

A Generic Audio Classification and Segmentation Approach for Multimedia Indexing and Retrieval

Serkan Kiranyaz, Ahmad Farooq Qureshi, and Moncef Gabbouj, *Senior Member, IEEE*

Abstract—We focus the attention on the area of generic and automatic audio classification and segmentation for audio-based multimedia indexing and retrieval applications. In particular, we present a fuzzy approach toward hierarchic audio classification and global segmentation framework based on automatic audio analysis providing robust, bi-modal, efficient and parameter invariant classification over global audio segments. The input audio is split into segments, which are classified as speech, music, fuzzy or silent. The proposed method minimizes critical errors of misclassification by fuzzy region modeling, thus increasing the efficiency of both pure and fuzzy classification. The experimental results show that the critical errors are minimized and the proposed framework significantly increases the efficiency and the accuracy of audio-based retrieval especially in large multimedia databases.

Index Terms—Automatic audio classification and segmentation, fuzzy modeling, multimedia indexing and retrieval, perceptual rule-based approach.

I. INTRODUCTION

AUDIO information often plays an essential role in understanding the semantic content of digital media and in certain cases, audio might even be the only source of information e.g., audio-only clips. Henceforth, audio information has been recently used for content-based multimedia indexing and retrieval. Audio may also provide significant advantages over the visual counterpart especially if the content can be extracted according to the human auditory perceptual system. This, on the other hand, requires efficient and generic audio (content) analysis that yields a robust and semantic classification and segmentation.

During the recent years, there have been many studies on automatic audio classification and segmentation using several techniques and features. Traditionally, the most common approach is speech/music classifications in which the highest accuracy has been achieved, especially when the segmentation information is known beforehand (i.e., manual segmentation). Saunders [21] developed a real-time speech/music classifier for audio in radio FM receivers based on features such as zero crossing rate (ZCR) and short-time energy. A 2.4-s window size was used and the primary goal of low computational complexity was achieved. Zhang and Kuo [28] developed a content-based audio retrieval system, which performs audio classification into basic types such as speech, music and noise.

In their latter work [27] using a heuristic approach and pitch tracking techniques, they introduced more audio classes such as songs, mixed speech over music. Scheirer and Slaney [22] proposed a different approach for the speech/music discriminator systems particularly for ASR. El-Maleh *et al.* [7] presented a narrow-band (i.e., audio sampled at 8 KHz) speech/music discriminator system using a new set of features. They achieved a low frame delay of 20 ms, which makes it suitable for real-time applications. A more sophisticated approach has been proposed by Srinivasan *et al.* [23]. They tried to categorize the audio into mixed class types such as music with speech, speech with background noise, etc. They reported over 80% classification accuracy. Lu *et al.* [14] presented an audio classification and segmentation algorithm for video structure parsing using a 1-s window to discriminate speech, music, environmental sound and silence. They proposed new features such as band periodicity to enhance the classification accuracy.

Although audio classification has been mostly realized in the uncompressed domain, with the emerging MPEG audio content, several methods have been reported for audio classification on MPEG-1 (Layer 2) encoded audio bit-stream [9], [16], [19], [24]. The last years have shown a widespread usage of MPEG Layer 3 (MP3) audio [4], [6], [18] as well as proliferation of several video content carrying MP3 audio. The ongoing research on perceptual audio coding yields a more efficient successor called (MPEG-2/4) Advanced Audio Coding (AAC) [2], [5]. AAC has various similarities with its predecessor but promises significant improvement in coding efficiency. In a previous work [10], we introduced an automatic segmentation and classification method over MP3 (MPEG-1, 2, 2.5 Layer-3) and AAC bit-streams. In this work, using a generic MDCT template formation extracted from both MP3 and AAC bit-streams, an unsupervised classification over globally extracted segments is achieved using a hierarchical structure over the common MDCT template.

Audio content extraction via classification and segmentation enables the design of efficient indexing schemes for large-scale multimedia databases. There might, however, be several shortcomings of the simple speech/music classifiers so far addressed in terms of extracting real semantic content, especially for multimedia clips that presents various content variations. For instance, most of speech/music discriminators work on the digital audio signals that are in the uncompressed domain, with a fixed capturing parameter set. Obviously, large multimedia databases may contain digital audio that is in different formats (compressed/uncompressed), encoding schemes (MPEG Layer-2, MP3, AAC, ADPCM, etc.), capturing and encoding parameters (i.e., sampling frequency, bits per sample,

Manuscript received July 5, 2004; revised December 21, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ananth Sankar.

The authors are with the Institute of Signal Processing, Tampere University of Technology, FIN-33101, Tampere, Finland (e-mail: serkan@cs.tut.fi).

Digital Object Identifier 10.1109/TSA.2005.857573

sound volume level, bit-rate, etc.) and durations. Therefore, the underlying audio content extraction scheme should be robust (invariant) to such variations since the content is independent from the underlying parameters that the digital multimedia world presents. For example, the same content of a speech may be represented by an audio signal sampled at 8 KHz or 32 KHz, in stereo or mono, compressed by AAC or stored in (uncompressed) PCM format, lasting 15 s or 10 min, etc.

A comparative study of several statistical, HMM and GMM and neural network based training models was carried out by Bugatti *et al.* [3]. Although such approaches may achieve a better accuracy for a limited set of collections, they are usually restricted to a focused domain and hence do not provide a generic solution for the massive content variations that a large-scale multimedia database may contain.

Another important drawback of many existing systems is the lack of global segmentation. Since classification and segmentation are closely related and dependent problems, an integrated and well-tuned classification and segmentation approach is required. Due to technical difficulties or low-delay requirements, some systems tend to rely on manual segmentation (or simply work over a short audio file having a single audio content type). The other existing systems use several segment features that are estimated over audio segments with a fixed duration (0.5–5 s) to accomplish a classification per segment. Although fixing the segment size brings many practical advantages in the implementation and henceforth improves the classification accuracy, its performance may suffer either from the possibly high resolution required by the content or from the lack of sufficient statistics needed to estimate the segment features due to the limited time span of the segment. An efficient and more natural solution is to extract global segments within which the content is kept stationary so that the classification method can achieve an optimum performance within the segment.

Almost all of the systems so far addressed do not have a bi-modal structure. That is, they are either designed in *bit-stream* mode where the bit-stream information is directly used (without decoding) for classification and segmentation, or in *generic* mode where the temporal and spectral information is extracted from the PCM samples and the analysis is performed afterwards. Usually, the former case is applied for improved computational speed and the latter for higher accuracy. A generic bi-modal structure, which supports both of the modes (possibly to some extent) is obviously needed in order to provide feasible solutions for the audio-based indexing of large multimedia databases. Such a framework can, for instance, work in *bit-stream* mode whenever the enhanced speed is required, especially for long clips for which the generic mode is not a feasible option for the underlying hardware or network conditions; or it can work in the *generic* mode whenever feasible or required.

Due to content variations, most of the existing works addressing just “speech and music” categories may not be satisfactory for the purpose of an efficient audio indexing scheme. The main reason for this is the presence of mixed audio types, such as speech with music, speech with noise, music with noise, environmental noise, etc. There might even be difficult cases where a pure class type ends up with an erroneous classification due

to several factors. For the sake of audio indexing overall performance, either new class types for such potential audio categories should be introduced, or such “mixed” or “erroneous” cases should be collected under a certain class category (e.g., *fuzzy*) so that special treatment can be applied while indexing such audio segments. Since high accuracy is an important and basic requirement for the audio analysis systems used for indexing and retrieval, introducing more class types might cause degradations in performance and hence is not considered as a feasible option most of the time for such generic solutions.

In order to overcome the aforementioned problems and shortcomings, in this paper we propose a generic audio classification and segmentation framework especially suitable for audio-based multimedia indexing and retrieval systems. The proposed approach has been integrated into the MUVIS [11], [12], [15], system. The proposed method is automatic and uses no information from the video signal. It also provides robust (invariant) solution for the digital audio files with various capturing/encoding parameters and modes such as sampling frequencies (i.e., 8 KHz up to 48 KHz), channel modes (i.e., mono, stereo, etc.), compression bit-rates (i.e., 8 kbps up to 448 kbps), sound volume level, file duration, etc. In order to increase accuracy, a *fuzzy* approach has been integrated within the framework. The main process is self-learning, which logically builds on the extracted information throughout its execution, to produce a reliable final result. The proposed method proceeds through logical hierarchical steps and iterations, based on certain perceptual rules that are applied on the basis of perceptual evaluation of the classification features and the behavior of the process. Therefore, the overall design structure is made suitable for human aural perception and for this, the proposed framework works on perceptual rules whenever possible. The objective is to achieve such a classification scheme that ensures a decision making approach suitable to human content perception.

The proposed method has a bi-modal structure, which supports both *bit-stream* mode for MP3 and AAC audio, and *generic* mode for any audio type and format. In both modes, once a common spectral template is formed from the input audio source, the same analytical procedure is performed afterwards. The spectral template is obtained from MDCT coefficients of MP3 granules or AAC frames in *bit-stream* mode and hence called as MDCT template. The power spectrum obtained from FFT of the PCM samples within temporal frames forms the spectral template for the *generic* mode.

In order to improve the performance and most important of all, the overall accuracy, the classification scheme produces only four class types per audio segment: *speech*, *music*, *fuzzy* or *silent*. *Speech*, *music* and *silent* are the *pure* class types. The class type of a segment is defined as *fuzzy* if it is either not classifiable as a pure class due to some potential uncertainties or anomalies in the audio source or it exhibits features from more than one pure class. For audio based indexing and retrieval in MUVIS system [8], a pure class content is only searched throughout the associated segments of the audio items in the database having the same (matching) pure class type, such as *speech* or *music*. All *silent* segments and *silent* frames within *nonsilent* segments are discarded from the audio indexing. As mentioned earlier, special care is taken for the *fuzzy* content,

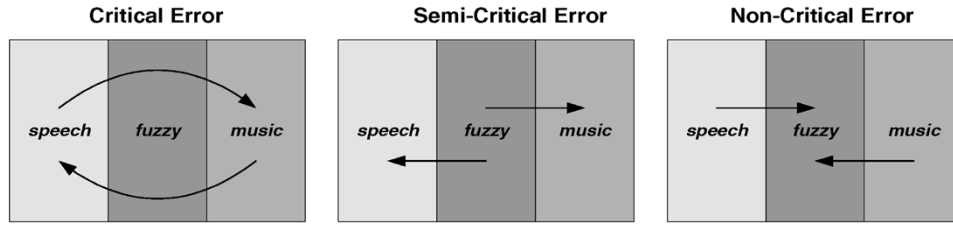


Fig. 1. Different error types in classification.

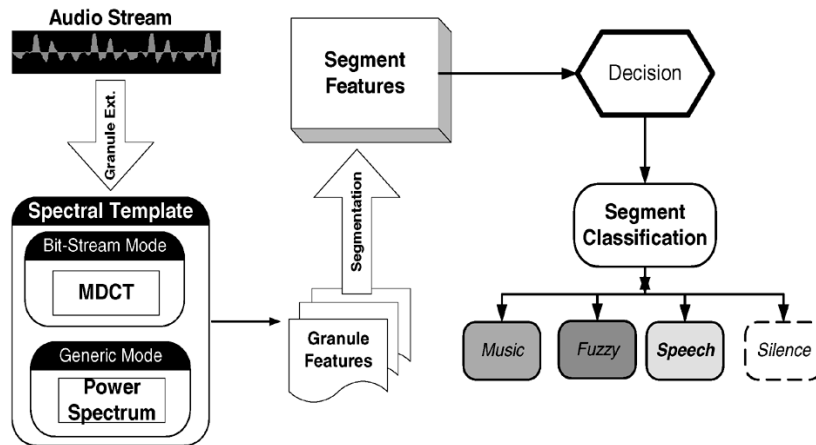


Fig. 2. Classification and segmentation framework.

that is, during the retrieval phase, the *fuzzy* content is compared with all relevant content types of the database (i.e., *speech*, *music* and *fuzzy*) since it might, by definition, contain a mixture of pure class types, background noise, aural effects, etc. Therefore, for the proposed method, any erroneous classification on pure classes is intended to be detected as *fuzzy*, so as to avoid significant retrieval errors (mismatches) due to such potential misclassification. In this context, three prioritized error types of classification, illustrated in Fig. 1, are defined.

- **Critical Errors:** These errors occur when one pure class is misclassified into another pure class. Such errors significantly degrade the overall performance of an indexing and retrieval scheme.
- **Semi-critical Errors:** These errors occur when a *fuzzy* class is misclassified as one of the pure class types. These errors moderately affect the performance of retrieval.
- **Non-critical Errors:** These errors occur when a pure class is misclassified as a *fuzzy* class. The effect of such errors on the overall indexing/retrieval scheme is negligible.

The paper is organized as follows. Section II is devoted to the description of the common spectral template formation depending on the mode. The hierarchic approach adopted for the overall feature extraction scheme and the perceptual modeling in the feature domain are introduced in Section III. Section IV describes the proposed framework with its hierarchical steps. The experimental results along with the evaluation of the proposed algorithm are reported in Section V. Finally, the concluding remarks are given in Section VI.

II. SPECTRAL TEMPLATE FORMATION

In this section, we focus on the formation of the generic spectral template, which is the initial and pre-requisite step in order to provide a bi-modal solution.

As show in Fig. 2 spectral template is formed either from the MP3/AAC encoded bit-stream in *bit-stream* mode or the power spectrum of the PCM samples in the *generic* mode. Basically, this template provides spectral domain coefficients, $SPEQ(w, f)$, (MDCT coefficients in *bit-stream* mode or power spectrum in *generic* mode) with the corresponding frequency values $FL(f)$ for each granule/frame. By using $FL(f)$ entries, all spectral features and any corresponding threshold values can be fixed independently from the sampling frequency of the audio signal. Once the common spectral template is formed the granule features can be extracted accordingly and thus, the primary framework can be built on a common basis, independent from the underlying audio format and the mode used.

