UNSUPERVISED SEGMENTATION AND CLASSIFICATION OVER MP3 AND AAC AUDIO BITSTREAMS

SERKAN KIRANYAZ, MATHIEU AUBAZAC, MONCEF GABBOUJ
Tampere University of Technology, Inst. of Signal Processing, Tampere, FINLAND.
E-mail: serkan@cs.tut.fi

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 [3], [4], [5]. Although audio classification has been mostly realized in 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 [3]-[5]. The last years have shown a widespread usage of MPEG Layer 3 (MP3) audio [1], [2] as well as proliferation of several video content carrying MP3 audio. The ongoing research on perceptual audio coding yield to a more efficient successor called (MPEG-2/4) Advanced Audio Coding (AAC) [2]. 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 bit-streams. 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.

2. Formalization of Compressed Domain Audio Features

The formalization of audio features is 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.

2.1. Forming the MDCT Template from MP3/AAC Bitstream

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 sized 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 \( f \) represents the line frequency index. Table 1 represents array dimensions \( NoW \) and \( NoF \) respectively.

Table 1. MDCT template array dimension with respect to Type and Mode.

<table>
<thead>
<tr>
<th>Compression Type and Windowing Mode</th>
<th>NoW</th>
<th>NoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3 Long Window</td>
<td>1</td>
<td>576</td>
</tr>
<tr>
<td>MP3 Short Window</td>
<td>3</td>
<td>192</td>
</tr>
<tr>
<td>MP3 Mixed Window</td>
<td>3</td>
<td>216</td>
</tr>
<tr>
<td>AAC Long Window</td>
<td>1</td>
<td>1024</td>
</tr>
<tr>
<td>AAC Short Window</td>
<td>8</td>
<td>128</td>
</tr>
</tbody>
</table>

Let \( f_s \) be the sampling frequency. Then according to Nyquist's theorem the maximum frequency \( f_{BW} \) of the audio signal will be: \( f_{BW} = \frac{f_s}{2} \). 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 Eq. (1).

\[
FL(f) = \begin{cases}
(f + 1) \times \frac{f_s}{NoF} & \text{if not MP3 Mixed Windowing Mode} \\
(f + 1) \times \frac{f_{BW}}{576} & \text{if MP3 Mixed Windowing Mode}
\end{cases}
\]

\( f < 36 \)  
\( \frac{576}{16} + \left( f - 35\right) \times \frac{f_s}{192} & \text{if MP3 Mixed Windowing Mode}
\]

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. In case of short window, there are three windows within a granule containing 6 coefficients. 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.
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.

2.2. Feature Extraction in Compressed Domain
The compressed domain features entirely depend on the MDCT template array
\( \text{MDCT} \ (w, f) \) along with the frequency line array \( \text{FL} \ (f) \), 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).
1) TFE Calculation: TFE can be calculated using Eq. (2). It is the primary feature
to detect silent granules/frames. Silence detection is also used for Pause Rate
calculation, which is one of the main features for classification of a segment.

\[
\text{TFE}_j = \sqrt{\sum_w \sum_f (\text{MDCT}_{(w, f)}^2)}
\]

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 \( f_c \left( f_c < f_{nw} \right) \), let \( \text{FL}(f_c) \) be the line frequency
index where \( \text{FL}(f_c) \leq f_c < \text{FL}(f_c+1) \). BER for a granule/frame \( j \)
can be calculated using Eq. (3).

\[
\text{BER}_{j, (f_c)} = \frac{\sum_w \sum_{f_{c-1}}^{f_c} (\text{MDCT}_{(w, f)}^2)}{\sum_w \sum_{f_{c+1}}^{f_{nw}} (\text{MDCT}_{(w, f)}^2)}
\]

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 nonharmonic 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 \( \Delta f = F(f) - F(f - 1) = f_{nyq}/N_{0F} \) 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: \( W(n) = n \times \Delta f \). We 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 \( f_{nyq} \) / \( N_p \) 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, it is chosen as a new peak and this process is repeated for all the partitions.

