LOW COMPLEXITY, LOW DELAY AND SCALABLE AUDIO CODING SCHEME BASED ON A NOVEL STATISTICAL PERCEPTUAL QUANTIZATION PROCEDURE

César Alonso Abad
Music Technology Group, Pompeu Fabra University, Barcelona, Spain
calonso@iua.upf.edu

Miguel Ángel Martín Fernández, Carlos Alberola López
Image Processing Laboratory, University of Valladolid, Valladolid, Spain
migmar@tel.uva.es, carlos@tel.uva.es

Keywords: fast perceptual quantization, perceptual audio coding, Huffman, histogram, scalability, low delay

Abstract: In this paper we present Fast Perceptual Quantization (FPQ), a novel procedure to quantize and code audio signals. It employs the same psychoacoustics principles used in the popular MPEG/Audio coders, but substantially simplifies the complexity and computational needs of the encoding process. FPQ is based on defining a hierarchy of privileged quantization values so that the masking threshold calculated through a psychoacoustic model is leveraged to quantize the real values to the privileged ones when possible. The computational cost of this process is very low compared to MP3's or AAC's quantization/coding loops. Experimental results show that it is possible to achieve nearly transparent coding using as few as approximately 100 quantization values. This leads to very efficient bit compaction using Huffman or arithmetic coding so that nearly state-of-the-art performance can be achieved in terms of quality/bit-rate trade-off. Since quantization and codification (bit compaction) procedures are completely independent here, efficient scalable decoding can be achieved either by parsing and entropy re-encoding the original quantized values or by coding the bit-planes independently and sorting them in order of perceptual significance. Very low delay performance is also possible to achieve, which makes the proposed coding scheme suitable for real-time applications.

1 INTRODUCTION

In this paper we introduce a perceptual quantization and coding scheme based on the MDCT and the Psychoacoustic Model 1 (ISO/MPEG, 1992) that achieves transparent coding at bit-rates comparable to those obtained using MP3 and also provides scalability and an algorithmic delay of about 5-10 ms. Our main contribution is the quantization procedure, namely FPQ (*Fast Perceptual Quantization*), which is rather simple but nevertheless leads to very interesting results in practice.

2 FAST PERCEPTUAL QUANTIZATION (FPQ)

FPQ is a novel quantization procedure that takes advantage of a tolerance margin below which quantization noise is not perceived or detected in such a way that the statistical properties of the quantized values make them especially suitable for efficient entropy coding. The idea behind most perceptual compression schemes is the following: given the modulus of an N-point spectral frame —from an FFT, MDCT, etc.— and an N point global masking threshold for that frame, we want to quantize the spectral coefficients with the largest quantization step —i.e. with the smallest number of quantization bits— that allows keeping quantization noise below the masking threshold. MPEG/Audio coders use Huffman coding iteratively to find the shortest codewords that quantize and encode the spectral values without exceeding the masking threshold for each scalefactor-band (ISO/MPEG, 1992; Bosi et al., 1997). Other recent approaches like the ones proposed in (Derrien et al., 2006; Kramer et al., 2004; Wabnik et al., 2006) take into account the statistical behavior of the quantization noise to simplify the quantization/coding iterative procedure.

The approach of FPQ is very related to these last ones, but instead of modeling the statistical properties

of the quantization noise, the idea is to take advantage of the masking threshold to *force* some interesting statistical property to occur. For example, it is very desirable for Huffman or arithmetic coding purposes to concentrate the probability of occurrence of quantization values in a small subset of "privileged" ones. In this way, we can use less bits to encode the most probable quantization values at the expense of using more bits to encode those quantization values that are rarely taken. And what is also important: coding bits are not spent in quantization values that are actually never taken.