A. Forming the MDCT Template From MP3/AAC Bit-Stream

1) *MP3 and AAC Overview:* MPEG audio is a group of coding standards that specify a high performance perceptual coding scheme to compress audio signals into several bit-rates. Coding is performed in several steps and some of them are common for all three layers. There is a perceptual encoding scheme that is used for time/frequency domain masking by a certain threshold value computed using psychoacoustics rules. The spectral components are all quantized and a quantization noise is therefore introduced.

MP3 is the most complex MPEG layer. It is optimized to provide the highest audio quality at low bit-rates. Layer 3 encoding process starts by dividing the audio signal into frames, each of

which corresponds to one or two granules. The granule number within a single frame is determined by the MPEG phase. Each granule consists of 576 PCM samples. Then a polyphase filter bank (also used in Layer 1 and Layer 2) divides each granule into 32 equal-width frequency subbands, each of which carries 18 (subsampling) samples. The main difference between MPEG Layer 3 and the other layers is that an additional MDCT transform is performed over the subband samples to enhance spectral resolution. A short windowing may be applied to increase the temporal resolution in such a way that 18 PCM samples in a subband is divided into three short windows with six samples. Then MDCT is performed over each (short) window individually and the final 18 MDCT coefficients are obtained as a result of three groups of six coefficients. There are three windowing modes in MPEG Layer 3 encoding scheme: *Long Windowing Mode*, *Short Windowing Mode* and *Mixed Windowing Mode*. In *Long Windowing Mode*, MDCT is applied directly to the 18 samples in each of the 32 subbands. In *Short Windowing Mode*, all of 32 subbands are short windowed as mentioned above. In *Mixed Windowing Mode*, the first two lower subbands are long windowed and the remaining 30 higher subbands are short windowed. Once MDCT is applied to each subband of a granule according to the windowing mode, the scaled and quantized MDCT coefficients are then Huffman coded and thus the MP3 bit-stream is formed.

There are three MPEG phases concerning MP3: MPEG-1, MPEG-2, and MPEG 2.5. MPEG-1 Layer 3 supports sampling rates of 32, 44.1, and 48 kHz and bit-rates from 32 to 448 kbps. It performs encoding on both mono and stereo audio, but not multi-channel surround sound. One MPEG-1 Layer 3 frame consists of two granules (1152 PCM samples). During encoding, different windowing modes can be applied to each granule. MPEG-2 Layer 3 is a backward compatible extension to MPEG-1 with up to five channels, plus one low frequency enhancement channel. Furthermore, it provides support for lower sampling rates such as 16, 22.05 and 24 kHz for bit-rates as low as 8 kbps up to 320 kbps. One MPEG-2 Layer 3 frame consists of one granule (576 PCM samples). MPEG 2.5 is an unofficial MPEG audio extension, which was created by Fraunhofer Institute to improve performance at lower bit-rates. At lower bit-rates, this extension allows sampling rates of 8, 11.025, and 12 KHz.

AAC and MP3 have mainly a similar structure. Nevertheless, compatibility with other MPEG audio layers has been removed and AAC has no granule structure within its frames whereas MP3 might contain one or two granules per frame depending on the MPEG phase as mentioned before. AAC supports a wider range of sampling rates (from 8 kHz to 96 kHz) and up to 48 audio channels. Furthermore it works at bit-rates from 8 kbps for mono speech and in excess of 320 kbps. A direct MDCT transformation is performed over the samples without dividing the audio signal in 32 subbands as in MP3 encoding. Moreover, the same tools (psychoacoustic filters, scale factors and Huffman coding) are applied to reduce the number of bits used for encoding. Similar to MP3 coding scheme, two windowing modes are applied before MDCT is performed in order to achieve a better time/frequency resolution: Long Windowing Mode or Short Windowing Mode. In Long Windowing Mode MDCT is directly applied over 1024 PCM samples. In Short

TABLE I
MDCT TEMPLATE ARRAY DIMENSION WITH RESPECT TO COMPRESSION TYPE AND WINDOWING MODE

Compression Type & Windowing Mode	NoW	NoF
MP3 Long Window	1	576
MP3 Short Window	3	192
MP3 Mixed Window	3	216
AAC Long Window	1	1024
AAC Short Window	8	128

TABLE II
TRANSITION PENALIZATION TABLE

Transition: $f_r \rightarrow f_r^{t+1}$	TP^r
<i>silent</i> \rightarrow <i>non-silent</i>	+1
<i>non-silent</i> \rightarrow <i>silent</i>	+1
<i>silent</i> \rightarrow <i>silent</i>	+1
<i>non-silent</i> \rightarrow <i>non-silent</i>	-1

Windowing Mode, an AAC frame is first divided into 8 short windows each of which contains 128 PCM samples and MDCT is applied to each short window individually. Therefore, in Short Windowing Mode, there are 128 frequency lines and hence the spectral resolution is decreased by 8 times whilst increasing the temporal resolution by 8. AAC has a new technique so called “Temporal Noise Shaping,” which improves the speech quality especially at low bit-rates. More detailed information about MP3 and AAC can be found in [2], [4], [5], [6] and [18].

The structural similarity in MDCT domain between MP3 and AAC makes developing generic algorithms that cover both MP3 and AAC feasible. So the proposed algorithm in this paper uses this similarity as an advantage to form a common spectral template based on MDCT coefficients. This template allows us to achieve a common classification and segmentation technique that uses the compressed domain audio features as explained in the next subsection.

2) *MDCT Template Formation*: The *bit-stream* mode uses the compressed domain audio features in order to perform classification and segmentation directly from the compressed bit-stream. Audio features are extracted using the common MDCT subband template. Hence, MDCT template is nothing but a variable size MDCT double array, $MDCT(w, f)$, along with a variable size frequency line array $FL(f)$, which represents the real frequency value of the each row entry in the MDCT array. The index w represents the window number and the index f represents the line frequency index. Table I represents array dimensions NoW and NoF with respect to the associated window modes of MP3 and AAC

Let f_s be the sampling frequency. Then according to Nyquist’s theorem the maximum frequency (f_{BW}) of the audio signal will be: $f_{BW} = f_s/2$. Since both AAC and MP3 use linearly spaced frequency lines, then the real frequency values f can be obtained from the $FL(f)$ using (1) shown at the bottom of the next page where f is the index from 0 to the corresponding NoF given in Table II.

The MDCT template array is formed from the absolute values of the MDCT subband coefficients, which are (Huffman) de-

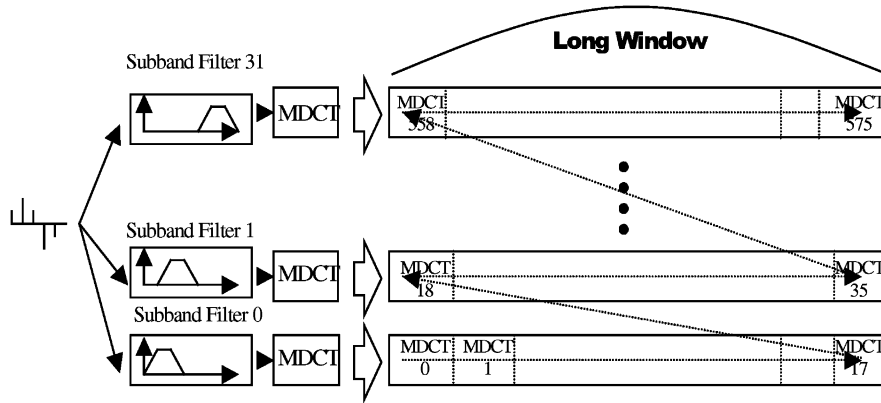


Fig. 3. MP3 long window MDCT template array formation from MDCT subband coefficients.

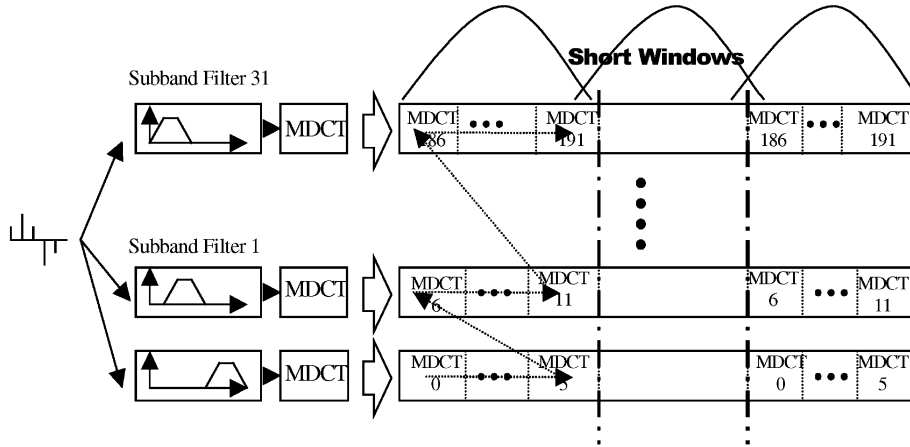


Fig. 4. MP3 short window MDCT template array formation from MDCT subband coefficients.

coded from the MP3/AAC bit-stream per MP3 granule or AAC frame. For each MP3 granule, the MDCT subband coefficients are given in the form of a matrix of 32 lines, representing the frequency subbands, with 18 columns each of which for every coefficient as shown in Fig. 3. In case of short window, there are three windows within a granule containing six coefficients. The template matrix formation for short window MP3 granules is illustrated in Fig. 4. In order to process the same algorithm for both encoding schemes, we apply a similar template formation structure to AAC frames. So in case of long window AAC frame, 1024 MDCT coefficient array is divided into 32 groups of 32 MDCT coefficients and the template matrix for AAC is formed by taking into account that the number of MDCT coefficients for a subband is not 18 (as in MP3) but now 32. Fig. 5 illustrates AAC long window template formation. In case of short window AAC frame, 1024 coefficients are divided into 8 windows of 128 coefficients each. We divide these 128 coefficients in 32 subbands and fill the matrix with 4 coefficients in every subband in order to have the same template as the MP3 short

window case. Fig. 6 shows how the subbands are arranged and the template array is formed by this technique.

B. Spectral Template Formation in Generic Mode

In the generic mode, the spectral template is formed from the FFT of the PCM samples within a frame that has a fixed temporal duration. In bit-stream mode, the frame (temporal) duration varies since the granule/frame size is fixed (i.e., 576 in MP3, 1024 in AAC long window mode). However, in this mode, we have the possibility to extract both fixed-size or fixed-duration frames depending on the feature type. For analysis compatibility purposes, it is a common practice to fix the (analysis) frame duration. If, however, fixed spectral resolution is required (i.e., for fundamental frequency estimation), the frame size (hence the FFT window size) can also be kept constant by increasing the frame size by zero padding or simply using the samples from neighbor frames.

$$FL(f) = \left\{ \begin{array}{ll} \left((f+1) \times \frac{f_{BW}}{NoF} \right) & \text{Short or Long Win. Mode} \\ \left(\begin{array}{l} \left((f+1) \times \frac{f_{BW}}{576} \right) f < 36 \\ \left(\frac{f_{BW}}{16} + \frac{(f-35) \times f_{BW}}{192} \right) f \geq 36 \end{array} \right) & \text{MP3 Mixed Win. Mode} \end{array} \right\} \quad (1)$$

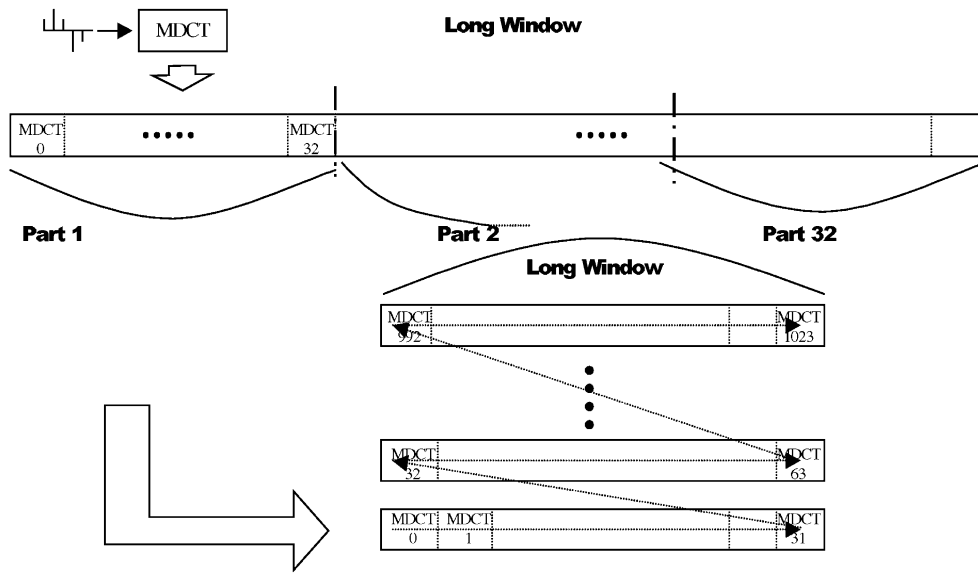


Fig. 5. AAC long window MDCT template array formation from MDCT subband coefficients.

