Optimisation technique for the implementation of two-dimensional recursive digital filters by sectioning

M.R. Azimi Sadjadi, M.Sc., D.I.C., Ph.D., Mem.I.E.E.E., R.A. King, M.A., D.I.C., Ph.D., C.Eng., F.I.E.E., and S.K. Pal, M.Tech., Ph.D., D.I.C., Mem.I.E.E.E.

Indexing term: *Filters and filtering*

Abstract: The problem of two-dimensional recursive digital filtering of large images by the sectioning (or block-mode) technique has been considered. It has been stated that, if the impulse-response sequence of the recursive digital filter dies down relatively sharply, by a suitable truncation, the available sectioning techniques can be used to perform the filtering operation, as with nonrecursive filters. An analytical approach is presented which formulates the upper bound on the norm of error resulting from the application of truncation. This bound is found to be independent of the size of blocks. In addition, an expression for the computation time is obtained, when a mixed radix two-dimensional FFT is used. Using these expressions, a simple procedure for minimising the truncation error and computation time is suggested which determines suitable values of the parameters involved. The effectiveness of the optimisation procedure developed here has been examined for smoothing operation of an X-ray image on a CDC 6500 computer.

1 Introduction

The sectioning technique [1–3] is a valuable aid in computing the convolution between two finite-area sequences in which one sequence (usually the input) is of considerably larger size than the other sequence (usually the impulse response). Applying this technique to picture processing, the input picture will be broken up into a number of blocks, each of which is of a size comparable with that of the impulse response. The FFT may then be employed to evaluate the circular convolution between each input block and the impulse response. There are basically two types of sectioning algorithms, known as 'select-save' and 'overlap-add', which can be used to eliminate the effect of wrap-around error inherent in the use of circular convolution. These techniques provide an efficient means of nonrecursive [finite impulse response (FIR)] digital filtering operation.

In some applications, need might arise to perform a recursive [infinite impulse response (IIR)] filtering operation by an FFT algorithm.

Under certain conditions, sectioning techniques would seem to be capable of being used in such applications. As Helms [4] suggested, in the one-dimensional case, if the impulse-response sequence of a recursive digital filter dies down fairly fast, with the aid of suitable truncation, the sectioning algorithms can be applied to perform the filtering operation within an acceptable accuracy.

Truncation necessarily produces error on the output elements, the magnitude of which is dependent on the degree of truncation chosen.

The aim of optimal sectioning is to determine the block sizes and truncation sizes in order to

(*a*) reduce truncation error

(*b*) reduce computation time

(*c*) use less computer memory.

The requirement on computer memory is strictly dependent on the type of computer used, but, in order to transfer the entire row of input blocks into the memory from secondary storage devices, the amount of the primary memory available must be adequate. An investigation has been carried out by Twogood, Ekstrom and Mitra [3] for when the amount of the primary memory is not sufficient to accommodate a complete row of blocks.

The computation time, on the other hand, is dependent on several parameters, such as picture size, block size, the size of the impulse-response sequence (or truncation size if a recursive filter is used), the radix of the FFT and the efficiency of the FFT algorithm.

The problem of minimising the computation time for the sectioning technique was extended to the two-dimensional case by Hunt [1, 2]. He has given an expression for the computation time T_{CPU} in terms of sectioning parameters. This expression is shown to be incorrect, because it does not include the needed rounding-up operations [3]. Moreover, in contrast with the one-dimensional case, two-dimensional filtering operations require substantial input/output (I/O) time, owing to the large amount of data involved. Recently Twogood, Ekstrom and Mitra [3] have considered a method of minimisation which also incorporates the I/O requirements.

The problem of finding the conditions for minimum computation time and truncation error in two-dimensional recursive filtering operation by sectioning techniques gives rise to contradictory results. That is, for a fixed block size, the computation time is minimum when the impulse-response sequence is severely truncated, and this consequently increases the corresponding error; conversely, the truncation error will be minimum when the impulse-response sequence is only slightly truncated, and this results in an increase in the computation time.

In this paper, we derive an expression for the upper bound on the norm of the truncation error using matrix representation of two-dimensional convolution. This bound is found to be independent of the block size. Thus optimisation with respect to truncation error can first be carried out without introducing the block sizes into the calculations. This optimum value of truncation size may then be used to obtain the relevant suitable value of block size which makes the computation time optimum.

In Section 2, a more general expression for computation time is obtained when a mixed-radix FFT is used.

