IDIAP RESEARCH REPORT

ENTROPY CODING OF QUANTIZED SPECTRAL COMPONENTS IN FDLP AUDIO CODEC

Petr Motlicek Sriram Ganapathy
Hynek Hermansky

Idiap-RR-71-2008

NOVEMBER 2008

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

ENTROPY CODING OF QUANTIZED SPECTRAL COMPONENTS IN FDLP AUDIO CODEC

Petr Motlicek[1], *Sriram Ganapathy*[1,2], *Hynek Hermansky*[1,2]

[1]Idiap Research Institute, Martigny, Switzerland
[2]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{motlicek,ganapathy}@idiap.ch, hynek@ieee.org

ABSTRACT

Audio codec based on Frequency Domain Linear Prediction (FDLP) exploits auto-regressive modeling to approximate instantaneous energy in critical frequency sub-bands of relatively long input segments. Current version of the FDLP codec operating at 66 kbps has shown to provide comparable subjective listening quality results to the state-of-the-art codecs on similar bit-rates even without employing strategic blocks, such as entropy coding or simultaneous masking. This paper describes an experimental work to increase compression efficiency of the FDLP codec provided by employing entropy coding. Unlike traditionally used Huffman coding in current audio coding systems, we describe an efficient way to exploit Arithmetic coding to entropy compress quantized magnitude spectral components of the sub-band FDLP residuals. Such approach outperforms Huffman coding algorithm and provides more than 3 kbps bit-rate reduction.

Index Terms— Audio Coding, Frequency Domain Linear Prediction (FDLP), Entropy Coding, Arithmetic Coding, Huffman Coding

1. INTRODUCTION

Traditionally, two-step process is carried out to perform source coding of analog audio/visual input signals. First, a lossy transformation of the analog input data into set of discrete symbols is performed. Second, lossless compression, often referred to as noiseless/entropy coding, is employed to further improve compression efficiencies. In many current audio/video codecs, such distinction does not exist or only one step is applied [1].

Traditionally, lossless coding is carried out by Huffman coding (e.g. [2]). Either the source symbols are compressed individually, or they are grouped to create symbol strings which are then processed by vector based entropy coder. Since the entropy of the combined symbols is never higher than the entropy of the elementary symbols (usually it is significantly lower), high compression ratios can be achieved [3]. However, a considerable lookahead is required. Therefore, vector based entropy coding is usually exploited for high quality audio coding where an algorithmic delay is available. Moreover, symbol grouping increases complexity which often grows exponentially with the vector size. Real-time coders therefore use per-symbol entropy coding for speed, simplicity, low delay and efficiency.

Recently, a new speech/audio coding technique based on approximating temporal evolution of the spectral dynamics was proposed [4, 5]. More particularly, this technique performs decomposition into Amplitude Modulation (AM) and Frequency Modulation (FM) components. Obtained AM/FM components are referred to as Hilbert envelope and Hilbert carrier estimates, respectively. The compression strategy is based on predictability of slowly varying amplitude modulations to encode audio/speech signals. On the encoder side, an input signal is split into frequency sub-bands roughly following critical sub-band decomposition provided by non-uniform Quadrature Mirror Filter (QMF) bank. In each sub-band, Hilbert envelope is estimated using Frequency Domain Linear Prediction (FDLP), which is an efficient technique for Auto-Regressive (AR) modeling of temporal envelopes of a signal [6]. Sub-band FDLP residuals are processed using Discrete Fourier Transform (DFT). Magnitude and phase spectral components are vector and scalar quantized, respectively. The process of quantization is controlled by perceptual model simulating temporal masking. The decoder inverts the steps from encoder to reconstruct the signal back.