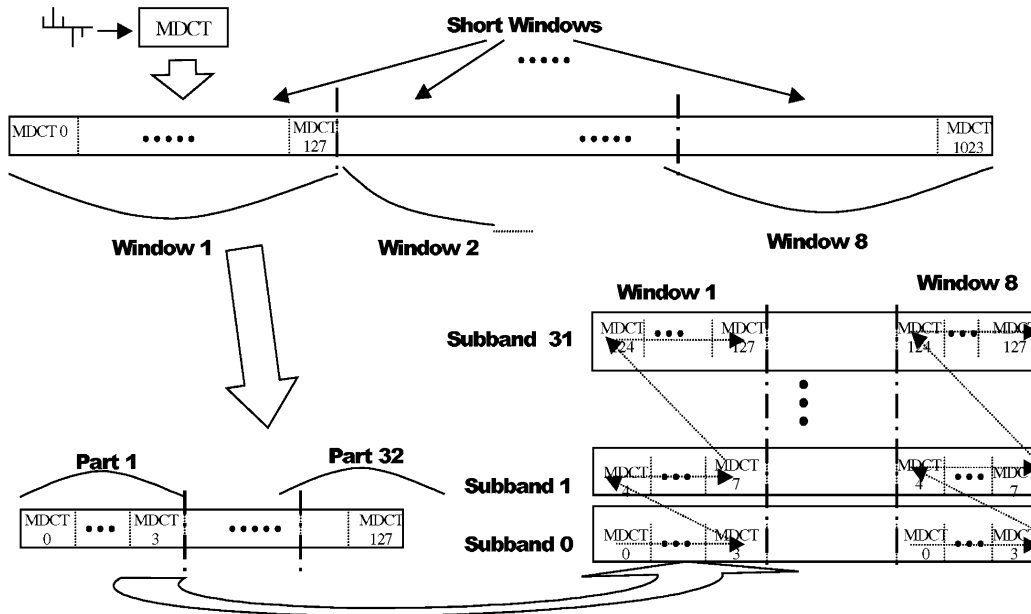


Fig. 6. AAC short window MDCT template array formation from MDCT subband coefficients.

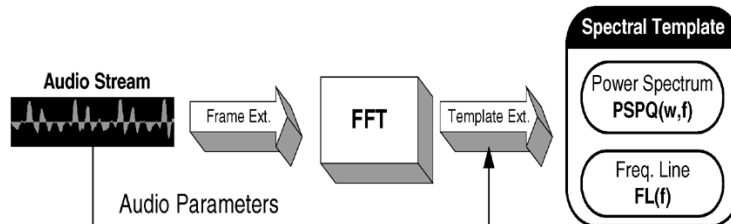


Fig. 7. Generic mode spectral template formation.

As shown in Fig. 7, the generic mode spectral template consists of a variable size power spectrum double array $PSPQ(w, f)$ along with a variable size frequency line array $FL(f)$, which represents the real frequency value of each row entry in the PSPQ array. The index w represents the window number and the index f represents the line fre-

quency index. In generic mode, $NoW = 1$ and NoF is the number of frequency lines within the spectral bandwidth: $NoF = 2^{(int)(\log_2(fr_{dur} f_s) - 1)}$ where fr_{dur} is the duration of one audio (analysis) frame and f_s is the sampling frequency.

Note that for both modes, the template is formed independently from the number of channels (i.e., stereo/mono) in the

audio signal. If the audio is stereo, both channels are averaged and used as the signal before the frame extraction is processed.

III. FEATURE EXTRACTION

As shown in Fig. 2, a hierarchic approach has been adopted for the overall feature extraction scheme in the proposed framework. First the frame (or granule) features are extracted using the spectral template and the segment features are derived afterwards in order to accomplish classification for the segments. In the following subsections we will focus on the extraction of the several frame features.

A. Frame Features

Granule features are extracted from the spectral template, $\text{SPEQ}(w, f)$ and $\text{FL}(f)$, where SPEQ can be assigned to MDCT or PSPQ depending on the current working mode.

We use some classical features such as Band Energy Ratio (BER), Total Frame Energy (TFE) and Subband Centroid (SC). We also developed a novel feature so called Transition Rate (TR) and tested it against the conventional counterpart, Pause Rate (PR). Since both PR and TR are segment features by definition, they will be introduced on the next section. Finally we proposed an enhanced Fundamental Frequency (FF) detection algorithm, which is based on the well-known HPS (harmonic product spectrum) technique [17].

1) *Total Frame Energy Calculation:* Total Frame Energy (TFE) can be calculated using (2). It is the primary feature to detect silent granules/frames. Silence detection is also used for the extraction of TR, which is one of the main segment features

$$\text{TFE}_j = \sqrt{\sum_w^{\text{NoW}} \sum_f^{\text{NoF}} (\text{SPEQ}_j(w, f)^2)}. \quad (2)$$

2) *Band Energy Ratio Calculation:* Band energy ratio (BER) is the ratio between the total energies of two spectral regions that are separated by a single cut-off frequency. The spectral regions fully cover the spectrum of the input audio signal. Given a cut-off frequency value $f_c (f_c \leq f_{\text{BW}})$, let $f \langle f_c \rangle$ be the line frequency index where $\text{FL}(f \langle f_c \rangle) \leq f_c < \text{FL}(f \langle f_c \rangle + 1)$, BER for a granule/frame j can be calculated using (3)

$$\text{BER}_j(f_c) = \frac{\sqrt{\sum_w^{\text{NoW}} \sum_{f=0}^{f \langle f_c \rangle} (\text{SPEQ}_j(w, f))^2}}{\sqrt{\sum_w^{\text{NoW}} \sum_{f=f \langle f_c \rangle}^{\text{NoF}} (\text{SPEQ}_j(w, f))^2}}. \quad (3)$$

3) *Fundamental Frequency Estimation:* If the input audio signal is harmonic over a fundamental frequency (i.e., there exists a series of major frequency components that are integer multiples of a fundamental frequency value), the real Fundamental Frequency (FF) value can be estimated from the spectral coefficients ($\text{SPEQ}(w, f)$). Therefore, we apply an adaptive peak-detection algorithm over the spectral template to check whether sufficient number of peaks around the integer multiple of a certain frequency (a candidate FF value) can be found or not. The algorithm basically works in three steps:

- adaptive extraction of the all the spectral peaks;
- candidate FF Peaks extraction via harmonic product spectrum (HPS);
- multiple peak search and fundamental frequency (FF) verification.

Especially the human speech have most of its energy at lower bands (i.e., $f < 500$ Hz.) and hence the absolute value of the peaks in this range might be significantly greater than the peaks in the higher frequency bands. This brings the need for an adaptive design in order to detect the major spectral peaks in the spectrum. We therefore, apply a nonoverlapped partitioning scheme over the spectrum and the major peaks are then extracted within each partition. Let N_P is the number of partitions each of which have the (f_{BW}/N_P) Hz bandwidth. In order to detect peaks in a partition, the absolute mean value is first calculated from the spectral coefficients in the partition and if a spectral coefficient is significantly bigger than the mean value (e.g., greater than twice the mean value), it is chosen as a new peak and this process is repeated for all the partitions. The maximum spectral coefficient within a partition is always chosen as a peak even if it does not satisfy the aforementioned rule. This is basically done to ensure that at least one peak is to be detected per partition. One of the main advantages of the partition based peak detection is that the amount of redundant spectral data is significantly reduced toward the major peaks, which are the main concern for FF estimation scheme.

The candidate FF peaks are obtained via HPS. If a frame is harmonic with a certain FF value, HPS can detect this value. However, there might be two potential problems. First HPS can be noisy if the harmonic audio signal is a noisy and mixed signal with significant nonharmonic components. In this case, HPS will not extract the FF value as the first peak in the harmonic product, but as the second or higher order peak. For this reason we consider a reasonable number (i.e., 5) of the highest peaks extracted from the harmonic product as the candidate FF values. Another potential problem is that HPS does not provide whether or not the audio frame is harmonic, since it always produces some harmonic product peak values from a given spectrum. The harmonicity should therefore be searched and verified among the peak values extracted in the previous step.

The multiple peak verification is a critical process for fundamental frequency (FF) calculation. Due to the limited spectral resolution, one potential problem might be that the multiple peak value may not be necessarily on the exact frequency line that spectral coefficient exists. Let the linear frequency spacing between two consecutive spectral coefficient be $\Delta f = \text{FL}(f) - \text{FL}(f - 1) = f_{\text{BW}}/\text{NoF}$ and let the real FF value will be in the $\{-\Delta f/2, +\Delta f/2\}$ neighborhood of a spectral coefficient at the frequency $\text{FL}(f)$. Then the minimum window width to search for n th (possible) peak will be: $W(n) = n \times \Delta f$. Another problem is the pitch-shifting phenomenon of the harmonics that especially occurs to harmonic patterns of the speech. Terhardt [25] proposed stretched templates for the detection of the pitch (fundamental frequency) and one simple analytical description of the template stretching is given in the following expression:

$$f_n = n^\sigma f_0 \Rightarrow W(n) = (n^\sigma - n)f_0 \quad \forall n = 2, 3, 4 \dots \quad (4)$$

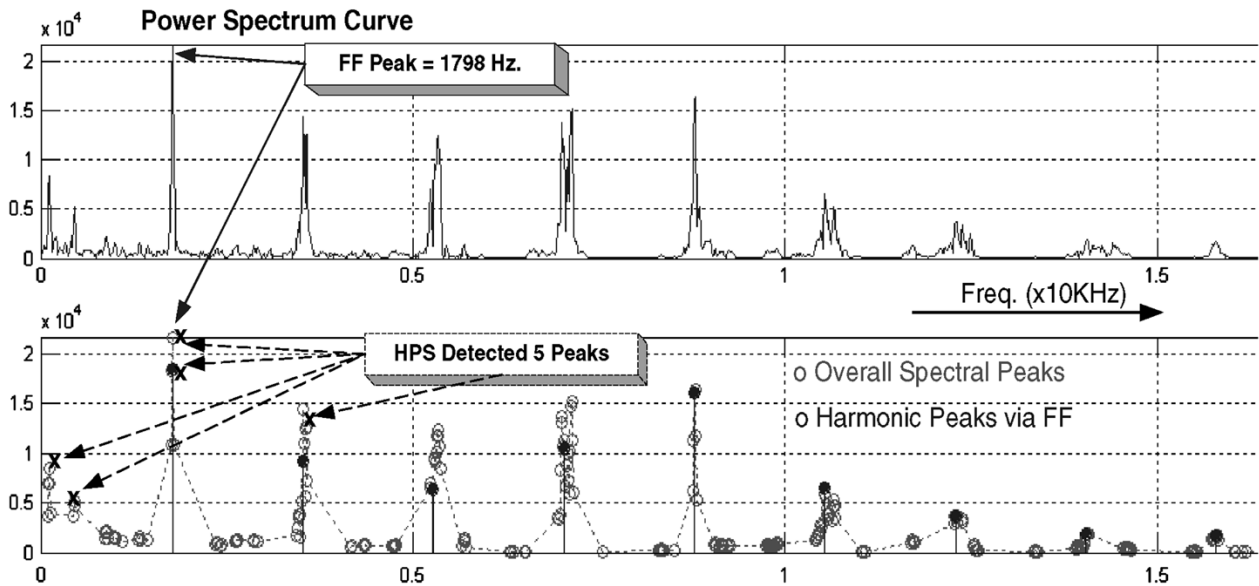


Fig. 8. FF detection within a harmonic frame.

where σ is the stretch factor with a nominal value 1 and f_0 is the perceived fundamental frequency. Practically $\sigma \approx 1.01$ for the human speech, which can therefore be approximated as a linear function (i.e., $f_n \cong n\sigma f_0$). Due to such harmonic shifts and the limited spectral resolution of the spectral template, an adaptive search window is applied in order not to miss a (multiple) peak on a multiple frequency line. On the other hand false detections might occur if the window width is chosen larger than necessary. We developed, tested, and used the following search window template for detection:

$$W(n) = n\Delta f\sigma \quad \forall n = 2, 3, 4, \dots \quad (5)$$

Note that the search window width is proportional to both sampling frequency of the audio signal and a stretch factor σ , and inversely proportional to the total number of frequency lines, both of which gives a good measure of resolution and provides a stretched template modeling for the perceived FF value.

Fig. 8 illustrates a sample peak detection applied on the spectral coefficients of an audio frame with the sampling frequency 44100 Hz. Therefore, $f_{BW} = 22050$ Hz but the sketch shows up to around 17000 Hz for the sake of illustration. The lower subplot shows the overall peaks detected in the first step (red), 5 candidate peaks extracted in the second step via HPS algorithm (black), the multiple peaks found (blue) and finally the FF value estimated accordingly ($FF = FL(18) = 1798$ Hz in this example).

4) *Subband Centroid Frequency Estimation*: Subband Centroid (SC) is the first moment of the spectral distribution (spectrum) or in compressed domain it can be estimated as the balancing frequency value for the absolute spectral values. Using the spectral template arrays, SC (f_{SC}) can be calculated using (6)

$$f_{SC} = \frac{\sum_w^{NoW} \sum_f^{NoF} (\text{SPEQ}(w, f) \times FL(f))}{\sum_w^{NoW} \sum_f^{NoF} \text{SPEQ}(w, f)} \quad (6)$$

B. Segment Features

Segment Features are extracted from the frame (or granule) features and mainly used for the classification of the segment. A segment, by definition, is a temporal window, which lasts a certain duration within an audio clip. There are basically two types: *silent* and *nonsilent* segments. The *nonsilent* segments are subject to further classification using their segment features. As mentioned before, the objective is to extract global segments, each of which should contain a stationary content along with its time span in the semantic point of view. Practically, there is no upper bound for the segments. For instance a segment may cover the whole audio clip if there is a unique audio category in it (i.e., MP3 music clips). However there is and should be a practical lower bound for the segments duration within which a perceptible content can exist (i.e., >0.6 s).

Total frame energy (TFE) is the only feature used for the detection of the *silent* segments. The segment features are then used to classify the *nonsilent* segments and will be presented in the following sections.

1) *Dominant Band Energy Ratio*: Since the energy is concentrated mostly on the lower frequencies for human speech, an audio frame can be classified as *speech* or *music* by comparing its band energy ratio (BER) value with an empirical threshold. This is an unreliable process when it is applied per-frame basis, but within a segment it can turn out to be an initial classifier by using the dominant (winning) class type within the segment. Experimental results show that dominant band energy ratio (DBER), as a segment feature, does not achieve as high accuracy as the other major segment features but it usually gives consistent results for the similar content. Therefore, we use DBER for the initial steps of the main algorithm, mainly for merging the immature segments into more global segments if their class types match with respect to DBER. One of the requirements of segment merging is to have same class types of the neighbor segments and DBER is consistent of giving same (right or wrong) result for the same content.

