Score extraction using MPEG-4 T/F partial encoding

Íñigo Barrera, Francesc Tarrés

Abstract-- **This paper describes the preliminary work in the development of a MPEG-4 audio transcoder between the Time/Frequency (T/F) and the Structured Audio (SA) formats. Our approach consists in not going from T/F format through to waveform data and back again to SA, but extracting the score information from an intermediate stage. For this intermediate form we have chosen the input of the Filterbank and Block Switching Tool, which consists on frequency data. This data is the result of windowing and applying the modified discrete cosine transform (MDCT) to the signal. The size of the window to be used is determined in a frame-by-frame basis by a psychoacoustics analysis of the data.**

In this paper we show that this approach is feasible by developing a system which extracts the score information from the Filterbank and Block Switching Tool output in a MPEG-4 T/F encoder by adapting and fine-tuning some existing processing techniques.

Index Terms—**score extraction, MPEG-4 audio**

I. Introduction

Many works exist which describe the process of obtaining a musical score from the raw waveform data. The most popular current trends use either patterns recognition techniques [1, 2], some kind of signal modeling and subsequent parameter estimation [3, 4, 5], or frequency-domain analysis [6, 7, 8].

Our work is a new and different approach. By developing a MPEG-4 audio transcoder between the T/F and SA formats, we will extract the score information from a coded signal, e.g. AAC, and re-code it back into a more abstract format, e.g. Structured Audio (SA). In order for this method to be efficient, we shouldn't go through to . waveform data and back again to SA (this would allow us to directly use any of the techniques mentioned above), but we should extract the score information from an intermediate form. This intermediate form will be the input of the Filterbank and Block Switching tool [9], which consists on frequency-domain data. This selection has several advantages:

- we can use a standard decoder to obtain this data in a very efficient and deterministic way [9]
- the knowledge and techniques developed in the frequency-domain analysis can be applied
- the block switching part can be used to our advantage to even improve the results of current methods

The aim of this paper is to show that this approach is feasible by developing a system which is able to extract the score information from the spectrum data, which will be fed with the output of the Filterbank and Block Switching Tool of a MPEG-4 T/F encoder. Thus, the system will perform the score extraction from some waveform data, using a partial MPEG-4 T/F encoding.

II. System Description

As shown in Figure 1, the system has five parts:

- psychoacoustic analysis
- MDCT and windowing (Filterbank)
- peak detection
- note extraction
- post-processing

The basic working is as follows: the system receives periodically a block of 1024 new input samples. This block is put together with the previous block to form a 2048-sample block which is fed into both the Psychoacoustic analysis

The authors are researchers at Teoría de la Senyal y Comunicaciones Department. Escola Universitaria Politècnica del Baix Llobregat (EUPBL). Universitat Politècnica de Catalunya (UPC)
C/ Generalitat s/n. Sant Just Desvern. 08960. Spain.

0-7803-6290-X/00/$10.00 ©2000 IEEE

part and (through a delay) the Filterbank part. The Psychoacoustic part output determines the size of the frames which will be coded in the Filterbank. For each 2048-sample block we can have one 1024-sample (long) spectrum or eight 128-sample (short) spectrums.

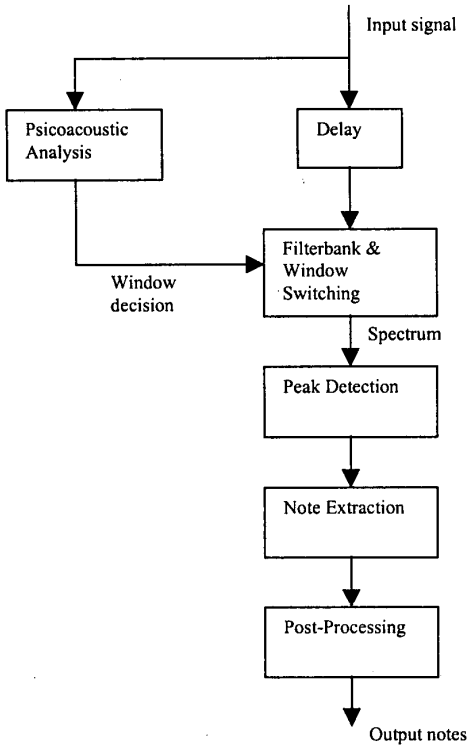

Figure 1. The system's block diagram

Every frame will go from the Filterbank to the Peak Detection part, which will discern significant peaks in the spectrum, then to the Note Extraction part which will first extract the fundamental frequencies in each frame, and finally will map the output to the SA note space. When all the frames are processed, we will apply high level post-processing to get the connection between the frames and extract the notes.

