NEURAL NETWORK BASED ARITHMETIC CODING FOR REAL-TIME AUDIO TRANSMISSION ON THE TMS320C6000 DSP PLATFORM

E. Pasero, A. Montuori

Dipartimento di Elettronica
Politecnico di Torino
E-mail: pasero@polito.it, montuori@lep.polito.it

ABSTRACT

We developed a real-time wideband speech codec adopting a wavelet packet based methodology. The transform domain coefficients were first quantized by using a psycho-acoustic model and then encoded with an arithmetic coding. The arithmetic coding was carried out by adapting the probability model of the quantized coefficients frame by frame by means of a competitive neural network, which was trained to detect regularities in the distribution of the wavelet packet coefficients. The weight matrix of the neural network is periodically updated during the compression in order to model better the speech characteristics of the current speakers. The coding/decoding algorithm was first written in C and then optimized on the TMS320C6000 DSP platform in a QoS-compliant fashion.

1. INTRODUCTION

Several applications such as teleconferencing, multimedia services and high-quality wideband telephony require advanced coding algorithms for wideband speech. In contrast to the standard telephony band of 200 to 3400 Hz, wideband speech is assigned the band 50 to 7000 Hz and is sampled at a rate of 16000 Hz for subsequent digital processing. The added low frequencies increase the voice naturalness whereas the added high frequencies make the speech sound more intelligible.

The uneven and time varying distribution of the wideband speech energy provides motivation for using adaptive subband coding. We developed a real-time wideband speech coder/decoder adopting a wavelet packet transform based methodology [20].

The transform domain coefficients were first quantized by means of a uniform quantizer on the basis of the psycho-acoustic masking phenomenon and then encoded with an arithmetic coding. If we can provide an accurate model for probability of occurrence of each possible symbol at every point in a frame, the encoding is very nearly optimal. An accurate probability model for the arithmetic coding can be build by using an *adaptive model*, i.e. by observing during the compression the probability of occurrence of each symbol. Since the receiver needs to know the probability model used for the arithmetic coding in order to decode the coefficients, each symbol must be encoded using the distribution of the part of the file already coded [4]. In this way no side information must be sent, because the receiver can reconstruct the probability table used by the transmitter. The poor error resistance of arithmetic coding does not make this approach very efficient for voice packet transmission, which is often characterized by packet losses and transmission errors. Better results can be achieved by using a fixed probability estimated on a robust and effective database of speech data. If the transformed speech were a statistically independent sequence of symbols the probability for each symbol could be set equal to its relative frequency in the database. Unfortunately this is not a realistic assumption for the speech wavelet transform, first of all because the frequency spectrum of each phoneme has peculiar characteristics which repeat themselves each time the phoneme is pronounced.

In our approach a competitive neural network was trained on the TIMIT corpus to detect regularities in the distribution of the wavelet packet coefficients. At the end of the training we obtained a codebook of probability tables. The probability table used for the arithmetic coding is updated frame by frame (i.e. packet by packet) by selecting it in the codebook. The trained network is also used to select frame by frame the best table in the codebook for the coding of the current speech frame. Only the code of the used probability table must be sent as side information so that the receiver decodes each packet correctly.

TMS320 DSPs have proven effective for real-time compression of audio signals [3][14][16]. The codec algorithm was first written in C and then optimized on the TMS320C6000 DSP platform.

2. THE SUBBAND DECOMPOSITION

In our coder the audio signal is transformed into a time-scale representation through a non-uniform wavelet packet decomposition.

The coding process first entails obtaining a frame of 128 speech samples, which are transformed into subband signals by means of a fast wavelet packet transform algorithm. The structure of the analysis tree is chosen so that the resulting 21 subbands [12] mimic the critical bands of the human auditory system for the 0-8 kHz bandwidth, which allows to make use of the spectral masking properties of the human ear to decrease the bit-rate of the encoder while perceptually hiding the quantization error.

