Modular Architectures for Two-Dimensional Digital Signal Processing

ALEXANDER C. P. LOUI, STUDENT MEMBER, IEEE,
ANASTASIOS N. VENETSANOPOULOS, SENIOR MEMBER, IEEE,
AND CHRYSOSTOMOS L. NIKIAS, MEMBER, IEEE

Abstract—This paper addresses the implementation aspects of highly modular two-dimensional (2-D) digital filters. The idea of matrix decomposition [1] is used to provide increased parallelism and regularity. Four different structures namely the transversal, the distributed arithmetic, the stored product and the systolic array, are considered. For comparison purposes, the direct implementation is included. Analysis is performed and comparisons are made in terms of hardware cost, cycle time, finite register length effects and regularity. From the analysis, it is found that the systolic array structure seems to offer the best compromise among the various conflicting figures of merit. The distributed arithmetic structure is shown to be superior in the case of low-order filters, while the stored product structure is preferable for high-order filters. The inherent parallelism and high throughput rate of these structures make them very suitable for real-time image processing applications.

I. INTRODUCTION

THE USE OF high speed two-dimensional (2-D) digital filters for processing sampled 2-D data in real-time has found important applications in many areas. Some applications which benefit from real-time digital image processing include [2] biomedical applications, such as angiocardiology, radiology and tomography; image transformation and video conferencing; robotics; and air reconnaissance. The computationally intensive nature of 2-D digital filters render them well suited for VLSI implementation. However, special purpose VLSI systems should have a design cost which is relatively small in order to be competitive with a general purpose system [3]. Also, in VLSI design, the routing cost normally dominates the power and area required for the implementation. As a consequence, modular structures that can be constructed using only a few simple standard cells with regular and simple communication and control are highly desirable. The modularity feature also implies that system cost can be adjusted to match system performance. In this paper, the idea of matrix decomposition [1], [4]–[6] is used for the realization of 2-D digital filters. Our objective here is to find the optimum structure which provides the best compromise among the many conflicting criteria such as cost,

speed and roundoff error. The comparisons can also be used as figures of merit in choosing one particular structure over the others.

The organization of this paper is as follows. In Section II, the realization of 2-D digital filters based on the singular value-lower upper (SV-LU) decomposition realization [5] is considered. Some realistic figures of merit for hardware implementation are introduced. In Section III, four structures for implementing the decomposed form are described. These are the transversal, the distributed arithmetic, the stored product and the systolic array. Analysis and comparisons of the proposed structures, are presented in Section IV, in terms of hardware cost, cycle time (reciprocal of throughput rate), finite register length effects and suitability for VLSI implementation. Finally, conclusions are drawn in Section V.

II. DECOMPOSITION REALIZATION OF TWO-DIMENSIONAL DIGITAL FILTERS

A. Objective

In this section, we summarize the decomposition technique [1], [5] for the realization of high speed 2-D digital filters. This approach leads to architectures which possess a highly parallel structure, suitable for VLSI implementation.

A causal 2-D FIR digital filter has the transfer function

$$H(z_1, z_2) = \sum_{i=1}^M \sum_{j=1}^N h_{i,j} z_1^{-i+1} z_2^{-j+1} \quad (1)$$

where z_1^{-1} and z_2^{-1} represent a space delay along a column and a space delay along a row, respectively. In matrix notation, (1) can be represented by

$$H(z_1, z_2) = \mathbf{Z}_1^T \mathbf{H} \mathbf{Z}_2 \quad (2)$$

where

$$\mathbf{Z}_1 = [1z_1^{-1} \cdots z_1^{-M+1}]^T \quad (3)$$

and

$$\mathbf{Z}_2 = [1z_2^{-1} \cdots z_2^{-N+1}]^T \quad (4)$$

and $\mathbf{H} = \{h_{i,j}\}$ is an $M \times N$ matrix. When the size of the matrix $\mathbf{H}$ is large, the 2-D convolution becomes computationally inefficient. The computational rate can be in-

Manuscript received July 2, 1986; revised January 29, 1987. This paper was recommended by Associate Editor, P. K. Rajan.

A. C. P. Loui and A. N. Venetsanopoulos are with the Department of Electrical Engineering, University of Toronto, Toronto, Ont., Canada M5S 1A4.

C. L. Nikias is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115.
IEEE Log Number 8717746.

creased by parallel processing and/or pipelining or arithmetic operations. The decomposition of the system matrix, $\mathbf{H}$ provides the parallelism required for high speed implementation. Specifically, the matrix $\mathbf{H}$ can be expressed in terms of a finite and converging sum of other block matrices, such that

$$H(z_1, z_2) = \sum_{i=1}^p H_i(z_1, z_2) \quad (5)$$

where p is the rank of the matrix $\mathbf{H}$ and $H_i(z_1, z_2)$ is a 2-D polynomial. Further simplification is possible as the polynomials $H_i(z_1, z_2)$ can always be decomposed into a product of 1-D polynomials, each one of which is a function of one variable only [1], i.e.,

$$H(z_1, z_2) = \sum_{i=1}^p a_i(z_1)b_i(z_2) \quad (6)$$

where

$$a_i(z_1) = a_{1,i} + a_{2,i}z_1^{-1} + \dots + a_{M,i}z_1^{-M+1} \quad (7a)$$

and

$$b_i(z_2) = b_{1,i} + b_{2,i}z_2^{-1} + \dots + b_{N,i}z_2^{-N+1}. \quad (7b)$$

The characteristic of the polynomials $a_i(z_1)$ and $b_i(z_2)$ depends on the type of decomposition used. Common types are singular-value (SV), lower upper triangular (LU), and Jordan decompositions. In this paper, a combination of SV and LU decompositions is used [4]. The SV decomposition is the optimum decomposition in the sense that it requires the least number of separable stages to achieve a given mean-squared error. The LU decomposition can then be used to realize a given structure with minimum amount of hardware. The number of stages required, k to achieve a satisfactory approximation to the original matrix $\mathbf{H}$, can be determined by the normalized error energy e_k which is defined as [15]

$$e_k = 1 - \frac{\sigma_1 + \sigma_2 + \dots + \sigma_k}{\sigma_1 + \sigma_2 + \dots + \sigma_p}, \quad k=1, 2, \dots, p \quad (8)$$

where σ_i are the singular values of $\mathbf{H}$ and the error e_k decreases as k increases and becomes zero when $k=p$. The realization procedure can be summarized as follows:

- i) apply singular-value decomposition on $\mathbf{H}$ to obtain $\hat{\mathbf{H}}$,
- ii) realize $\hat{\mathbf{H}}$ using the LU decomposition.

In this paper, the implementation aspects of the SV-LU decomposition realization will be considered. In particular, the implementation of the 2-D FIR filters will be emphasized. Extension to IIR filters can be easily obtained by applying the same procedures to the recursive part. In this paper, we will concentrate our analysis on the following structures:

- 1) Transversal structure;
- 2) Distributed arithmetic structure;
- 3) Stored product structure;
- 4) Systolic array structure.