2) *Transition Rate versus Pause Rate*: Pause rate (PR) is a well-known feature as a speech/music discriminator and basically it is the ratio between the numbers of *silent* granules/frames to total number of granules/frames in a *nonsilent* segment. Due to natural pauses or unsound consonants that occur within any *speech* content, *speech* has a certain level of PR level that is usually lacking in any *music* content. Therefore, if this ratio is over a threshold (T_{PR}), then the segment is classified as a *speech* segment, otherwise *music*.

PR usually achieves a significant performance in discriminating *speech* from *music*. However, its performance is degraded when there is a fast speech (without sufficient amount of pauses), a background noise or when the speech segment is quite short (i.e., < 3 s). Since PR is only related with the amount (number) of *silence* (*silent* frames) within a segment, it can lead to critical misclassifications.

In a natural human *speech*, due to the presence of the unsound consonants, the frequency of the occurrence of the silent granules is generally high even though their total time span (duration) might be still low as in the case of a fast *speech*. On the other hand, in some classical music clips, there might be one or a few intentional *silent* sections (*silent* passes) that may cause misclassification of the whole segment (or clip) as *speech* due to the long duration of such passes. These erroneous cases lead us to introduce an improved measure, transition rate (TR), which is based on the transitions, occurs between consecutive frames. TR can be formulated for a segment as in (7)

$$TR(S) = \frac{\text{NoF} + \sum_i^{\text{NoF}} TP^i}{2 \text{NoF}} \quad (7)$$

where NoF is the number of frames within segment S , i is the frame index and TP^i is the transition penalization factor that can be obtained from Table II.

Note that although the total amount of silent frames is low for a fast speech or in short speech segment, the transition rate will be still high due to their frequent occurrence.

3) *Fundamental Frequency Segment Feature*: Fundamental frequency (FF) is another well-known music/speech discriminator due to the fact that *music* is more harmonic than the *speech* in general. Pure *speech* contains a sequence of harmonic tonals (vowels) and inharmonic consonants. In *speech*, due to the presence of inharmonic consonants, the natural pauses and the low-bounded FF values (i.e., < 500 Hz) the average FF value within a segment tend to be quite low. Since the presence of the continuous instrumental notes results large harmonic sections with unbounded FF occurrences in *music*, the average FF value tends to be quite high. However, the average FF value alone might result in classification failures in some exceptional cases. Experiments show that such misclassifications occur especially in some harmonic female *speech* segments or in some hard-rock *music* clips with saturated beats and base-drums.

In order to improve the discrimination factor from FF segment feature, we develop an enhanced segment feature based on conditional mean, which basically verifies strict FF tracking (continuity) within a window. Therefore, FF value of a particular frame will be introduced in the mean summation only if its

nearest neighbors are also harmonic, otherwise discarded. The conditional mean based FF segment feature is formulated in (8)

$$FF(S) = \frac{\sum_i^{\text{NoF}} \begin{pmatrix} FF_i & \text{if } FF_j \neq 0 \forall j \in NN(i) \\ 0 & \text{otherwise} \end{pmatrix}}{\text{NoF}} \quad (8)$$

where FF_i is the FF value of the i th frame in segment S and j represents the index of the frames in the nearest neighbor frame set of the i th frame $NN(i)$.

Due to frequent discontinuities in the harmonicity such as pauses (*silent* frames) and consonants on a typical speech segment, the conditional mean results in a significantly low FF segment value for pure *speech*. *Music* segments, on the other hand, tend to have higher FF segment values due to their continuous harmonic nature even though the beats or base-drums might cause some partial losses on the FF tracking. The experimental results approve the significant improvement obtained from the conditional mean based FF segment feature and thus FF segment feature become one of the major features that we use for the classification of the final (global) segments.

4) *Subband Centroid Segment Feature*: Due to the presence of both voiced (vowels) and unvoiced (consonants) parts in a speech segment, the average Subband Centroid (SC) value tend to be low with a significantly higher standard deviation and *vice versa* for *music* segments. However the experimental results show that some *music* types can also present quite low SC average values and thus SC segment feature used to perform classification is the standard deviation alone with one exception: The mean of SC within a segment is only used when it gives such a high value (forced-classification by SC) indicating the presence of the music with a certainty.

Both of SC segment features are extracted by smoothly sliding a short window through the frames of the nonsilent segment. The standard deviation of the SC is calculated using local windowed mean and windowed standard deviation of SC in the segment and formulated as in (9)

$$\sigma_{SC}(S) = \sqrt{\frac{\sum_j^{\text{NoF}} (SC_j - \mu_j^{SC})^2}{\text{NoF}}} \quad (9)$$

where $\mu_i^{SC} = \frac{\sum_{j \in W_i}^{\text{NoW}_i} SC_j}{\text{NoW}}$

where μ_i^{SC} is the windowed SC mean of the i th frame calculated within a window W_i with NoW frames. $\sigma^{SC}(S)$ is the SC segment feature of the segment S with NoF frames.

Such adaptive calculation of the segment feature improves the discrimination between *speech* and *music* and therefore, SC is used as the third major feature within the final classification scheme.

C. Perceptual Modeling in Feature Domain

The primary approach in the classification and segmentation framework is based on the perceptual modeling in the feature domain that is mainly applied on to the major segment features: FF, SC, and TR. Depending on the nature of the segment feature, the model provide a perceptual-rule based division in the feature space as shown in Fig. 9. The forced-classification occurs if that particular feature results such an extreme value that perceptual

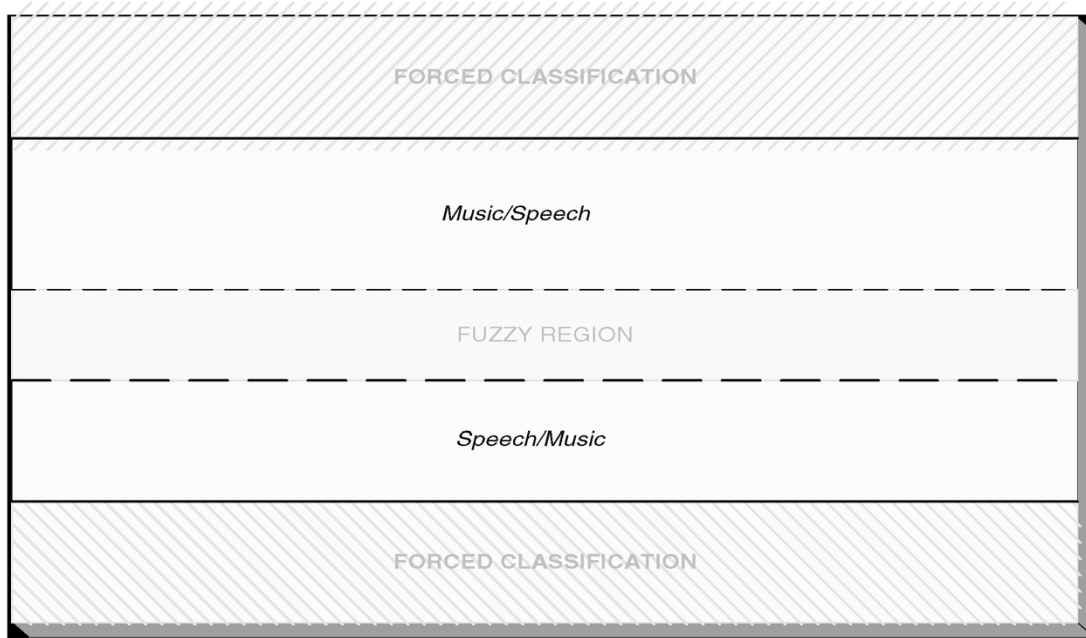


Fig. 9. Perceptual modeling in feature domain.

certainty about content identification is occurred. Therefore, it overrides all the other features so that the final decision is made with respect to that feature alone. Note that the occurrence of a forced classification, its region boundaries and its class category depend on the nature of the underlying segment feature. In this context each major segment feature has the following forced-classification definitions.

- TR has a forced *speech* classification region above 15% due to the fact that only pure *speech* can yield such a high value within a segment formation.
- FF has a forced *music* classification with respect to its mean value that is above 2 KHz. This makes sense since only pure and excessively harmonic *music* content can yield such an extreme mean value.
- SC has two forced-classification regions, one for music and the other for speech content. The forced *music* classification occurs when the SC mean exceeds 2 KHz and the forced *speech* classification occurs when the primary segment feature of SC, the adaptive σ^{SC} value, exceeds 1200 Hz.

Although the model supports both lower and upper forced-classification regions, only the upper regions are so far used. However we tend to keep the lower region in case the further experimentations might approve the usage of that region in the future.

The region below forced-classification is where the natural discrimination occurs into one of the pure classes such as *speech* or *music*. For all segment features the lower boundary of this region is tuned so that the feature would have a typical value that can be expected from a pure class type but still quite far away having a certainty to decide the final classification alone.

Finally there may be a *fuzzy* region where the feature value is no longer reliable due to various possibilities such as the audio class type is not pure, rather mixed or some background noise is

present causing ‘blurring’ on the segment features. So for those segment features that are examined and approved for the *fuzzy* approach, a *fuzzy* region is formed and tuned experimentally to deal with such cases. There is, on the other hand, another advantage of having a *fuzzy* region between the regions where the real discrimination occurs. The *fuzzy* region prevents most of the critical errors, which might occur due to noisy jumps from one (pure class) region on to another. Such noisy cases or anomalies can be handled within the *fuzzy* region and a critical error turns out to be a noncritical error for the sake of audio-based indexing and retrieval. Furthermore, there are still other features that might help to clarify the classification at the end. Experimental results show that FF and SC segment features are suitable for *fuzzy* region modeling. However TR cannot provide a reliable *fuzzy* model due to its nature. Although TR can achieve probably the highest reliability of distinguishing *speech* from the other class types, it is practically blind of categorization of any other *nonspeech* content (i.e., *fuzzy* from *music*, *music* from *speech* with significant background noise, etc.). Therefore, *fuzzy* modeling is not applied to TR segment feature to prevent such erroneous cases.

IV. GENERIC AUDIO CLASSIFICATION AND SEGMENTATION

The proposed approach is mainly developed based on the aforementioned fact: automatic audio segmentation and classification are mutually dependent problems. A good segmentation requires good classification and vice versa. Therefore, without any prior knowledge or supervising mechanism, the proposed algorithm proceeds in an iterative way, starting from granule/frame based classification and initial segmentation, the iterative steps are carried out until a global segmentation and thus a successful classification per segment can be achieved at the end. Fig. 10 illustrates the 4-steps iterative approach to the audio classification and segmentation problem.

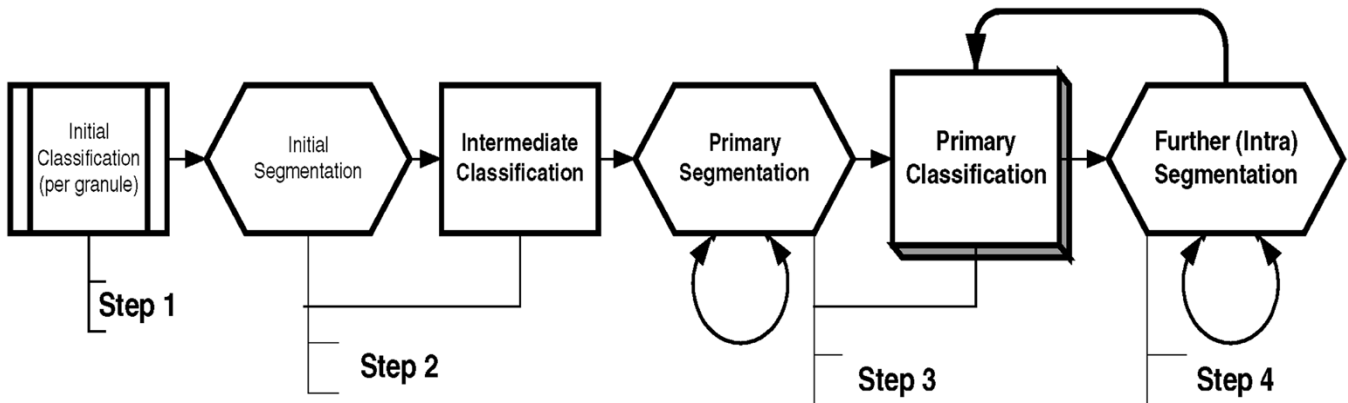


Fig. 10. Flowchart of the proposed approach.

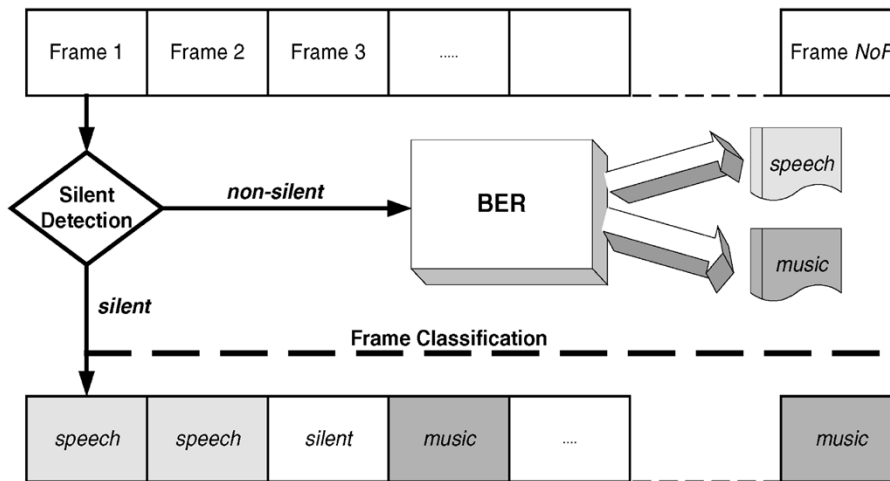


Fig. 11. Step 1.