This paper describes the flexible Arithmetic coding algorithm used in the FDLP audio codec to encode selected codebook indices obtained using Vector Quantization (VQ). VQ is employed to quantize magnitude spectral components of the sub-band FDLP residuals. More particularly, sufficiently low quantization noise as well as acceptable computational load is achieved by split VQ [7]. On the other hand, the distribution of phase spectral components was found to be close to uniform. Their correlation across time is minor. Therefore a uniform Scalar Quantization (SQ) is performed without applying additional entropy coding. Since Arithmetic coding has advantageous properties for small alphabets [8], VQ codebooks are first pruned down (without the significant increase of quantization noise). Created input sequences provided by successive VQ indices are then split into two sub-streams (with reduced alphabets) which are then independently entropy compressed. Finally, achieved compression efficiencies of Arithmetic coder are compared with traditional Huffman coding algorithm on challenging audio/speech data.

This paper is organized as follows. Section 2 describes the basic structure of the FDLP audio codec operating at medium bit-rates. In Section 3, Arithmetic coding algorithm is briefly described with concentration on the FDLP audio compression needs. Here, we also mention an experimental setup proposed for the entropy coding experiments. Experimental results are given in Section 4, followed by discussions and conclusions.

2. STRUCTURE OF THE FDLP CODEC

FDLP codec is based on processing long (hundreds of ms) temporal segments. As described in [5], the full-band input signal is decomposed into non-uniform frequency sub-bands. In each sub-band, FDLP is applied and Line Spectral Frequencies (LSFs) approxi-

This work was partially supported by grants from ICSI Berkeley, USA and the Swiss National Center of Competence in Research (NCCR) on "Interactive Multi-modal Information Management (IM)2"; managed by the IDIAP Research Institute on behalf of the Swiss Federal Authorities.

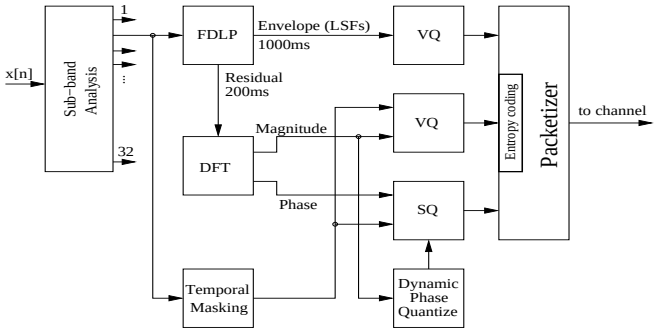

Fig. 1. *Scheme of the FDLP encoder with block of entropy coding.*

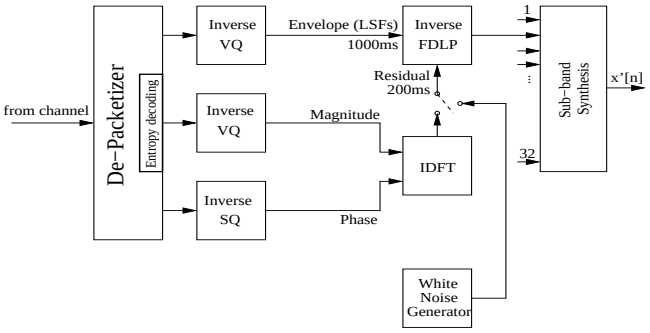

Fig. 2. *Scheme of the FDLP decoder with block of entropy decoding.*

mating the sub-band temporal envelopes are quantized. The residuals (sub-band carriers) are obtained by filtering sub-band signals through corresponding AR model reconstructed from the quantized LSF parameters (quantization noise is introduced using analysis-by-synthesis approach). Then these sub-band residuals are segmented into 200ms long segments and processed in DFT domain. Magnitude and phase spectral components are quantized using VQ and SQ, respectively. Graphical scheme of the FDLP encoder is given in Fig. 1.

In the decoder, shown in Fig. 2, quantized spectral components of the sub-band carriers are reconstructed and transformed into time-domain using inverse DFT. The reconstructed FDLP envelopes (from LSF parameters) are used to modulate the corresponding sub-band carriers. Finally, sub-band synthesis is applied to reconstruct the full-band signal. The final version of the FDLP codec operates at 66 kbps.