In addition, the direct implementation (no decomposition) of 2-D FIR filter is included for comparison purposes.

B. Figures of Merit

The results of the analysis provide the designer with knowledge of the tradeoffs among the criteria that are important for VLSI implementation. The criteria that will be employed are:

- a) Component cost;
- b) Cycle time;
- c) Finite register length effect;
- d) Regularity.

The definition of each of the above criteria is now given.

(a) Component Cost:

In order to provide a fair comparison between the different structures, a cost function derived in [7] will be adopted here for component cost evaluation. This cost function has the advantage of separating the technology dependence from the algorithm efficiency and it can be applied to different types of implementations. This cost function is defined as

$$C_s = G + K_{\text{RAM}}B_{\text{RAM}} + K_{\text{ROM}}B_{\text{ROM}} \quad (9)$$

where

- G total number of gates in the network,
- $B_{\text{RAM}}, B_{\text{ROM}}$ number of bits of RAM and ROM,
- $K_{\text{RAM}}, K_{\text{ROM}}$ relative cost of one bit of RAM and ROM to one logic gate.

Thus the overall component cost is

$$C = C_s P_{\text{tech}} f \quad (10)$$

where f is the speed of the logic gates in hertz and C is measured in watts or silicon area if P_{tech} is the technology price in units of watts per gate hertz or silicon area per gate hertz. It should be noted that the cost function C_s does not take into account the interconnection or communication cost of the system.

The basic elements for implementing a 2-D digital filter consist of ALU's, multipliers, RAM's, ROM's, and shift registers/latches. General formulas for evaluating the number of logic gates for each of the arithmetic units are given in [7] and they are outlined as follows:

- a) An ALU is assumed to have a gate complexity of $16b$ logic gates, where b is the input wordlength in bits, i.e.,

$$G_a = 16b. \quad (11)$$

- b) A multiplier with b -bit inputs has a $2b$ -bit output. The logic gate complexity is assumed to be

$$G_m = 10b^2 + 30b + 80. \quad (12)$$

- c) The relative cost in terms of power consumption of one bit of memory to one logic gate is assumed to be 0.3 for RAM and 0.03 for ROM, i.e.,

$$K_{\text{RAM}} = 0.3, \quad K_{\text{ROM}} = 0.03 \quad (13)$$

- d) A single bit shift register is assumed to have a logic gate complexity of

$$G_{\text{sr}} = 10b. \quad (14)$$

(b) Cycle Time:

The cycle time is defined as the time lag between two consecutive input samples of a system. Hence, the throughput rate of a particular structure is just the reciprocal of the cycle time. In terms of the cycle times of the arithmetic units, the cycle time of a structure, T_c is defined as

$$T_c = N_m T_m + N_a T_a + N_{RAM} T_{RAM} + N_{ROM} T_{ROM} + N_{sr} T_{sr} \quad (15)$$

where

N_m	number of multiplications,
N_a	number of additions,
N_{RAM}, N_{ROM}	number of RAM and ROM accesses,
N_{sr}	number of shift register delays,
T_m	multiplication time,
T_a	addition time,
T_{RAM}, T_{ROM}	RAM and ROM access times,
T_{sr}	shift register propagation delay.

Similar to the cost function, the following cycle time assumptions are made about the arithmetic units [7]:

a) The ALU is assumed to have an equivalent 5 gate delays (t) addition time with “carry look-ahead”, i.e.,

$$T_a = 5t. \quad (16)$$

b) The multiplication time is assumed to be

$$T_m = 4bt. \quad (17)$$

c) The RAM and ROM access times are approximated given by

$$T_{RAM} = [5 + \log_2(\text{number of addresses})] t_{RAM} \quad (18a)$$

$$T_{ROM} = [5 + \log_2(\text{number of addresses})] t_{ROM} \quad (18b)$$

where t_{RAM} and t_{ROM} are the respective gates delays for the RAM and ROM.

d) The single bit shift register is assumed to have a delay of

$$T_{sr} = 5t. \quad (19)$$

(c) Finite Register Length Effect:

The output signal quality due to the finite register length effect is always one of the most important factors affecting a particular implementation. Longer registers in general will improve the signal quality at the expense of increasing hardware cost and cycle time. Hence, it is important to understand the different trade offs between these criteria. To simplify the analysis, only the roundoff accumulation error will be considered, as it is the major contributor to the total roundoff error [8]. The roundoff accumulation error consists mainly the errors due to rounding or truncation after multiplications and additions. In addition, two's complement fixed-point arithmetic is assumed to be used in all the implementations.

(d) Regularity:

In the design of VLSI system, modularity and localized data transfer are two main factors affecting both the architecture and the total hardware cost of the system. It

Fig. 1. SV-LU decomposition realization for 2-D FIR digital filter.

should be noted that the component cost described in II-B(a). does not reflect the total hardware cost as communication and routing costs in VLSI system are very expensive in terms of area, power and time consumption. In addition, the utilization of modular structure implies a few types of standard cells can be used repetitively in a regular fashion for the construction of the system. Hence, the cost-effectiveness of a design is strongly related to the regularity of the architecture.

III. STRUCTURES FOR THE DECOMPOSED FILTERS

In this paper, the image is assumed to be processed in a column-wise way. Hence, m and n represent the row and column indices respectively. The image size is assumed to be $L \times L$ pixels. In addition, the input and output samples are assumed to have a precision of b_1 bits and the intermediate results will have a precision of b_2 bits. Using the SV-LU decomposition approach, (7) can be rewritten as follows:

$$u_i(z_1) = u_{i,i} z_1^{\lambda_i} + u_{i,i+1} z_1^{\lambda_{i+1}} + \dots + u_{M,i} z_1^{\lambda_M} \quad (20a)$$

and

$$v_i(z_2) = v_{i,i} z_2^{\mu_i} + v_{i,i+1} z_2^{\mu_{i+1}} + \dots + v_{i,N} z_2^{\mu_N} \quad (20b)$$

where $i = 1, \dots, k$. The superscripts λ_i and μ_i take on the appropriate values from the sets $[0, -1, \dots, -M+1]$ and $[0, -1, \dots, -N+1]$, respectively, depending on whether the rows and columns of the truncated matrix $\mathbf{H}$ are needed to be permuted. The structure described by (20) is depicted in Fig. 1. The number of stages in the truncated matrix $\mathbf{H}$ is assumed to be k . Let's denote the output of the row filter at stage l be

$$w_{m,n-\mu_l}^{l,s} = \sum_{i=l}^M u_{i,l} x_{m-\lambda_i,n} \quad (21)$$

and the output of the column filter at stage l be

$$y_{m,n}^l = \sum_{j=l}^N v_{l,j} w_{m,n-\mu_j}^l. \quad (22)$$

Then the final output is given by

$$y_{m,n} = \sum_{l=1}^k y_{m,n}^l. \quad (23)$$

In the sequel, we will present the four different structures for the implementation of the 1-D polynomials $u_i(z_1)$ (row filter) and $v_i(z_2)$ (column filter).