The choice of the prototype filter of the transform, as well as

0-7803-7663-3/03/$17.00 ©2003 IEEE

its length, influences the separation of the subband signals and the compression performance. The filters proposed by Daubechies are the ones that best preserve frequency selectivity as the number of stages of the DWPT increases. This is due to their regularity property [2]. We have obtained excellent performances with biorthogonal filters, specifically with the Filters 3 of the best biorthogonal filter banks of Villasenor [19].

3. PERCEPTUAL NOISE MASKING

The samples of the time-scale representation were quantized to reduce the amount of data sent to the transmission channel. The allocation of the bits to the subbands considered the perceptual noise masking characteristics of the human ear. The noise threshold, i.e. the maximum noise that can be inaudibly inserted into the signal was used as quantization error.

The calculus of the noise threshold involved several steps [7]. First we calculated the energy in each critical band. To estimate the effects of masking across critical bands the spreading function given in [1][15] was used. After this step the signal to masking ratio was calculated for each subband by evaluating the tonality of the signal. In order to determine the *noiselike* or *tonelike* nature of the signal, the spectral flatness measure [6], computed by means of the wavelet packet transform, was used. Finally the maximum between the masking threshold and the threshold in quiet was taken.

The frame length was set equal to 128 samples (8 ms). Since we used symmetric extension of frames and a biorthogonal filter bank, which has impulse responses perfectly symmetric (or antisymmetric), the analysis and synthesis window lengths were equal to 128 samples too and no overlap was used between frames, leading to an algorithm delay of only 8 ms. The use of symmetric extensions of the frames caused the incorrect calculation of the masking threshold, but the use of the psycho-acoustic model still proved advantageous.

The transform domain coefficients were quantized by means of a mid-tread quantizer [11] and encoded with the arithmetic coding.

4. THE COMPETITIVE NEURAL NETWORK

The neurons of competitive networks learn to recognize groups of similar input vectors.

The topology of the network we used is showed in Fig. 1. In our competitive network the distance between the N inputs p_i, representing the probability table of the quantized wavelet coefficient of a speech frame, and vectors formed from the columns of the input weight matrix W_{ij}, was calculated by means of the following equation,

$$D_j = -\left(\sum_{i=1}^{N} p_i \bullet \log_2 W_{ij} - \sum_{i=1}^{N} p_i \bullet \log_2 p_i \right) \qquad (1)$$

The quantity D_j in the equation (1) represents the difference between the following two quantities:

- $-\sum_{i=1}^{N} p_i \bullet \log_2 W_{ij}$, which represents the mean number of bits we would code the quantized wavelet coefficients of that frame by using an optimal arithmetic coding with the probability table [$W_{1j}, W_{2j}, \ldots, W_{Nj}$];

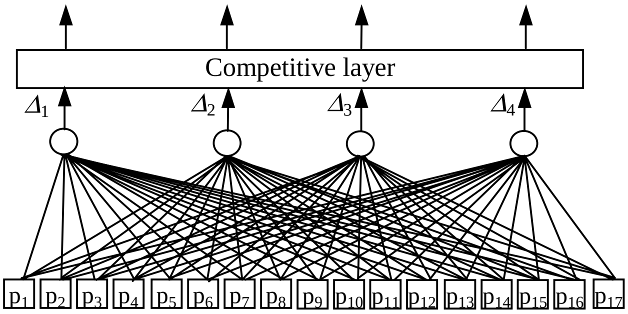

Fig. 1. The neural network topology with N=17 input symbols and M=4 output neurons.

- $-\sum_{i=1}^{N} p_i \bullet \log_2 p_i$, which represents the entropy of the probability distribution [$p_1, p_2, \ldots, p_N$].

