APPROXIMATE SIGNAL PROCESSING USING INCREMENTAL REFINEMENT AND DEADLINE-BASED ALGORITHMS

S. Hamid Nawab and Joseph M. Winograd

ECS Department, Boston University, Boston, MA 02215

ABSTRACT

A framework for approximate signal processing is introduced which can be used to design novel classes of algorithms for performing DFT and STFT calculations. In particular, we focus on the derivation of multi-stage incremental refinement algorithms that meet a variety of design criteria on the tradeoff achieved at each stage between solution quality and computational cost.

1. INTRODUCTION

In any given problem-solving domain, an approximation to a given algorithm may be defined as an algorithm which offers a reduced computational cost but produces a lower quality answer according to some standard of accuracy, certainty, and/or completeness. The approximate algorithm may be said to carry out *approximate processing* in the domain under consideration. Such algorithms have previously been studied in the context of various applications, including real-time vehicular tracking [1, 2] and real-time database query processing [3].

In real-time applications, any individual task must generally be performed within a time interval whose duration may or may not be determined prior to its execution. In the case of a predetermined time allocation, an algorithm which produces an approximate solution may be used to obtain the requisite computational efficiency by sacrificing the quality of the answer obtained. We refer to this as *deadline-based* approximate processing. In cases where the time allocation is not predetermined, it is desirable to use an approximate processing technique which produces an answer of improving quality as a function of time. This allows the processing to be terminated whenever desired and the quality of the resulting answer is directly proportional to the actual execution time. These types of algorithms are said to carry out *incremental refinement* of their answers.

2. DEADLINE-BASED ALGORITHMS

We have previously reported [4, 5] results on the design of deadline-based algorithms for the DFT and the short-time Fourier transform (STFT). That work has recently been extended to include a class of multi-stage algorithms for incremental DFT and STFT refinement [6]. This new approach can be used to generate a large number of algorithms whose stages possess different cost versus quality tradeoff characteristics. In this paper, we demonstrate the application of a mathematical framework for quality and cost assessment to the design of incremental DFT refinement algorithms whose tradeoffs meet a variety of design criteria. Since the STFT can be decomposed into a series of DFT calculations, these design techniques may be applied to the derivation of incremental STFT refinement algorithms as well.

3. INCREMENTAL DFT REFINEMENT

Assume the N-point signal $x(n)$ under analysis to be real-valued and represented in B-bit 2's complement binary fraction format. We denote bit b of the nth sample as $x_b(n)$, so that

$$x(n) = -x_0(n) + \sum_{b=1}^{B-1} x_b(n) 2^{-b} \tag{1}$$

We may define a space of algorithms which perform incremental DFT refinement for which the result of the ith successive approximation is given by:

$$\hat{X}_i(k) = \sum_{b=0}^{v_i-1} \sum_{n=0}^{r_i-1} g_b(n) G_{n,b}(k), \quad l_i \leq k \leq u_i \tag{2}$$

where $g_b(n)$ is the first circular backward difference of $x_b(n)$ and

$$G_{n,b}(k) = \begin{cases} -e^{-j\frac{2\pi}{N}kn}/(1-e^{-j\frac{2\pi}{N}k}), & b = 0 \\ 2^{-b}e^{-j\frac{2\pi}{N}kn}/(1-e^{-j\frac{2\pi}{N}k}), & 1 \leq b \leq B-1 \end{cases} \tag{3}$$

The ith successive approximation is part of a sequence of successive approximations whose results are defined through the following relationships between the indexing bounds of equation (2): $1 \leq v_{i-1} \leq v_i \leq B$, $1 \leq r_{i-1} \leq r_i \leq N_w$, $1 \leq l_i \leq u_i \leq N/2$, $u_{i-1} \leq u_i$, $l_{i-1} \geq l_i$, and $v_i + r_i + u_i - l_i < v_{i+1} + r_{i+1} + u_{i+1} - l_{i+1}$. These constraints ensure that solution quality improves after each successive approximation. Each successive approximation may be implemented through a vector summation process we have previously reported [4] for similar approximations in the context of deadline-based algorithms. When all the successive approximations in a particular sequence are implemented this way, we refer to the implementation of the entire sequence as a single incremental refinement algorithm. Since equation (2) forms the basis for generating many different sequences of successive approximations and the cor-

This work was sponsored in part by the Department of the Navy, Office of the Chief of Naval Research, contract number N00014-93-1-0686 as part of the Advanced Research Projects Agency's RASSP program.