Fig. 2. Row and column filters of Stage l of the transversal structure.

A. Transversal Structure

The transversal structure is the most common structure that is being used for the implementation of digital filters. Its popularity is mainly due to its simple relation to the impulse response of the system. A schematic block diagram of this structure is depicted in Fig. 2. The total number of gates or cost function of the transversal structure is given by

$$C_{tr} = [10(M + N - k + 1)k]b_2^2 + \left[30(M + N - k + 1)k + 16 \sum_{i=0}^{k-1} (M + N - 2(i + 1)) + 16(k - 1) + 10Lk \left(N - \frac{(k + 1)}{2} \right) + 10k \right] b_2 + \left[10k \left(M - \frac{(k + 1)}{2} \right) \right] b_1 + [80(M + N - k + 1)k]. \quad (24)$$

The cycle time assuming the pipelining of the output stage is given by

$$T_{tr} = [8b_2 + 5\lceil \log_2 M \rceil + 5\lceil \log_2 N \rceil + 10]t \quad (25)$$

where $\lceil x \rceil$ denotes the smallest integer greater than x .

The roundoff accumulation error in this case will consist of the quantization error due to multiplications in the column filters; the propagation of the quantization error in the row filters through the column filters, as well as the output quantization error. Assuming the error due to each stage is uncorrelated, the variance of the output roundoff error is given by [9]

$$\sigma_{tr}^2 = \sum_{l=1}^k \left[(M - l + 1) \left(\frac{2^{-2b_2}}{3} \right) + \sum_{j=l}^N v_{l,j}^2 + (N - l + 1) \left(\frac{2^{-2b_2}}{3} \right) \right] + \frac{1}{3} (2^{-2b_1} - 2^{-2b_2}) \quad (26)$$

where $v_{l,j}$'s are the coefficients of the column filters. The details of the derivation is given in Appendix A.

B. Distributed Arithmetic Structure

The distributed arithmetic structure [10] has gained a lot of attention in recent years due to the fact that it requires

no multipliers in its implementation. The hardware implementation of a 2-D second-order recursive distributed filter has been reported in [11], [12]. In this section, the decomposed form is investigated.

Let all the input and output signals be represented in two's complement code bounded between ± 1 . Except for the input and output samples which have b_1 bits of precision, all the other signals will have a precision of b_2 bits. The input $x_{m,n}$ and the respective outputs of each of the row and column filters, $w_{m,n}^l$ and $y_{m,n}^l$ can be written as

$$x_{m,n} = -x_{m,n}^0 + \sum_{s=1}^{b_1-1} (x_{m,n}^s)2^{-s} \quad (27)$$

$$w_{m,n}^l = -w_{m,n}^{l,0} + \sum_{s=1}^{b_2-1} (w_{m,n}^{l,s})2^{-s} \quad (28)$$

$$y_{m,n}^l = -y_{m,n}^{l,0} + \sum_{s=1}^{b_2-1} (y_{m,n}^{l,s})2^{-s}. \quad (29)$$

In addition, the partial outputs $w_{m,n-\mu_j}^{l,s}$ and $y_{m,n}^{l,s}$ are given by (21) and (22), respectively. Substituting (27) into (21) and rearranging the orders of summation yields the following equation:

$$w_{m,n}^l = \sum_{s=1}^{b_1-1} \left[\sum_{i=l}^M (u_{i,l} x_{m-\lambda_i,n}^s) 2^{-s} \right] - \sum_{i=l}^M u_{i,l} x_{m-\lambda_i,n}^0. \quad (30)$$

Equation (30) can be simplified by defining a function $p_l^s(\star)$, such that

$$p_l^s(x_{m-\lambda_l,n}^s, x_{m-\lambda_{l+1},n}^s, \dots, x_{m-\lambda_M,n}^s) = u_{l,l} x_{m-\lambda_l,n}^s + u_{l+1,l} x_{m-\lambda_{l+1},n}^s + \dots + u_{M,l} x_{m-\lambda_M,n}^s. \quad (31)$$

Replacing the bracketed term of (30) with (31) yields

$$w_{m,n}^l = \sum_{s=1}^{b_1-1} p_l^s(\star) 2^{-s} - p_l^0(\star). \quad (32)$$

Similarly, for the column filters, substituting (28) into (22) and reordering the summations, (33) is obtained:

$$y_{m,n}^l = \sum_{s=1}^{b_2-1} \left[\sum_{j=l}^N (v_{l,j} w_{m,n-\mu_j}^{l,s}) 2^{-s} \right] - \sum_{j=l}^N v_{l,j} w_{m,n-\mu_j}^{l,0}. \quad (33)$$

Fig. 3. Row and column filters of Stage l of the distributed arithmetic structure.

Fig. 4. Memory address partition method for memory reduction.

Defining another function $q_l^s(*)$ such that

$$q_l^s(w_{m,n-\mu_l}^{l,s}, w_{m,n-\mu_{l+1}}^{l,s}, \dots, w_{m,n-\mu_N}^{l,s}) = v_{l,l} w_{m,n-\mu_l}^{l,s} + v_{l,l+1} w_{m,n-\mu_{l+1}}^{l,s} + \dots + v_{l,N} w_{m,n-\mu_N}^{l,s} \quad (34)$$

and substituting it into (33) yields

$$y_{m,n}^l = \sum_{s=1}^{b_2-1} q_l^s(*) 2^{-s} - q_l^0(*). \quad (35)$$

Finally, the output $y_{m,n}$ is obtained from (23). Equations (32) and (35) are the basic equations describing the distributed arithmetic realization. The outcomes of the functions $p_l^s(*)$ and $q_l^s(*)$ are stored into memories and the arguments of the functions will form the address to the appropriate memory. The bit shift 2^{-s} associated with the functions $p_l^s(*)$ and $q_l^s(*)$ can be preprogrammed into the memory to reduce the hardware complexity. An illustration of one stage of this structure is shown in Fig. 3. The effect of the LU decomposition is to reduce the amount of memory size in the successive stages of the realization.

One major drawback of this approach is that as the order of the filter increases, the amount of memory requirement increases as $(b_1 2^M + b_2 2^N) \times b_2$. Hence, for high order filter, the large memory requirement can make this approach impractical. One way to reduce the amount of memory is to use an address partition technique [13], which trades off memory with extra additions. With this modification, the effective memory size decreases while number of memory addresses remains the same. For example, $2^{32} \times 16$ bits of memory can be implemented using two $2^{16} \times 16$ bits of memory plus one 16 bits adder as shown in

Fig. 4. In this structure, the address partition method is used whenever the memory size becomes unmanageable.