Among important blocks of the FDLP codec belong:

- *Non-uniform QMF decomposition*: A perfect reconstruction filter-bank is used to decompose a full-band signal into 32 (roughly critically band-sized) frequency sub-bands.
- *Temporal masking*: First order forward masking model of the human hear is implemented. This model is employed in encoding the sub-band FDLP residuals.
- *Dynamic Phase Quantization (DPQ)*: DQP enables non-uniform scalar quantization of spectral phase components to reduce their bit-rate consumption.
- *Noise substitution*: FDLP filters in frequency sub-bands above 12 kHz (last 3 sub-bands) are excited by white noise in the decoder. This has shown to have a minimum impact on the quality of reconstructed signal.

2.1. Quantization of spectral magnitudes in the FDLP codec

Spectral magnitudes together with corresponding phases represent 200 ms long segments of the sub-band FDLP residuals. At the encoder side, spectral magnitudes are quantized using VQ (corresponding codebooks generated using LBG algorithm).

VQ is well known technique which provides the best quantization scheme for a given bit-rate. However, a full-search VQ exponentially increases computational and memory requirements of vector quantizers with the bit-rate. Moreover, usually large amount of training data is required. Therefore, a sub-optimal (split) VQ is employed in the FDLP codec. Each vector of spectral magnitudes is split into a number of sub-vectors and these sub-vectors are quantized separately (using separate VQ). Due to unequal width of frequency sub-bands introduced by non-uniform QMF decomposition, vector lengths of spectral magnitudes differ in each sub-band. Therefore, number of splits differs, as well. In addition, more precise VQ (more splits) is performed in lower frequency sub-bands where the quantization noise has shown to be more perceptible than in higher sub-bands.

Finally, codebook pruning is performed in lower frequency sub-bands (bands $1-26$) in order to reduce their size and to speed up VQ search. Objective quality listening tests proved that 25% codebook reduction (i.e. the least used centroids are removed based on statistical distribution estimated on training data) has a minimum impact on resulting quality.

3. ARITHMETIC CODING

Arithmetic Coding (AC) has been selected to perform additional (lossless) compression applied in the FDLP audio codec. Main advantage of AC is that it can operate with symbols (to be encoded) by a fractional number of bits [9], as opposed to well-known Huffman coding. In general, AC can be proven to reach the best compression ratio possible introduced by the entropy of the data being encoded. AC is superior to the Huffman method and its performance is optimal without the need for grouping of input data. AC is also simpler to implement since it does not require to build a tree structure. Simple probability distribution of input symbols needs to be stored at encoder and decoder sides, which possibly allows for dynamic modifications based on input data to increase compression efficiency.

AC processes the whole sequence of input symbols in one time by encoding symbols using fragments of bits. In other words, AC represents an input sequence by an interval of real numbers between 0 and 1. As a sequence becomes longer, the interval needed to represent this sequence becomes smaller. Therefore, the number of bits to specify given interval grows.

Nowadays, AC is being used in many applications especially with small alphabets (or with an unevenly distributed probabilities) such as compression standards G3 and G4 used for fax transmission. In these cases, AC is maximally efficient compared to well-known Huffman coding algorithm. It can be shown that Huffman coding never overcomes a compression ratio of $(0.086 + P_{max})H_M(S)$ for an arbitrary input sequence S with P_{max} being the largest of all occurring symbol probabilities [10]. $H_M(S)$ denotes the entropy of the sequence S for a model M. It is obvious that for large alphabets, where P_{max} reaches relatively small values, Huffman algorithm achieves better compression efficiencies. Therefore, this gives

