

# Unsupervised Segmentation and Classification over MP3 and AAC Audio Bitstreams

Serkan Kiranyaz, Mathieu Aubazac, Moncef Gabbouj, Tampere University of Technology

## Abstract

The paper presents a novel classification and segmentation scheme for *MP3* and *AAC* audio in the compressed domain. The input audio is split into speech, music and silent segments using features such as total energy, band energy ratio, pause rate, subband centroid and fundamental frequency. Simulation results show the efficiency of the proposed algorithm.

## 1 Introduction

Audio information has been recently used for content-based multimedia indexing and retrieval systems. As for audio segmentation and classification, several methods have been recently reported [7]-[11]. Some of them are supervised methods, which depend on the feedback from the video information in order to segment audio. Although audio classification has been mostly realized in uncompressed domain with the emerging MPEG audio [1], [2] content, several methods have been reported for audio classification on MPEG-1 (Layer 2) encoded audio bitstream. The last years have shown a widespread usage of MPEG Layer 3 (*MP3*) [1], [3], [6] audio as well as proliferation of several video content carrying *MP3* audio (i.e. MPEG-4 video interleaved with *MP3* audio). Because of its high quality for low bit-rate encoding schemes, *MP3* is becoming a common basis wherever the digital audio is concerned. The ongoing research on perceptual audio coding yield to a more efficient successor called (MPEG-2/MPEG-4) Advanced Audio Coding (*AAC*) [4], [5], [6]. *AAC* has various similarities with its predecessor but promises significant improvement in coding efficiency.

This paper describes a method for audio classification and segmentation method directly from *MP3* and *AAC* bitstreams. The proposed method is unsupervised meaning that it does not get any feedback from video and therefore, it can also be applied to any standalone *MP3/AAC* audio clip or to any media primitive that carries *MP3/AAC* audio. For each audio segment the classification will result into one of the following types: speech/music/silent. The method is designed in such a way that it can provide global and reliable solutions for the various encoding parameters and modes such as sampling frequencies (i.e. 8KHz up to 48 KHz), channel modes (i.e. mono, stereo, etc.), compression bit-rates (i.e. 8kbps up to 448kbps). It has a hierarchic structure so that segmentation is performed in four iterative steps whilst the features needed for classification is extracted iteratively.

## 2 MP3 and AAC Overview

MPEG audio is a group of coding standards that specify a high performance perceptual coding scheme to compress audio signal into several bit-rates. It 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 the 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) basically 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 might 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 6 samples. Then MDCT is performed to each (short) window individually and the final 18 MDCT coefficients are obtained as a result of three groups of 6 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 performed directly to 18 samples from each of 32 subbands. In *Short Windowing Mode*, all of 32 subbands are short windowed as the aforementioned way. In *Mixed Windowing Mode*, first two lower subbands are long windowed and the remaining 30 higher subbands are short windowed. Once MDCT is performed to each subband of a granule according to the windowing mode, then the scaled and quantized MDCT coefficients are then Huffman coded and thus the *MP3* bitstream 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 the encoding different windowing modes can be applied to each granule. MPEG-2 Layer 3 is a backwards compatible extension to MPEG-1 with up to five channels, plus one low frequency enhancement channel. In addition to that it provides support for lower sampling rates 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.

Mainly AAC has a similar structure with MP3. 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, scalefactors and Huffman coding) are applied to reduce the number of bits used for encoding. Another important difference is that AAC has a better frequency resolution up to 1024 frequency lines compared to 576 for MP3. 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 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. There are other new tools to enhance the audio coding quality (i.e. Enhanced Block Switching). More detailed information about MP3 and AAC can be found in [6].

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 template based on MDCT coefficients. This template allows us to achieve a generic classification and segmentation technique that uses the compressed domain audio features as explained in the next section.

### 3 Formalization of Compressed Domain Audio Features

The proposed method uses on the compressed domain audio features in order to perform classification and segmentation directly from the compressed bit-stream. The formalization of such audio features are based on forming of a generic MDCT sub-band template. Once the MDCT template formation is completed then the proposed algorithm can be applied to both types of bit-streams independent from the underlying encoding scheme. In the following sub-section this template formation is explained in detail.

#### 3.1 Forming the MDCT Template from MP3/AAC Bitstream.

As explained in section 2, due to the variations among several parameters and modes such as sampling frequency, windowing type and audio channel number (mono or stereo) requires a formation of a common template in order to achieve a generic feature extraction technique. This so called 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. The Table 1 represents array dimensions  $NoW$  and  $NoF$  respectively.