Section 3 is devoted to the formulation of an upper bound on the norm of the truncation error based on matrix representation of two-dimensional convolution [5]. Using these formulations, a simple approach towards obtaining optimal sectioning parameters is proposed. In Section 4, the implementation of two-dimensional recursive digital filtering by the

Paper 2013F, received 27th January 1982

The authors are with the Electrical Engineering Department, Imperial College of Science & Technology, London SW7 2BT, England. Dr. Pal is on leave from the Electronics & Communication Unit, Indian Statistical Institute, Calcutta 700035, India

sectioning technique for an X-ray image is examined on a large scientific CDC 6500 computer. A procedure for determining the optimal sectioning parameters based on the techniques developed in this paper is given which leads to an optimum result; this has been illustrated and compared with other nonoptimum ones.

2 Computation time

Consider the following two-dimensional convolution:

$$y(k,l) = \sum_{m=0}^{k} \sum_{n=0}^{l} h(k-m, l-n)\, x(m,n) \tag{1}$$

where $\{x(k,l)\}$, $\{y(k,l)\}$ and $\{h(k,l)\}$ are the input, output and impulse-response sequences, respectively.

Let us assume the input sequence to be of dimensions $P \times Q$, the truncated impulse response to be of dimensions $M \times N$ and the blocks of input to be of dimensions $D_1 \times D_2$. Then, using the select-save or overlap-add method [1, 3], the acceptable part of the blocks which is processed in each iteration of the sectioning algorithm is a subsection of size $(D_1 - M + 1) \times (D_2 - N + 1)$.

Therefore the total number of blocks to be processed is given by

$$N_{blocks} = \left\lceil \frac{P}{(D_1 - M + 1)} \right\rceil \left\lceil \frac{Q}{(D_2 - N + 1)} \right\rceil \tag{2}$$

where the quantity in the square bracket must be rounded up to the next integer above its value.

Then the total number of complex multiplications required is

$$N_{total} = N_{blocks}(2N_{FFT} + D_1 D_2) \tag{3}$$

where N_{FFT} is the number of complex multiplications to compute the mixed-radix FFT of each block. The total time required for processing to a first approximation (excluding the time for index interchanging and additions) is

$$T_{CPU} = \gamma N_{total} \tag{4}$$

where γ is a proportionality constant and is dependent on the type of FFT algorithm and computer used; for instance, the value of γ for the Singleton algorithm [6] on CDC 6500/6400 computers is approximately 25 μs.

In most image processing problems, owing to the large amount of data to be processed, the total I/O time necessary to fetch an input row of blocks from disc or tape and to store the output row of blocks on another disc or tape, after processing, constitutes a substantial part of the computational time required for the filtering operation. There are basically two timings which contribute to the total I/O time: access time T_{acc} and the transfer rate per word T_{trans} [3].

Owing to the complexity of the operating system in a multijob environment, the access time is very difficult to predict and virtually impossible to measure. Moreover, depending on the type of the tape drivers used and their relevant densities, a particular transfer rate can be produced. In a multijob environment, the user is not provided with a facility to choose the disc unit, and so I/O transfer rates will vary wildly, even with the same job running on two different occasions.

The CDC 6500 computer is a multijob computer running under NOS operating system;§ hence, unlike the CDC 7600 [3], which executes only one job at a time, the I/O timings cannot be included in the total real time.

§ Imperial College Computer Centre: Private communication

In order to find N_{FFT} for the mixed-radix FFT in terms of D_1 and D_2, in Appendix 1, the FFT algorithm has been considered within the framework of matrix decomposition [7]. Since the two-dimensional FFT can be regarded as repeated applications of a one-dimensional FFT along all the rows and columns of the two-dimensional array, then without loss of generality we may focus our attention on the decomposition of a one-dimensional transform. The total number of complex multiplication N_{multi} when using the mixed-radix one-dimensional FFT is (Appendix 1)

$$N_{multi} = N \sum_{i=1}^{m} n_i - (m+2)N + N/n_m + 1 \tag{5}$$

where it is assumed that N may be factorised into a number m of prime numbers n_i as

$$N = \prod_{i=1}^{m} n_i \tag{6}$$

In the special case for $N = p^m$, where p is an odd prime, the total number of operations for performing the ith transform step, excluding the number of operations for twiddle factors, is

$$N(n_i - 1) - \frac{N(n_i - 1)}{n_i} = \frac{N(n_i - 1)^2}{n_i}$$

which in this case becomes $N(p-1)^2/p$; hence the total number of operations (excluding the twiddle factors) will be $mN(p-1)^2/p$. Singleton [6] has shown that the complex transform of dimension p, for p odd, can be computed with $(p-1)^2$ real multiplications, or equivalently $(p-1)^2/4$ complex multiplications. With this modification, the total number of complex multiplications given above will become $mN(p-1)^2/4p$, which is the result given in Reference 6.

