(From *Proc. Second ARPA RASSP Conference,* (Washington, D.C.), pp. 257–261, July 1995.)

Approximate Processing and Incremental Refinement Concepts

J. Winograd*, J. Ludwig, H. Nawab*, A. Chandrakasan, A. Oppenheim
Massachusetts Institute of Technology, RLE, Cambridge, MA 02139
*Boston University, ECS Department, Boston, MA 02215

Abstract

Approximate processing and incremental refinement concepts are needed for applications where it is desirable to provide a systematic tradeoff between the quality of signal processing results and the availability of resources, such as time, bandwidth, memory, and power. We examine the impact of these concepts for three distinct application areas: (1) low-power frequency-selective FIR filtering, (2) real-time time-frequency analysis of signals and (3) DCT-based image encoding/decoding. Results from approximate processing of signal data illustrate the practical utility of these types of systems.

1 Introduction

Approximate processing of signals may be used by a signal processing system to potentially achieve a wide range of possible tradeoffs between utilization of system resources (cost) and system performance (output quality). We have formulated the concept of *incremental refinement* (IR) structures for signal processing transformations which in conjunction with appropriate *control strategies* may be used in designing approximate signal processing systems for various applications. These IR structures have the property that at intermediate stages of computation they produce approximations to the final output. In this paper, we report results from our exploration of three categories of practical applications for IR-based approximate processing systems.

2 A Low-Power Application

Recent investigations [1] [2] [3] related to low-power CMOS implementations of signal processing systems have been motivated by the growing demand for portable multi-media devices. A key issue in the design of such devices is to minimize the total power consumption in order to maximize the run time or minimize battery size. In this section we develop an algorithm designed for low-power frequency-selective FIR filters, an essential element in many communication devices. A key feature of this algorithm is that the number of filter taps used is dynamically varied to provide stopband attenuation in proportion to a simple estimate of the time-varying energy in the undesired components of the input signal.

To first order, the average power required to perform a signal processing task is: $P = \sum_i N_i C_i V_{dd}^2 f_s$, where C_i is the average capacitance switched per operation of type i (corresponding to addition, multiplication, storage, etc.), N_i is the number of operations of type i performed per sample, V_{dd} is the operating supply voltage, and f_s is the sampling frequency. One approach to power reduction is to lower N_i, which in the case of FIR filtering corresponds to the number of taps used to produce each output sample.

Consider a tapped delay line implementation (Fig. 1) of an FIR filter whose taps correspond to a rectangularly windowed version of the impulse response of an ideal frequency selective filter. This tapped delay line may be viewed as an incremental refinement structure for carrying out frequency selective filtering. If a subset of the taps are *powered down* (effectively disabling portions of the tapped delay line) in such a way that the length of the corresponding ideal impulse response is further truncated, then the net stopband attenua-

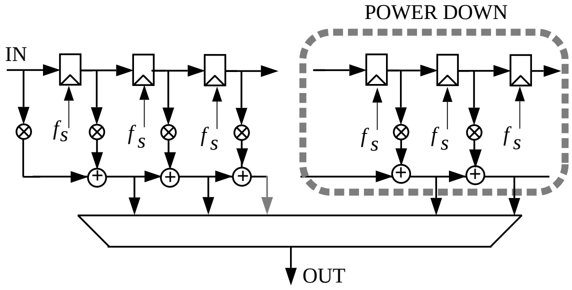

Figure 1: Illustration of dynamic tapped delay line

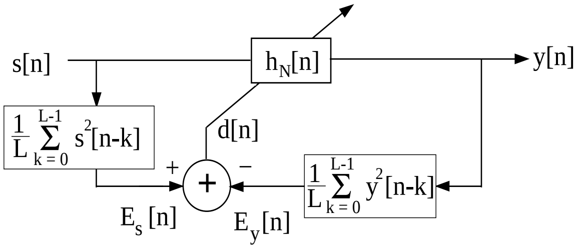

Figure 2: Overview of approximate filtering strategy

tion in the FIR filter output decreases proportionately. Conversely, *powering up* of additional taps can be used to increase the stopband attenuation in the FIR filter output. We have developed a *control strategy* to utilize this incremental refinement property of the tapped delay line for constructing approximate filtering systems whose stopband attenuation (and the corresponding power consumption) is adapted to the time-varying stopband energy of the input signal.