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 MDCT template arrays, SC frequency \( f_{sc} \) can be calculated using Eq. (4).

\[
f_{sc} = \frac{\sum_{w} \sum_{f} (MDCT(w, f) \times FL(f))}{\sum_{w} \sum_{f} MDCT(w, f)}
\]  

(4)

3. 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 1 illustrates our iterative approach to the audio classification and segmentation problem.

![Figure 1: The flowchart of the proposed method.](image)
We apply 4-steps algorithm 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 ($T_{TFF}$) to the total energy. $T_{TFF}$ 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, $T_{TFF} = E_{min} + \lambda \times (E_{\mu} - E_{min})$, where $\lambda (0 < \lambda \leq 1)$ 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 $T_{TFF}$, then it is classified as silent, otherwise non-silent. If the granule/frame is not classified as silent, 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.

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. 0.2sec) 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 are then considered as non-silent segments. Once all non-silent segments are formed, then the classification of these segments is performed using the following features:

- **Dominant BER Classification:** In Step 1 all granules/frames are already classified with respect to BER. In this step, for each non-silent segment, the dominant classifier type (the largest number of granule/frame type) will determine the segment type.
- **Pause Rate Classification:** Pause Rate (PR) is the ratio between the number of silent granules/frames to total number of granules/frames in a non-silent segment. If this ratio is over a threshold ($T_{PR}$), then the segment is classified as a speech segment, otherwise music.

3) **Segment Merging and Global Classification:** After Step 2 is performed, some of the silent segments might still be quite small and negligible for the sake of segmentation. Such small silent segments reduce the duration of the non-silent segments and thus they lead to erroneous calculations for some features such as Pause Rate or Subband Centroid. Therefore, they need to be eliminated to yield a
global segmentation that would indeed result in a better classification. There are two conditions in order to eliminate a silent segment:

i) \( \text{Silent} \) segment duration is below a threshold value,

ii) Neighbor non-silent segment types are matching.

After merging some of non-silent segments, the overall segmentation scheme is changed and the features have to be re-extracted over the new (emerged) non-silent segments. For all the non-silent segments, \( PR \) and Dominant BER are re-calculated and all non-silent segments are re-classified. This new classification of non-silent 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 needed to eliminate all possible small silent segments. The iteration is carried out till all small silent segments are eliminated and non-silent segments are merged to global segments that have a unique classification type.

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 segments. The first part in this step tests if the non-silent segment duration is significantly larger than a given threshold (i.e. 4sec). 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.

4. Experimental 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 1-5 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 number of audio channels on the performance. The following threshold values are used:

\[ T_{\text{SNR}} = 2, \quad T_{\text{FR}} = 350 \ \text{Hz}, \quad T_{\text{M}} = 2.75 \ \text{Hz}, \quad T_{\text{R}} = 8 \%, \quad \lambda = 15 \%, \quad f_c = 500 \ \text{Hz}, \quad N_s = 20 \]

In total measures, the method is applied onto 224 (~38 hours) MP3 and 185 (~16 hours) AAC clips. Table 2 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.

<table>
<thead>
<tr>
<th></th>
<th>Only Speech</th>
<th>Only Music</th>
<th>Speech and Music</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MP3 Segmentation</strong></td>
<td>94%</td>
<td>98.5%</td>
<td>92.4%</td>
</tr>
<tr>
<td><strong>MP3 Classification</strong></td>
<td>91.3%</td>
<td>88.1%</td>
<td>97%</td>
</tr>
<tr>
<td><strong>AAC Segmentation</strong></td>
<td>96.2%</td>
<td>96%</td>
<td>91%</td>
</tr>
<tr>
<td><strong>AAC Classification</strong></td>
<td>98.4%</td>
<td>91%</td>
<td>90.2%</td>
</tr>
</tbody>
</table>

The proposed algorithm uses directly the information from the encoded bitstream without processing full decoding. A generic 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.

**References**