The major contributor to the cost function are the memory arrays. The amount of memory depends on the arithmetic precision, as well as the order of the filter. The second major component of the cost function are the ALU's. The number of ALU's in the case depends mainly on arithmetic precision. The cost function for the distributed arithmetic structure with address partition is given as follows:

$$C_{da} = \left[16k(p-1) + 0.3 \sum_{l=1}^k D_l \right] b_2^2 + \left[16k(p-1) + 0.3 \sum_{l=1}^k C_l \right] b_1 b_2 + \left[10Lk \left(N - \frac{(k+1)}{2} \right) - 6k - 16 \right] b_2 + \left[10k \left(M - \frac{(k-1)}{2} \right) \right] b_1 \quad (36)$$

where $p \geq 1$ is the number of memory address partitions ($p=1$ implies no memory partition is used). The term C_l represents the amount of memory per input bit ($x_{m,n}^s$) in stage l for the row filter. Similarly, D_l is the amount of memory per input bit ($w_{m,n}^{l,s}$) in stage l for the column filter.

The cycle time assuming the use address partition and a pipelined output summation stage is given by

$$T_{da} = [20 + M_p + N_p + 5([\log_2 b_1] + [\log_2 b_2] + [\log_2 p])] t \quad (37)$$

where M_p and N_p are the largest memory banks after partition in the row and column filters, respectively. Also, the gate delays for the RAM, t_{RAM} is assumed to be the same as the logic gate delay, t .

The roundoff accumulation error in this case consists of the quantization due to finite precision of the memory outputs, the propagation of the row filters' errors through the column filters, as well as the output quantization error. The variance of the final output error is shown to be given

Fig. 5. Row and column filters of Stage l of the stored product structure.

by

$$\begin{aligned} \sigma_{da}^2 = & \sum_{l=1}^k \left[\frac{2^{-2b_2}}{3} \left(b_1 + \frac{4}{3}(1-2^{-2b_1})(p^2-1) \right) \sum_{j=l}^N v_{l,j}^2 \right. \\ & \left. + \frac{2^{-2b_2}}{3} \left(b_2 + \frac{4}{3}(1-2^{-2b_2})(p^2-1) \right) \right] \\ & + \frac{1}{3}(2^{-2b_1} - 2^{-2b_2}). \end{aligned} \quad (38)$$

For $p=1$, (38) reduces to

$$\begin{aligned} \sigma_{da}^2 = & \sum_{l=1}^k \left[b_1 \left(\frac{2^{-2b_2}}{3} \right) \sum_{j=l}^N v_{l,j}^2 + b_2 \left(\frac{2^{-2b_2}}{3} \right) \right] \\ & + \frac{1}{3}(2^{-2b_1} - 2^{-2b_2}). \end{aligned} \quad (39)$$

The details of the derivation is given in Appendix B.

C. Stored Product Structure

The stored product structure is another structure which does not require the use of digital multipliers. In this approach, all possible combinations of the product of the input signal and the filter coefficients are stored in memory. Hence, each coefficient requires $2^{b_1} \times b_2$ or $2^{b_2} \times b_1$ bits of memory. The total amount of memory depends on the number of coefficients and hence, the order of the filter. A possible realization of the stored product structure is shown in Figure 5 [13]. In this case, only a single stage of the decomposed structure is shown. In this configuration, all the RAMs are connected at a common node and are addressed by the same input signal.

Similar to the distributed arithmetic structure, the major factors in the cost function are the memory arrays and the adders. The amount of memory required depends on the order of the filter as well as the input word length and the arithmetic precision. Again address partition is used to reduce the memory size when needed. The cost function for the stored product arithmetic structure with address

partition is given as follows:

$$\begin{aligned} C_{sp} = & \left[0.3k \left(\left(M - \frac{(k-1)}{2} \right) p_1 2^{b_1/p_1} \right. \right. \\ & \left. \left. + \left(N - \frac{(k-1)}{2} \right) p_2 2^{b_2/p_2} \right) \right. \\ & \left. + 16 \left(\sum_{i=0}^{k-1} (M + N - 2(i+1)) + k - 1 \right) \right. \\ & \left. + 10k \left(M - \frac{k+1}{2} \right) \right. \\ & \left. + 10Lk \left(N - \frac{k+1}{2} \right) + 10k \right] b_2 \end{aligned} \quad (40)$$

where $p_1 \geq 1$ and $p_2 \geq 1$ are the number of memory address partitions in the row filter and column filter respectively. To simplify the analysis, it is assumed that p_1 divides b_1 and p_2 divides b_2 .

The cycle time is determined as follows. From Fig. 5, the output of the row filter is obtained after one memory access, one addition and one shift register delay. A similar amount of time is required for the signal at the input of the column filter to appear at the output of the column filter. Hence, the cycle time of this structure assuming the pipelining of the output stage is given by

$$T_{sp} = \left[30 + \frac{b_1}{p_1} + \frac{b_2}{p_2} + \lceil \log_2 p_1 \rceil + \lceil \log_2 p_2 \rceil \right] t. \quad (41)$$

One advantage of using the stored product structure is that coefficient quantization error is eliminated. Since we only consider arithmetic roundoff error here, the increase of accuracy will not be obvious. Nevertheless, the variance of the final output roundoff error is given by

$$\begin{aligned} \sigma_{sp}^2 = & \sum_{l=1}^k \left[p_1^2 (M - l + 1) \left(\frac{2^{-2b_2}}{3} \right) \sum_{j=l}^N v_{l,j}^2 \right. \\ & \left. + p_2^2 (N - l + 1) \left(\frac{2^{-2b_2}}{3} \right) \right] + \frac{1}{3}(2^{-2b_1} - 2^{-2b_2}). \end{aligned} \quad (42)$$

Fig. 6. Block diagram of the decomposed implementation using the systolic array structure.

Note: All shift registers are $1 \times b_1$ or $1 \times b_2$ bits except for the one indicated by L which is $L \times b_2$ bits.

Fig. 7. Systolic implementation of a single stage first-order 2-D decomposed filter.

The details of the derivation of (42) is given in Appendix C.

D. Systolic Array Structure

In this section, a systolic array [14] structure for the implementation of the row and column filters of the decomposed realization is presented. Modularity and pipelining are the main features of this structure. This implies that the realization of Fig. 1 can be constructed using a few standard cells which can be locally interconnected. Hence, except for the global system clock line, all data transfer and communications can be done in the local neighborhood of the individual cell. In addition, full utilization of all the cells can be achieved.