When $N = 2^m$, a case often used in practice, the elements of the transform matrices T_i are either 1 or -1; thus performing $T_i x^{(i)}$ can be done without multiplications, and the total number of multiplications in this case will be reduced to the number of complex multiplications needed for the twiddle factors only, i.e. $mN/2$, $m = \log_2 N$. Thus

$$N_{multi} = \frac{N}{2} \log_2 N \tag{7}$$

The idea of matrix decomposition for a one-dimensional transform may easily be applied to the two-dimensional case; hence, for a two-dimensional FFT of size $D_1 \times D_2$, the total number of complex multiplications is

$$N_{FFT} = N_{multi(1)} D_2 + N_{multi(2)} D_1 \tag{8}$$

where

$$N_{multi(1)} = D_1 \sum_{i=1}^{p} d_{1i} - (p+2)D_1 + \frac{D_1}{d_{1p}} + 1$$

$$D_1 = \prod_{i=1}^{p} d_{1i}$$

and

$$N_{multi(2)} = D_2 \sum_{i=1}^{q} d_{2i} - (q+2)D_2 + \frac{D_2}{d_{2q}} + 1$$

$$D_2 = \prod_{i=1}^{q} d_{2i}$$

d_{1i} and d_{2i} being prime numbers.

For the radix-2 transform, we have

$$N_{FFT} = \frac{D_1 D_2}{2} \log_2 D_1 D_2 \tag{9}$$

3 Upper bound on the norm of truncation error

As stated earlier, sectioning techniques may be used to implement two-dimensional recursive digital filtering operation if the impulse-response sequence is truncated to a proper degree. In this regard, Helms [4] has proposed two different methods in the one-dimensional case for determining the suitable amount of truncation. The degree of truncation, according to the Helms first method, can be obtained by inspecting the impulse-response sequence (in the time domain) and choosing the maximum number of contiguous values of the impulse-response sequence which are not 'very small'. A suitable amount of this truncation may then be found by increasing this value until further increases no longer produce significant changes in the output. This method is impractical for various reasons. First, what is 'very small' and how should it be chosen? Secondly how can one be assured that the suitable degree of truncation chosen by this method is really suitable? It is not clear how the overall effect of changes on the output can be determined. If this suitable value is chosen according to the effect which it has only on one output block, this does not imply that it is also desirable for the other blocks. Again, if this suitable value is selected based on the overall effect on the whole output for each truncation degree, the computation time required is several times greater than that of the filtering operation itself.

Helms has also proposed another method when the filtering operation is specified as a frequency response (i.e. an amplitude response or a phase response or both). This method, which is named the 'four-Ts' (i.e. transform-truncate-transform-test) method, involves determining the appropriate truncation degree by computing the amplitude and/or phase response after the truncation and then comparing the result with the predetermined amplitude and/or phase response at each iteration of the algorithm until satisfactory agreement between the two frequency responses is achieved.

Again this method is impractical, first because the effect of the input on the output is ignored, secondly it requires computing a number of DFTs and IDFTs, and thirdly the block size needs to be predetermined, whereas the optimum block size is dependent on the truncation degree.

These problems, which play important roles in two-dimensional filtering by sectioning techniques, have been closely investigated in this Section. A method which circumvents all the above mentioned problems associated with the Helms method is introduced, which without any difficulty gives suitable values of truncation degree.

In what follows, the formulation of the upper bound on the norm of the truncation error is made using matrix representation of two-dimensional convolution [8]. With this formulation, the overall effect of the truncation error on the output elements can be determined, and the choice of suitable degree of truncation may then be made possible.

Let us consider the input to be a picture array of dimensions $P \times Q$, sectioned into nonoverlapping blocks of size $K \times L$ (K and L should be greater than the order of the filter). Moreover, it will be assumed that the picture size is increased (if necessary) so that P and Q are exactly divisible by K and L by adding an appropriate set of zeros at the extreme boundaries. Then writing the convolution eqn. 1 in matrix form we have

$$Y = GX \tag{10}$$

where X and Y are vectors of dimension PQ representing the input and output ordered lexicographically, and G is a lower triangular block Toeplitz matrix of dimensions $PQ \times PQ$.

