim of replacing beats that were lost during an Internet transmission of a pop-song with previously stored beat samples of that song. They use the window type information of Layer 3 files and the band energy of four frequency ranges for beat detection. The four frequency ranges are: the full-band energy and the frequency intervals 0-459, 3405-7462, and 7463-22050 Hz. The middle frequency ranges usually give poor beat information because other instruments and singing are more dominant in these ranges. When restricting the search for beats to the most probable times after inter beat intervals (IBIs), they detect most beats.

7 Conclusions

In this paper we have given an overview of the kind of features that have been extracted in the MPEG-1 audio compressed domain. Considering the amount of MPEG-1 Layer 3 (MP3) files available nowadays, audio analysis on compressed files is bound to be in great demand soon. Research in this field is still in its infancy and there are still many opportunities to pursue for fundamental research. Audio analysis results can be used more powerfully when used in conjunction with video analysis results to achieve automatic extraction of more abstract concepts. On its own it can be used for sound-based audio search engines no more based on textual queries and filenames but on the audio content.

ACKNOWLEDGEMENTS

We thank Conrad Parker and Stephen Barrass for their proof-reading and feedback to this paper.

REFERENCES

[BAR97] Stephen Barrass “Bilby – A tool for foraging in MPEG audio”, Technical Report No. 01/98, CSIRO Mathematical and Information Sciences, Nov. 1997 (published June 2001).

[BOC99] G. Boccignone, M. De Santo, and G. Percannella, "Joint Audio-Video Processing of MPEG Encoded Sequences", Proc. IEEE Intl. Conf. on Multimedia Computing and Systems (ICMCS), Vol. 2, 1999, pp. 225-229.

[HAS97] Haskell, Puri, and Netravali, "Digital Video: An Introduction to MPEG-2, Chapter 4, 1997, Chapman & Hall, New York.

[ISA93] ISO, International Organization for Standardization, "International Standard 11172-3, Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 MBit/s - Part 3: Audio", 1993.

[NAK99] Y. Nakajima, Y. Lu, M. Sugano, A. Yoneyama, H. Yanagihara, and A. Kurematsu, "A Fast Audio Classification From MPEG", Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Vol. IV, 1999, Phoenix, Arizona, USA, pp. 3005-3008.

[NOL97] P. Noll, "MPEG Digital Audio Coding", IEEE Signal Processing Magazine, Sept. 1997, pp. 59-81.

[PAN95] D. Pan, "A Tutorial on MPEG/Audio Compression", IEEE Multimedia, Vol. 2 No. 2, summer 1995, pp. 60-74.

[PAT96] N.V. Patel, and I.K. Sethi, "Audio characterization for video indexing", Proc. SPIE, Storage and Retrieval for Still Image and Video Databases IV, Vol. 2670, 1996, San Jose, CA, USA, pp. 373-384.

[PFE01] Silvia Pfeiffer “Pause concepts for audio segmentation at different semantic levels”, Proc. ACM Multimedia 2001, Sept 30 – Oct 5, Ottawa, Ontario, Canada, pp. 187-193, 2001.

[PFE99] S. Pfeiffer, J. Robert-Ribes, and D. Kim, "Audio Content Extraction from MPEG-encoded sequences", Proc. Fifth Joint Conference on Information Sciences, pp. 513-516, 1999, Vol. II, Atlantic City, New Jersey.

[TZA00] G. Tzanetakis, and P. Cook, "Sound analysis using MPEG compressed audio", Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing ICASSP 2000, pp. 761-764, Vol. 2, Istanbul, Turkey.

[VEN99] S. Venugopal, K.R. Ramakrishnan, S.H. Srinivas, and N. Balakrishnan, "Audio scene analysis and scene change detection in the MPEG compressed domain", IEEE Third Workshop on Multimedia Signal Processing, MMSP 19999, pp. 191-196.

[WAN01] Ye Wang, Miikka Vilermo “A compressed domain beat detector using MP3 audio bitstreams”, Proc. ACM Multimedia 2001, Sept 30 – Oct 5, Ottawa, Ontario, Canada, pp. 194-202, 2001.

[YAP97] L. Yapp, and G. Zick, "Speech recognition on MPEG/audio encoded files", Proc. IEEE Intl. Conf. on Multimedia Computing and Systems (ICMCS), pp. 624-625, 1997, Ottawa, Canada.