Compression Type and 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 1: The MDCT template array dimension with respect to Compression Type and Windowing Mode.**

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 uses linearly spaced frequency lines, then the real

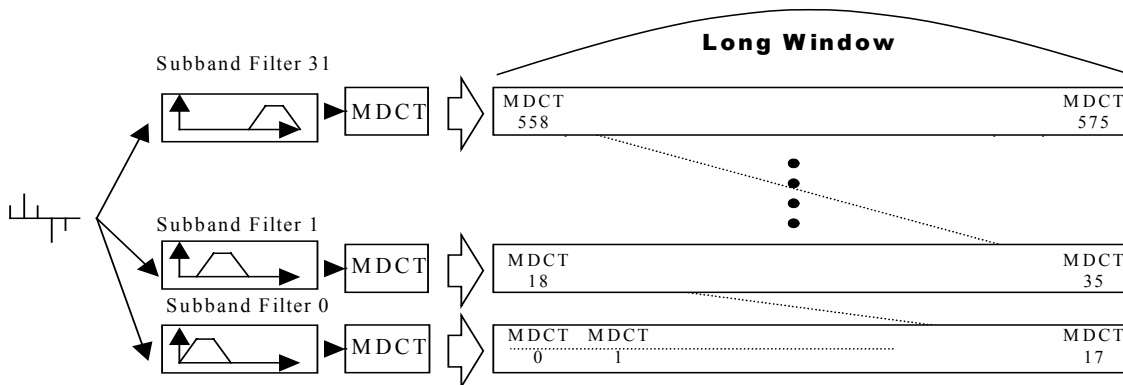
frequency values to which the index  $f$  represents can be obtained from the array  $FL(f)$  using the following equation:

$$FL(f) = \begin{cases} \left( (f+1) \times \frac{f_{BW}}{NoF} \right) & \text{if not MP3 Mixed Windowing Mode} \\ \left( (f+1) \times \frac{f_{BW}}{576} \right) f < 36 \\ \left( \frac{f_{BW}}{16} + \frac{(f-35) \times f_{BW}}{192} \right) f \geq 36 \end{cases} \text{ if MP3 Mixed Windowing Mode}$$

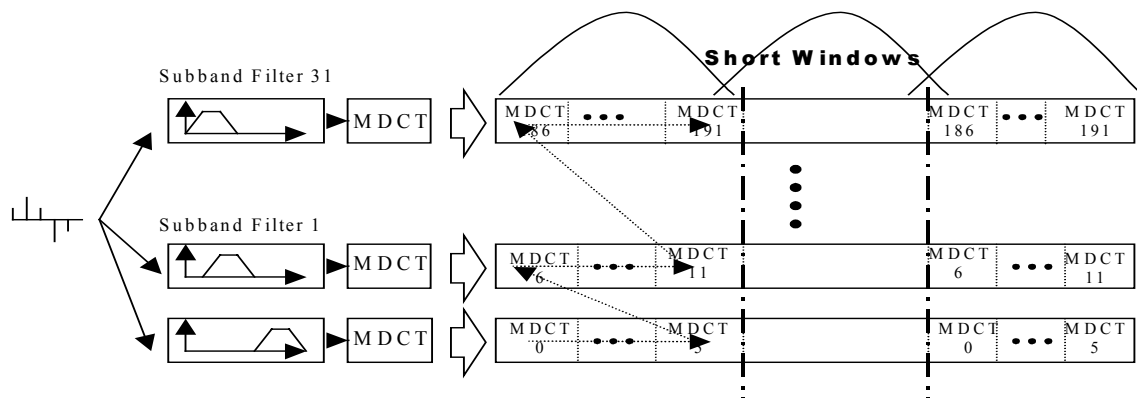
**Equation 1: The line frequency array formation.**

where  $f$  is the index from 0 to corresponding  $NoW$  given in Table 1.

The MDCT template array is formed from the absolute values of the MDCT subband coefficients, which are (Huffman) decoded from the MP3/AAC bitstream per MP3 granule or AAC frame. For each MP3 granule, the MDCT subband coefficients are directly under the form of a matrix of 32 lines, representing the frequency subbands, with 18 columns each of which for every coefficient as shown in Figure 1. In case of short window, there are three windows within a granule containing 6 coefficients. The template matrix formation for short window MP3 granules is illustrated in Figure 2. 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. Figure 3 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. Figure 4 shows how the subbands are arranged and the template array is formed by this technique.