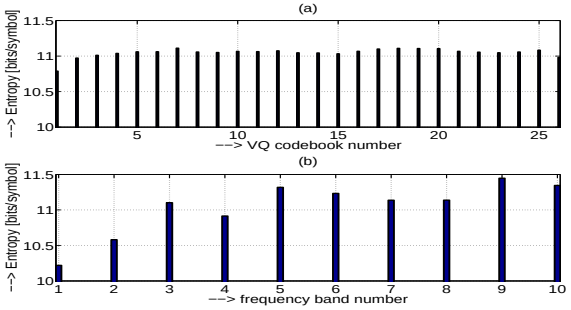

Fig. 3. *Mean entropy of VQ indices of the first 10 sub-bands estimated: (a) for each VQ codebook (codebook dependent sequences), (b) for each sub-band (sub-band dependent sequences).*

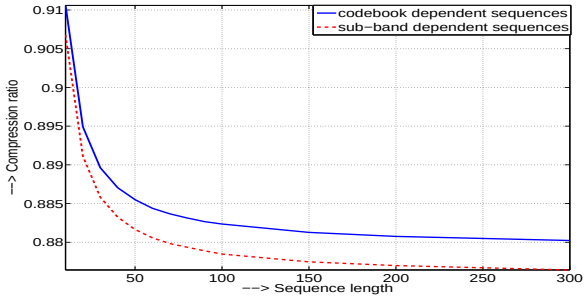

Fig. 4. *Compression ratio of Arithmetic coder for different lengths of input sequences. Input sequences are generated: (a) for each codebook (codebook dependent sequences), (b) for each sub-band (sub-band dependent sequences).*

a good justification for such a technique on large alphabets. However, for small alphabet applications, which lead to bigger symbol occurrence probabilities, AC is more efficient.

3.1. Experimental data

Entropy coding experiments are performed on audio/speech data sampled at 48 kHz. In all experiments, fixed model based entropy coding algorithms are used. Unlike Huffman algorithm which requires to generate a tree structure to be shared by the encoder and the decoder, AC requires only probabilities of input symbols to be estimated from training data.

In our experiments, training data consists of 47 audio recordings (19.5 minutes), mainly downloaded from several internet databases. The content is distributed among speech, music and radio recordings. Test data consists of 28 recordings (7.25 minutes) with mixed signal content from MPEG database for "explorations in speech and audio coding" [11].

3.2. Experimental setup

Entropy coding is applied on spectral magnitudes of the sub-band FDLP residuals in all 32 sub-bands. Size of VQ codebooks employed in the FDLP codec differs for lower and higher frequency bands. Codebooks in bands 1-26 and 27-32 contain 3096 and 512 centroids, respectively. This corresponds to 11.5962 bits/symbol and 9 bits/symbol, respectively.

Several experiments are conducted to optimize performances of AC. In order to reduce time complexity of these experiments, VQ indices (symbols) only from the first 10 sub-bands ($0 \sim 4$ kHz) are used to form the input sequences for AC. Sub-bands $1-10$ utilize 26 (band independent) VQ codebooks to quantize magnitude spectral components. Since AC operates over sequences of symbols, it matters how these symbol sequences are generated. We experiment with two ways:

- Input sequences comprise symbols generated by the same VQ codebook (codebook dependent sequences): Fixed probability model *for each VQ codebook* is estimated from training data. Mean entropy estimated from training data is shown in Fig. 3 (a). Different lengths of input test sequences are created from test data to be then encoded by AC. Achieved compression ratios for different test sequence lengths are shown in Fig. 4.
- Input sequences comprise symbols belonging to the same sub-band (sub-band dependent sequences): Fixed probability model *for each sub-band* is generated from training data. Mean entropy estimated from training data is shown in Fig. 3 (b). Achieved compression ratios for different test sequence lengths created from test data are given in Fig. 4.

Compression ratios given in Fig. 4 clearly show that AC is more efficient in the second mode, i.e. when applied independently in each frequency sub-band. This means that applied entropy coding can better exploit similarities in input data distribution generated by identical frequency sub-band.