Finding the distances D_j and subtracting the biases b_j, we compute the Δ_j elements

$$\Delta_j = D_j - b_j \qquad (2)$$

The competitive transfer function returns neuron outputs of 0 for all neurons except for the *winne*r, the neuron associated with the minimum element Δ_j. The winner's output is 1. The condition

$$\sum_{i=1}^{N} \mathrm{W_{ij}} = 1 \ \ \forall j \qquad (3)$$

is imposed on the weight matrix and on the inputs. We used the Kohonen learning rule [8] to adapt the weights of the winning neuron *j*,

$$W_{ij}(t) = W_{ij}(t-1) + \lambda \bullet (p_i(t) - W_{ij}(t-1)) \qquad (4)$$

This rule preserves the condition (3).

The biases b_i are updated during the training to force each neuron to classify roughly the same percentage of input vectors[18]. If M is the number of the output neurons, the probability table is encoded with only $\log_2$M bits.

The neural network is also trained during the compression in order to learn the speech characteristics of the current speakers. If the performances of the new codebook look meaningfully different from the currently used codebook performances, we can change the codebook sending the new codebook to the receiver.

5. REAL TIME IMPLEMENTATION

Real-Time voice transport introduces tight constraints on QoS with respect to delay, jitter, loss and/or error, due to the limited tolerance of the human listener to both the average delay and the fluctuation of delay. The overall delay should not exceed 200-250 ms, but a delay of 200 ms to 800 ms is conditionally acceptable for a short portion of the conversation, when such delays are rare and far apart [10]. Due to fluctuations of the network delay, buffering is needed at the receiver.

We implemented our tool on the TMS320C6701 Evaluation Module. The TMS320C6701 Evaluation Module is equipped with a Peripheral Component Interconnect (PCI) interface, which supports high-speed modes of data transfers.

The input voice was sampled at 16 kHz and digitalized by means of the 16-bit A/D-D/A converter, which is on the DSP board. In order to transfer continuously the digitalized voice from the A/D serial port to the CPU memory without loading the CPU, we performed this operation by programming the Direct Memory Access controller (DMA). Although the CPU and the DMA controller function independently of one another, when both are performing simultaneous data accesses it is necessary to properly schedule and configure them in order to minimize conflict and waiting while meeting real-time requirements. To allow the CPU activity to be distanced from the DMA activity we implemented a *ping-pong buffering* technique [17]. In ping-pong buffering there are two sets of data buffers for all incoming and outgoing data streams. While the DMA is transferring data to the ping buffers, the CPU is compressing the data in the pong buffers. When both CPU and DMA activity completes, they switch. The compressed signal is sent in real-time to an host computer through the onboard DSP Host Port Interface. The HPI is a parallel 16-bit port through which a host processor can directly access the internal and external memory space of the DSP CPU. The host computer packets the compressed voice and sends it to the Internet. In order to test the application an UDP/IP socket was used on 10 Mbps LAN. On the host computer a server application is present too, which receives speech packets from the Internet, unpacks them and sends the compressed speech to the DSP CPU in real-time through the HPI, where it is decompressed by the CPU and sent by the DMA to the serial port connected to D/A converter.

In order to meet the QoS constraints the receiving procedure was implemented by programming the DMA on a circular N-dimensional buffer array. Each element of the buffer array contains the data of a packet, so that the total buffer delay is N*8 ms and values of N up to 25 can be used. In this way we can send the voice to the speakers continuously, without silence gaps caused by the network jitter. Furthermore the buffer permits of reordering out of sequence packets. Packet desequencing was eliminated by using a sequence number inserted in each packet during the compression step.

The arithmetic coding was implemented by using integer arithmetic to partition the cumulative frequency distribution table used at each stage. Not only this is more efficient than using floating point arithmetic, but avoids that different round-off errors can make different machines encode differently [13].

6. MAIN RESULTS

A reconstructed signal with a segmental SNR greater than 16.5 dB was achieved at 32 kbit/s. It is possible to reduce the bit rate by using a quantization step greater than the masking noise. In this way the coding is not any more perceptually transparent, but the quantization noise is tolerable up to 8-10 kbit/s.