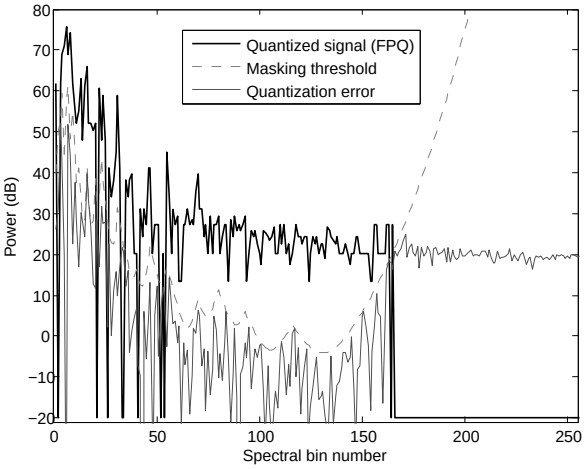

Figure 1: Short spectral frame of a music signal quantized using FPQ. Note that quantization error does not exceed the masking threshold at any point.

One simple and efficient way to achieve this concentration of probability in a small subset of quantization values is to quantize each spectral bin according to the following procedure:

1. First we try the largest possible quantization step, i.e. we try to quantize the actual value either to zero or the maximum of the dynamic range. Typically, the dynamic range would be 0-65535 (16 quantization bits).
2. We calculate the quantization error as the absolute value of the difference between the actual value and the quantized one using the actual quantization step.
3. If this error level is below the value of the masking threshold for the particular spectral bin considered, the distortion is assumed to be inaudible and the bin is quantized with the actual quantization step.
4. If not, we divide the quantization step by 2 and repeat the process from the second step.

These steps are repeated for each and every spectral value until all of them are quantized in such a way that quantization error never exceeds the masking threshold, as it is shown in Figure 1. This simple algorithm has several advantages:

- The quantization procedure has a very low computational cost and can be implemented using a few simple operations. Only one loop is necessary to quantize each value, with 15 iterations at the most (using 16 quantization bits) but about 3-4 iterations in mean.
- Entropy coding is only made once, when all the values are quantized, as opposed to MPEG/Audio quantization/coding nested loops, in which Huffman coding is performed in every iteration until any of the termination conditions of the loops are met.
- Spectral bins tend to quantize to powers of 2 or multiples of powers of 2. If we use natural binary coding, most of the values are expressed with binary words with few ones surrounded by zeros. This sparsity is interesting for compressing each bit-plane separately for scalability purposes.
- Experimental results show that it is possible to achieve transparent coding of modern music audio signals using as low as about 100 quantization values, which is a dramatic dimensionality reduction from the initial 65536 possible if we use 16 quantization bits.
- Spectral bins in every frame are quantized one by one and not at sub-band level. In other words, they are quantized to the maximum possible frequency resolution given a window length. This allows for high coding gains even when using short analysis windows, what is desirable for very low delay applications.
- As quantization is independent from the later entropy coding procedure used, we can achieve scalable decoding easily. For example by compressing the bit-planes separately, taking advantage of the aforementioned sparsity of the values quantized with FPQ expressed in natural binary form, and selecting them in order of significance.
- Fixed bit-rate streaming can be also achieved by selecting the proper number of bit-planes for each temporal frame.

3 ENCODING SCHEME

To test the properties of FPQ embedded in a complete coding scheme, we developed a prototype of a simple perceptual audio compression system using MATLAB. The scheme is shown in Figure 2. An input PCM signal sampled at 44100 Hz and quantized with 16 bits is divided into N-point 50% overlapped frames to calculate the MDCT. N should not be very large to avoid pre-echoes. In our experiments each

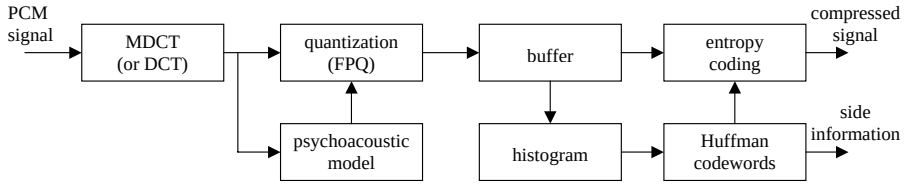

Figure 2: Proposed scheme for a simple encoder that makes use of FPQ and Huffman entropy coding.

spectral frame had 256 points ($N = 512$). For each spectral frame, the masking threshold or JND level is calculated, as shown in Figure 1. This is done through Psychoacoustic Model 1 but using the modulus of the MDCT coefficients instead of making a separate FFT analysis. The modulus of every single bin is quantized independently using FPQ, according to the value of the masking threshold for this particular bin. The signs are compressed separately using a lossless scheme.