Our control strategy for approximate filtering is depicted in Figure 2. Note that only two additions and two multiplications are required to obtain $E_s[n+1]$ from $E_s[n]$. To analyze the control strategy, we assume that the input signal $s[n] = x[n] + w[n]$, where $x[n]$ and $w[n]$ correspond to wide sense stationary, independent random processes. The output $y[n]$ is produced by filtering $x[n]$ at each time, n, with an $N[n]$-tap filter having frequency response $H_{N[n]}(\omega)$. Given that the output power spectral density is

$$P_y(\omega) = P_s(\omega)|H_{N[n]}(\omega)|^2 + P_w(\omega)|H_{N[n]}(\omega)|^2, \quad (1)$$

our objective is to choose $N[n]$ to be the smallest odd integer which assures that

$$\int_{SB} P_y(\omega)|H_{N[n]}(\omega)|^2 d\omega < \gamma, \quad (2)$$

where the parameter γ is the maximum tolerable stopband energy in the output, and SB denotes the stopband region in frequency. Defining $A_{SB}(N[n]) = \int_{SB} |H_{N[n]}(\omega)|^2 d\omega$ and $Q = d[n]A_{SB}[n-1]$, the decision rule for choosing $N[n]$ is:

$$\begin{array}{rcl} Q > \gamma & \longrightarrow & \text{increase } N \text{ by } N_0 \\ \gamma - \delta < Q < \gamma & \longrightarrow & N \text{ unchanged} \\ Q < \gamma - \delta & \longrightarrow & \text{decrease } N \text{ by } N_0 \end{array} \quad (3)$$

The parameters δ and N_0 control the sensitivity of the dynamically-varying filter length. The values of $A_{SB}(N[n])$ can be precalculated and stored in a lookup table. In the case of stationary interferences, the filtering algorithm will converge to the length which satisfies eqn. (2) and remain fixed thereafter. For nonstationary interferences, the filter length will dynamically vary to approach the appropriate length at each time.

We have successfully applied our approximate filtering system to the problem of separating a low-frequency FDM voice channel from a high-frequency one and demonstrated that significant power savings over conventional methods may be obtained. The incorporation of adaptive approximate filters into a binary-tree structured filterbank for low-power source coding applications has also been successfully explored.

3 A Real-Time Application

The discrete Fourier transform (DFT) is an important component of many real-time signal processing systems. For those in which real-time deadlines vary or computational resources are dynamically allocated, the incremental refinement approach to DFT computation provides a framework in which the availability of an approximate result can be guaranteed across a wide range of resource allocations.

In our recent work, we have developed a class of incremental refinement structures for DFT computation. These structures allow many different quality/cost refinement paths to be achieved. Such structures may be derived by considering the real-valued N-point signal under analysis to be represented in radix complement form using a fixed-point mixed-radix encoding[1] [4]. When a D-digit signal representation based upon the radices $(m_{D-1}, m_{D-2}, \ldots, m_1, m_0)$ is employed and m_{D-1} is even, the value of each signal point $x(n)$ can be related to the value of the digits with which it is encoded via:

$$x(n) = \sum_{d=0}^{D-1} \alpha_d(x_d(n))\beta_d \quad (4)$$

where $x_d(n)$ is the dth digit from the least-significant digit of the word representing $x(n)$,

$$\alpha_d(x) = \begin{cases} x, & (d \neq D-1)\vee \\ & (0 \leq x \leq (m_{D-1}/2)-1), \\ x - m_{D-1}, & (d = D-1)\wedge \\ & (m_{D-1}/2 \leq x \leq m_{D-1}-1), \end{cases} \quad (5)$$

and

$$\beta_d = \begin{cases} 1, & d = 0, \\ \prod_{j=0}^{d-1} m_j, & 1 \leq d \leq D-1, \end{cases} \quad (6)$$

[1] We note that fixed-point binary and, in some instances, floating-point representations can be considered as special cases of mixed-radix.