The table 1 shows the results obtained for 5 different kind of filter banks. As showed in the second column of table 1, Vaidynathan and Battle-Lemarie filters have a long impulse response and are very frequency selective [9]. Anyway their performance are similar to Symmlet 6, Villasenor 3 and Daubechies 5 filters, which are shorter and then computationally more appreciable. The Villasenor 3 filter bank is biorthogonal, so it can be advantageously used in symmetric windowing.

Filter bank	*Length*	SNR_{SEG} at 32 kbit/s (dB)	SNR_{SEG} at 24 kbit/s (dB)	SNR_{SEG} at 16 kbit/s (dB)
Vaidynathan	24	16.52	14.65	12.62
Daubechies 5	10	16.32	14.30	12.31
Symmlet 6	12	16.49	14.43	12.58
Battle-Lemarie	41	16.57	14.77	12.37
Villasenor 3	6/10	16.53	14.67	12.39

Tab. 1. Segmental signal to noise ratio for different filter banks at different bit rates.

The ITU-T recommendation for 7 kHz bandwidth audio signals working at 24 or 32 kbit/s (for use in hands-free applications such as conferencing) is the G.722.1, a digital coder based on transform coding as well, using a Modulated Lapped Transform [5]. Because the transform window (basis function length) is 640 samples and a 50 per cent (320 samples) overlap is used between frames, the total algorithmic delay of G.722.1 is 40 ms. The overlapping avoids the blocking artifacts that can be listened in most DCT-based compression system, but introduces an algorithmic delay that could be unacceptable in real-time applications, such as video-conferencing, in which the telecommunication system can create echo problems.

Three neural networks were trained for the following subbands: 0-2kHz, 2kHz-4kHz and 4kHz-8kHz. Codebooks of different sizes were obtained by using M=2,4,8,16 output neurons and N=9,17,33 input symbols p_i. We used as training set the TIMIT corpus sentences spoken by 64 different speakers from 8 major dialect regions of the United States. A separate set of sentences spoken by 8 different speakers was used as test set. The best trade-off between bit rate, computation complexity and quality of the reconstructed speech was M=4 neurons and N=17 or N=33 symbols. For this configuration the overhead for the transmission of a probability table code is only 250 bit/s.

The performance of the neural network were evaluated on the test set for this configuration by comparing the bit rate obtained using the learned probability tables with the bit rate of 2 alternative approaches:

- the bit rate obtained using a fixed probability table calculated on the same data used for the network training.
- the bit rate obtained using an adaptive model .

In both alternative approaches the probability of each symbol was set equal to its relative frequency. Two different implementation of the adaptive model were made. In the first one the model was restarted every 100 frames (800 ms) and in the second one the model was restarted every 10 frames (80 ms) in order to reduce the effect of errors in the decoding caused by packet losses. The results, showed in table 2, demonstrate that the performances of the neural network approach are better then the other approaches (the Villasenor 3 filter bank was used).

The test set was also used in order to measure the suitability of the wavelet transform to concentrate speech information. We found that nearly the 90% of the Villasenor 3 normalized wavelet coefficients are below 0.05 in module. This result was essential for a low bit rate representation of the coefficients using the arithmetic coding.

Neural network (kbit/s)	Fixed probability table (kbit/s)	Adaptive model restarted every 100 frames (kbit/s)	Adaptive model restarted every 10 frames (kbit/s)
16.0	17.0	17.7	18.8
24.0	25,3	26.2	27.4
32.0	33.6	34.6	35.7

Tab. 2. Bit rates obtained with different arithmetic coding approaches.

The processing delay of the coding algorithm with the TMS320C6701 CPU, which has 8 independent functional units, a 5 ns cycle time and is designed to perform up to eight 32 bit instruction per cycle, was less than 2 ms.

7. CONCLUSIONS AND FUTURE WORK

A wavelet based real-time speech coder has been proposed and a real-time implementation on the TMS320C6000 platform has been tested. Since the coder does not rely on any source model of the signal, it can be used for all audio signals if a new training of the neural network is made.

The low algorithmic delay (8 ms) makes the coder suitable for real-time applications in which the telecommunication system can create echo problems.