**Figure 1: MP3 Long Window MDCT template array formation from MDCT subband coefficients.**



**Figure 2: MP3 Short Window MDCT template array formation from MDCT subband coefficients.**

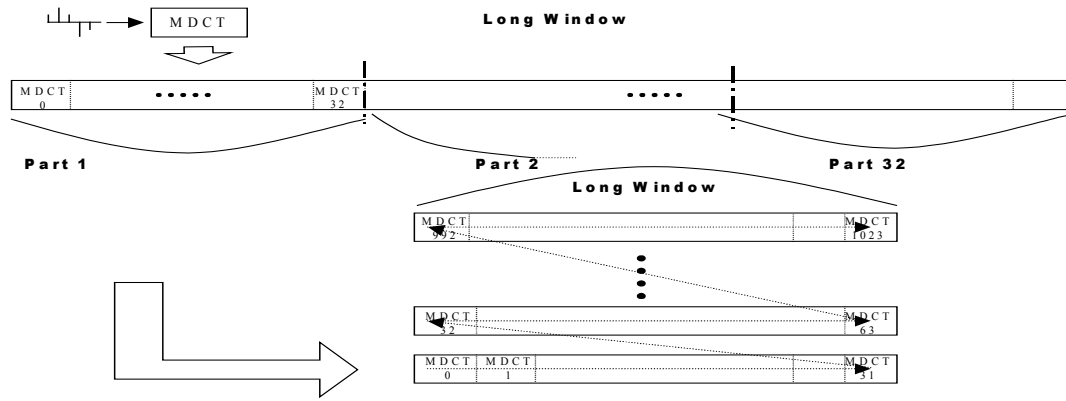


Figure 3: AAC Long Window MDCT template array formation from MDCT subband coefficients.

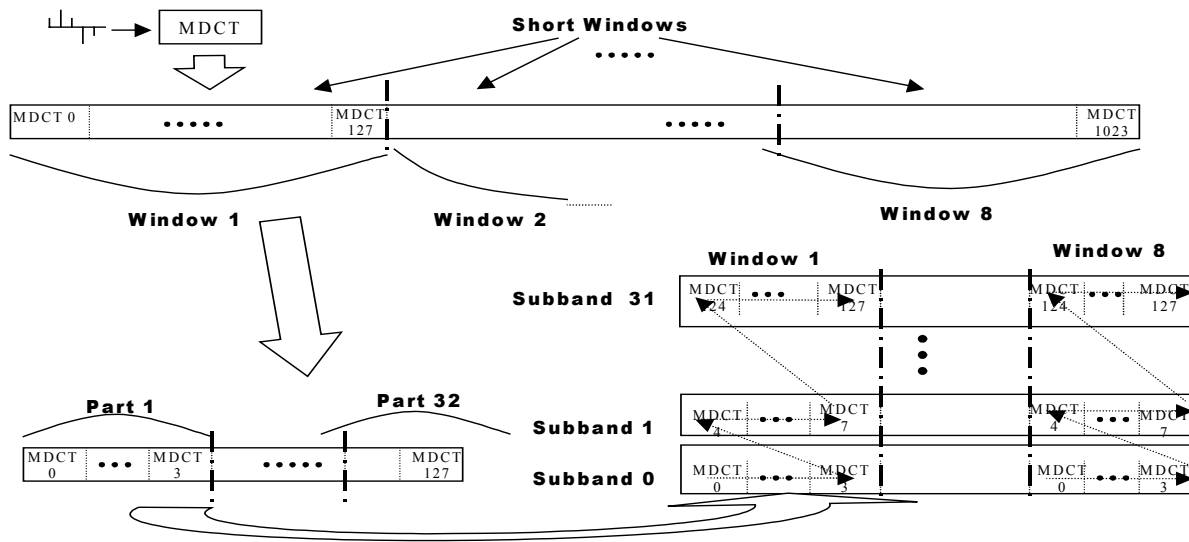


Figure 4: AAC Short Window MDCT template array formation from MDCT subband coefficients.

### 3.2 Feature Extraction in Compressed Domain

The compressed domain features entirely depend on the MDCT template array  $MDCT(w, f)$  along with the frequency line array  $FL(f)$ , both of which are formed per MP3 granule or AAC frame. For this reason, there are limited number of feasible features that can be extracted using only MDCT template such as features based on *Total Frame Energy (TFE)*, *Band Energy Ratio (BER)*, *Fundamental Frequency (FF)* and *Subband Centroid (SC)*. The proposed technique uses these features to accomplish audio segmentation and classification (per segment accordingly).