A block diagram of the decomposed implementation using the systolic array structure is depicted in Fig. 6. The number of systolic blocks or cells in the row and column filters in stage l are equal to $M-l+1$ and $N-l+1$ respectively. For the row filters, the standard cell consists of one multiplier, one ALU and single bit registers. The standard cell of column filters have similar configurations, except that one of the registers is L bits long in order to store the previous column of the image data. The regularity of the row and column standard cells enables the filter's order as well as the number of stages to be expanded by simply adding more cells to the original struc-

ture. The implementation of a first-order 2-D FIR filter using the proposed structure, is shown in Fig. 7. It is assumed that the system matrix H is approximated using only one stage. Hence, according to (21) and for $k=1$, we have

$$w_{m,n}^1 = \sum_{i=1}^2 u_{i,1} x_{m-i+1,n} \quad (43)$$

and from (22) and (23) the output is given by

$$\begin{aligned} y_{m,n} &= y_{m,n}^1 \\ &= \sum_{j=1}^2 v_{1,j} w_{m,n-j+1}^1 \end{aligned} \quad (44)$$

where λ_i and μ_j are assumed to be equal to $i-1$ and $j-1$, respectively. In order to illustrate how the signal processing is done in this fully pipelined structure, let's consider the corresponding space-time diagram of Fig. 8. The space-time diagram indicates the flow of the input image sample at different points in time and location. Let's determine the value of $y_{2,2}$. According to (44),

$$y_{2,2} = v_{1,1} w_{2,2}^1 + v_{1,2} w_{2,1}^1 \quad (45)$$

and from (43),

$$w_{2,2}^1 = u_{1,1} x_{2,2} + u_{2,1} x_{1,2} \quad (46a)$$

Fig. 8. Space-time diagram for the systolic implementation of the single stage first-order 2-D decomposed filter.

and

$$w_{2,1}^1 = u_{1,1}x_{2,1} + u_{2,1}x_{1,1}. \quad (46b)$$

Hence,

$$y_{2,2} = u_{1,1}v_{1,1}x_{2,2} + u_{2,1}v_{1,1}x_{1,2} + u_{1,1}v_{1,2}x_{2,1} + u_{2,1}v_{1,2}x_{1,1}. \quad (47)$$

From Fig. 8, it can be verified that the correct value of $y_{2,2}$ is obtained at time T_9 in position P_{10} , which corresponds to the output register. Notice that the register in position P_7 is needed to ensure data transfer from the row filter is synchronized with the data flow in the column filter. This data transfer occurs between positions P_4 and P_5 and is indicated by the upward arrow in the middle of Fig. 8. As seen from Fig. 8, all the cells are busy at each cycle and one output is produced at every cycle. The cycle time is determined by the arithmetic unit which requires the longest time to accomplish an operation.

The cost function of this structure is shown to be [9]

$$G_{sa} = [10(M + N - k + 1)k]b_2^2 + \left[10k^2 + 36(k - 1) + 66k(M + N - k + 1) + 10k \left(N - \frac{(k - 1)}{2} \right) (L + 1) - 10kL \right] b_2 + [10k(2M - k)]b_1. \quad (48)$$

The cycle time of the systolic array structure is determined by the multiplication time plus a shift register delay, hence,

$$T_{sa} = [4b_2 + 5]t. \quad (49)$$

Fig. 9. Direct form implementation of a 2-D second-order FIR filter.

The output roundoff error of this structure is the same as the one for the transversal structure and the error variance is given by (26).

Finally, for the purpose of comparison, a direct form implementation is included. The structure of the direct implementation is shown in Fig. 9 for $M = N = 3$. The cost

Fig. 10. Cost function versus filter order up to 256×256 for $b_1 = 8$, $b_2 = 16$, and $k = 4$.

function and cycle time of the direct form implementation are given by

$$C_{df} = [10MN]b_2^2 + [46MN - 16]b_2 + [10(L(N-1) + N(M-1))]b_1 + 80MN \quad (50)$$

and

$$T_{df} = [4b_2 + 5[\log_2(MN)] + 5]t. \quad (51)$$

The roundoff accumulation error in this case is mainly due to the rounding or truncation of the multiplications, as well as the rounding or truncation of the output sample. The output error variance is shown to be given by

$$\sigma_{df}^2 = \frac{1}{3} [MN(2^{-2b_2}) + (2^{-2b_1} - 2^{-2b_2})]. \quad (52)$$

The details of the derivation is shown in Appendix D.

IV. ANALYSIS AND COMPARISONS

In this section, comparisons will be made between the various structures presented in the preceding sections. Fig. 10 shows the behaviors of each of the cost functions as a function of the filter's order. In Fig. 10, the arithmetic precisions b_1 and b_2 are equal to 8 and 16, respectively. The number of stages used for the analysis is $k = 4$. The input image size is assumed to be 512×512 pixels. Also, for the distributed arithmetic structure, memory address partition is used for high order filters. The partition is done in unit of 256K bits of memory. From Fig. 10, the following observations can be made:

i) The cost of the systolic array structure is only slightly higher than the transversal structure.

ii) The cost of the distributed arithmetic structure is lower than that of the systolic array and the transversal structures for low order filters.

iii) The cost of the stored product structure with no address partitions is lower than that of the distributed arithmetic with address partitions for high order filters.

iv) The direct form implementation has the lowest cost function for filter order less than about 2^6 .

The characteristics of the cycle time as a function of the filter's order for $b_1 = 8$ and $b_2 = 16$ are depicted in Fig. 11. On the basis of these graphs, the following observations can be made.

i) The stored product structure has the lowest cycle time followed by the systolic array structure.

ii) The cycle time of the distributed arithmetic structure is lower than the systolic array structure for low-order filters.

iii) The cycle time or throughput rate of the systolic array and the stored product structures is independent of the filter's order.

The behavior of each of the error variances is analyzed using a gravity filter taken from [15]. The impulse response of this digital gravity filter has a size of 21×21 and is given by

$$h_{i,j} = \frac{\pi}{2} \frac{h}{(h^2 + i^2 + j^2)^{3/2}} \quad (53)$$

where

$$h = 1.5 \\ i = -10, -9, \dots, 0, \dots, 10$$

Fig. 11. Cycle time versus filter order for $b_1 = 8$, $b_2 = 16$, and $k = 4$.

TABLE I
OUTPUT ERROR VARIANCES OF THE FINITE ARITHMETIC
IMPLEMENTATIONS OF THE GRAVITY FILTER

Structure	Output Error Variance (σ^2) ($k=5$)			
	$b_1=8$		$b_1=16$	
	$b_2=8(\times 10^{-3})$	$b_2=16(\times 10^{-5})$	$b_2=16(\times 10^{-4})$	$b_2=32(\times 10^{-9})$
Direct Form	2.24304200	.51204115	3.42261050	.77610223
Transversal	.48319505	.50935584	.73729714	.77610216
Distributed Arithmetic	.30517427	.50939464	.77610224	.77610218
Stored Product	.48319505	.50935584	2.9491885	.77610242
Systolic Array	.48319505	.50935584	.73729714	.77610216

TABLE II
COMPARISON BETWEEN THE CRITERIA FOR $\sigma^2 = 0.50935584 \times 10^{-5}$
AND $b_1 = 8$ BITS FOR THE GRAVITY FILTER