Now let an array of size $M \times N$ contain all the elements of the impulse-response array which may be considered to be of significant magnitude, and let the impulse-response array be truncated to a size (m, n), where $M \leqslant m < K$ and $N \leqslant n < L$. For this truncated impulse response, G can be defined as

$$G = (G_{i,j}) \qquad i = 1, 2, \ldots, Q/L$$
$$j = 1, 2, \ldots, Q/L$$

where

$$G_{i,j} = \begin{cases} G_0 & i-j = 0 \\ G_1 & i-j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The block matrices G_0 and G_1, which are of dimensions $PL \times PL$, may also be defined similarly, i.e.

$$G_0 = (G_{0i,j}) \qquad i = 1, 2, \ldots, P/K$$
$$j = 1, 2, \ldots, P/K$$

where

$$G_{0i,j} = \begin{cases} S_0 & i-j = 0 \\ S_1 & i-j = 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$G_1 = (G_{1i,j}) \qquad i = 1, 2, \ldots, P/K$$
$$j = 1, 2, \ldots, P/K$$

where

$$G_{1i,j} = \begin{cases} S_0' & i-j = 0 \\ S_1' & i-j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Here S_0, S_1, S_0' and S_1' are block Toeplitz matrices of dimensions $KL \times KL$. We shall now study the effect on the output error of increasing (m, n) to $(m + a, n + b)$, where a and b are preferably large numbers, and $a < K - m$, $b < L - n$. Then the value of (m, n) is chosen such that the change of error due to this transition is negligible and also in order to ensure that the specified value is acceptable in that further transitions should result in decreasing error.

Let the output vector resulting from (m, n) truncation of the impulse response be $Y^{m,n}$. Then the error vector may be defined as

$$\epsilon^{m,n} = Y^{m+a,n+b} - Y^{m,n}$$
$$= (G^{m+a,n+b} - G^{m,n})X \triangleq \hat{G}X \tag{13}$$

The bound on the error may be derived from the Euclidean norm [9] of eqn. 13:

$$\|\epsilon^{m,n}\| \leqslant \|\hat{G}\| \, \|X\| \tag{14}$$

where

$$\|\hat{G}\|^2 = \text{trace}\,(\hat{G}^t \hat{G}) \tag{15}$$

If we further assume that the input sequence $\{x_{i,j}\}$ is bounded, i.e. $|x_{i,j}| \leqslant B$, the average error ξ for each output element

must satisfy

$$\xi \leqslant \frac{B\|\hat{G}\|}{(PQ)^{1/2}} \tag{16}$$

In Appendix 2, the norm of $\hat{G}$ is expressed in terms of the elements of the truncated impulse-response sequence. Since the upper bound given in expr. 16 is independent of the block size, a simpler procedure for obtaining optimal sectioning parameters is suggested.

4 Implementation and results

The optimal sectioning procedure can be organised in accordance with the following steps:

(i) Choose the initial values of m and n as $m = M$, $n = N$. The bound on the norm of error for the transition to $m + a$, $n + b$ is computed. If this error is less than β (the maximum acceptable error for each output element) and the errors due to the subsequent transitions are decreasing, the selected values of m, n are acceptable. Otherwise, m, n will be substituted by $m + a, n + b$ and then incremented by a, b and the process will be repeated until the error is less than β. The final suitable values of m, n will then be placed as M, N in eqn. 2 for the next step.

(ii) Having determined the optimal values of truncation degree, the optimisation procedure with respect to the block size for minimum computation time is then carried out.

It must be mentioned that the optimisation procedure for computation time presented here is based on minimising N_{total} rather than T_{CPU}, because the proportionality constant γ is dependent on the type of computer and the efficiency of the algorithm for a particular radix. This means that different radices could produce some variation in γ and consequently in T_{CPU}.

The effectiveness of the optimisation technique developed so far, when the lowpass filtering (smoothing) operation is performed by the sectioning technique, has been examined for an X-ray image of Fig. 1. This picture shows the radiograph of a part of the wrist containing the radius (with epiphysis and metaphysis) and a part of two small carpal bones taken from a boy in the 10–12 year age group. The digitised version of the picture is represented by a two-dimensional 128 x 145 ($= P \times Q$) array having 32 ($= B + 1$) grey levels. The image, as seen from the histogram of Fig. 2, contains five regions of pixel intensity. They are approximately:

(*a*) 6 to 10
(*b*) 10 to 12
(*c*) 12 to 17
(*d*) 17 to 21
(*e*) 21 to 25.