A. Step 1: Initial Classification

As for the first step the objective is to extract *silent* and *nonsilent* frames and then obtain an initial categorization for the *nonsilent* frames in order to proceed with an initial segmentation on the next step. Since the frame-based classification is nothing but only needed for an initial segmentation, there is no need of introducing *fuzzy* classification in this step. Therefore, each granule/frame is classified in one of three categories: *speech*, *music* or *silent*. Silence detection is performed per granule/frame by applying a threshold (T_{TFE}) to the total energy as given in (2). T_{TFE} is calculated adaptively in order to take the audio sound volume effect into account. The minimum (E_{min}), maximum (E_{max}) and average (E_{μ}) granule/frame energy values are first calculated from the entire audio clip. An empirical all-mute test is performed to ensure the presence of the audible content. Two conditions are checked.

- I) $E_{max} > \text{Min. Audible Frame Energy Level.}$
- II) $E_{max} \gg E_{min}.$

Otherwise the entire clip is considered as all mute and hence no further steps are necessary. Once the presence of some nonsilent granules/frames is confirmed then T_{TFE} is calculated according to (10)

$$T_{TFE} = E_{min} + \lambda_s \times (E_{\mu} - E_{min}),$$

where $0 < \lambda_s \leq 1$ (10)

where λ_s is the *silence* coefficient, which determines the *silence* threshold value between E_{min} and E_{μ} . If the total energy of a granule/frame is below T_{TFE} , then it is classified as *silent*, otherwise *nonsilent*. If a granule/frame is not classified as *silent*, the BER is then calculated for a cut-off frequency of 500 Hz due to the fact that most of speech energy is concentrated below 500 Hz. If BER value for a frame is over a threshold (i.e., 2%) that granule/frame is classified as *music*, otherwise *speech*. Fig. 11 summarizes the operation performed in Step 1.

B. Step 2

In this step, using the frame-based features extracted and classifications performed in the previous step the first attempts for the segmentation has been initiated and the first segment features are extracted from the initial segments formed. To begin with *silent* and *nonsilent* segmentations are performed. In the previous step, all the *silent* granules/frames have already been found. So the *silent* granules/frames are merged to form *silent* segments. An empirical minimum interval (i.e., 0.2 s) is used to assign a segment as a *silent* segment if sufficient number of *silent* granules/frames merges to a segment, which has the duration greater than this threshold. All parts left between *silent* segments can then be considered as *nonsilent* segments. Once all *nonsilent* segments are formed, then the classification of these segments is performed using DBER and TR. The initial segmentation and segment classification (via

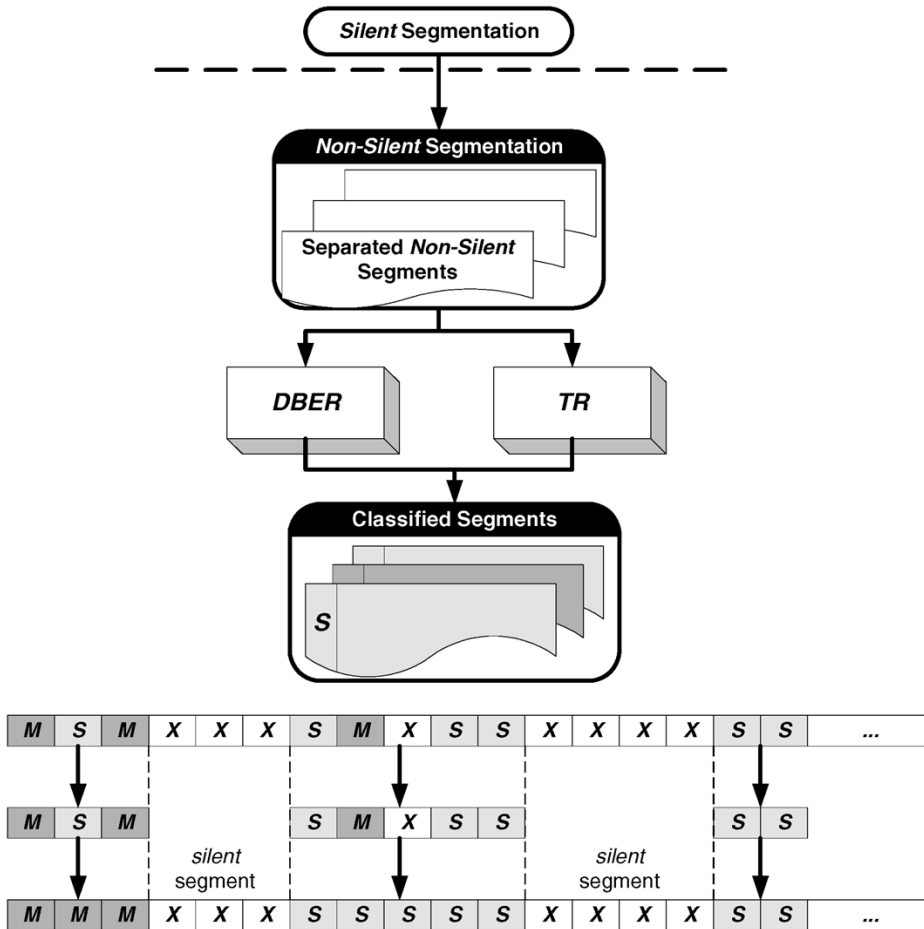


Fig. 12. Step 2.

DBER) is illustrated in Fig. 12 with a sample segmentation and classification example at the bottom. Note that there might be different class types assigned independently via DBER and TR for the nonsilent segments. This is done on purpose since the initial segment classification performed in this step with such twofold structure is nothing but a preparation for the further (toward a global) segmentation efforts that will be presented in the next step.

C. Step 3

This is the primary step where most of the efforts toward classification and global segmentation are summed up. The first part of this step is devoted to a merging process to obtain more global segments at the end. The silent segments extracted in the previous step might be ordinary local pauses during a natural speech or they can be the borderline from one segment to another one with a different class type. If the former argument is true, such silent segments might still be quite small and negligible for the sake of segmentation since they reduce the duration of the nonsilent segments and hence they might lead to erroneous calculations for the major segment features. Therefore, they need to be eliminated to yield a better (global) segmentation that would indeed result in a better classification. There are two conditions in order to eliminate a *silent* segment and merge its nonsilent neighbors.

- i) Its duration is below a threshold value.
- ii) Neighbor nonsilent segment class types extracted from DBER and TR are both matching.

After merging some of *nonsilent* segments, the overall segmentation scheme is changed and the features have to be re-extracted over the new (emerged via merging) *nonsilent* segments. For all the *nonsilent* segments, PR and DBER are re-calculated and then they are re-classified. This new classification of *nonsilent* segments may result into such classification types that allow us to eliminate further *silent* segments. (In the first step they may not be eliminated because the neighbor classification types did not match). So an iterative loop is applied to eliminate all possible small *silent* segments. The iteration is carried out till all small *silent* segments are eliminated and *nonsilent* segments are merged to have global segments, which have a unique classification type.

Once the merging loop is terminated, there might still exist some short nonsilent segments that are not merged to any neighbor global segments. Such short *nonsilent* frames are naturally false segments and therefore, should also be eliminated by forcefully merging them to a neighbor *nonsilent* segment. After the elimination of such short *nonsilent* segments, the formation of *nonsilent* segments is global enough to contain one single class type, which will then be extracted using the major features.

TABLE III
GENERIC DECISION TABLE

TR	FF	SC	Decision
speech	speech	speech	Speech
speech	speech	music	Speech
speech	speech	fuzzy	Speech
speech	music	speech	Speech
speech	music	music	Music
speech	music	fuzzy	Fuzzy
speech	fuzzy	speech	Speech
speech	fuzzy	music	Fuzzy
speech	fuzzy	fuzzy	Fuzzy
music	speech	speech	Speech
music	speech	music	Music
music	speech	fuzzy	Fuzzy
music	music	speech	Music
music	music	music	Music
music	music	fuzzy	Music
music	fuzzy	speech	Fuzzy
music	fuzzy	music	Music
music	fuzzy	fuzzy	Fuzzy

Due to the perceptual modeling in the feature domain any segment feature may fall into forced-classification region and overrides the common decision process with its decision alone. Otherwise a decision look-up table is applied for the final classification as given in Table III. This table is formed up considering all possible class type combinations. Note that the majority rule is dominantly applied within this table, that is, if the majority of the segment features favor a class type, and then that class type is assigned to the segment. If a common consensus cannot be made, then the segment is set as *fuzzy*.

D. Step 4

This step is dedicated to the intra segmentation analysis and mainly performs some post processing to improve the overall segmentation scheme. Once final classification and segmentation is finished in step 3 (Section III-C), *nonsilent* segments with significantly long duration might still need to be partitioned into new segments if they consist of two or more subsegments (without any silent part in between) with different class types. For example within a long segment there might be subsegments that include both pure *music* and pure *speech* content without a silent separation in between. Due to the lack of (sufficiently long) silent segment separation in between, the early steps failed to detect those subsegments and therefore, a further (intra) segmentation is performed in order to separate those subsegments.

We developed two approaches to perform intra segmentation. The first approach divides the segment into two and uses SC segment feature to see the presence of a significant difference in between (unbalanced subsegments). The second one attempts to detect the boundaries of any potential subsegment with different class type by using Subband Centroid (SC) frame feature. Generally speaking, the first method is more robust on detecting the major changes since it uses the segment feature that is usually robust to noise. However it sometimes introduces significant offset on the exact location of the subsegments and therefore

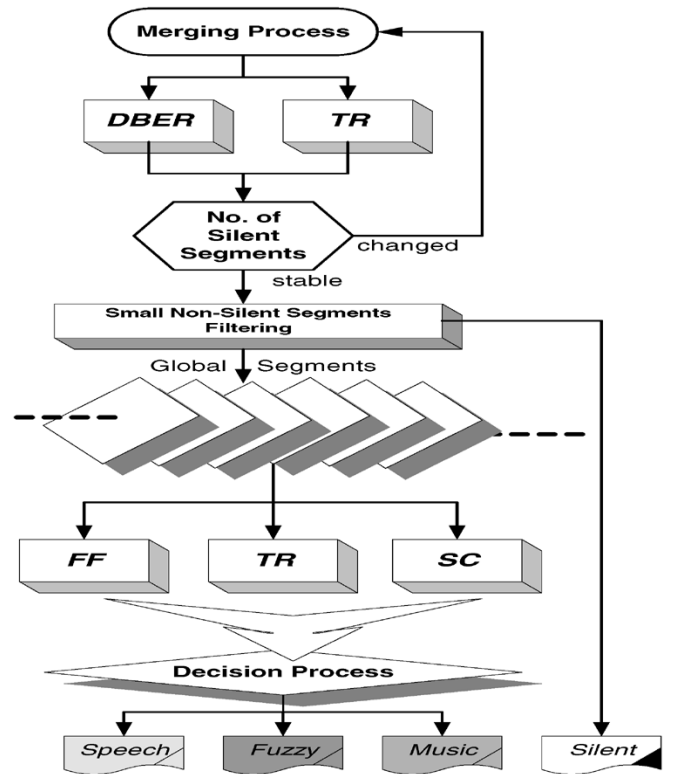


Fig. 13. Step 3.

causes severe degradations on the temporal resolution and segmentation accuracy. This problem is mostly solved in the second method but it might increase the amount of false detections of the subsegments especially when the noise level is high. In the following subsections both methods will be explained in detail.

1) *Intra Segmentation by Binary Division*: The first part in this method tests if the nonsilent segment is significantly longer than a given threshold, (i.e., 4 s). Then we start by dividing the segment into two subsegments and test whether their SC segment feature values are significantly differing from each other. If not, we keep the parent segment and stop. Otherwise we execute the same operation over the two child segments and look for the one, which is less balanced (the one which has higher SC value difference between the left and the right child-segments). The iteration is carried out till the child segment is small enough and breaks the iteration loop. This gives the subsegment boundary and then Step 3 is reperformed over the subsegments in order to make the most accurate classification possible. If Step 3 does not assign a different class type for the potential subsegments detected, then the initial parent segment is kept unchanged. This means a false detection has been performed in Step 4. Fig. 14 illustrates the algorithm in detail. The function *local_change()* performs SC based classification for the right and left child segments within the parent segment (without dividing) and returns the absolute SC difference between them.

