A Pipelined Adaptive Differential Vector Quantizer for Low-power Speech Coding Applications[1]

Naresh R. Shanbhag Keshab K. Parhi

Dept. of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract

A fine-grain pipelined adaptive differential vector quantizer architecture is proposed for low-power speech coding applications. The pipelined architecture is developed by employing the relaxed look-ahead technique. The hardware overhead due to pipelining is only the pipelining latches. Simulations with speech sampled at 8 Khz show that, for a vector dimension of 8, the degradation in the signal-to-noise ratio (SNR) due to pipelining is negligible. Furthermore, this degradation is independent of the level of pipelining. Thus, the proposed architecture is attractive from an integrated circuit implementation point of view.

1 Introduction

In the past, pipelined digital signal processing (DSP) algorithms have been developed for high-speed applications such as video compression. However, in [1] the use of pipelining for reducing power consumption in VLSI chips has been described. This fact motivates the need for designing pipelined algorithms for low-power applications such as speech codecs for personal communication systems. Vector quantization (VQ) [2] is an attractive technique for source coding at low bit rates. However, for a fixed rate, the encoder complexity increases exponentially with the vector dimension. The adaptive differential vector quantizer (ADVQ) [3]-[4] belongs to the class of feedback vector quantizers, which have reduced complexity and are attractive from an implementation point of view.

The recursive structure of ADVQ along with the predictor adaptation makes it difficult to pipeline it. Unlike [3]-[4], we consider continuous predictor adaptation. In the past, algorithm transformation techniques [5] such as *look-ahead* [6] have been succesfully employed for pipelining of numerous recursive DSP algorithms at expense of a hardware overhead. Recently, in [7], we proposed an approximate form of look-ahead called the *relaxed look-ahead*, which results in negligible hardware overhead. The relaxed look-ahead maintains the functionality of the algorithm instead of the exact input-output behavior and, therefore, results in hardware efficient pipelined algorithms with negligible degradation in performance. Employing the relaxed look-ahead technique, we proposed a fine-grain pipelined ADVQ (PIPADVQ) architecture [8] for real-time video compression. In this paper, we apply the PIPADVQ algorithm in [8] for low-power speech coding at rates $R = 8$ Kbps and $R = 6$ Kbps [9].

There are significant differences between the PIPADVQ architecture in [8] and the PIPADVQ architecture addressed in this paper. The PIPADVQ architecture in [8] exploited the two-dimensional nature of video signals by employing a row-by-row vector order and a linear vector topology. This cannot be done in the present context of speech compression. In this paper, we show that the degradation in the signal-to-noise ratio (SNR) due to pipelining for any rate R decreases as the vector dimension is increased and is negligible for a vector dimension of 8. Furthermore, the SNR degradation remains more or less constant as the speed-up increases. This is verified by simulations on real speech data sampled at 8 KHz.

2 The Pipelined ADVQ Architecture

The derivation of PIPADVQ using relaxed look-ahead is given in [8]. A M-level PIPADVQ coder (see Fig.1(a)) is described by the following equations

$$\hat{s}_i(n+M) = \mathbf{W}_i^T(n-1)\tilde{\mathbf{S}}_i(n-1) \quad i=1 \quad \text{to} \quad K \tag{2.1}$$

$$\mathbf{e}(n) = \mathbf{S}(n) - \hat{\mathbf{S}}(n) \tag{2.2}$$

$$\mathbf{e}_q(n) = \beta(\gamma(\mathbf{e}(n))) \tag{2.3}$$

$$\tilde{\mathbf{S}}(n) = \hat{\mathbf{S}}(n) + \mathbf{e}_q(n) \tag{2.4}$$

$$\mathbf{W}_i(n) = \mathbf{W}_i(n-1) + \mu_p \mathbf{e}_{qi}(n)\tilde{\mathbf{S}}_i(n-M-1)$$

$$i = 1 \quad \text{to} \quad K. \tag{2.5}$$

[1] This research was supported by the army research office by contract number DAAL03-90-G-0063.