These regions relate to small variations in grey level, corresponding to soft tissue, single bone, superimposed bones, palmar and dorsal surfaces [10]. The first and the last regions correspond to soft tissue, and palmar and dorsal surfaces, respectively.

The smoothing operation is performed by a two-dimensional recursive lowpass filter (see example 2 of Reference 11) with the following transfer function:

$$H(z^{-1}, w^{-1}) = 0.0122 \frac{[1 \quad z^{-1} \quad z^{-2}] \begin{bmatrix} 1.0 & 0.410191 & 0.594957 \\ 0.240013 & -0.887865 & 0.423221 \\ 0.560841 & 0.453500 & 0.360962 \end{bmatrix} \begin{bmatrix} 1 \\ w^{-1} \\ w^{-2} \end{bmatrix}}{[1 \quad z^{-1} \quad z^{-2}] \begin{bmatrix} 1.0 & -0.500549 & -0.138282 \\ -0.690435 & -0.195020 & 0.346731 \\ -0.043308 & 0.342758 & -0.093572 \end{bmatrix} \begin{bmatrix} 1 \\ w^{-1} \\ w^{-2} \end{bmatrix}}$$

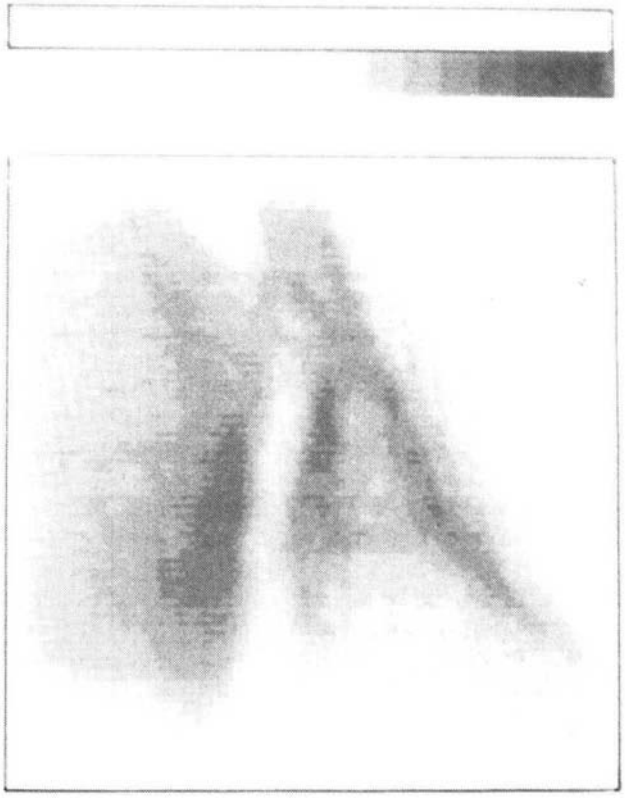

Fig. 1 *Original input image*

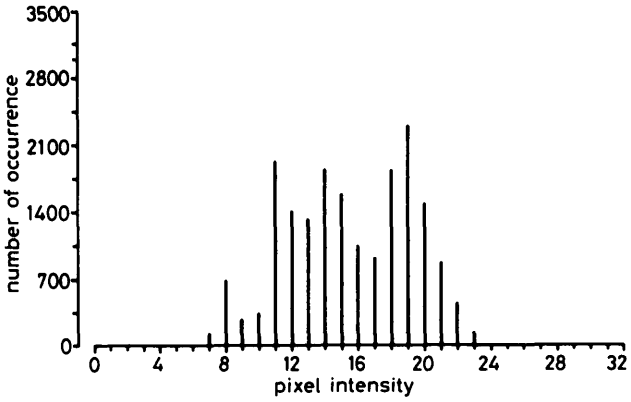

Fig. 2 *Histogram of original input image of Fig. 1*

The optimisation procedure is first carried out for the truncation error, and gives $(m, n) = (11, 11)$, which is the optimum value for truncation of the impulse-response sequence; truncation values less than this lead to output pictures which have lost some desired information. Different block sizes, namely 32 x 32, 30 x 30, . . . , 20 x 20, have been considered in order to examine their effects on the computation time and also on the quality of the smoothed image. The computation times for different block sizes are tabulated in the third column of Table 1 for several runs of the program on the CDC 6500