0-7803-2431-5/95 $4.00 © 1995 IEEE

responding incremental refinement algorithms, we refer to it as the *generator equation.*

4. QUALITY AND COST ASSESSMENT

The quality of an approximate result can be characterized using an appropriately selected ensemble of numeric quality measures. Since a technique for estimating the center frequency of the energy distribution of $x(n)$ can be applied to position l_i and u_i in a data dependent manner [4, 5], the quality of $\hat{X}_i(k)$ with respect to frequency coverage is related to the number of frequency bins approximated at the completion of stage i. Defining c_i as the number of samples of the DFT approximated at the completion of stage i, or $c_i = u_i - l_i + 1$, we assert that after stage i, the frequency coverage of $\hat{X}_i(k)$ is

$$q_{c,i} = \frac{2\pi c_i}{N} \tag{4}$$

where $q_{c,i}$ is measured in rad/sample. In terms of the number of resolvable frequency components, the frequency resolution can be shown to be approximately

$$q_{r,i} = r_i \tag{5}$$

where $q_{r,i}$ is the number of resolvable components after stage i. It can also be shown that the additive noise due to the quantization in stage i is approximately

$$q_{v,i} = 6v_i \tag{6}$$

with $q_{v,i}$ representing the SNR in dB.

Also of importance in the analysis of approximate processing algorithms is the systematic evaluation of their computational cost. Let us consider the space of algorithms represented by the evaluation of the generator equation using the vector summation process described in [4]. We first note that the values of $G_{n,b}(k)$ can be pre-computed and stored in memory, allowing $\hat{X}_i(k)$ to be calculated using only additions. The storage of $BN_wN/2$ complex values is required.[1]

The number of vector additions required through stage i depends upon the "density" of non-zero values in the portion of $g_b(n)$ included in the calculation. That density, which we denote as γ_i, is

$$\gamma_i = \frac{1}{r_i v_i} \sum_{b=0}^{v_i-1} \sum_{n=0}^{r_i-1} |g_b(n)| \tag{7}$$

and γ_i takes on values in the range $[0,1]$. *A priori* estimates of γ_i allow us to calculate the expected cost through stage i. If we assume a uniform and independent distribution of quantization levels in $x(n)$, the expected value of γ_i can be easily shown to be 0.5. Positive correlation between adjacent signal samples will reduce this value and a frequency reversal technique [5] can be applied to the evaluation of equation (2) to obtain a similar reduction when they are negatively correlated. We now define a cost function which estimates the number of complex additions performed through stage i as[2]

$$k_i = c_i r_i v_i \hat{\gamma}_i \tag{8}$$

where $\hat{\gamma}_i$ is an estimate of γ_i and k_i is the cost estimate.

5. ALGORITHM DESIGN

The characterizations from section 4 can be used to find algorithms from among those generated by equation (2) that satisfy various types of constraints on their cost and quality tradeoffs. In this section we present three design problems which can be solved analytically within our framework.

The examples share a common approach, which begins with the interpretation of all design constraints as mathematical constraints on the values of cost and quality at each stage of incremental refinement. Combining equations (4)-(6) and (8), we derive the quality/cost tradeoff which governs the family of algorithms generated by equation (2),

$$k_i = \frac{N}{12\pi} q_{c,i} q_{r,i} q_{v,i} \hat{\gamma}_i \tag{9}$$

Although only a discrete set of quality/cost points which satisfy this relation are achievable through equation (2), we initially assume c_i, r_i, and v_i (and, consequently, the quality and cost functions) to be continuous-valued. We also assume that $\hat{\gamma}_i$ is independent of i. The quality/cost points that satisfy both equation (9) and the design constraints are then determined analytically, and the corresponding (continuous) values of c_i, r_i, and v_i are found using equations (4)-(6). These results are then rounded appropriately to integral values which, through equation (2), define the resulting incremental refinement algorithm.