The use of a neural network approach for the arithmetic coding of the quantized coefficients led to coding performances better than alternative approaches, as adaptive models or the use of a fixed probability table, in which the probability of each symbol is set equal to its relative frequency.

Future work includes the incorporation of a temporal masking model, the analysis of the coder performance with music signals sampled at 44.1 kHz, a mean opinion score (MOS) evaluation procedure and the use of a real-time network protocol able to minimize the audible artifacts which are caused by packet losses and jitter in the IP network.

8. REFERENCES

[1] B. Carnero, A. Drygajlo, "Perceptual Speech Coding and Enhancement Using Frame-Synchronized Fast Wavelet Packet Transform Algorithms", IEEE Trans. on Signal Processing, vol. 47, no. 6, June 1999.

[2] Daubechies, I., *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.

[3] Fu, X., Z. Zhang, *TMS320C6000 DSP Multichannel Vocoder Technology Demonstration Kit Host Side Design*, Texas Instruments Appl. Rep. SPRA558B- February 2000.

[4] P.G. Howard, J.S. Vitter, “Practical Implementations of Arithmetic Coding”, in *Image and Text Compression*, James A. Storer, ed., Kluwer A. P., Norwell, MA, pp. 85-112, 1992.

[5] ITU-T: G.722.1, *Coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss*, in Series G: Transmission Systems and Media, Digital Systems and Networks, 1999.

[6] Jayant, N.S., P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] J.D. Johnston, "Transform coding of audio signals using perceptual noise criteria” Select. Areas Commun., vol. 6, pp. 314–323, Febr. 1988.

[8] Kohonen, T., *Self-Organization and Associative Memory*, 2nd Edition, Berlin: Springer-Verlag, 1987.

[9] Mallat, S., *A wavelet tour of signal processing*, Academic Press, Second Edition 1998.

[10] Minoli, D., E. Minoli, *Delivering Voice over IP Network*, John Wiley & Sons, New York, 1998.

[11] Papamichalis, P.E., *Practical approaches to speech coding*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[12] E. Pasero, A. Montuori, "Wavelet Based Wideband Speech Coding on the TMS320C67 for Real Time Transmission", in Proceedings of IEEE MTAC 2001, Multimedia Technology and Applications Conference, Irvine, 8-11, Nov. 2001.

[13] Press, W.H., S.A. Teukolsky, W.T. Vetterling, B.P Flannery, *Numerical Recipes in C: The Art of Scientific Computing* , Cambridge University Press, pp. 910-915, 1988.

[14] D. Quaglia, A. Montuori, J.C. De Martin, E. Pasero, “Interactive DSP Educational Platform for Real-Time Subband Audio Coding”, in Proc. of ICASSP 2002, International Conf. on Acoustics, Speech, and Signal Processing, May 2002.

[15] M.R. Schroeder, B.S. Atal, J.L. Hall, “Optimizing digital speech coders by exploiting masking properties of the human ear” , JASA, vol.66, no. 6, pp. 1647–1652, Dec. 1979.

[16] I. Singh, P. Agathoklis, A. Antoniou, "Wavelet-based compression of speech signals on the TMS320C30 digital signal processor", Advances in Digital Filtering and Signal Processing, 1998 IEEE Symposium on, pp.178 –182, 1998.

[17] Texas Instruments Inc., *TMS320C6000 Peripherals Reference*, Literature Number SPRU190D, Febr. 2001.

[18] The Mathworks Inc., *Neural networks toolbook*, Sept 2000.

[19] J.D. Villasenor, B. Belzer, J. Liao, "Filter Evaluation and Selection in Wavelet Image Compression", IEEE Data Compression Conference, DCC '94. Proc., pp. 351 -360, 1994.

[20] M.V. Wickerhauser, *INRIA Lectures on wavelet packet algorithms, Problemes Non Lineaires Appliques, Ondellettes and Paquets d’Onde,* P.L. Lions Ed., Roquencourt, France,1991.