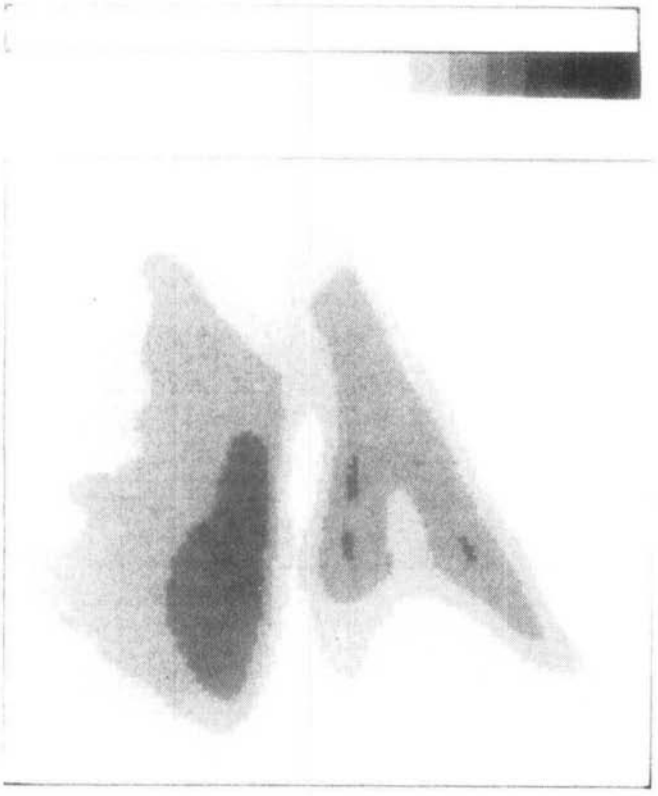

Fig. 3A *Smoothed image using 32 x 32-block impulse response truncated to 11 x 11*

Fig. 3B *Smoothed image using 26 x 26-block impulse response truncated to 11 x 11*

computer. The theoretical results for the number of multiplications are then calculated using eqns. 2, 3 and 8 and tabulated in the second column of the Table. The FFT algorithm used here is Singleton's algorithm [6]; therefore, when calculating the number of multiplications, N_{FFT} should be computed in accordance with the necessary modifications.

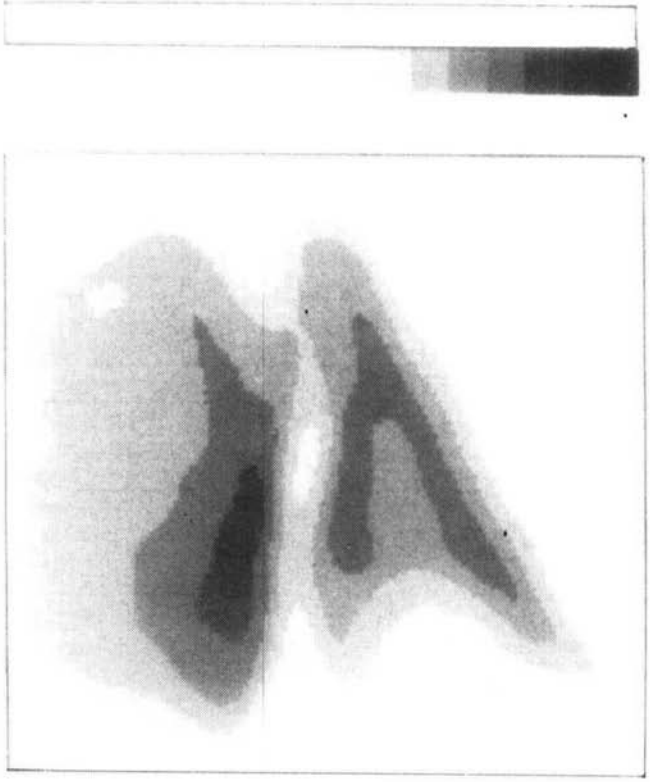

Fig. 3C *Smoothed image using 20 x 20-block impulse response truncated to 11 x 11*

The measured timings are the total time for processing of the image, including the time needed for computations and also the time for index interchanging. Moreover, there are other factors, such as the efficiency of the algorithm for a particular radix [6], which can contribute to the total CPU time. Nevertheless, the theoretical and experimental results in Table 1 are shown to be in good agreement. Figs. 3*a–c* demonstrate three such typical instances of the smoothed images for block sizes 32 x 32, 26 x 26 and 20 x 20, respectively. The corresponding changes in histogram are shown in Figs. 4*a–c*. For a fixed truncated impulse response, the variation of block size does not lead to a significant change in the quality of the smoothed image. Small changes in grey levels as seen from the histograms are mainly due to round-off error in computation. Throughout this experiment we considered a truncated impulse response of dimensions 11 x 11. A block size of dimensions 32 x 32 is found to provide optimum result with regard to computation time and truncation error.