5.1. Refinement in Frequency Coverage and Frequency Resolution with Uniform Incremental Cost

Suppose we wish to generate an incremental DFT refinement algorithm whose stages have equal quality increments in coverage and resolution, constant SNR, and an approximately uniform incremental cost. We shall mathematically interpret the constraint on quality to restrict our solutions to those whose quality falls on the line through the quality space for which

$$q_{c,i} = \frac{\pi}{N_w} q_{r,i} \tag{10}$$

We seek an algorithm whose cost increments are such that

$$k_i = ei \tag{11}$$

where e is the desired number of complex additions per stage. Combining these constraints with the quality and cost measures and solving for the c_i and r_i gives

$$c_i = \sqrt{\frac{Nei}{2N_w v_i \hat{\gamma}_i}} \tag{12}$$

$$r_i = \sqrt{\frac{2N_w ei}{N v_i \hat{\gamma}_i}} \tag{13}$$

[1] Alternatively, we may store the $N_wN/2$ complex values of $W_n(k)$ and compute $G_{n,b}(k)$ as part of the algorithm. In this case, each complex addition becomes 2 real power-of-2 multiplies and a complex addition.

[2] We have omitted the cost of computing the backward-difference of $x_b(n)$ from this relation for simplicity.

This result will now be applied to the design of an algorithm. Consider the task of producing an incremental refinement DFT with $N = 256$, $N_w = 128$, $\hat{\gamma}_i = 0.26$, and $q_{v,i} = 12$ dB for which each stage costs approximately 1500 complex additions and coverage and resolution improve commensurately. Evaluating equations (12)-(13) and rounding down to integer values gives

$$\{c_i\} = \{r_i\} = \{53, 75, 93, 107, 120, 128\} \tag{14}$$

Clearly, we use $v_i = 2$ for all stages. A contour plot of the coverage/resolution versus cost tradeoff characteristic for this design is shown in Fig. 1. Note that our derivation has bounded the cumulative cost through stage i, resulting in some of the cost increments (i.e. per stage costs) exceeding the desired amount. An alternate approach bounding the cost increments can be derived, but has limited importance unless the estimate of γ_i is of high quality.

5.2. Refinement in Frequency Coverage with Exponentially Decreasing Incremental Cost

In a real-time environment, it may be useful to apply an algorithm for incremental DFT refinement which produces an initial approximation within some fixed cost, say a complex additions, and then refines that solution in stages with decreasing cost increments, approaching the incremental cost b at an exponential rate. We derive expressions here for the design of algorithms based on equation (2) which have exponentially decreasing incremental cost and perform refinement in frequency coverage while frequency resolution and SNR are held constant.

The cost constraint can be mathematically expressed as

$$k_i = \sum_{n=1}^{i} \left((a-b)e^{-\alpha(n-1)} + b\right) \tag{15}$$

where the exponential rate constant $\alpha > 0$. Combining this with the measures of cost and quality and solving for c_i gives

$$c_i = \frac{6}{q_{r,i}q_{v,i}\hat{\gamma}_i} \sum_{n=1}^{i} \left((a-b)e^{-\alpha(n-1)} + b\right) \tag{16}$$

which may be used to design an algorithm.

To illustrate, consider the design of an algorithm with $N = 256$, $N_w = 128$, $\hat{\gamma}_i = 0.26$, $q_{r,i} = 128$ components, and $q_{v,i} = 12$ dB for which the cost of the first stage is approximately 2000 complex additions and the incremental cost of subsequent stages decreases towards 500 complex additions at the exponential rate of $\alpha = 0.6$. Such an algorithm, generated from equation (2), will have $r_i = 128$ and $v_i = 2$ for all stages. Applying equation (16) and rounding down to the nearest integer we find

$$\{c_i\} = \{30, 54, 74, 90, 105, 117, 128\} \tag{17}$$

The coverage versus cost tradeoff achieved at each refinement stage of the designed algorithm is shown in Fig. 2. As in the previous example, the design criteria have been met with respect to the cumulative cost. A variant approach which bounds the incremental cost estimate may also be derived for this case.