0-7803-1254-6/93$03.00 © 1993 IEEE

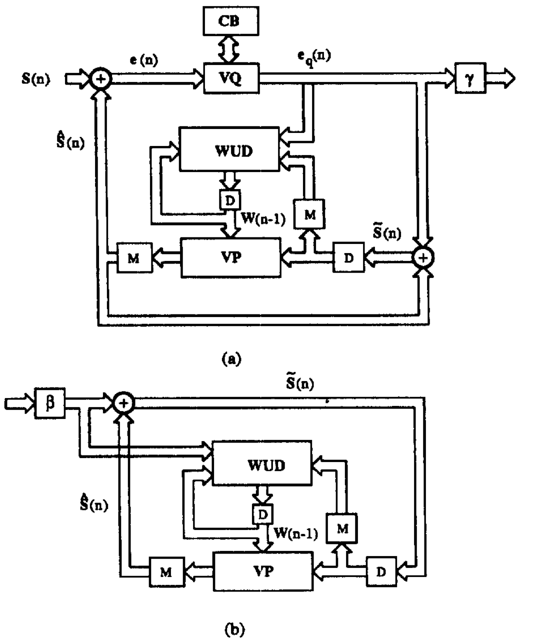

Fig. 1

Figure 1: The PIPADVQ : (a) coder and (b) decoder.

where n is the time index, $\mathbf{S}(n) = [s_1(n), s_2(n), \ldots, s_K(n)]^T$ is the input vector, K is the vector dimension, $\hat{\mathbf{S}}(n) = [\hat{s}_1(n), \hat{s}_2(n), \ldots, \hat{s}_K(n)]^T$ is the predicted vector, $\mathbf{e}(n)$ is the prediction error vector and $\tilde{\mathbf{S}}(n) = [\tilde{s}_1(n), \tilde{s}_2(n), \ldots, \tilde{s}_K(n)]^T$ is the reconstructed signal. In addition, the encoder function $\gamma(.)$ assigns to each $\mathbf{e}(n)$ the codeword index as the channel symbol and $\beta(.)$ is the decoder read-out function which returns a value from the reproduction alphabet for a given channel symbol. The predictor weight vector for the i^{th} dimension ($i = 1, ..., K$) is denoted by $\mathbf{W}_i(n)$. Finally, $\tilde{\mathbf{S}}_i(n)$ is the vector of past reconstructed samples employed for predicting $s_i(n)$.

We employed an improved version of the stochastic gradient algorithm (SGA) in [10]. This algorithm was also employed in [8]. Our algorithm adapts the predictor to minimize the prediction error (see (2.5), while the codebook adaptation (given below) minimizes the reconstruction error.

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu_c(\mathbf{e}(n) - \mathbf{c}(n-1)), \tag{2.6}$$

where μ_c is the codebook adaptation step-size and $\mathbf{c}(n)$ is the codevector to which the prediction error $\mathbf{e}(n)$ has been mapped. This algorithm has the advantage of being less sensitive to the changes in the step-sizes μ_p and μ_c and also offers an SNR improvement. Furthermore, analysis for joint stability of the predictor and codebook adaptations was also carried out in [8].

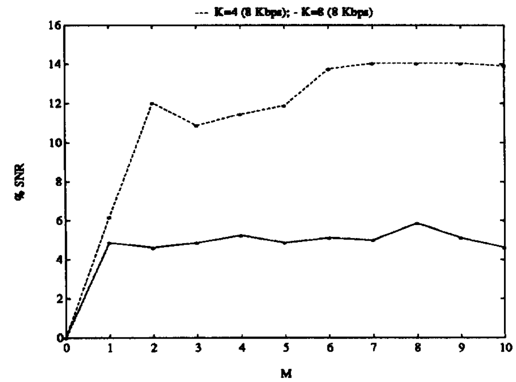

Figure 2: ΔSNR vs. M at $R = 8$ Kbps.

In Fig.1, the prediction error vector quantizer **VQ**, the codebook **CB**, the vector predictor **VP** and the weight-update block **WUD** are shown. The PIPADVQ has M delays in the most critical loop and these can be retimed to pipeline all the computational blocks and the two vector adders. Note that, the only hardware overhead due to relaxed look-ahead are the $2MK$ pipelining latches. However, employing relaxed look-ahead has resulted in the use of $M + 1$ *vector-step* forward prediction (see Fig.1). This corresponds to $(M+1)K$ *scalar step* forward prediction. Clearly, as M (or K) increases, the output SNR would degrade due to the increase in the prediction error power. However, from rate-distortion theory [11], the output SNR should improve as K increases. Thus, it is important to investigate whether the degradation in SNR due to increase in M is offset by an increase in the value of K.

As a measure of performance, we employ the relative SNR degradation ΔSNR defined as

$$\Delta SNR = \frac{SNR_{serial} - SNR_{pipe}}{SNR_{serial}}, \tag{2.7}$$

where SNR_{serial} and SNR_{pipe} are the SNR's of the serial ADVQ and the PIPADVQ, respectively.