Structure	$\sigma^2 = 0.50935584 \times 10^{-5}$			
	b_2	$C_s (\times 10^7)$	$T_c (t)$	$F (\times 10^9)$
Direct Form	18	.2682	122	.3272
Transversal	16	.8021	188	1.5079
Distributed Arithmetic	17	.8969	86	.7713
Stored Product	16	3.7436	54	2.022
Systolic Array	16	.8111	69	.5597

and

$$j = -10, -9, \dots, 0, \dots, 10.$$

The rank of the filter is 11 and it is found that a MSE of less than 5×10^{-8} is achieved using only $k = 5$ stages to approximate the original system matrix $\mathbf{H} = \{h_{i,j}\}$. Using the approximated matrix $\hat{\mathbf{H}}$, the variances of the five structures for $b_1 = 8$, $b_2 = 8$ and 16, as well as for $b_1 = 16$, $b_2 = 16$ and 32 are tabulated in Table I. In the case of $b_1 = 16$, address partition is assumed to be used in both the distributed arithmetic and the stored product structures. It is observed from Table I that for $b_1 = 8$, the error variances tend to be the same for all structures as the arithmetic precision b_2 increases. In the case of $b_1 = 16$, similar

results are obtained. In addition, the error variances for $b_1 = 16$ are much smaller than that for $b_1 = 8$, as expected.

In order to provide some insights on the relationship of the different criteria, a comparison of internal precision requirements, component cost and cycle time for a given output error variance is shown in Table II. A new factor F , defined as the product of C_s and T_c , is also included in this table. Hence, the smaller the factor F , the less costly and faster the structure is. From Table II, for a variance of $\sigma^2 = 0.5093558 \times 10^{-5}$, the direct form has the smallest F followed by the systolic array and the distributed arithmetic structures. Although the stored product structure has the highest throughput rate, it has the largest F . However, the use of address partitions should reduce the cost function of the stored product structure and hence, F . Finally,

TABLE III
COMPARISON BETWEEN THE VARIOUS STRUCTURES IN TERMS OF
PARALLELISM AND LOCALIZED DATA TRANSFER

Structure	Localized Data Transfer	Parallelism
Direct Form	No	No
Transversal	No	Yes
Distributed Arithmetic	No	Yes
Stored Product	No	Yes
Systolic Array	Yes	Yes

by trading off some flexibility, the cost function and the cycle time of the systolic array structure can be further reduced if the multipliers are replaced by memory look-up similar to the stored product approach.

In addition to the three criteria discussed above, it is important to consider another criterion which will provide some information regarding the suitability of each structure for efficient VLSI implementation. In VLSI implementation, regularity of the structure is of main concern. Based on the description of the structures presented in Section III, as well as Figs. 2-7 and Fig. 9, a comparison of the five structures for efficient VLSI implementation is given in Table III. From this table we observe that the

more robust to components failure than the direct form implementation. This is due to the fact that the malfunction of a particular stage will not affect the operation of other stages and hence, the output will still retain some degree of correctness.

Four different implementations of the SV-LU decomposed form have been developed and compared, in terms of cycle time, cost (number of logic gates), variance of filter output roundoff errors, and regularity. For comparison purposes, the direct form implementation has been included. From the analysis, it is found that the systolic array structure seems to offer the best compromise among the various conflicting figures of merit. The systolic array structure is also best suited for VLSI implementation due to the effective use of standard cells. The throughput rate of both the systolic array and the stored product structures is found to be independent of the filter's order. Furthermore, the throughput rate of the systolic array structure is determined only by the longest arithmetic operation between two latches and hence high operating speed can be attained. Finally, the distributed arithmetic structure is found to be more suitable for low-order filter implementation, while stored product structure is more preferable for high order filter implementation.

APPENDIX A ROUND OFF ERROR ANALYSIS OF THE TRANSVERSAL STRUCTURE

Consider the following error model:

direct form implementation is neither parallel nor its data transfer is localized. The transversal and distributed arithmetic stored product structures possess parallelism, but they still require global data communications. Finally, the systolic array structure exhibits both the parallel and local transfer properties due to the effective use of standard cells.

V. CONCLUSIONS

The implementations of 2-D digital filters based on the SV-LU decomposition realization have been presented in this paper. This approach results in structures that are

Denoting

$\delta_{m,n}^l$ the error due to the row filter ($u_l(z_1)$) alone,
 $\alpha_{m,n}^l$ the error due to the column filter ($v_l(z_2)$) alone.

The error variance at the output of stage l is given by

$$\sigma_l^2 = \sigma_{\delta_{m,n}^l}^2 \sum_{j=l}^N v_{l,j}^2 + \sigma_{\alpha_{m,n}^l}^2. \quad (\text{A.1})$$

Since the number of coefficients decreases for the row and column filters as l increases, the error variances of $\delta_{m,n}^l$ and $\alpha_{m,n}^l$ are then given by

$$\sigma_{\delta_{m,n}^l}^2 = (M - l + 1) \sigma_{e_{b_1}}^2 \quad (\text{A.2})$$

and

$$\sigma_{\alpha_{m,n}}^2 = (N-l+1)\sigma_{\epsilon_{b_2}}^2 \quad (\text{A.3})$$

and $\sigma_{\epsilon_{b_2}}^2 = 2^{-2b_2}/3$ is the usual quantization error due to multiplication. Hence, the final error variance of the structure assuming the uncorrelatedness of the individual errors is given by

$$\sigma_{\text{tr}}^2 = \sum_{l=1}^k \sigma_l^2 + \sigma_{\epsilon_{b_2, b_1}}^2 \quad (\text{A.4})$$

where $\sigma_{\epsilon_{b_2, b_1}}^2$ is the error variance due to the output rounding or truncation of b_2 bits to b_1 bits. Substituting the appropriate values for the variances, (A.4) becomes

$$\sigma_{\text{tr}}^2 = \sum_{l=1}^k \left[(M-l+1) \left(\frac{2^{-2b_2}}{3} \right) \cdot \sum_{j=l}^N v_{l,j}^2 + (N-l+1) \left(\frac{2^{-2b_2}}{3} \right) \right] + \frac{1}{3} (2^{-2b_1} - 2^{-2b_2}). \quad (\text{A.5})$$

APPENDIX B ROUND OFF ERROR ANALYSIS OF THE DISTRIBUTED ARITHMETIC STRUCTURE

Row Filters:

The output signal $w_{m,n}^l$ from the infinite length register row filter is given by

$$w_{m,n}^l = \sum_{s=1}^{b_1-1} p_l^s (x_{m-\lambda_l, n}^s, x_{m-\lambda_{l+1}, n}^s, \dots, x_{m-\lambda_M, n}^s) 2^{-s} - p_l^0 (x_{m-\lambda_l, n}^0, x_{m-\lambda_{l+1}, n}^0, \dots, x_{m-\lambda_M, n}^0). \quad (\text{B.1})$$