5.3. Refinement in Frequency Resolution with Path Approximation

One may envisage a situation in which it is desirable to produce an incremental DFT refinement algorithm which gives a cost and quality tradeoff that is as near as possible to a numerically specified characteristic. In this section, we demonstrate a technique for designing algorithms which approximate an arbitrary refinement path in frequency resolution and cost while maintaining constant frequency coverage and SNR.

We address this design problem in the following way. Assume that the specification is given as a series of points in the quality/cost space which describe the desired refinement path. For each point on this path, we find the point on the surface defined by the quality/cost relation given in equation (9) which minimizes a Euclidean distance measure. For this measure, we use

$$d = \sqrt{s_q^2(q_{r,i} - q'_{r,i})^2 + (k_i - k'_i)^2} \tag{18}$$

where $(q'_{r,i}, k'_i)$ is the desired tradeoff of resolution and cost at stage i, $(q_{r,i}, k_i)$ is a point which satisfies equation (9) (with $q_{c,i}$, $q_{v,i}$, and $\hat{\gamma}_i$ as known quantities, constant w.r.t. i), and s_q is a normalization constant which allows a correction for the differing units of measure in the various dimensions of cost and quality. The ordered pair (x, y), here, represents a solution with x resolvable frequency components produced at a cumulative cost of y complex additions. The point that minimizes d is found to be the one for which

$$q_{r,i} = 12\pi \frac{12\pi s_q^2 q'_{r,i} + N q_{c,i} q_{v,i} \hat{\gamma}_i k'_i}{(12\pi s_q)^2 + (N q_{c,i} q_{v,i} \hat{\gamma}_i)^2} \tag{19}$$

This result may be rounded to the nearest integer to derive the associated value of r_i for use with equation (2).

Let us consider the design of an algorithm using this path approximation technique. Suppose that we require an incremental DFT refinement algorithm with $N = 256$, $N_w = 128$, $\hat{\gamma}_i = 0.26$, $q_{c,i} = \pi$ rad/sample, and $q_{v,i} = 12$ dB which has four stages with associated quality and cost that are as close as possible to the resolution/cost points (15, 2000), (40, 4000), (90, 5500), and (128, 6000). From equations (4) and (6), we see that $c_i = 128$ and $v_i = 2$ should be used. Application of equation (19) requires that a suitable value for the normalization constant s_q be determined. Since the range of $q_{r,i}$ and k_i are bounded in accordance with the constraints on the generator equation, we use the value

$$s_q = \frac{k_{max}}{q_{r,max}} = 66.6 \tag{20}$$

to normalize the resolution and cost axes over the full range of refinement. Applying equation (19) we find

$$\{r_i\} = \{23, 50, 86, 109\} \tag{21}$$

Fig. 3 compares the desired and achieved refinement paths for this design.

The approach demonstrated in this example has also been successfully applied to the approximation of an arbitrary refinement path in two dimensions of quality. It requires, however, the numerical solution of a quintic polynomial equation. Techniques for solving the path approximation problem in more than two dimensions of quality are currently under investigation.

6. REFERENCES

[1] V. R. Lesser, J. Pavlin, and E. Durfee. Approximate processing in real-time problem solving. *AI Magazine*, pages 49–61, Spring 1988.

[2] K. Decker, A. Garvey, M. Humphrey, and V. Lesser. A blackboard system for real-time control of approximate processing. In *Proc. 25th Hawaii Int. Conf. Sys. Sci.*, volume 3, pages 532–541, Kauai, Hawaii, 1992.

[3] S. V. Vrbsky and J. W. S. Liu. An object-oriented query processor that produces monotonically improving approximate answers. In *Proc. 7th Int. Conf. Data Eng.*, pages 472–481, Kobe, Japan, 1991.

[4] E. Dorken and S. H. Nawab. Frame–adaptive techniques for quality versus efficiency tradeoffs in STFT analysis. In *IEEE ICASSP '94*, Adelaide, April 1994.