With respect to the theoretical insights of AC mentioned in Sec. 3, we further perform alphabet reduction. It is achieved by simple splitting each input sequence comprising 12-bit symbols into two independent 6-bit symbol sub-sequences. Training data is used to estimate two independent probability models from 6-bit symbol distributions. During encoding, each input test sequence of 12-bit symbols is split into two 6-bit symbol sub-sequences which are then encoded independently by two ACs employing two different probability models. Finally, obtained compressed bit-streams are merged to create one bit-stream to be transmitted over the channel. Achieved compression ratios (for the first 10 sub-bands) are given in Fig. 5. This figure compares performances for the case when AC employs the reduced and the full alphabet. As can be seen, proposed alphabet reduction provided by splitting of 12-bit symbol sequences into two 6-bit symbol sub-sequences significantly increases compression efficiency.

4. EXPERIMENTAL RESULTS

In previous section, we described the experimental procedure to exploit AC in the FDLP audio codec. In order to reduce computational complexities and to be able to quickly summarize achieved results, the experiments were performed with data (VQ indices) coming from the first 10 frequency sub-bands. The best performances were obtained for the case when AC was applied independently in each frequency sub-band (regardless to VQ codebook assignment). Furthermore, reduced alphabet provided better compression efficiency in all frequency sub-bands compared to the original (full) alphabet. Next, this configuration is used to test the efficiency of AC applied to encode VQ indices from all 32 frequency sub-bands (although AC is eventually not employed in the last 3 sub-bands in the FDLP codec).

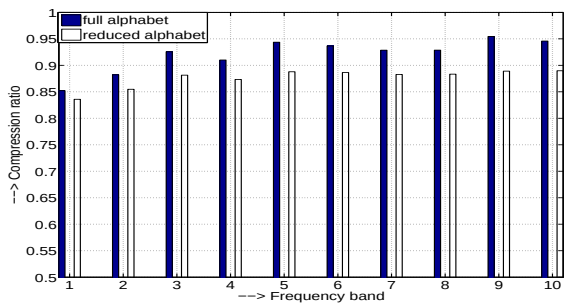

Fig. 5. *Compression ratio of Arithmetic coder operating on the full and the reduced alphabet.*

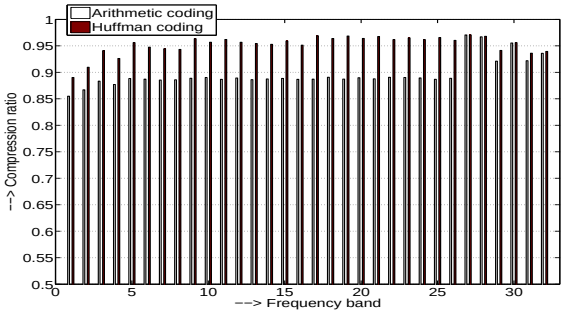

Fig. 6. *Compression ratio of Arithmetic and Huffman coding for different frequency sub-bands.*

Resulting compression ratios are given in Fig. 6. In these final experiments, test sequence lengths are chosen to equal 50 (number of successive VQ indices forming input sequences for AC).

Lastly, AC performances are compared with performances of Huffman coding, traditionally applied in the state-of-the-art audio systems. The same training data is used to generate a fixed model provided by a tree structure shared by the Huffman encoder and the decoder. Since better performances of Huffman coding are obtained for large alphabets [10], original 12-bit alphabet is used. Similarly to AC, Huffman coding is also applied independently in each frequency sub-band (Huffman tree structure is generated for each frequency sub-band). Performances of Huffman based entropy coder for different frequency sub-bands are also given in Fig. 6.