Once every bin on every frame is quantized with FPQ, the resulting data are very highly compressible by almost any entropy coder. We tried Huffman and arithmetic coding, PNG (*Deflate*), RAR and ZIP, achieving in all cases compression ratios between 1:7 and 1:10 for nearly transparent coding of mono signals. In other words: this simple scheme is comparable to MP3 compression in terms of quality/bitrate trade off. The decoding procedure is also very simple: the entropy-coded spectral coefficients are uncompressed and the inverse MDCT is performed. Note that the Huffman codewords are the only side information needed for decoding.

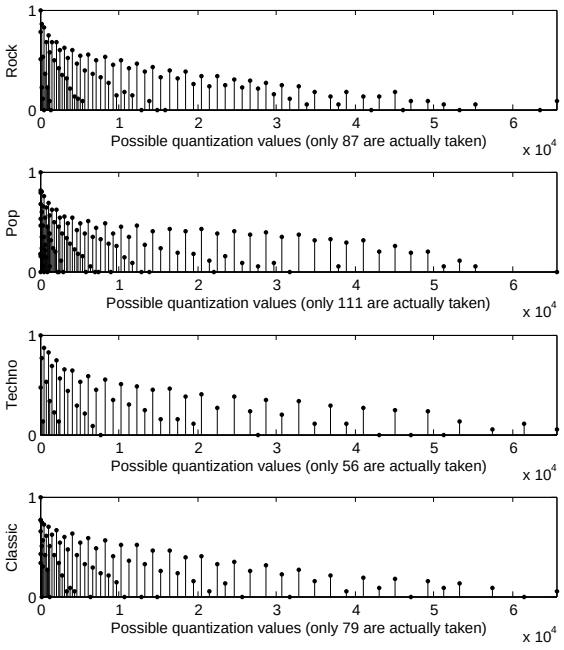

Figure 3: Histogram of the spectral quantization values for nearly transparent coding using FPQ for four 8-second musical excerpts named *Rock*, *Pop*, *Techno*, and *Classic* according to their music style. Values are quantized using 16 bits (the dynamic range is therefore 0-65535). Vertical axis is in logarithmic scale and normalized to 1 for better visualization.

4 EXPERIMENTAL RESULTS

4.1 Non-real-time performance

When low delay is not important, we can use the buffer in Figure 2 to calculate the histogram and therefore the optimum codewords for entropy coding the signal considered. As it has been said, FPQ quantizes the modulus of spectral audio coefficients in such a way that a few quantization values —say about 100— are enough to cover the whole dynamic range without introducing audible distortion. It has been found experimentally that the probability distribution of the quantized spectral values for all the music signals we used for testing is rather similar, as shown in Figure 3. In all cases approximately the same few values (most of them powers of two and multiples of powers of two) are largely the most probable.

To measure the compressibility of the coefficients quantized using FPQ and the performance of the entire prototype of the encoder here proposed, we used several 8-second music excerpts, representative of various music styles. First we compressed them using MP3 and AAC coders at the less possible bitrate which produced nearly transparent coding, that is, distortion which is perhaps slightly perceptible, but not annoying. Then we used the MATLAB prototype based on FPQ described in section 3 to obtain the corresponding quantized MDCT coefficients for each signal. The global masking threshold level used for quantization was adjusted in order to obtain decoded audio files with a similar quality as those coded with MP3 and AAC in the previous step. Fi-

nally we compressed the resulting quantized spectral coefficients using Huffman, arithmetic, RAR and ZIP entropy coders. The resulting file lengths were very close to those obtained with MP3 coding in all cases. AAC compression was significantly better for similar audio quality.