[5] S. H. Nawab and E. Dorken. A framework for quality versus efficiency tradeoffs in STFT analysis. To appear in *IEEE Trans. Signal Processing*, April 1995.

[6] J. M. Winograd and S. H. Nawab. Incremental refinement of DFT and STFT approximations. To appear in *IEEE Signal Processing Letters*.

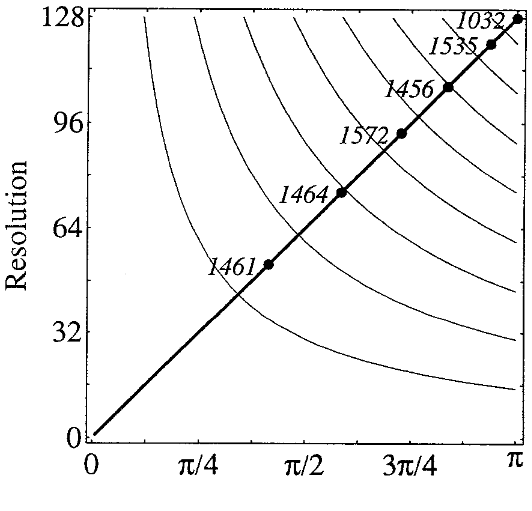

Figure 1: Refinement path in frequency coverage and frequency resolution of algorithm designed with uniform cost increments of 1500 complex additions. Each point indicates the output quality after a particular stage is completed and is labeled with its associated cost increment. Equal cost contours are shown at multiples of 1000 complex additions.

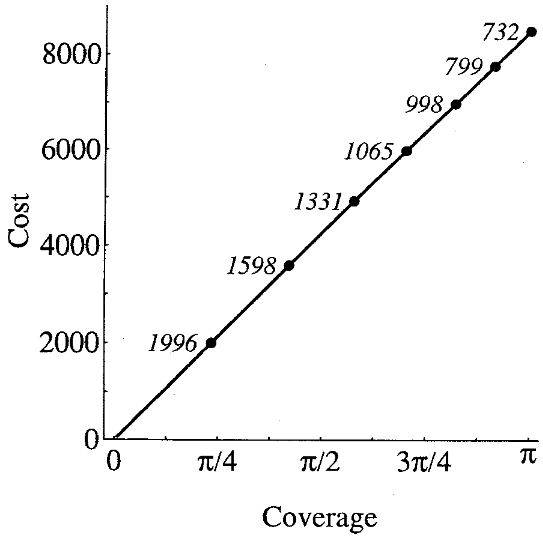

Figure 2: Refinement path in frequency coverage of algorithm designed with exponentially decreasing cost increments. Each point indicates the output quality after a particular stage is completed and is labeled with its associated cost increment in complex additions.

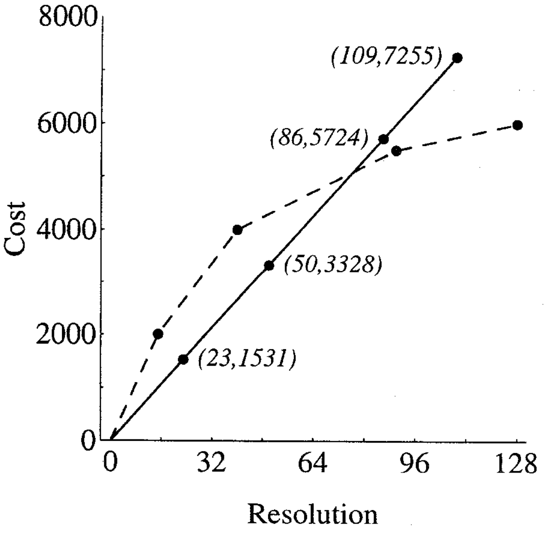

Figure 3: Refinement path in frequency resolution of algorithm (solid line) designed to approximate a given path (dashed line). Each point indicates the output quality after a particular stage is completed and is labeled with the cumulative cost incurred through that stage in complex additions.