5 Conclusion

The implementation of two-dimensional recursive digital filtering of a large image using the sectioning technique is considered. In particular, the sectioning approach is investigated within the context of the total computation time and truncation error. Expressions are then derived which enable us to determine the optimal sectioning parameters. Using these expressions, the optimisation procedure can be carried out for minimising the computation time, and also the truncation error, independently, because the bound on the norm of incremental error due to the truncation of the impulse response is found to be independent of the size of the sections. This consequently results in a simpler and more efficient algorithm for obtaining suitable values of sectioning parameters. The effectiveness of this algorithm has been examined for the smoothing operation of an X-ray image on the CDC 6500 computer for different block sizes and truncation degrees.

Table 1: Number of computations N_{total} and computation time T_{total} for different values of block sizes

(D_1, D_2)	N_{total} (theoretical)	T_{total} (experimental)
		s
(20, 20)	670800	66
(21, 21)	797328	102
(22, 22)	880880	101
(23, 23)	1451760	124
(24, 24)	517440	64
(25, 25)	616500	52
(26, 26)	798720	86
(27, 27)	532656	58
(28, 28)	604800	69
(30, 30)	530880	53
(32, 32)	134400	32

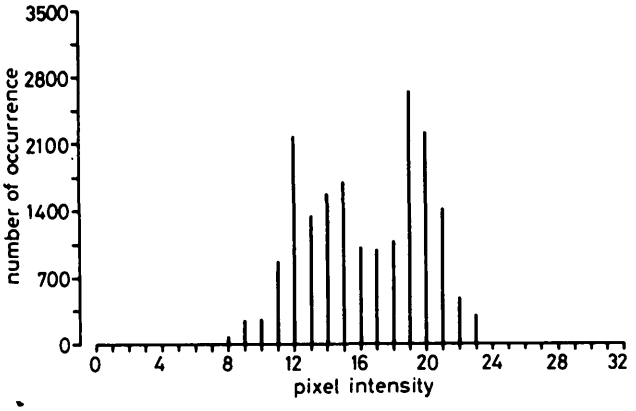

Fig. 4A *Histogram of Fig. 3A*
$(D_1, D_2) = (32, 32)$ $(M, N) = (11, 11)$

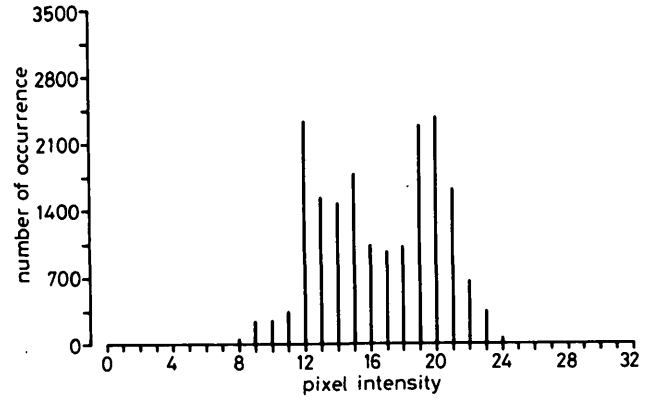

Fig. 4B *Histogram of Fig. 3B*
$(D_1, D_2) = (26, 26)$ $(M, N) = (11, 11)$

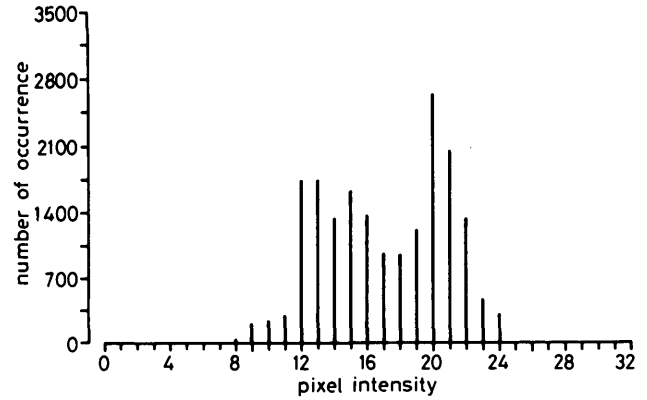

Fig. 4C *Histogram of Fig. 3C*
$(D_1, D_2) = (20, 20)$ $(M, N) = (11, 11)$

The resultant smoothed images are then demonstrated. The optimum result obtained from the application of the algorithm developed here is shown to provide optimum computation time and truncation error.