#### 3.2.1 TFE Calculation

*TFE* can be calculated using Equation 2. It is the primary feature to detect silence granules/frames. Silence detection is also used for *Pause Rate* calculation, which is one of the main features for classification of a segment.

$$TFE_j = \sqrt{\sum_w \sum_f^{NoW NoF} (MDCT_j(w, f))^2}$$

Equation 2: TFE calculation for granule/frame j.

#### 3.2.2 BER Calculation

*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

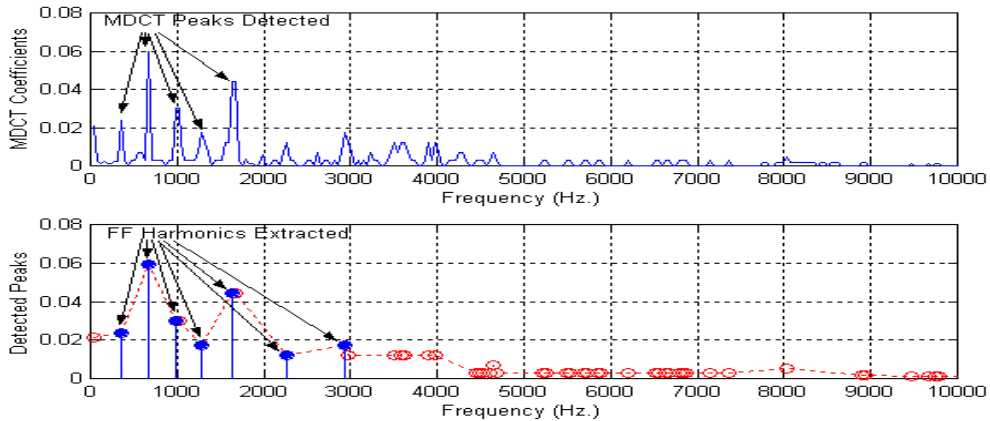
$f_c$  ( $f_c \leq f_{BW}$ ), let  $f\langle f_c \rangle$  be the line frequency index where  $FL(f\langle f_c \rangle) \leq f_c < FL(f\langle f_c \rangle + 1)$ , BER for a granule/frame  $j$  can be calculated using Equation 3.

$$BER_j(f_c) = \frac{\sqrt{\sum_w^{NoW} \sum_{f=0}^{f\langle f_c \rangle} (MDCT_j(w, f))^2}}{\sqrt{\sum_w^{NoW} \sum_{f=f\langle f_c \rangle}^{NoF} (MDCT_j(w, f))^2}}$$

**Equation 3: BER calculation for granule/frame  $j$ .**

### 3.2.3 FF Estimation

If the input audio signal encoded by AAC or MP3 is harmonic over a fundamental frequency (i.e. there exists a series of major frequency components that are integer multiples of a fundamental frequency), the real FF can be estimated from the MDCT coefficients that are nothing but the linearly spaced spectral components. Therefore, we apply an adaptive peak-detection algorithm over the MDCT template to check whether sufficient number of peaks around the integer multiple of a certain frequency can be found or not. The peak detection is a critical process for FF calculation. One potential problem might be that the peak value may not be necessarily on the frequency line that MDCT coefficient exists and therefore, an adaptive search window should be applied in order not to miss a peak on a multiple frequency line. On the other hand a non-harmonic audio frame might have other major spectral components that might fall into the range of the search window if the window width is chosen larger than necessary. Let the linear frequency spacing between two consecutive MDCT coefficient be  $\Delta f = FL(f) - FL(f - 1) = f_{BW} / NoF$  and let the real FF value will be in the  $\{-\Delta f/2, +\Delta f/2\}$  neighborhood of a MDCT coefficient at the frequency  $FL(f)$ . Then the minimum window width to search for  $n^{th}$  (possible) peak will be:  $W(n) = n \times \Delta f$ . Especially the human speech have most of its energy at lower bands (i.e.  $f < 500\text{Hz}$ .) 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 another adaptive detection design in order to detect the major spectral peaks in the spectrum. We therefore, apply a non-overlapped 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 MDCT coefficients in the partition and if a MDCT coefficient is significantly bigger than the mean value (i.e. greater than three times the mean value), it is chosen as a new peak and this process is repeated for all the partitions. The maximum MDCT 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 data is significantly reduced and the major peaks are extracted. Figure 5 illustrated a sample peak detection applied on the MDCT coefficients of a long windowed MP3 granule with sampling frequency 44100 Hz. Therefore,  $f_{BW} = 22050 \text{ Hz}$  but the sketch shows up to 10000 Hz for the sake of illustration. The lower subplot shows the peaks detected and finally the FF value estimated accordingly ( $FF = FL(8) = 351 \text{ Hz}$  in this example).