5. DISCUSSIONS AND CONCLUSIONS

In this paper the entropy coder based on Arithmetic Coding (AC) algorithm was proposed to be implemented in the FDLP audio codec initially operating at 66 kbps. Only VQ codebook indices of magnitude spectral components of the sub-band FDLP residuals from 0 to 12 kHz were entropy encoded. Overall bit-rate reduction achieved by AC is 3 kbps. This corresponds to 11% bit-rate reduction to compress VQ indices of spectral magnitudes of the sub-band FDLP residuals. Substantially larger entropy compression efficiencies cannot probably be achieved since a significant reduction is already captured by split VQ. However, VQ indices of spectral magnitudes consume only $\sim 30\%$ of the total bit-rate (compared to $\sim 60\%$ assigned to entropy uncompressed spectral phase components). Therefore, different transform to replace DFT may be of interest to avoid phase coefficients inapplicable for entropy compression. AC outperforms traditionally used Huffman coding (only ~ 1 kbps bit-rate reduction). Although AC requires at the input a sequence of symbols to be encoded, it does not increase computational delay of the whole system. The entropy decoding can start immediately with the first bits transmitted over the channel. In our work, AC did not exploit adaptive probability model, which could significantly increase performances. In this case, AC would be a powerful technique, which does not require complex changes of the structure, as opposed to Huffman coding.

Objective and subjective listening tests were performed and described in [5] to compare FDLP codec with LAME-MP3 (MPEG 1 Layer 3) [12] and MPEG-4 HE-AAC v1 [13], both operating at 64 kbps. Since AC is a lossless technique, previously achieved audio quality results are valid. In overall, the FDLP audio codec achieves similar subjective qualities as the state-of-the-art codecs on medium bit-rates. Additional improvements can potentially be obtained by employing simultaneous masking module.

6. REFERENCES

[1] P. A. Chou, T. Lookabaugh, R. M. Gray, "Entropy Constrained Vector Quantization", *in Trans. Acoust. Sp. and Sig. Processing*, 37(1), January 1989.

[2] S. R. Quackenbush, J. D. Johnston, "Noiseless coding of quantized spectral components in MPEG-2Advanced Audio Coding", *in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, October 1997.

[3] Y. Shoham, "Variable-size vector entropy coding of speech and audio", *in International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 769-772, Salt Lake City, USA, May 2001.

[4] P. Motlicek, H. Hermansky, H. Garudadri, N. Srinivasamurthy, "Speech Coding Based on Spectral Dynamics", Proceedings of TSD 2006, LNCS/LNAI series, Springer-Verlag, Berlin, pp. 471-478, September 2006.

[5] S. Ganapathy, P. Motlicek, H. Hermansky, H. Garudadri, "Autoregressive Modelling of Hilbert Envelopes for Wide-band Audio Coding", *Audio Engineering Society*, 124th Convention, Amsterdam, Netherlands, May 2008.

[6] M. Athineos, D. Ellis, "Frequency-domain linear prediction for temporal features", *Automatic Speech Recognition and Understanding Workshop IEEE ASRU*, pp. 261-266, December 2003.

[7] K. Paliwal, B. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame", *in IEEE Transactions on Speech and Audio Processing*, vol. 1, No. 1, pp. 3-14, January 1993.

[8] E. Bodden, M. Clasen, and J. Kneis, "Arithmetic Coding revealed - A guided tour from theory to praxis", *Technical report*, Sable Research Group, McGill University, No. 2007-5, 2007.

[9] J. Rissanen, G. G. Langdon, Jr., "Arithmetic Coding", *in IBM Journal of Res. & Dev.*, vol. 23, No. 2., 3/79.

[10] Khalid Sayood. "Introduction to data compression", (2nd ed.), Morgan Kaufmann Publishers Inc., 2000.

[11] ISO/IEC JTC1/SC29/WG11: "Framework for Exploration of Speech and Audio Coding", MPEG2007/N9254, Lausanne, Switzerland, July 2007.

[12] LAME MP3 codec: <*http://lame.sourceforge.net*>

[13] 3GPP TS 26.401: "Enhanced aacPlus General Audio Codec".