The rounded value of $p_l^s(\cdot)$ due to finite word length of the memory is given by

$$\tilde{p}_l^s (x_{m-\lambda_l, n}^s, x_{m-\lambda_{l+1}, n}^s, \dots, x_{m-\lambda_M, n}^s) = p_l^s (x_{m-\lambda_l, n}^s, x_{m-\lambda_{l+1}, n}^s, \dots, x_{m-\lambda_M, n}^s) + p \eta_{m,n}^s \quad (\text{B.2})$$

where $\eta_{m,n}^s$ is due to roundoff and $p \geq 1$ is the number of the address partitions. The rounded output is represented by

$$v_{m,n}^l = \sum_{s=1}^{b_1-1} \left[\tilde{p}_l^s (x_{m-\lambda_l, n}^s, \dots, x_{m-\lambda_M, n}^s) 2^{-s} + \epsilon^{b_2+s, b_2} \right] - \tilde{p}_l^0 (x_{m-\lambda_l, n}^0, \dots, x_{m-\lambda_M, n}^0) \quad (\text{B.3})$$

where ϵ^{b_2+s, b_2} is the quantization error when $\tilde{p}_l^s(\cdot) 2^{-s}$ is rounded from (b_2+s) to b_2 bit. Let $\delta_{m,n}^l$ be the accumulated roundoff error at the output of the row filter in stage l , we have

$$\delta_{m,n}^l = w_{m,n}^l - v_{m,n}^l = - \sum_{s=1}^{b_1-1} (p \eta_{m,n}^s 2^{-s} + \epsilon^{b_2+s, b_2}) + \eta_{m,n}^0. \quad (\text{B.4})$$

Column Filters:

Using similar analysis as in the row filters, the accumulated roundoff error at the output of the column filter in stage l is given by

$$\alpha_{m,n}^l = - \sum_{s=1}^{b_2-1} (\gamma_{m,n}^s 2^{-s} + \epsilon^{b_2+s, b_2}) + \gamma_{m,n}^0 \quad (\text{B.5})$$

where $\gamma_{m,n}^s$ is roundoff error at the output of the column filter's memory and ϵ^{b_2+s, b_2} is the same as defined before.

The variance of the output roundoff error of the row filter is given by

$$\sigma_{\delta_{m,n}^l}^2 = E[(\delta_{m,n}^l)^2] - E^2[\delta_{m,n}^l]. \quad (\text{B.5})$$

Assume all errors are uncorrelated, (B.5) reduced to

$$\sigma_{\delta_{m,n}^l}^2 = \sum_{s=0}^{b_1-1} [p^2 \sigma_{\eta_{m,n}^s}^2 2^{-2s} + \sigma_{\epsilon^{b_2+s, b_2}}^2]. \quad (\text{B.6})$$

Similarly, the variance of the output roundoff error due to the column filter only is given by

$$\alpha_{m,n}^l = \sum_{s=0}^{b_2-1} [p^2 \sigma_{\gamma_{m,n}^s}^2 2^{-2s} + \sigma_{\epsilon^{b_2+s, b_2}}^2]. \quad (\text{B.7})$$

Hence, substituting (B.6) and (B.7) into (A.4) and with the various variances defined as

$$\sigma_{\gamma_{m,n}^s}^2 = \frac{2^{-2b_2}}{3} \quad (\text{B.8a})$$

$$\sigma_{\eta_{m,n}^s}^2 = \frac{2^{-2b_2}}{3} \quad (\text{B.8b})$$

and

$$\sigma_{\epsilon^{b_1+s, b_2}}^2 = \frac{1}{3} (2^{-2b_2} - 2^{-2(b_2+s)}) \quad (\text{B.8c})$$

the final output error variance is given by

$$\sigma_{da}^2 = \sum_{l=1}^k \left[\frac{2^{-2b_2}}{3} \left(b_1 + \frac{4}{3} (1 - 2^{-2b_1}) (p^2 - 1) \right) \sum_{j=l}^N v_{l,j}^2 + \frac{2^{-2b_2}}{3} \left(b_2 + \frac{4}{3} (1 - 2^{-2b_2}) (p^2 - 1) \right) \right] + \frac{1}{3} (2^{-2b_1} - 2^{-2b_2}). \quad (\text{B.9})$$

APPENDIX C ROUND OFF ERROR ANALYSIS OF THE STORED PRODUCT STRUCTURE

The derivation of the output error variance in this case is similar to that of Appendix A. The difference here is that RAMs are used instead of multipliers. Also, address partition is considered here. Let $\delta_{m,n}^l$ be the accumulated roundoff error at the output of the row filter in stage l , we have

$$\delta_{m,n}^l = - \sum_{i=l}^M p_1 \eta_{m,n}^i \quad (\text{C.1})$$

where $p_1 \geq 1$ is the number of address partitions in the row filter and $\eta_{m,n}^i$ is the roundoff error due to finite word length of the memory in the row filter. Similarly, the accumulated roundoff error at the output of the column filter in stage l is given by

$$\alpha_{m,n}^l = - \sum_{i=l}^N p_2 \gamma_{m,n}^i \quad (C.2)$$

where $p_2 \geq 1$ is the number of address partitions in the column filter and $\gamma_{m,n}^i$ is the roundoff error due to finite word length of the memory in the column filter.

The variance of the output error of the row filter is given by

$$\sigma_{\delta_{m,n}}^2 = E[(\delta_{m,n}^l)^2] - E^2[\delta_{m,n}^l]. \quad (C.3)$$

Assume all errors are uncorrelated, (C.3) reduces to

$$\sigma_{\delta_{m,n}}^2 = \sum_{i=l}^M p_1^2 \sigma_{\eta_{m,n}^i}^2. \quad (C.4)$$

Similarly, the variance of the output roundoff error due to the column filter only is

$$\alpha_{m,n}^l = \sum_{i=l}^N p_2^2 \sigma_{\alpha_{m,n}^i}^2. \quad (C.5)$$

Finally, according to (A.4), the variance of the final output roundoff error is given by

$$\sigma_{sp}^2 = \sum_{l=1}^k \left[p_1^2 (M-l+1) \frac{2^{-2b_2}}{3} \sum_{j=l}^N v_{l,j}^2 + p_2^2 (N-l+1) \frac{2^{-2b_2}}{3} \right] + \frac{1}{3} (2^{-2b_1} - 2^{-2b_2}). \quad (C.6)$$

APPENDIX D

ROUND OFF ERROR ANALYSIS OF THE DIRECT IMPLEMENTATION WITHOUT DECOMPOSITION

The roundoff accumulation error in this case is mainly due to the rounding or truncation of the multiplications as well as the rounding or truncation of the output sample. The output $\tilde{y}_{m,n}$ from the finite length register filter is given by

$$\tilde{y}_{m,n} = \left[\sum_{i=1}^M \sum_{j=1}^N [h_{i,j} x_{m-i+1, n-j+1}]_{b_2} \right]_{b_1} \quad (D.1)$$

where $b_2 \geq b_1$ and $[*]_b$ implies the content $*$ is rounded or truncated to b bits. Hence, the error at the output is

$$\begin{aligned} f_{m,n} &= y_{m,n} - \tilde{y}_{m,n} \\ &= MN\epsilon^{b_2} + \epsilon^{b_2, b_1}. \end{aligned} \quad (D.2)$$