**Figure 5: A sample FF detection applied on a MP3 granule.**

### 3.2.4 SC Frequency Estimation

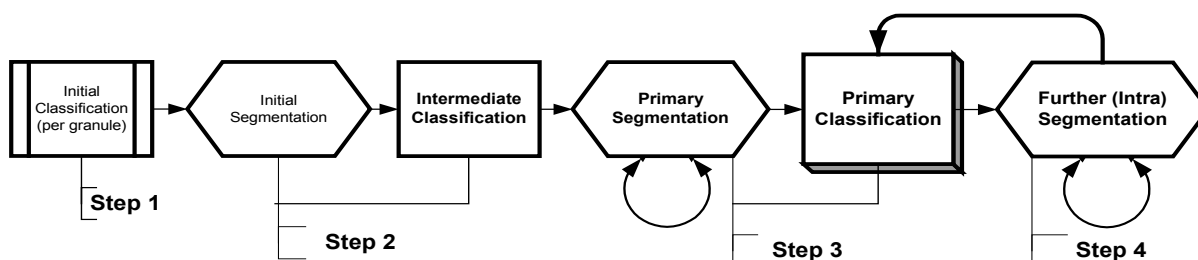
SC is the first moment of the spectral distribution (spectrum) or in compressed domain it can be estimated as the balancing frequency value for the absolute MDCT values. Using the MDCT template arrays, SC frequency ( $f_{SC}$ ) can be calculated using Equation 4.

$$f_{SC} = \frac{\sum_w \sum_f^{NoW} (MDCT(w, f) \times FL(f))}{\sum_w \sum_f^{NoW} MDCT(w, f)}$$

**Equation 4: SC frequency calculation**

## 4 MP3/AAC Classification and Segmentation

Audio segmentation and classification are closely related and internally dependent problems. Achieving 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 frame based classification and initial segmentation, to ensure a global segmentation outcome and thus a successful classification per segment at the end. Figure 6 illustrates our iterative approach to the audio classification and segmentation problem.



**Figure 6: The flowchart of the proposed method.**

As shown in Figure 6, there are 4 hierarchical steps for MP3/AAC classification and segmentation:

**1) Initial Classification:** 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 Equation 2.  $T_{TFE}$  is adaptively calculated in order to take the volume effect into account. The minimum ( $E_{min}$ ), maximum ( $E_{max}$ ) and average ( $E_{\mu}$ ) granule/frame energy values are first calculated from the entire audio clip. If there exists a significant difference (i.e. at least 10 times) between the minimum and maximum granule/frame energy values, then the audio clip consists of both *silent* and *non-silent* granules/frames, otherwise the entire clip is considered as *silent*. Once the presence of *non-silent* granules/frames is confirmed then  $T_{TFE}$  is calculated according to Equation 5.

$$T_{TFE} = E_{min} + \lambda_s \times (E_{\mu} - E_{min}), \text{ where } 0 < \lambda_s \leq 1$$

**Equation 5: Silence/Non-silence threshold calculation.**

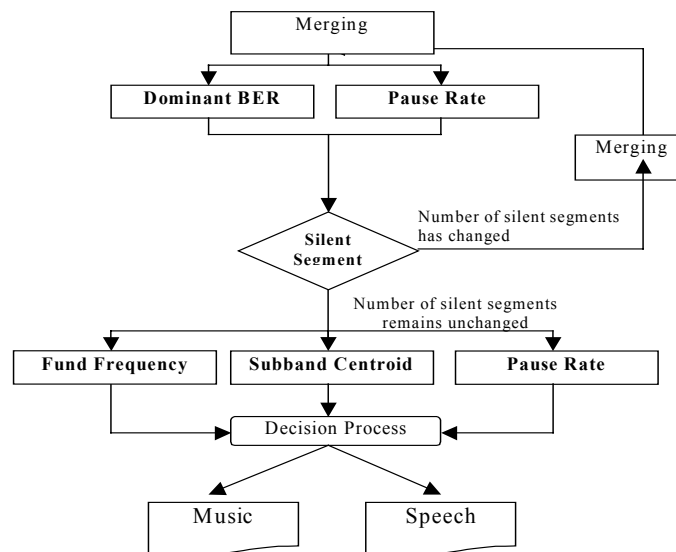
$\lambda_s$  is the silence coefficient, which determines the silence threshold value between  $E_{min}$  and  $E_{\mu}$ . If the total energy of a granule/frame is below  $T_{TFE}$ , then it is classified as *silent*, otherwise *non-silent*. If the granule/frame is not classified as *silent*, the band energy ratio (**BER**) is calculated for each granule/frame. If this ratio is over a threshold value ( $T_{BER}$ ) the granule/frame is then classified as *music*, otherwise *speech*. Figure 7 summarizes the operation performed in Step 1.