2) *Intra Segmentation by Breakpoints Detection*: This is a more traditional approach performed in many similar works. The same test as in the previous approach is performed to test whether the segment has a sufficiently long duration. If so, using a robust frame feature (i.e., SC), a series of breakpoints (certain frame locations) where the class types of the associated frames

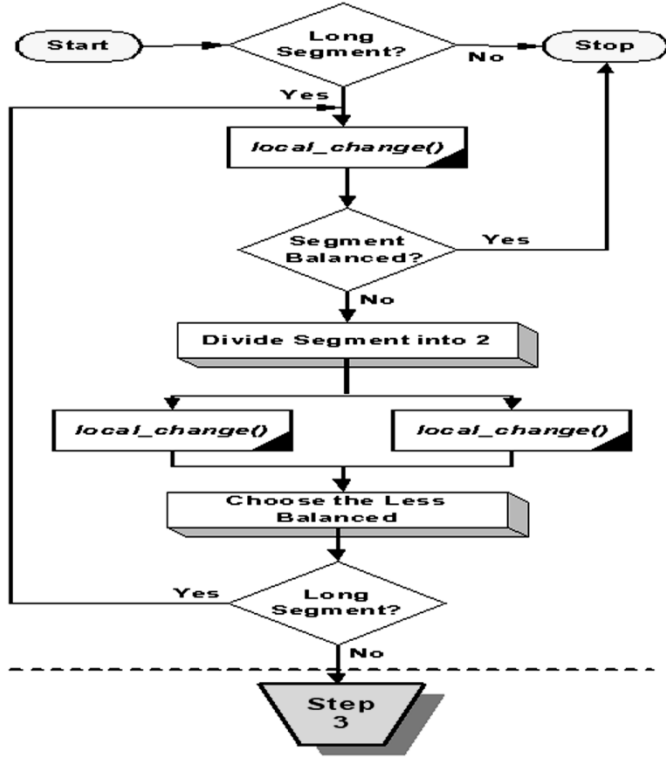


Fig. 14. Intrasegmentation by binary subdivision in Step 4.

according to the SC frame feature alternate with respect to the class type of the parent segment, are detected. The alternation occurs first with a particular frame giving such a SC feature value indicating a class type that is different from the class type of the parent segment. Then it may swing back to original class type of its parent after a while or ends up with the parent segment boundary. SC segment feature is the windowed standard deviation used for the classification of the segment. Keeping the same analogy, we use windowed standard deviation calculated per frame and via comparing it with the SC segment feature, the breakpoints can be detected. Windowed standard deviation of SC, σ_i^{SC} , for frame i can be calculated as in (11)

$$\sigma_i^{SC} = \sqrt{\frac{\sum_{j \in W_i}^{NoW_i} (SC_j - \mu_i^{SC})^2}{NoW}}$$

where

$$\mu_i^{SC} = \frac{\sum_{j \in W_i}^{NoW_i} SC_j}{NoW}. \quad (11)$$

So the pair of breakpoints can be detected via comparing σ_i^{SC} for all the frames within the segment with the SC segment feature σ_{SC} (i.e., $\sigma_i^{SC} > \sigma_{SC} \rightarrow \sigma_{i+NoF_{SS}}^{SC} < \sigma_{SC}$).

After all the potential breakpoints are detected, the perceptually relevant ones are kept for further analysis and the rest is filtered out. A pair of breakpoints can be kept as a relevant subsegment if it lasts long enough (i.e., > 1 s) since several breakpoints can occur due to noisy nature of the process. Another difficulty arises on the exact location of the breakpoint. The intersection between the SC segment threshold value and the slope that occurs on the intermediate region from one class type to another

is not the accurate boundary, which represents the pure class region (subsegment). The real boundary lies at the global minima (or the maxima). In order to detect this simple roll-down algorithm is applied toward to local minima and hence the more accurate boundary detection is performed. An example of initial breakpoint detection and the roll-down algorithm are shown in Fig. 15.

Once the subsegment boundaries (breakpoints) are found, the same approach explained in the previous method is then performed: Step 3 is re-visited for the detected subsegments in order to make the most accurate classification possible for them and as before, Step 3 may or may not approve the birth of the new subsegments detected in this step.

V. EXPERIMENTAL RESULTS

As mentioned earlier, this framework is integrated under MUVIS to test and evaluate the audio classification and segmentation performance and its effects on audio based multimedia indexing and retrieval. MUVIS framework aims to bring a unified and global approach to indexing, browsing and querying of various digital multimedia types such as audio/video clips and digital images.

As shown in Fig. 16, MUVIS framework is based on three applications, each of which has different responsibilities and functionalities. *AVDatabase* is mainly responsible for real-time audio/video database creation with which audio/video clips are captured, (possibly) encoded and recorded in real-time from any peripheral audio and video devices connected to a computer. It is an essential tool for real-time audio and video capturing, encoding by last generation codecs such as MPEG-4, H.263+, MP3 and AAC. *DbsEditor* performs the creation and indexing of the multimedia databases and therefore, offline feature extraction process over the multimedia collections is its main task. *MBrowser* is the primary media browser and retrieval application, which supports two query mechanisms: *Normal Query* (NQ), which employs exhaustive search over databases with no indexing structure and *Progressive Query* (PQ) [13], which can provide periodic retrieval updates to the user and can be used in any MUVIS database. In order to query an audio/video clip, it should first be appended to a MUVIS database upon which the query will be performed. This can be done using *AVDatabase* application via capturing, encoding and recording in real-time one or more audio/video files (with possibly different capturing and encoding parameters) and appending to any MUVIS database one by one and in a sequential order. The other alternative is appending the available files using the *DbsEditor* application, possibly by converting them into one of the native formats supported within MUVIS framework. By this way, any multimedia file, alien or native, can be appended into a MUVIS database. Using the visual/Aural Feature eXtraction modules (FeX/AFeX modules) the visual and aural features are extracted and appended to the database structure along with multimedia items. Once this phase is completed, the database becomes ready for any type of query operation (visual and/or aural).

By means of MUVIS framework, we collected databases of multimedia clips offering various capturing and encoding parameters, practically an infinite content supply and different

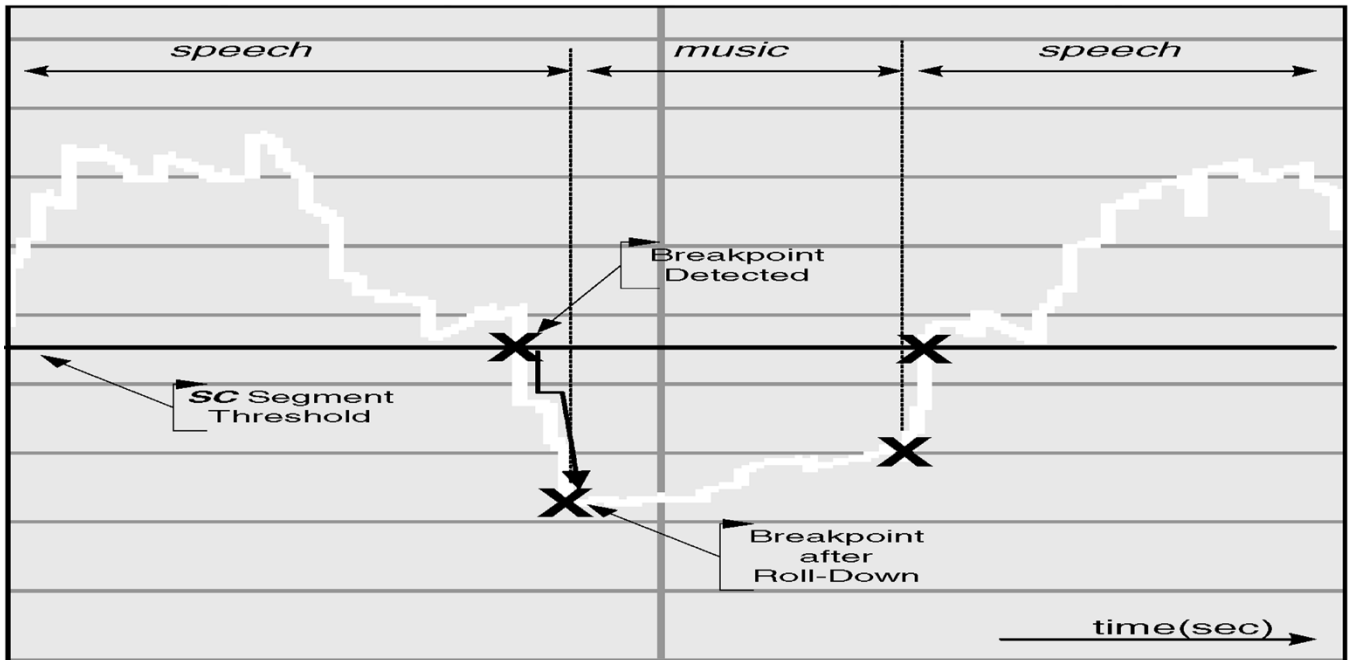


Fig. 15. Windowed SC standard deviation sketch (white) in a *speech* segment. Breakpoints are successfully detected with roll-down algorithm and music subsegment is extracted.

formats and durations. Particularly, three different MUVIS databases are used in the experiments performed in this section.

- 1) **Open** Video Database: This database contains 1500 video clips, obtained from “The Open Video Project” web site [26]. The clips are quite old (from 1960s or older) but contain color video with sound. The total duration of the database is around 46 h and the content mostly contains documentaries, talk shows and commercials. The language is English.
- 2) **Real World** Audio/Video Database: There are 800 audio-only clips and video clips in the database with a total duration of over 36 h. They are captured from several TV channels and the content is distributed among news, commercials, talk shows, cartoons, sports and music clips. The speech is distributed among English, Turkish, French, Swedish, Arabic, and Finnish languages.
- 3) **Music** Audio Database: There are 550 MP3 music files mostly downloaded from the Internet. The music clips are among Classical, Techno, Rock, Metal, Pop, and some other native music types.

All experiments are carried out on a Pentium-4 3.06 GHz computer with 2048 MB memory. The evaluation of the performance is carried out subjectively using only the samples containing straightforward (obvious) content. In other words, if there is any subjective ambiguity on the result such as an insignificant (personal) doubt on the class type of a particular segment (e.g., *speech* or *fuzzy*?) or the relevancy of some of the audio-based retrieval results of an aural query, etc., then that sample is simply discarded from the evaluation. Therefore, the experimental results presented in this section depend only on the decisive subjective evaluation via ground truth and yet they are meant to be evaluator-independent (i.e., same subjective decisions are guaranteed to be made by different evaluators).

This section is organized as follows: Section V-A presents the performance evaluation of the enhanced frame features, their discrimination factors and especially the proposed fuzzy modeling and the final decision process. The accuracy analysis via error distributions and the performance evaluation of the overall segmentation and classification scheme is given in Section V-B. Since the proposed scheme is primarily designed to improve the audio based multimedia retrieval performance, in Section V-C we shall present the role of proposed scheme in audio indexing mechanism and then present the performance improvements on the retrieval among several numerical and visual experiments.

A. Feature Discrimination and Fuzzy Modeling

The overall performance of the proposed framework mainly depends on the discrimination factors of the extracted frame and segment features. Furthermore, the control over the decisions based on the each segment feature plays an important role in the final classification. In order to have a solid control, we only used certain number of features giving significant discrimination for different audio content due to their improved design, instead of having too many traditional ones. As shown in a sample automatic classification and segmentation example in Fig. 17, almost all of the features provide a clean distinction between pure *speech* and *music* content. Also intra segmentation via breakpoints detection works successfully as shown in the upper part of Fig. 17 and false breakpoints are all eliminated.

Yet there are still weak and strong points of each feature used. For instance TR is perceptually blind in the discrimination between *fuzzy* and *music* content as explained before. Similarly FF segment feature might fail to detect harmonic content if the *music* type is *Techno* or *Hard Rock* with saturated beats and base drums. In the current framework such a weak point of a particular feature can still be avoided by the help of the others during

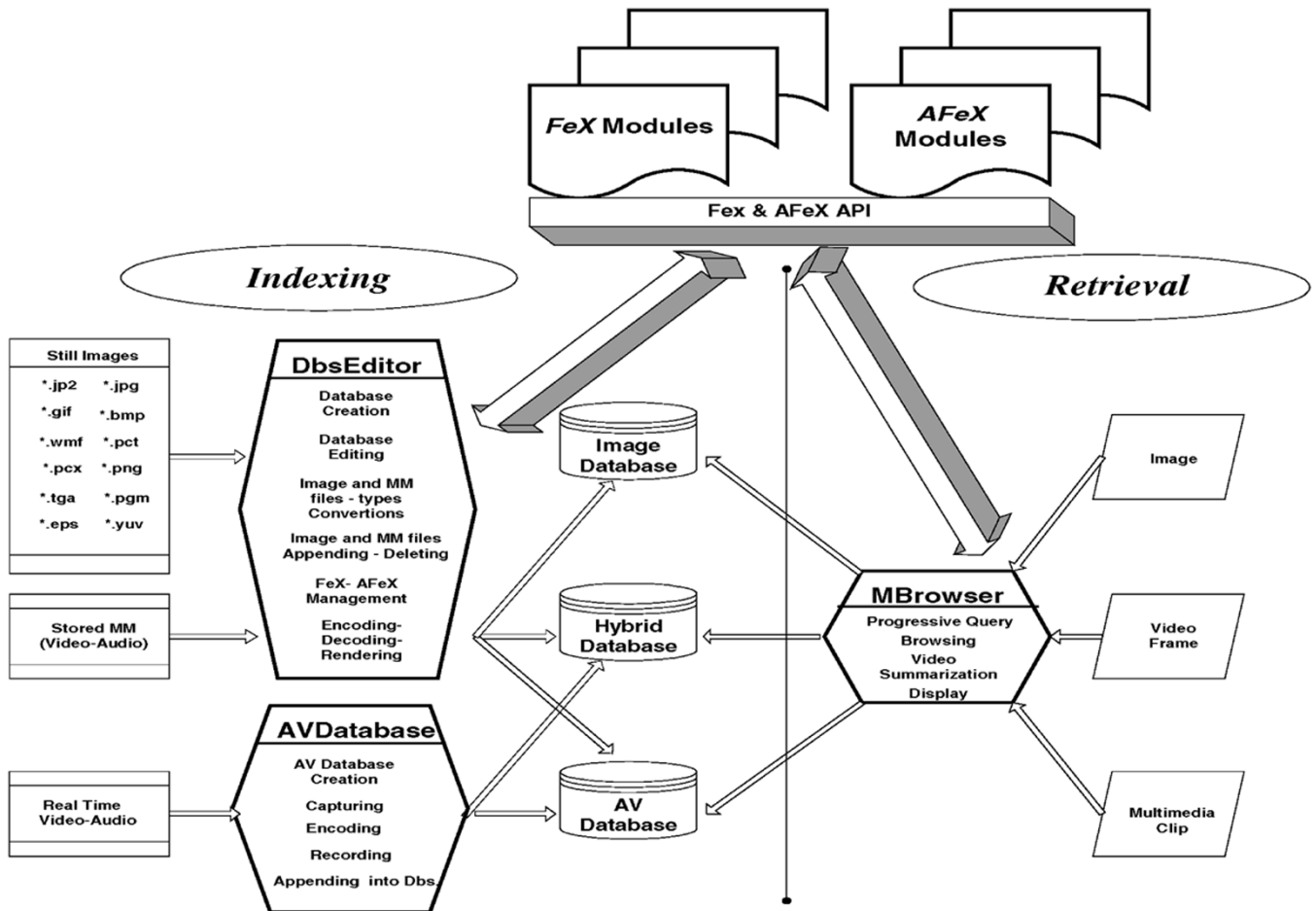


Fig. 16. Generic overview of MUVIS framework.

the final decision process. One particular example can be seen in the example in Fig. 17: FF segment feature wrongly classifies the last (*music*) segment, as *speech*. However, the overall classification result is still accurate (*music*) as can be seen at the top Fig. 17 since both SC and TR features favor *music* that overrides the FF feature (by majority rule) in the end.