Assume all the errors are uncorrelated, the output error variance is given by

$$\begin{aligned} \sigma_{df}^2 &= MN\sigma_{\epsilon^{b_2}}^2 + \sigma_{\epsilon^{b_2, b_1}}^2 \\ &= \frac{1}{3} [MN(2^{-2b_2}) + (2^{-2b_1} - 2^{-2b_2})] \end{aligned} \quad (D.3)$$

REFERENCES

- [1] A. N. Venetsanopoulos and B. G. Mertzios, "A decomposition theorem and its implications to the design and realization of two-dimensional filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-33, pp. 1562-1575, Dec. 1985.
- [2] J. R. Adams, E. C. Driscoll, and C. Reader, "Image processing system," in *Digital Image Processing Techniques*, New York, Academic, 1984.
- [3] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980.
- [4] A. N. Venetsanopoulos and B. G. Mertzios, "A general implementation technique for 2-D digital filters," in *Proc. 6th Summer Symp. Circuit Theory*, Prague, Czechoslovakia, pp. 176-180, July 12-16, 1982.
- [5] C. L. Nikias, A. P. Chrysafis, and A. N. Venetsanopoulos, "The LU decomposition theorem and its implications to the realization of two-dimensional digital filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-33, June 1985.
- [6] H. H. Chiang, C. L. Nikias, and A. N. Venetsanopoulos, "Efficient implementation of quadratic digital filters," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1511-1528, Dec. 1987.
- [7] J. S. Ward, P. Barton, J. B. G. Roberts, and B. J. Stanier, "Figures of merit for VLSI implementations of digital signal processing algorithms," *Proc. Inst. Elect. Eng.*, vol. 131, p. F, no. 1, pp. 64-70, Feb. 1984.
- [8] A. N. Venetsanopoulos, B. G. Mertzios and S. H. Mneney, "Effects of finite precision in two-dimensional recursive digital filters," *Int. J. Electronics*, vol. 58, no. 1, pp. 159-174, 1985.
- [9] A. C. P. Loui, "Modular implementation of two-dimensional and quadratic digital filters," M.A.Sc. thesis, Dep. Elect. Eng. Univ. of Toronto, Toronto, Canada, 1986.
- [10] A. Peled and B. Liu, "A new hardware realization of digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 456-462, 1974.
- [11] A. C. P. Loui and A. N. Venetsanopoulos, "Distributed arithmetic two-dimensional digital filters for real-time image processing," in *IEEE Proc. Electronic*, vol. 1, pp. 198-201, Toronto, Canada, Oct. 7-9, 1985.
- [12] H. Jaggernauth, A. C. P. Loui, and A. N. Venetsanopoulos, "Real-time image processing by distributed arithmetic implementation of two-dimensional digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1546-1557, Dec. 1985.
- [13] W. Steenaart, D. Dubois, and O. Monkewich, "Stored product digital filtering, structures, potential and applications," in *Proc. 1981 European Conf. on Circuit Theory and Design*, The Hague, The Netherlands, pp. 118-120, Aug. 25-28, 1981.
- [14] H. T. Kung, "Why systolic architectures," *IEEE Computer*, vol. 15, pp. 37-46, Jan. 1982.
- [15] S. Treitel and J. L. Shanks, "The design of multistage separable planar filters," *IEEE Trans. Geosci. Electron.*, vol. GE-9, pp. 10-27, Jan. 1971.

✱

Alexander C. P. Loui (S'82) received the B.A.Sc. (honor) and M.A.Sc. degrees in electrical engineering from the University of Toronto, Canada, in 1983 and 1986, respectively. He is now working towards the Ph.D. degree in electrical engineering, also from the same university.

In the summer of 1982, he worked at the Atomic Energy of Canada Ltd., as a Research Assistant. He has been a Teaching Assistant with the Department of Electrical Engineering, University of Toronto, Canada since 1983. From 1983-1986, he was awarded the Postgraduate Scholarship from the

Natural Sciences and Engineering Research Council of Canada. In 1987, he was the recipient of the University of Toronto Doctoral Fellowship. His research interests include digital signal/image processing, architectures for image processing and knowledge-based image processing systems.

Venetsanopolos was a Fulbright Scholar, A. F. Schmitt Scholar and recipient of the J. Vakis Award. Member of the New York Academy of Sciences, Sigma Xi, the International Society for Optical Engineering and the Technical Chamber of Greece; he is a Fellow of the Engineering Institute of Canada, and Registered Professional Engineering in Ontario and Greece.

Anastasios N. Venetsanopoulos (S'66-M'69-SM'79) received the B.S. degree from the National Technical University of Athens [NTU], Greece [1965], and the M.S. [1966], M. Phil. [1968] and Ph.D. [1969] degrees in electrical engineering, all from Yale University. He joined the University of Toronto, Canada, in September 1968, where he is now Professor at the Department of Electrical Engineering. He also served as Chairman of the Communications Group (1974-78 and 1981-86), and as Associate Chair-

man of the Department of Electrical Engineering (1978-79). He was Invited Professor at NTU, the Swiss Federal Institute of Technology, the Federal University of Rio de Janeiro, and the University of Florence. He was on research leave at the University of Grenoble, the Imperial College of Science and Technology, and was Adjunct Professor at Concordia University. He served as lecturer of numerous short courses to industry and continuing education programs; is contributor to ten books and has published over 200 papers in digital signal and image processing, and digital communications; he also served as consultant to several organizations; Editor of the Canadian Electrical Engineering Journal [1981-83]; President of the Canadian Society for Electrical Engineering and Vice-President of the Engineering Institute of Canada [1983-86]. Dr.

Chrysostomos L. Nikias (S'76-M'82) received the diploma in electrical and mechanical engineering from the National Technical University of Athens, Greece, in 1977 and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York at Buffalo, in 1980 and 1982, respectively.

He is Associate Professor and Director of the Communications and Digital Signal Processing (CDSP) Laboratory, Department of Electrical and Computer Engineering, Northeastern University, Boston, MA. His research interests lie in the area of digital signal processing with applications. He has organized and taught extensively short courses devoted to continuing engineering education. He is the designer and instructor of a videotape seminar (15 hours) entitled MODERN SPECTRUM ESTIMATION AND ARRAY PROCESSING, Network Northeastern Boston, MA: 1987.

Dr. Nikias is a member of Tau Beta Pi and the Technical Chamber of Greece. He has previously served as Associate Editor of the IEEE Transactions on ASSP (1985-87) and as the Co-Chairman of the Third ASSP Workshop on Spectrum Estimation and Modeling (1986).