**2) Segmentation and Feature extraction per Segment:** In this step, first of all *silent* and *non-silent* 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. A preset threshold value (i.e. 0.2s) 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



Once the merging loop is terminated, there might still exist some short non-silent segments that are not merged to any neighbor global segments. Such short *non-silent* frames are naturally false segments and therefore, should also be eliminated. After the elimination of the short *non-silent* segments, the remaining *non-silent* segments are global enough to contain one single classification type. This is further important to have global segments to run primary feature extraction techniques based on Subband Centroid (SC) and Fundamental Frequency (FF) estimation. The voiced speech tends to have lower fundamental frequency values compared to music. The unvoiced speech is not harmonic at all (FF=0 Hz). Due to the presence of unvoiced speech parts together with the natural speech pauses within a speech segment, the percentage of the granules that have a non-zero fundamental frequency value will also be lower than the ones in a music segment. By taking these two criteria into account, the multiplication of the average fundamental frequency value with the percentage of granules having non-zero fundamental frequency within a non-silent segment is chosen to be the feature based on FF estimation. This feature value is then compared to a threshold ( $T_{FF}$ ) for each *non-silent* segment in the clip and classification based on FF estimation is performed accordingly. *Figure 11* middle section shows a sample sketch of the feature based on FF estimation over an entire MP3 audio clip with both *speech* and *silent* segments that can be differentiated clearly.

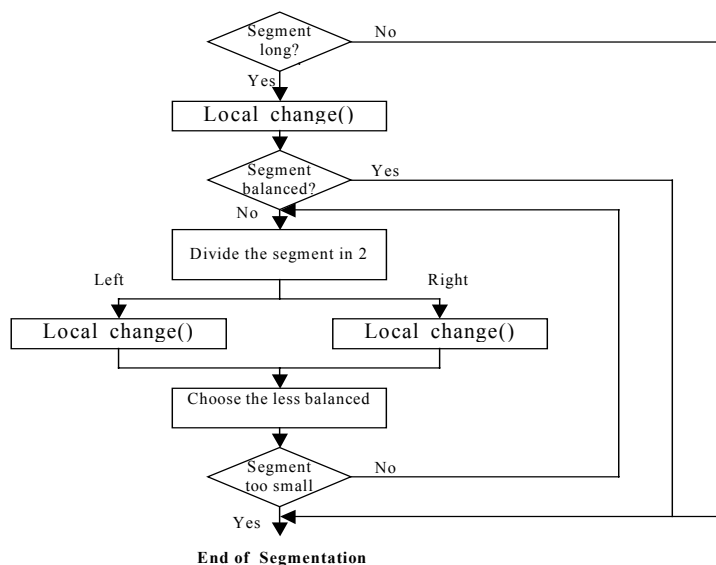
A similar argument can be made for the feature extraction based on SC estimation, that is, due to the presence of both voiced and unvoiced speech parts in a speech segment, the average SC value tend to be smaller with a significantly bigger standard deviation value compared to music segments. This can be observed on the SC curve in *Figure 11*. Therefore, the feature used for the classification based on SC estimation is fixed as the ratio between the mean and standard deviation of the SC values within a non-silent segment. This feature value is then compared to a threshold ( $T_{SC}$ ) for each non-silent segment in the clip and hence the classification based on SC estimation is performed accordingly. The final classification type is classification by **Pause Rate**. As a result, the majority of three classification results from each technique (**Pause Rate**, **Fundamental Frequency** and **Subband Centroid**) makes the final decision over a segment. *Figure 9* summarizes all operations in Step 3.