Using a framework for deriving successive approximations to the DFT [5] based on a backward differencing approach [6] to DFT evaluation, a family of structures which perform incremental refinement can be implemented from the following update equations. The ith successive approximation, $\hat{X}_i(k)$, is computed from the previous approximation $\hat{X}_{i-1}(k)$ by:

$$\hat{X}_i(k) = \begin{cases} \hat{X}_{i-1}(k) + C_i(k), & c_{i-1} < k \leq c_i, \\ \hat{X}_{i-1}(k) + R_i(k) + V_i(k), & 1 \leq k \leq c_{i-1}, \end{cases} \tag{7}$$

where $C_i(k)$ is the *coverage update*, which is defined as

$$C_i(k) = \sum_{d=D-v_i}^{D-1} \sum_{n=0}^{r_i} g_d(n) G_{n,d}(k), \tag{8}$$

$R_i(k)$ is the *resolution update*, which is defined as

$$R_i(k) = \sum_{d=D-v_i}^{D-1} \sum_{n=r_{i-1}+1}^{r_i} g_d(n) G_{n,d}(k), \tag{9}$$

and $V_i(k)$ is the *SNR update*, which is defined as

$$V_i(k) = \sum_{d=D-v_i}^{D-v_{i-1}-1} \sum_{n=0}^{r_{i-1}} g_d(n) G_{n,d}(k). \tag{10}$$

Here, we define $\hat{X}_0(k) = 0$ for all k,

$$g_d(n) = \begin{cases} \alpha_d(x_d(0)) - \alpha_d(x_d(N-1)) & n = 0, \\ \alpha_d(x_d(n)) - \alpha_d(x_d(n-1)) & 1 \leq n \leq N-1 \end{cases} \tag{11}$$

and

$$G_{n,d}(k) = \beta_d \frac{e^{-j2\pi kn/N}}{1 - e^{-j2\pi k/N}} \tag{12}$$

The variables c_i, r_i, and v_i represent control parameters which determine the solution quality achieved in the ith successive approximation. Metrics relating c_i to frequency coverage, r_i to frequency resolution, and v_i to the SNR of the analysis have been derived [4]. The computational cost of producing an approximation of a given quality has also been analyzed, and was shown [4] to depend both on the input data as well as the radix used for signal encoding when the algorithm of [6] is used for implementing the updates of eqs. (7)-(10). A policy for selecting the most efficient set of radices has been developed [4] and, in the fixed-radix case, a variety of different solutions have been obtained [7] to the problem of selecting control parameter values that can be expected to meet specific design constraints.

As in the case of our incremental refinement structures for approximate filtering, the above DFT structures may be used in conjunction with appropriate control strategies to obtain IR-based systems. For example, we have developed IR-based systems for real-time STFT computation in which the DFT computation for each frame produces an approximate answer whose quality depends upon the characteristics of the input signal and the available computation time.

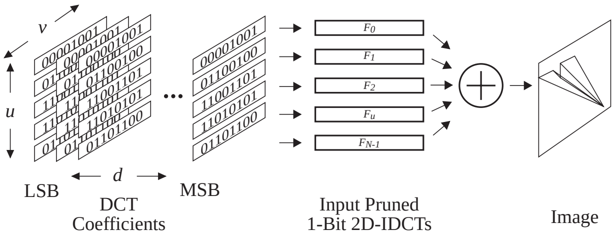

Figure 3: Schematic diagram of the proposed architecture for incremental refinement of 2D-IDCT approximations.

4 A Communications Application

The phenomenal rate of growth in high-capacity telecommunication networks has brought the issue of heterogeneity to the forefront of the communications field. One instance where approximate processing is relevant is when a message is broadcast across a variable bandwidth network to receivers whose characteristics are not known to the sender and which possess a wide range of capabilities. Reduction of video quality may be required at the receivers due to local performance limitations or in order to adjust to variable data rates. In either of these cases, the use of an incremental refinement structure for the decoder implementation enables performance to be easily adapted.