Simulations with speech data sampled at 8 KHz indicate (see Fig.2) that as the vector dimension K is increased from 4 to 8 (at $R = 8$ Kbps), ΔSNR drops from approximately 12.22% to 5.01%. Furthermore, for a given vector dimension K, ΔSNR is approximately constant for pipelining levels of up to 10. We have confirmed that ΔSNR for $K = 8$ does not change even at values of M as high as 20. Note that $M = 20$ (with $K = 8$) corresponds to a 168 scalar-step forward prediction. Clearly, the increased vector dimension compensates for the increased prediction error. To further confirm these results, we repeated the simulations for $R = 6$ Kbps. Again (see Fig.3), we find that the average ΔSNR drops from 13.88% for $K = 4$

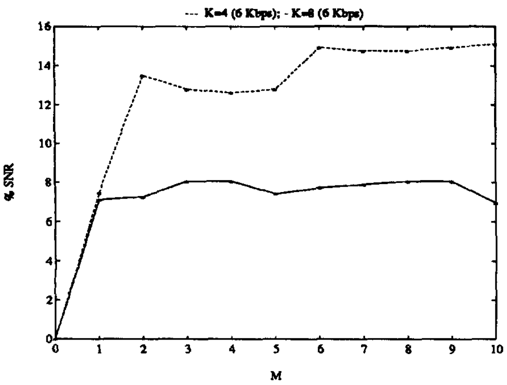

Figure 3: ΔSNR vs. M at $R = 6$ Kbps.

to 7.66% for $K = 8$. Furthermore, with $K = 8$, the ΔSNR remains approximately constant for values of M upto atleast 20.

3 Summary and Conclusions

The PIPADVQ architecture derived by the application of relaxed look-ahead offers substantial speed-ups at no extra hardware cost (except for the pipelining latches). Furthermore, the SNR degradation due to pipelining is independent of the speed-up. The shorter critical path in the pipelined architecture can be charged or discharged at the same speed as the non-pipelined system using a lower supply voltage. This voltage reduction can lead to low power consumption. Furthermore, the pipelined architecture can also be unfolded [12]-[13] systematically to design parallel or block processing architectures which can also be used for low-power speech coding though at the expense of an increase in silicon area. Thus, the PIPADVQ can be employed in a low-power speech coding environment.

References

[1] A. P. Chandrakasan, S. Sheng and R. W. Brodersen, "Low power CMOS digital design", *IEEE J. of Solid-State Circuits*, vol. 27, pp. 473-484, April 1992.

[2] R. M. Gray, "Vector quantization", *IEEE ASSP Mag.*, pp. 4-29, Apr. 1984.

[3] V. Cuperman and A. Gersho, "Adaptive differential vector coding of speech", *Conference Record, GlobeCom 82*, pp. 1092-1096, Dec. 1982.

[4] J. W. Modestino and Y. H. Kim, "Adaptive entropy-coded predictive vector quantization of images", *IEEE Trans. on Signal Processing*, vol. SP-40, no. 3, pp. 633-644, March 1992.

[5] K. K. Parhi, "Algorithm transformation techniques for concurrent processors", *Proceedings of the IEEE*, vol. 77, pp. 1879-1895, Dec. 1989.

[6] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters - Part I : Pipelining using scattered look-ahead and decomposition", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 37, pp. 1099-1117, July 1989.

[7] N. R. Shanbhag and K. K. Parhi, "A pipelined adaptive lattice filter architecture", *IEEE Trans. on Signal Processing*, May 1993.

[8] N. R. Shanbhag and K. K. Parhi, "A pipelined differential vector quantizer architecture for real-time video applications", in *Proc. IEEE Workshop on Visual Signal Processing and Communications*, Raleigh, North Carolina, Sept. 1992, pp. 9-14.

[9] N. R. Shanbhag and K. K. Parhi, "A pipelined adaptive differential vector quantizer for low-power speech coding applications", *IEEE Trans. on Circuits and Systems*, Jan. 1993.

[10] P.-C. Chang and R. M. Gray, "Gradient algorithms for designing predictive vector quantizers", *IEEE Trans. on Acoust., Speech and Signal Proc.* , vol. ASSP-34, no. 4, pp. 679-690, August 1986.

[11] T. Berger, *Rate Distortion Theory*, Prentice-Hall Inc., Englewood Cliffs, NJ (1971).

[12] K. K. Parhi and D. G. Messerschmitt, "Static rate-optimal scheduling of iterative data-flow programs via optimal unfolding", *IEEE Trans. on Comput.*, vol. 40, no. 2, pp. 178-195, Feb. 1991.

[13] K. K. Parhi, "A systematic approach for the design of digit-serial signal processing architectures", *IEEE Trans. on Circuits and Systems*, vol. 38, no. 4, pp. 358-375, April 1991.