**Figure 9: Primary Segmentation and Classification in Step 3**

**4) Further Segmentation (Sub-Segment Analysis):** Once final classification and segmentation is finished in Step 3, a further segmentation is performed in order to separate sub-segments, which are not separated by silent parts. For example within a segment there might be sub-segments that include music and speech without a silent part in between. The first part in this step tests if the non-silent segment is significantly larger than a given threshold, (i.e. typically 20 times the threshold chosen for silent segmentation). Then we start by dividing the segment into two sub-segments and test if the subband centroid feature values are nearly the same for the new parts. If so, we keep the large segment and stop. Otherwise we execute the same operation over the two new segments and look for the one, which is less balanced (the one which has larger centroid frequency difference between the left and the right child-segments). The iteration is carried out till the segment is small enough and breaks the iteration loop. This gives the sub-segment boundary and then Step 3 is re-performed over the new segments in order to make the correct classification. If Step 3 results in the same segment types for both sub-segments, then the big segment is kept unchanged. This means a false detection has been performed in Step 4. *Figure 10* illustrates the algorithm in detail.

The function *local\_change()* performs **SC** based classification for the right and left segments and returns the absolute centroid frequency difference between them.



**Figure 10: Further Segmentation in Step 4.**

## 5 Simulation Results and Conclusion

Experiments are carried out on both standalone *MP3* and *AAC* audio clips, and *AVI* and *MP4* files containing *MPEG-4* video along with *MP3* or *AAC* audio. Some of the clips containing *MP3* and all of clips containing *AAC* are real-time recorded from TV channels showing News, Cartoon, Talk Show, Music Clips and Commercials. The rest of them are ordinary *MP3* clips downloaded from Internet. The duration of clips are varying between 1-3 minutes up to 2 hours. The clips are recorded using several sampling frequencies from 16 KHz to 44.1 KHz so that both *MPEG 1* and *MPEG 2* phases are thus tested for Layer 3 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 is mono. Some clips are both recorded in mono and stereo in order to test the effect of channel numbers on the performance. The following threshold values are used:  $T_{BER} = 2$ ,  $T_{FF} = 350\text{Hz}$ ,  $T_{SC} = 2.75$ ,  $T_{PR} = 8\%$ ,  $\lambda_s = 15\%$ ,  $f_c = 500\text{Hz}$ ,  $N_p = 20$ . In total measures, the method is applied onto 214 (~28 hours) *MP3* and 85 (~6 hours) *AAC* clips. The following table presents the success rate achieved separately on *AAC* and *MP3* clips with respect to several test cases such as audio with only speech, audio with only music and audio with both speech and music.

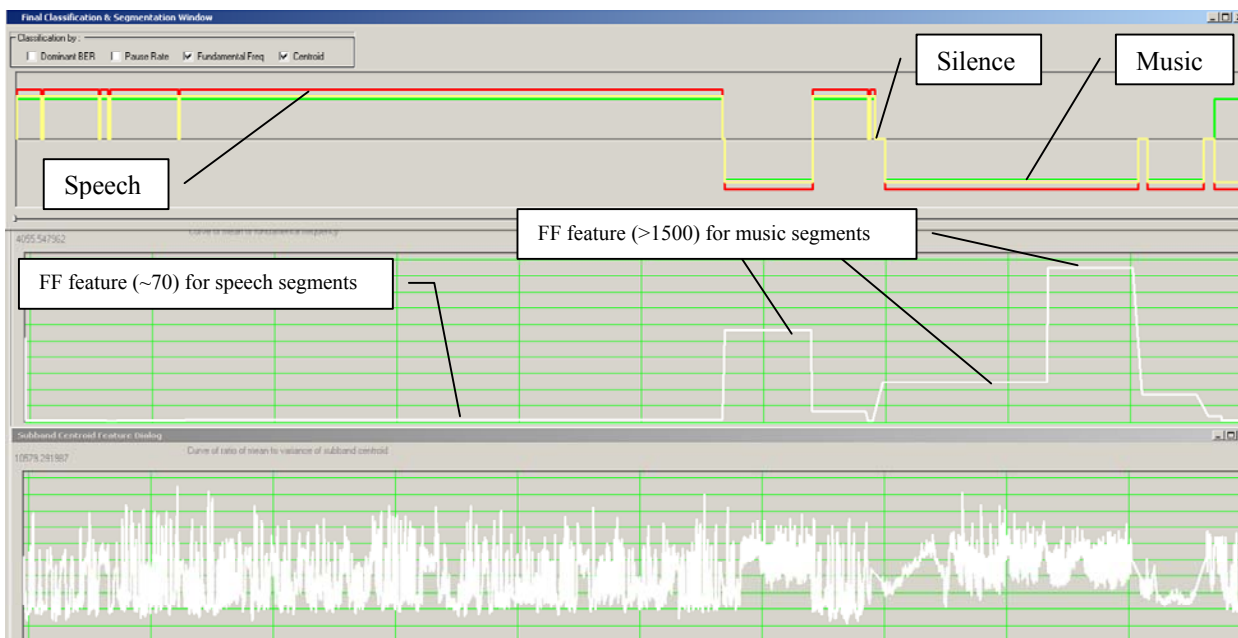
	Only Speech	Only Music	Speech and Music
MP3 Segmentation Success %			
MP3 Classification Success %			
AAC Segmentation Success %			
AAC Classification Success %			