We have recently developed an incremental refinement structure for the two-dimensional inverse discrete cosine transform (2D-IDCT). The energy compaction properties of the DCT make it a popular tool for image and video coding. Accordingly, IDCT computations comprise a significant proportion of the computational effort required in the decompression of the most widely used image and video coding standards.

Our incremental refinement structure for the 2D-IDCT has the distributed arithmetic (DA) [8] architecture shown schematically in Fig. 3. Other structures for computing the 2D-IDCT have previously been developed [9] [10] [11] using DA but they do not have the IR property.

A primary difference between our IR structure for the 2D-DCT and the other structures lies in the bit-serial ordering in which the distributed arithmetic operation is performed. Our architecture begins processing at the most significant bit of the input words, advancing progressively towards the least significant. With this approach, the intermediate results obtained at the output of the DA sub-system represent an approximation of the exact result based on the quantization of the input data to a fewer number of representation levels. When the standard DA procedure of beginning at the least significant bit is employed, the intermediate values of the DA process have no general utility. This observation is illustrated in Fig. 4.

Another important innovation in our work lies in the basic manner in which distributed arithmetic has been applied to the 2D-IDCT. Previously reported implementations [9] [10] [11] are based upon the decomposition of the 2D-IDCT into the 1D-IDCT of the rows of the input data followed by the 1D-IDCT of each of the columns. Obtaining satisfactory incremental refinement behavior from this architecture is hindered by the fact that even if the MSB-to-LSB bit ordering is used, the intermediate results produced by the first stage of row 1D-IDCT processing do not represent approximations to the desired output.

As in all applications of distributed arithmetic, the selection of an appropriate DA structure is strongly influenced by tradeoffs between performance and memory usage. For instance, a direct DA implementation of the 8x8 2D-IDCT would require an astronomical 2^{64} words of ROM. In contrast, the architecture described here, with no memory saving optimizations applied, needs 2^{17} words (128K) of ROM. Due to the periodic structure of the IDCT basis functions, there exists considerable potential for reducing this memory requirement further. Such techniques have been successfully applied in the separable 2D-IDCT implementation [9], for which the memory requirements for the 16x16 transform were reduced from 2^{21} words (2M) to 2^{10} words (1K).

To examine our 2D-IDCT IR structure, consider the NxN 2D-IDCT of $X(u,v)$:

$$x(i,j) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)X(u,v) \times \cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (13)$$

where $C(0)=1/\sqrt{2}$ and $C(u)=C(v)=1$ for $u,v\neq 0$. Throughout our derivation, u, v, i, $j \in \{0..N-1\}$. When $X(u,v)$ is encoded in two's complement binary we have, using the notation of section 3, $\forall d : m_d = 2$ and the 2D-IDCT can be written as:

$$x(i,j) = \frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)C(v)\sum_{d=0}^{D-1}\alpha_d(X_d(u,v))\beta_d \times \cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (14)$$

with $X_d(u,v)$ denoting the dth bit of the binary representation of $X(u,v)$. We can now express the 2D-IDCT in a form suitable for applying distributed arithmetic:

$$x(i,j) = -\sum_{u=0}^{N-1} F_u(X_{D-1}(u,v),i,j)\beta_{D-1} + \sum_{d=0}^{D-2}\sum_{u=0}^{N-1} F_u(X_d(u,v),i,j)\beta_d \quad (15)$$

with

$$F_u(X_d(u,v),i,j) = C(u)\frac{2}{N}\sum_{v=0}^{N-1} C(v)X_d(u,v) \times \cos\left[\frac{(2i+1)u\pi}{2N}\right]\cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (16)$$

The arguments to each function F_u are a row vector of N bits (indexed by v) taken from the d-th position of the u-th row of $X_d(u,v)$, and a coordinate of $x(i,j)$. It's output is the 2D-IDCT of the given row vector of bits evaluated at position (i,j). By pre-computing and storing in memory the values of F_u, and implementing separately the F_u functions as shown in Fig. 3, the entire summation over u in eq. (15) can be evaluated in parallel for a single value of d. Thus, at each stage of computation (i.e. for each value of d) the structure updates its previous result with the 2D-IDCT corresponding to an entire additional bit plane of the input coefficients. The scaling associated with β_d in eqn. (15) is implemented via bit shifting in the output accumulators.