B. Overall Classification and Segmentation Performance

The evaluation of the proposed framework is carried out on the standalone MP3, AAC audio clips, AVI and MP4 files containing MPEG-4 video along with MP3, AAC, ADPCM (G721 and G723 in 3–5 bits/sample) and PCM (8 and 16 bits/sample) audio. These files are chosen from **Open**, **Real World**, and **Music** databases as mentioned before. The duration of the clips are varying between 1–5 min up to 2 h. The clips are captured using several sampling frequencies from 16 KHz to 44.1 KHz so that both MPEG 1 and MPEG 2 phases are tested for Layer 3 (MP3) audio. Both MPEG-4 and MPEG-2 AAC are recorded with the *Main* and *Low Complexity* profiles (object types). TNS (Temporal Noise Shaping) and M/S coding schemes are disabled for AAC. Around 70% of the clips are stereo and the rest are mono. The total number of files used in the experiments is above 500 and in total measures, the method is applied to 260 (>15 h) MP3, 100 (>5 h) AAC and 200 (>10 h) PCM (uncompressed) audio clips. Neither the classification and segmentation parameters such as threshold values, window duration, etc., nor

any part of the algorithm are changed for those aforementioned variations in order to test the robustness of the algorithm. The error distributions results, which belong to both *bit-stream* and *generic* modes, are provided in Table IV and V. These results are formed, based on the deviation of the specific content from the ground-truth classification, which is based on subjective evaluation as explained earlier.

In fact, for each and every particular audio clip within the database, the output classification and especially the segmentation result are completely different. Furthermore, as classification and segmentation are mutually dependent tasks, there should be a method of evaluating and reporting the results accordingly. Owing to the aforementioned reasons, neither size nor the number of segments can be taken as a unit on which errors are calculated. Therefore, in order to report the combined effect of both classification and segmentation, the output error distribution, ε_{c^*} , is calculated and reported in terms of the total misclassification-time of a specific class, c^* , per total ground-truth-time (total actual time) of that content within the database formulated as follows:

$$\varepsilon_{c^*}(\%) = 100 \times \frac{\sum_D t(c) \Big|_{c \in (C - c^*)}}{\sum_D t(c^*)} \quad (12)$$

where C represents the set of elements from all class types, t represents time, while D represents the experimental database.

TABLE IV
ERROR DISTRIBUTION TABLE FOR *BIT-STREAM* MODE

BS Type	Speech		Music		Fuzzy
	Critical	Non-Critical	Critical	Non-Critical	Semi-Critical
MP3	2.0 %	0.5 %	5.8 %	10.3 %	24.5 %
AAC	1.2 %	0.2 %	0.5 %	8.0 %	17.6 %

TABLE V
ERROR DISTRIBUTION TABLE FOR GENERIC MODE

Speech		Music		Fuzzy
Critical	Non-Critical	Critical	Non-Critical	Semi-Critical
0.7 %	4.9 %	5.1 %	22.0 %	23.4 %

For example in Table IV, 2.0% for MP3 critical errors in *speech* basically means that if there were 100 min of ground-truth verified *speech* content in the database, 2 min out of those are misclassified as *music*. This error calculation approach makes possible that the results stay independent from the effects of segmentation i.e., the errors are calculated and reported similarly for cases such as the misclassification of a whole segment or the misclassification of a fraction of a segment due to wrong intra-segmentation.

From the analysis of the simulation results, we can see that the primary objective of the proposed scheme, i.e., minimization of critical errors on classification accuracy, is successfully achieved. As a compromise of this achievement, most of the errors are semi-critical and sometimes, as intended, noncritical. Semi-critical errors, despite of having relatively higher values, are still useful, especially considering the fact that the contribution of *fuzzy* content toward the overall size of a multimedia database (also in the experimental database) is normally less than 5%, which in turn means that the overall effect of these high values on the indexing and retrieval efficiency is minor. The moderately valued noncritical errors, as the name suggests, are not critical with respect to the audio-based multimedia retrieval performance because of the indexing and retrieval scheme as discussed in detail in the next section.

C. Experiments on Audio-Based Multimedia Indexing and Retrieval

As mentioned before, this framework is integrated under MUVIS to test and evaluate the audio classification and segmentation role over audio based multimedia indexing and retrieval. As shown in Fig. 18, audio indexing scheme in MUVIS is performed in 4 steps and classification and segmentation of the audio stream is the first step. Once the audio is segmented into 4 class types, the audio frames among three class types (*speech*, *music* and *fuzzy*) are used for indexing. *Silent* frames are simply discarded since they do not carry any content information. The frame conversion is applied in step 2 due to possible difference in duration between the frames used in classification and segmentation and the latter feature extraction modules. The boundary frames, which contain more than one class types are assigned as *uncertain* and also discarded from indexing since

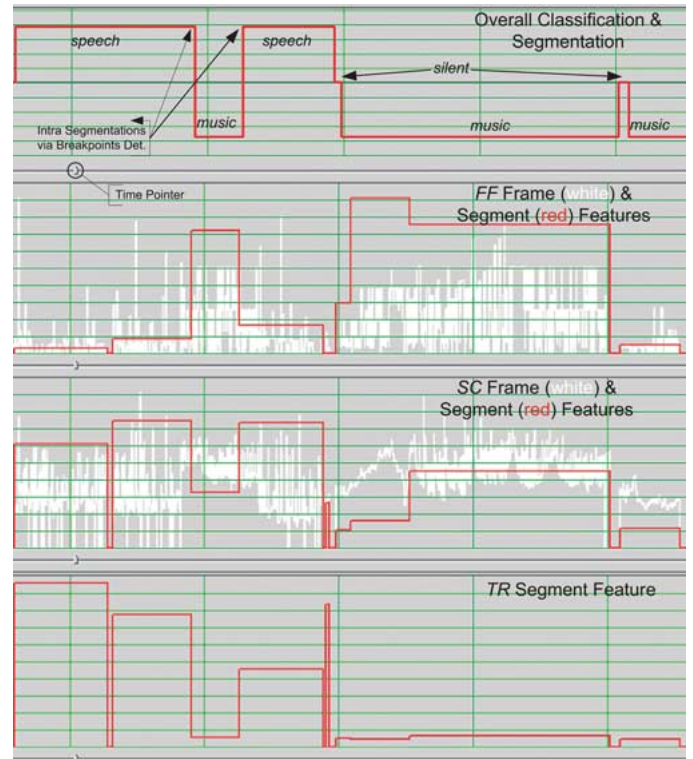


Fig. 17. Frame and Segment Features on a sample classification and segmentation.

their content is not pure, rather mixed and hence do not provide a clean content information. The remaining pure *speech*, *music*, and *fuzzy* frames (within their corresponding segments) are each subjected to audio feature extraction (AFEX) modules and their corresponding feature vectors are indexed into descriptor files separately after a clustering (key-framing) process. Further details about audio indexing and retrieval scheme in MUVIS framework can be found in [8].

As mentioned before, during the retrieval process of a queried audio clip, its feature vectors belonging to a particular class type are only compared with the database items' feature vectors of the corresponding (matching) class type. There is, however, one exception to this rule, which is, the *fuzzy* class type. All frames belonging to *fuzzy* class type are compared with all frames present since by definition *fuzzy* content can carry relevant information from both of the pure class types.

Several experiments are carried out in order to assess the performance of the proposed method. The sample databases are indexed with and without the presence of audio classification and segmentation scheme, which is basically a matter of including/excluding Step-1 (the classification and segmentation module) from the indexing scheme. Extended experiments on audio based multimedia query retrievals using the proposed audio classification and segmentation framework during the indexing and retrieval stage, approve that significant gain is achieved due to filtering the perceptually relevant audio content from a semantic point of view. The improvements in the retrieval process can be described based on each of the following factors.

- *Accuracy*: Since only multimedia clips, containing matching (same) audio content are to be compared

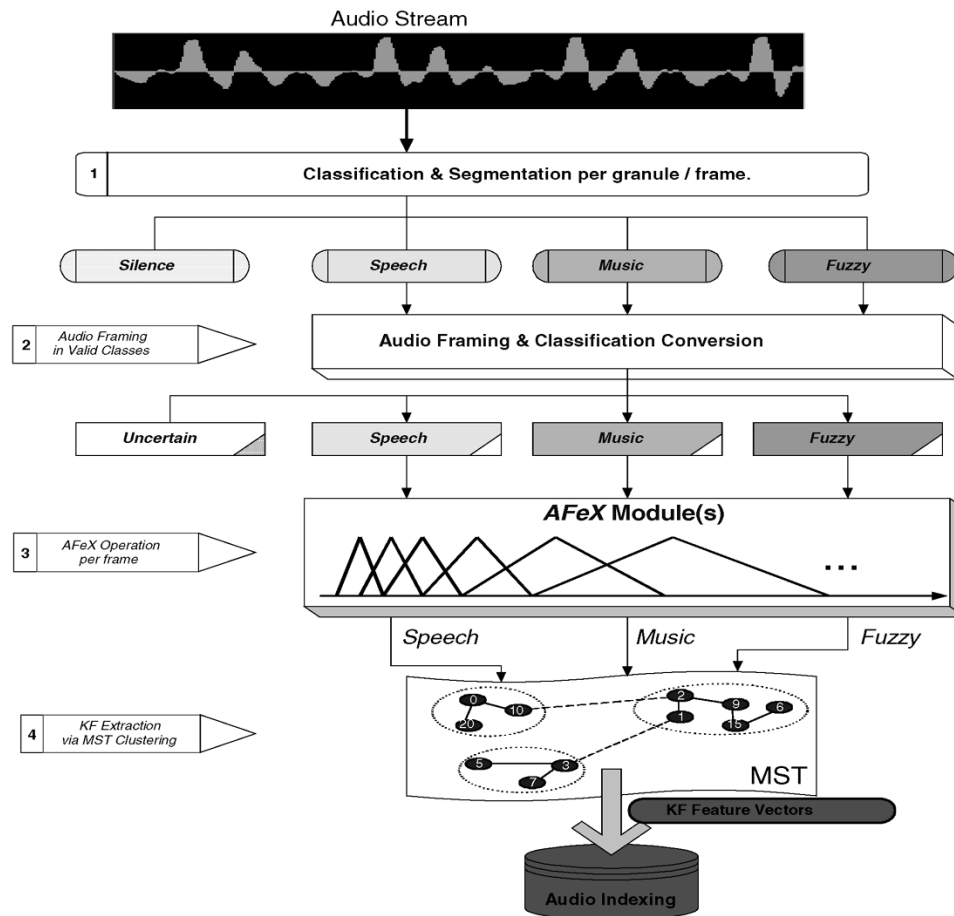


Fig. 18. Audio indexing scheme in MUVIS.

with each other (i.e., *speech* with *speech*, *music* with *music*, etc.) during the query process, the probability of erroneous retrievals can be reduced. The accuracy improvements are observed within 0–35% range for the average retrieval precision. One typical PR (Precision-Recall) curve for an audio-based retrieval of a 2 min multimedia clip bearing pure *speech* content within **Real World** database is shown in Fig. 19. Note that in the left part of Fig. 19, 8 relevant clips are retrieved within 12 retrievals using the proposed scheme. Without using the proposed method, one relevant retrieval is clearly missed in the right side.

- **Speed:** The total elimination of *silent* parts from the indexing scheme reduces the amount of data for indexing and retrieval and hence improves the overall retrieval speed. Moreover, the filtering of irrelevant (different) class types during the retrieval process significantly improves the speed by reducing the CPU time needed for similarity distance measurements and the sorting process afterwards. In order to verify this expectation experimentally and obtain a range for speed improvement, we have performed several aural queries on **Real World** database indexed with and without the proposed method. Among these retrievals we have chosen 10 of them, which have the same precision level in order to have an unbiased measure. Table VI. presents the total retrieval time (the time passed from

the moment user initiates an aural query till the query is completed and results are displayed on the screen) for both cases. As a result the query speed improvements are observed within 7–60% range whilst having the same retrieval precision level.

- **Disk Storage:** Fewer amounts of data are needed and henceforth recorded for the audio descriptors due to the same analogy given before. Furthermore the silent parts are totally discarded from the indexing structure. Yet it is difficult to give a exact numerical figure showing how much disk space can be saved using the proposed method because this clearly depends on the content itself and particularly the amount of silent parts that the database items contain. The direct comparison between the audio descriptor file sizes of the same databases indexed with and without the proposed method shows that above 30% reduction can be obtained.