We will now go through all the parts for the details.

A. Psychoacoustic Analysis

The only reason for implementing this part in our system is the Block-Switching tool. This tool relies on the result of the psychoacoustic analysis to decide which type of window (size and shape) will be applied to the current block.
In order to do this, we will calculate the Perceptual Entropy (PE) of the current block:

$$PE = \sum_b \Delta W \cdot \log\left(\frac{E(b)}{T(b)}\right)$$

Where b is the partition, ΔW is the wide-band of the partition, E(b) the energy and T(b) the energy psychoacoustic threshold [9].
If the PE is bigger than a given threshold, the entropy is high, so the signal is changing from a steady state to another one, and we will use short frames to follow the changes with a better time resolution. If it's lower than the threshold, the signal is stable, so we will use long frames, with the best frequency resolution.
The origin of this block switching is to avoid pre-echoes, but it's very convenient for us because usually these switchings happen in the transitions between notes (see Fig. 2). That means we will have a good frequency resolution in the middle of the notes, and very good time resolution at the edges, so we will have a very good estimation of the attacks and decays of the notes.

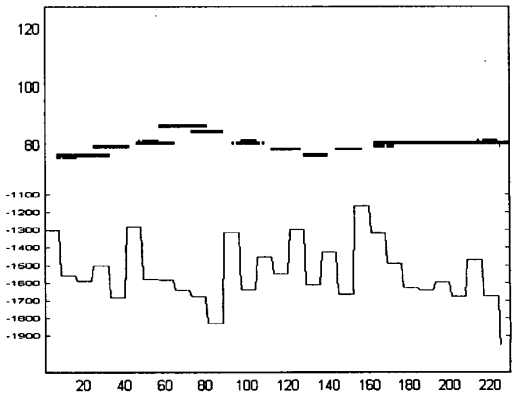

Figure 2. PE evolution with the notes

B. Filterbank

1) Block-switching and Windowing

Although the frame length is already given by the psychoacoustic analysis, it's very important to keep a smooth transition between frames. This is done here, by using a proper window shape in the way defined in the MPEG-4 T/F standard [9].

Just notice that in order to determine the transition shape, we need to know the type of the current frame, the previous frame and the *future* frame. This means that the psychoacoustic analysis will have to go one frame in advance of the filterbank. This is the reason for the delay block in the block diagram.

2) MDCT

After windowing, the MDCT is applied. For short blocks we'll have 128-sample spectrums, and for long blocks we have 1024-sample spectrums. It's easy to see now what the time and frequency resolutions are for each sampling frequency. In Table 1 we have them for Fs = 44100 Hz (CD quality).

(Fs = 44'1 KHz)	Freq. Res. (Hz)	Time res. (msecs.)
Long Frames	43.04	23.22
Short Frames	344.53	2.90

Table 1. Time and frequency resolutions

This compromise has always been a problem in the frequency-domain score extraction techniques. Sometimes it's been solved through high-level processing [7], or by using several resolutions and interpolating [6], but we have here an adaptive technique which selects dynamically which is the best resolution for each frame. This leads to an optimal-effort algorithm which could be very interesting for a possible real-time application.

C. Peak detection

The next step is to find the energy peaks in the frame. We work directly with the absolute value of the MDCT, since we are not concerned with the value itself of the energy, but just on the peaks.

After getting all the peaks, we discard the most important spurious by using inter-peak masking. This is accomplished by rejecting all the peaks which are lower than the masking of their adjacent counterparts.

D. Note extraction

1) Fundamental frequencies extraction

Once peak estimation has been obtained, the fundamental frequencies should be descerned from the partials and the noise. This is done by taking into account both the peak level and the existence of peaks in multiples of some arbitrary frequency. Perhaps the most interesting part of the algorithm is the use of a window of dynamic size for searching for the partials, to compensate for the low frequency resolution.

2) Note space mapping

The final step in the frame processing is the mapping into the SA note space. In SA (same as MIDI), the notes are numbered from 0 to 127, corresponding note 0 to C0 (8.17 Hz), and note 127 to G10 (12543.89 Hz). This mapping is not symetric, being the application function like this:

$$N = 12\log_2\left(\frac{f}{440.0}\right) + 69$$

This means that usually for low frequencies, a frequency line corresponds to several notes, and for high frequencies, several frequency lines map into the same note. The first case poses a problem qhich will be addressed in future papers by means of high-level processing (e.g. harmony or music theory rules).