The IR structure outlined above for the 2D-IDCT may be used in a practical DCT-based image encoding/decoding system. An appropriate control strategy may be used by each receiver for terminating the decoding process at any intermediate stage in accordance with the availability of system resources and/or the desired quality of the decoded image.

References

[1] J. G. Ackenhusen, *Signal Processing Technology and Applications*, IEEE Press, 1995.

[2] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power Digital CMOS Design," *IEEE Journal of Solid State Circuits,* pp. 473-484, April 1992.

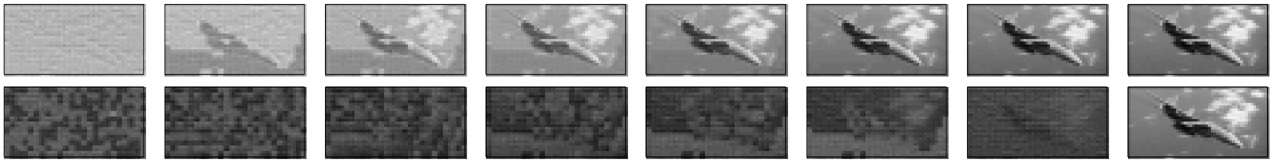

Figure 4: The top row of images illustrates the approximate results obtained after 8 successive stages of 2D-IDCT refinement using the distributed arithmetic architecture described in the text on all 16x16 pixel blocks of a 384x192 pixel 8-bit image. The bottom row of images depicts the corresponding results obtained using a standard distributed arithmetic approach to performing the 2D-IDCT.

[3] B. M. Gordon and T. Meng, "A Low Power Subband Video Decoder Architecture," *International Conference on Acoustics, Speech, and Signal Processing,* April, 1994.

[4] J. M. Winograd and S. H. Nawab, "Mixed-radix approach to incremental DFT refinement," in *Advanced Signal Processing Algorithms* (F. T. Luk, ed.), pp. 418–429, Proc. SPIE 2563, 1995.

[5] J. M. Winograd and S. H. Nawab, "Incremental refinement of DFT and STFT approximations," *IEEE Signal Processing Letters*, vol. 2, pp. 25–28, Feb. 1995.

[6] S. H. Nawab and E. Dorken, "Efficient STFT approximation using a quantization and differencing method," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, (Minneapolis), Apr. 1993.

[7] S. H. Nawab and J. M. Winograd, "Approximate signal processing using incremental refinement and deadline-based algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, vol. 5, (Detroit), pp. 2857–2860, May 1995.

[8] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust. Speech and Signal Processing*, vol. ASSP-22, pp. 456–462, Dec. 1974.

[9] M.-T. Sun, T.-C. Chen, and A. M. Gottlieb, "VLSI implementation of a 16x16 discrete cosine transform," *IEEE Trans. on Circ. and Sys.*, vol. 36, pp. 610–617, Apr. 1989.

[10] S. Uramoto, Y. Inoue, A. Takabatake, J. Takeda, Y. Yamashita, H. Terane, and M. Yoshimoto, "A 100-MHz 2-D discrete cosine transform core processor," *IEEE J. Solid State Circuits*, vol. 27, pp. 492–498, Apr. 1992.

[11] H. Fujiwara, M. L. Liou, M.-T. Sun, K.-M. Yang, M. Maruyama, K. Shomura, and K. Ohyama, "An all-ASIC implementation of a low bit-rate video codec," *IEEE Trans. on Circ. and Sys. for Video Tech.*, vol. 2, pp. 123–133, June 1992.