We used also another method to entropy code the quantized values for scalability purposes. Taking advantage of the sparse structure that quantized values using FPQ present, we compressed each bit-plane using PNG for binary images. The resulting file lengths obtained were between 5% and 23% bigger than those obtained with MP3. Nevertheless, the files thus compressed have the advantage of being able to generate scalable bit-streams without any additional computational cost. It is possible to achieve fine-grain scalability by further subdividing the bit-planes into smaller binary images. Experiments show that graceful degradation is obtained by successively removing the less significant bit-planes.

4.2 Real-time performance

When low-delay is required each frame must be coded and sent immediately. Using 512-point temporal windows for the MDCT, the minimum algorithmic delay is 11.6 ms for a sampling rate of 44100 Hz. Using DCT with non-overlapped 256-point windows, delay can be reduced to 5.8 ms. As long as FPQ and Huffman coding are very low costing operations, only the psychoacoustic model would increase these delay values significantly. However, as FPQ uses the masking threshold to the maximum possible frequency resolution for a given a window length, it is expected that even shorter windows could be used while maintaining acceptable coding gains.

The price we have to pay for low-delay performance is that the Huffman codewords must be actualized and sent to the decoder each and every time new quantization values appear in the incoming frames. After an initial start-up stage in which Huffman codewords must be sent for every frame, the histogram becomes approximately stable. In this steady-state scenario, the same codewords are nearly optimal for the frames transmitted thereafter. If new values appear, a new Huffman table including these values must be generated. Fortunately, this happens only occasionally after the first frames are coded and sent, and the overload penalty is only an increase of 10-15% in the amount of information that must be transmitted. Another possibility is to predefine a fixed "standard" Huffman table for all the cases, taking into account that histograms of very different signals are actually not so different, as shown in Figure 3: approximately the same values (powers of 2 and multiples of powers of 2) are the most probable in all cases.

5 CONCLUSIONS

In this paper, a novel perceptual quantization method for audio coding called FPQ has been proposed. This method is an alternative to those used in the most important perceptual audio coders and has several advantages with respect to them: it is much simpler and computationally inexpensive; it can produce scalable bit-streams with no additional computational cost and it is suitable for real-time applications.

For the purpose of measuring the FPQ capabilities, a simple audio compression prototype was built in MATLAB. The results of the experiments showed that the quantized spectral values given by FPQ are very well suited for compression using almost any entropy coding system. Compression rates comparable to those obtained using MP3 coding for the same audio quality were achieved. Very low delay and scalability is also achievable at the expense of slightly higher bit-rates. It is expected that the compression capacity of the system will be significantly improved using more refined psychoacoustic models and entropy coders, while keeping the advantages of being simple, computationally inexpensive and scalable.

REFERENCES

Bosi, M., Brandenburg, K., Quackenbush, S., Fielder, L., Akagiri, K., Fuchs, H., Dietz, M., Herre, J., Davidson, G., and Oikawa, Y. (1997). ISO/IEC MPEG-2 advanced audio coding. *J. Audio Eng. Soc.*, 45(10):789–814.

Derrien, O., Duhamel, P., Charbit, M., and Richard, G. (2006). A New Quantization Optimization Algorithm for the MPEG Advanced Audio Coder Using a Statistical Subband Model of the Quantization Noise. *IEEE Transactions on Audio, Speech and Language Processing*, 14(4):1328–1339.

ISO/MPEG (1992). Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s—Part 3: Audio. *IS11172-3 (MPEG-1)*.

Kramer, U., Schuller, G., Wabnik, S., Klier, J., and Hirschfeld, J. (2004). Ultra Low Delay audio coding with constant bit rate. *117th AES Convention*.

Wabnik, S., Schuller, G., Hirschfeld, J., and Kraemer, U. (2006). Different quantisation noise shaping methods for predictive audio coding. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse*.