**Table 2: The success rate of the proposed method on various test cases.**

Figure 11 gives one example of a clip containing both music and speech. Several classification curves are displayed at top section: Classification based on **FF** (green), **SC** (yellow) and final result (red curve) where as the value +1 is used for speech, 0 for silent and -1 for music.

The proposed algorithm uses directly the information from the encoded bit-stream without processing full decoding. In fact only the huffman decoding is performed to extract MDCT coefficients and then the MDCT template is formed to achieve generic structure for both *MP3* and *AAC* audio. In order to achieve global segments with logical classification, the method has been designed in a hierarchical structure. At each step some classification is applied in the non-silent segments and then new non-silent segments are re-formed within a merging loop.

Therefore, the method achieves a logical iteration in order to obtain global segmentation and successful classification per segment within the audio clip.



**Figure 11: Final Classification of a MP3 clip (top), FF based features per segment (middle), SC curve (bottom).**

## 6 Conclusion

In this paper an unsupervised classification and segmentation algorithm has been proposed for MP3 and AAC audio. The algorithm uses directly the information from the encoded bit-stream without processing full decoding. In fact only the huffman decoding is performed to extract MDCT coefficients and a subband template is formed to achieve generic structure, which makes the proposed algorithm generic for both MP3 and AAC audio. In order to achieve global segments with logical classification, the method has been designed in a hierarchical structure. At each step some classification is applied in the non-silent segments and then new non-silent segments are re-formed within a merging loop. Therefore, the method contains a logical iteration in order to obtain global segments within the audio clip.

In the future, further feature extraction schemes will be applied over the segments found by this method in order to use this method in a content-based multimedia indexing and retrieval system. For the moment we have achieved to segment and classify the audio information in an accurate way. The accuracy can also be improved if the threshold values are adaptively set.

## 7 References

- [1] ISO/IEC 11172-3, Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s, Part 3: Audio, 1992.
- [2] ISO/IEC 13818-3:1997, Information technology -- Generic coding of moving pictures and associated audio information -- Part 3: Audio, 1997.
- [3] D. Pan, "A tutorial on MPEG/Audio Compression", IEEE Multimedia, pp 60-74, 1995.
- [4] ISO/IEC CD 14496-3 Subpart4: 1998, Coding of Audiovisual Object Part 3: Audio, 1998.
- [5] M. Watson and P. Buettner, "Design and Implementation of AAC Decoders", IEEE Trans. on Consumer Electronics, 2000.
- [6] Karl-Heinz Brandenburg, "MP3 and AAC Explained", AES 17th International Conference, Florence, Italy, September 1999.
- [7] Z. Liu, J. Huang, Y. Wang and T. Chen, "Audio Feature Extraction and Analysis for Scene Classification", Electr. Proc. Of IEEE Workshop on Multimedia Signal Processing, Princeton, pp. 1-6, NJ, June 1997.
- [8] Nakayima Y, Lu Y, Sugano M, Yoneyama A, Yanagihara H, Kurematsu A "A Fast Audio Classification from MPEG Coded Data". In Proceedings of Int. Conf. on Acoustics, Speech, and Signal Proc., vol. 6, pp 3005-3008, Phoenix, AZ, March 1999.
- [9] Patel N, Sethi I, "Audio Characterization for video indexing" in Proc. of SPIE Conference on Storage and Retrieval for Still Image and Video Databases, vol.2670, pp. 373-384, San Jose, 1996.
- [10] Zhang T, Jay Kuo C-C "Hierarchical Classification of Audio Data for Archiving and Retrieving", Proc. ICASSP'99, Vol. 6, pp. 3001-3004, Phoenix, Mar. 1999.
- [11] R. M. Aarts and R. T. Dekkers, "A Real-Time Speech-Music Discriminator", J. Audio Eng. Soc., Vol 47, No 9, pp. 720-725, Sept. 1999.