Fig. 20 shows two examples of audio-based query retrievals of two video clips from **Open** Video database using the **MBrowser** application of MUVIS. In the left example an audio clip with male and female *speech* content is queried and the results ranked with the relevancy from left to right and top to bottom. The first row gives three clips with the same speakers, and the rest of the rows give similar male-female or only male—only female speeches. The right example is another query example for a video clip containing a documentary pro-

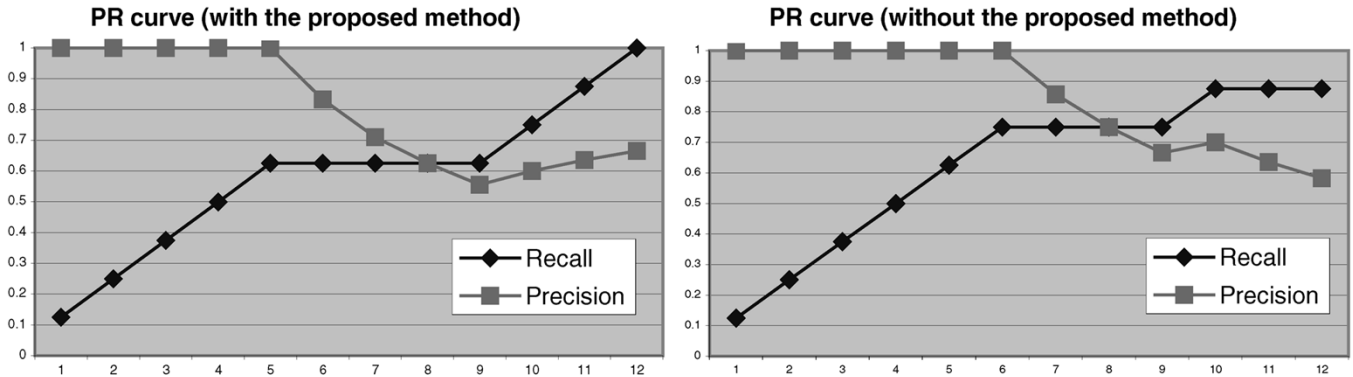


Fig. 19. PR curves of an aural retrieval example within Real World database indexed with (left) and without (right) using the proposed method.

TABLE VI
QTT (QUERY TOTAL TIME) IN SECONDS OF 10 AURAL RETRIEVAL EXAMPLES FROM REAL WORLD DATABASE

Aural Retrieval No.	1	2	3	4	5	6	7	8	9	10
QTT (with proposed method)	47.437	28.282	42.453	42.703	43.844	42.687	46.782	45.814	44.406	41.5
QTT (without proposed method)	30.078	26.266	19.64	39.141	18.016	16.671	31.312	30.578	20.006	37.39
QTT Reduction (%)	36.59	7.12	53.73	8.34	58.9	60.94	33.06	33.25	54.94	9.90

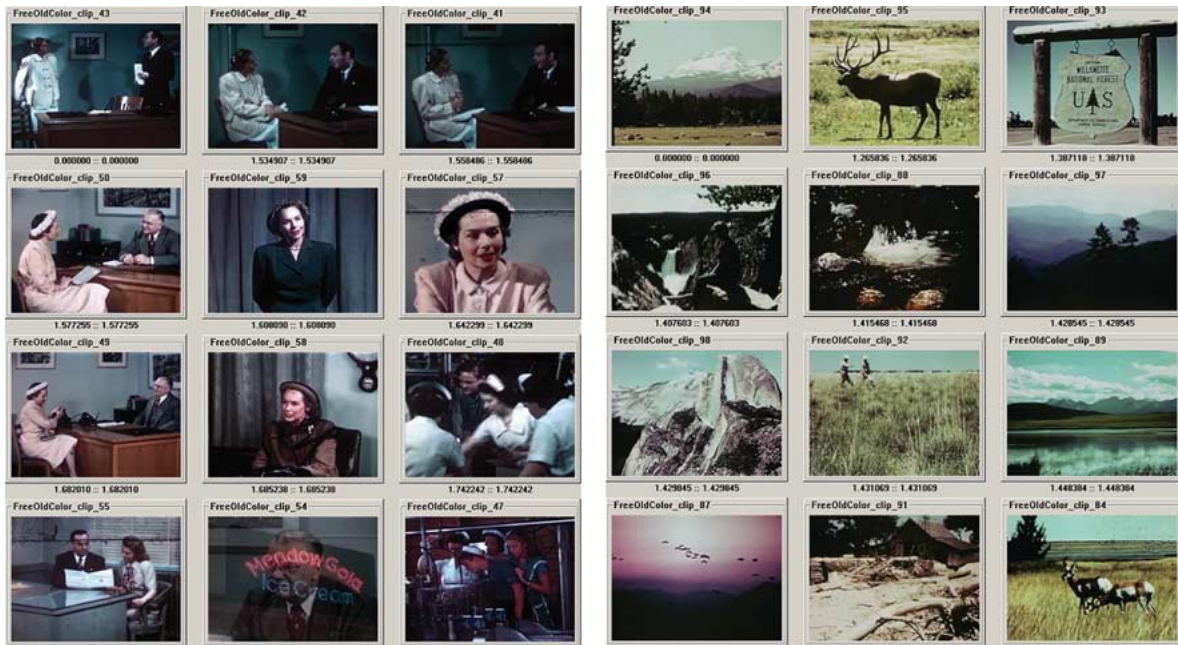


Fig. 20. Audio-based video query retrieval examples. Left-top clip is initially queried.

gram about wild life in the U.S. The audio contains speech with either environmental sounds or an accompanying music and therefore classified (and indexed) mostly as *fuzzy* content. Due to the similar audio content in similar clips, all-relevant results are ranked as the 12-best retrievals presented on the first page.

VI. CONCLUSION

In this paper we presented a study on automatic audio content analysis and a generic framework developed for audio-based in-

dexing and retrieval. Once the shortcomings of the existing systems are addressed and the due requirements are set, we have focused our efforts on providing the structural details of a generic, robust, unsupervised and bi-modal system so as to accomplish a perceptual rule-based approach. We have achieved good results with respect to our primary goal of being able to minimize the critical errors on audio content classification by introducing *fuzzy* modeling in the feature domain and shown the important role of having the global and perceptually meaningful segmentation on the accurate classification (and vice versa) in this con-

text. Furthermore, we have shown that high accuracy can be achieved with sufficient number of enhanced features that are all designed according to the perceptual rules in a well-controlled manner, rather than using a large number of features.

The proposed work achieves significant advantages and superior performance over existing approaches for automatic audio content analysis, especially, in the context of audio-based indexing and retrieval for large-scale multimedia databases. Future work will focus on improving the intra segmentation algorithm and further reduction of semi-critical and noncritical error levels.

REFERENCES

- [1] R. M. Aarts and R. T. Dekkers, "A real-time speech-music discriminator," *J. Audio Eng. Soc.*, vol. 47, no. 9, pp. 720–725, Sep. 1999.
- [2] K.-H. Brandenburg, "MP3 and AAC explained," in *Proc. AES 17th Int. Conf.*, Florence, Italy, Sep. 1999.
- [3] A. Bugatti, A. Flammini, and P. Migliorati, "Audio classification in speech and music: A comparison between a statistical and a neural approach," *Eurasip J. Appl. Signal Process.*, pt. 1, vol. 2002, no. 4, pp. 372–378, Apr. 2002.
- [4] *Coding of Moving Pictures and Associated Audio for Digital Storage Media up to About 1.5 Mbit/s, Part 3: Audio*, 1992. ISO/IEC 11172-3.
- [5] *Coding of Audiovisual Object Part 3: Audio*, 1998. ISO/IEC CD 14496-3 Subpart4: 1998.
- [6] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information—Part 3: Audio*, 1997. ISO/IEC 13818-3:1997.
- [7] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal, "Speech/music discrimination for multimedia applications," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Istanbul, Turkey, 2000, pp. 2445–2448.
- [8] M. Gabbouj, S. Kiranyaz, K. Caglar, E. Guldogan, O. Guldogan, and F. A. Qureshi, "Audio-based multimedia indexing and retrieval scheme in MUVIS framework," in *Proc. Int. Symp. Intelligent Signal Processing and Communication Systems (ISPACS)*, Awaji Island, Japan, 2003.
- [9] R. Jarina, N. Murphy, N. O'Connor, and S. Marlow, "Speech-music discrimination from MPEG-1 bitstream," in *Advances in Signal Processing, Robotics, and Communications*, V. V. Kluev and N. E. Mastorakis, Eds. New York: WSES, 2001, pp. 174–178.
- [10] S. Kiranyaz, M. Aubazac, and M. Gabbouj, *Unsupervised Segmentation and Classification Over MP3 and AAC Audio Bit-Streams*. London, U.K., 2003, pp. 338–345.
- [11] S. Kiranyaz, K. Caglar, O. Guldogan, and E. Karaoglu, "MUVIS: A multimedia browsing, indexing and retrieval framework," in *Proc. 3rd Int. Workshop on Content Based Multimedia Indexing, CBMI 2003*, Rennes, France, Sep., 22–24 2003.
- [12] S. Kiranyaz, K. Caglar, E. Guldogan, O. Guldogan, and M. Gabbouj, "MUVIS: A content-based multimedia indexing and retrieval framework," in *Proc. 7th Int. Symp. Signal Processing and Its Applications, ISSPA 2003*, Paris, France, Jul. 1–4, 2003, pp. 1–8.
- [13] S. Kiranyaz and M. Gabbouj, "A novel multimedia retrieval technique: Progressive query (why wait?)," in *Proc. 5th Int. Workshop on Image Analysis for Multimedia Interactive Services*, Lisboa, Portugal, Apr., 21–23 2004. WIAMIS 2004.
- [14] L. Lu, H. Jiang, and H. J. Zhang, "A robust audio classification and segmentation method," in *Proc. ACM 2001*, Ottawa, ON, Canada, 2001, pp. 203–211.
- [15] MUVIS [Online]. Available: <http://muvis.cs.tut.fi/>
- [16] Y. Nakayima, Y. Lu, M. Sugano, A. Yoneyama, H. Yanagihara, and A. Kurematsu, "A fast audio classification from MPEG coded data," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, Phoenix, AZ, Mar. 1999, pp. 3005–3008.
- [17] M. Noll, "Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate," in *Proc. Symp. Computer Processing Communications*, Brooklyn, NY, 1969, pp. 770–797. Polytechnic Inst. Of Brooklyn.
- [18] D. Pan, "A tutorial on MPEG/audio compression," *IEEE Multimedia*, pp. 60–74, 1995.
- [19] S. Pfeiffer, J. Robert-Ribes, and D. Kim, "Audio content extraction from MPEG encoded sequences," in *Proc. 5th Joint Conf. Information Sciences*, vol. II, 1999, pp. 513–516.
- [20] S. Pfeiffer, S. Fischer, and W. Effelsberg, "Automatic audio content analysis," in *Proc. ACM Int. Conf. Multimedia*, 1996, pp. 21–30.
- [21] J. Saunders, "Real time discrimination of broadcast speech/music," in *Proc. ICASSP96*, vol. II, Atlanta, GA, May 1996, pp. 993–996.
- [22] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature, speech/music discriminator," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Proc.*, Munich, Germany, Apr. 1997, pp. 1331–1334.
- [23] S. Srinivasan, D. Petkovic, and D. Ponceleon, "Toward robust features for classifying audio in the cuevideo system," in *Proc. 7th ACM Int. Conf. Multimedia*, Ottawa, ON, Canada, 1999, pp. 393–400.
- [24] G. Tzanetakis and P. Cook, "Sound analysis using MPEG compressed audio," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP 2000*, vol. II, Istanbul, Turkey, 2000, pp. 761–764.
- [25] E. Terhardt, "Pitch shifts of harmonics, an explanation of the octave enlargement phenomenon," in *Proc. 7th Int. Congr. Acoustics*, Budapest, Hungary, 1971, pp. 621–624.
- [26] The Open Video Project [Online]. Available: <http://www.open-video.org/>
- [27] T. Zhang and J. Kuo, "Video content parsing based on combined audio and visual information," *Proc. SPIE*, vol. IV, pp. 78–89, 1999.
- [28] T. Zhang and C.-C. J. Kuo, "Hierarchical classification of audio data for archiving and retrieving," in *Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Proc.*, Phoenix, AZ, Mar. 1999, pp. 3001–3004.



Serkan Kiranyaz was born in Turkey in 1972. He received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 1994 and M.S. degree in signal and video processing from the same university in 1996.

He was a Researcher with Nokia Research Center and later with Nokia Mobile Phones, Tampere, Finland. He is currently working as a Researcher with the Signal Processing Department, Tampere University of Technology, where he is the Architect and Principal Developer of the ongoing content-based multimedia indexing and retrieval framework, MUVIS. His research interests include, content-based multimedia indexing, browsing and retrieval algorithms, audio analysis and audio-based multimedia retrieval, video summarization, automatic subsegment analysis from the edge field and object extraction, motion estimation and VLBR video coding, MPEG4 over IP, and multimedia programming and processing.



Ahmad Farooq Qureshi was born in Sheikhpura, Pakistan, in 1979. He received the B.S. degree in electronics and communications from the University of Engineering and Technology, Taxila, Pakistan, in 2002. He is currently pursuing the M.S. degree in information technology at Tampere University of Technology, Tampere, Finland.

His research interests include multimedia processing, multimedia content analysis, multimodal signal processing, multimedia on demand, and content-based multimedia indexing and retrieval. He is currently carrying out his research activities as a Researcher at the Digital Media Institute (DMI), Tampere University of Technology.



Moncef Gabbouj (SM'95) received the B.S. degree in electrical engineering in 1985 from Oklahoma State University, Stillwater, and the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1986 and 1989, respectively.

He is currently a Professor and Head of the Institute of Signal Processing of Tampere University of Technology, Tampere, Finland. His research interests include nonlinear signal and image processing and analysis, content-based analysis, and retrieval and video coding. He is coauthor of over 250 publications.

Dr. Gabbouj is the Chairman of the IEEE-EURASIP NSIP (Nonlinear Signal and Image Processing) Board. He was the Technical Committee Chairman of COST 211 and MC Vice-Chair of COST 292. He served as associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and was co-guest editor of the *European Journal of Applied Signal Processing* special issues on multimedia interactive services and signal processing and special issue on nonlinear digital signal processing. He was the TPC Chair of EUSIPCO 2000 and the DSP track chair of the 1996 IEEE ISCAS and the program chair of NORSIG'96. He was also member of EURASIP AdCom. He was co-recipient of the Myril B. Reed Best Paper Award from the 32nd Midwest Symposium on Circuits and Systems and co-recipient of the NORSIG 94 Best Paper Award.