E. Post-processing

Once the processing for each frame has been carried out, we start the inter-frame processing. We can clean-up the map easily by applying some standard morphologic techniques, to get, so to say, "straight lines". The next step is to extract the notes segments and associate an energy to each segment, for finally rejecting the segments with too low energy or too short length values. We can see the outputs of the system for a sample signal (short and quick guitar melody) before the post-processing (Fig. 3), after the morphologic processing (Fig. 4) and the final result (Fig. 5). The coding into SA (or MIDI) format is immediate and not considered in this paper.

III. RESULTS

As we can see in Fig. 5, the results are excellent, and truly excellent if we think of how simple the post-processing is. In Fig. 6 we can see the result for a full octave, which is also very good.

Simulations show that this approach is feasible, and the results are good enough to go on for

further research. We could even think of using the system described here directly as a score extractor from a waveform signal, due to the advantages that represent the block switching part.

There are some points, though, which should be studied further: The PE threshold is a critical value. The quality of the original encoder of the input signal could affect the quality of the result of our transcoder; and the bad frequency resolution for low notes must be compensated with high-level processing

IV. REFERENCES

[1] SIEGER, TEWFIK: "Audio coding for conversion to MIDI" IEEE Work. Mult. SP 97

[2] KASHINO, MURASE: "Music recognition using note transition context" IEEE 98

[3] CHOI: "Real-Time Fundamental Frequency Estimation by Least-Square Fitting" IEEE Trans. S& A Proc. Mar97

[4] DING, QIAN: "Estimating sinusoidal parameters of musical tones based on global waveform fitting" IEEE Work. Mult. SP 97

[5] CANO: "Fundamental Frequency Estimation in the SMS - Analysis" DAFX 98

[6] DAVIES, ETTER; "An Adaptative Technique for Automated Recognition of Musical Tones" IEEE 97

[7] FERNANDEZ-CID, CASAJUS-QUIROS: "Multi-pitch estimation for polyphonic musical signals" IEEE ICASSP 98

[8] MODEGI, IISAKU: "Proposals for MIDI Coding and its Application for Audio Authoring" IEEE Int. Conf. on Multimedia Computing & Systems 98

[9] ISO/IEC JTC1/SC29/WG11: "ISO/IEC CD 14496-3 Subpart 4 (MPEG-4/Audio/Time-Frequency Coding)" N2203TF

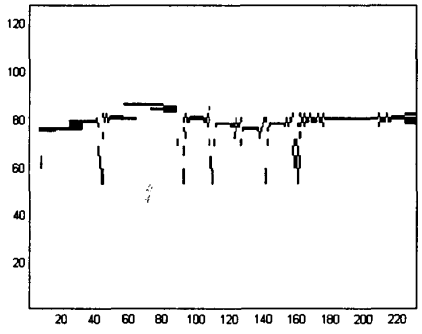

Figure 3. Result before post-processing

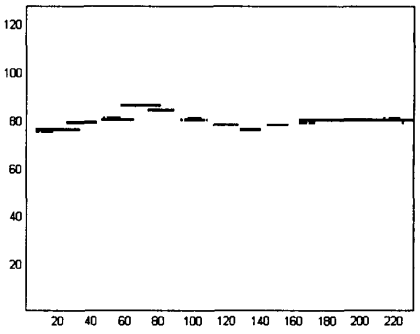

Figure 4. Result after morphological processing

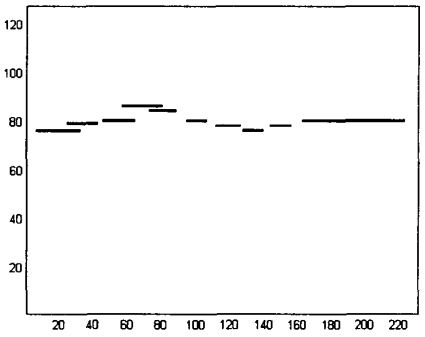

Figure 5. Final result

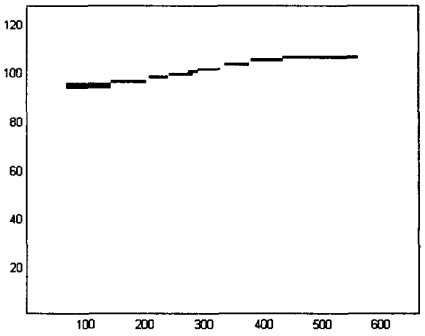

Figure 6. Result for an octave