The two-dimensional recursive digital filter used for the smoothing operation here is a lowpass filter for which the impulse-response sequence dies down fairly quickly. Although in most applications this is generally the case, if the filter impulse-response sequence in the time domain contains a large number of elements with significant magnitudes, the filtering operation by means of sectioning will lose its validity. An alternative and more efficient implementation scheme for such cases is developed in Reference 8.

6 References

1 HUNT, B.R.: 'Block-mode digital filtering of pictures', *Math. Biosci.*, 1971, **11**, pp. 343–354
2 HUNT, B.R.: 'Minimization of the computation time for using the technique of sectioning for digital filtering of pictures', *IEEE Trans.*, 1972, **C-21**, pp. 1219–1222
3 TWOGOOD, R.E., EKSTROM, M.P. and MITRA, S.K.: 'Optimal sectioning procedure for the implementation of 2-D digital filters', *ibid.*, 1978, **CAS-25**, pp. 260–268
4 HELMS, H.: 'Fast Fourier transform method of computing difference equations and simulating filters', *ibid.*, 1967, **AU-15**, pp. 85–90
5 AZIMI SADJADI, M.R. and King, R.A.: 'Truncation error in 2-D block-mode filtering', *Electron. Lett.*, 1981, **17**, pp. 217–218
6 SINGLETON, R.C.: 'An algorithm for computing the mixed radix fast Fourier transform', *IEEE Trans.*, 1969, **AU-17**, pp. 93–103
7 KAHANER, D.K.: 'Matrix description of the fast Fourier transform', *ibid.*, 1970, **AU-18**, pp. 442–450
8 AZIMI SADJADI, M.R., and KING, R.A.: 'Block implementation of 2-D digital filters'. 22nd Midwest symposium on circuits and systems, 1979, pp. 658–662
9 LANCASTER, P.: 'Theory of matrices' (Academic Press, 1969)
10 TANNER, J.M., WHITEHOUSE, R.H., MARSHALL, W.A., HEALY, M.J.R., and GOLDSTEIN, H.: 'Assessment of skeletal maturity and prediction of adult height (TW2 method)' (Academic Press, 1975)
11 ALY, S.A.H., and FAHMY, M.M.: 'Design of two-dimensional recursive digital filters with specific magnitude and group delay characteristics', *IEEE Trans.*, 1978, **CAS-25**, pp. 908–916
12 GENTLEMAN, W.M., and SANDE, G.: 'Fast Fourier transforms for fun and profit'. 1966 Fall joint computer conference, AFIPS Proc., 29, 1966, pp. 563–578

7 Appendixes

7.1 Appendix 1

The N-point DFT is defined as

$$X(k) = \sum_{l=0}^{N-1} x(l) W_N^{kl} \tag{17}$$

where

$$k, l \in [0, N-1]$$

and

$$W_N = \exp\left(\frac{-2\pi j}{N}\right) \qquad j = \sqrt{-1}$$

Eqn. 17 may be represented in matrix form:

$$X = Ax$$

where

$$X = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} \qquad x = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix} \tag{18}$$

and where A is an $N \times N$ matrix with elements

$$a_{k,l} = W_N^{kl \bmod N}$$

If N can be factorised as

$$N = \prod_{i=1}^{m} n_i$$

where n_i are prime numbers, then using Sande's version of factorisation [12] we have

$$\begin{aligned} k &= k_m n_{m-1} n_{m-2} \ldots n_1 + k_{m-1} n_{m-2} \ldots n_1 + \ldots \\ &\quad + k_2 n_1 + k_1 \\ l &= l_1 n_2 n_3 \ldots n_m + l_2 n_3 n_4 \ldots n_m + \ldots \\ &\quad + l_{m-1} n_m + l_m \end{aligned} \tag{19}$$

where

$$k_i, l_i \in [0, n_i - 1] \qquad i = 1, 2, \ldots, m$$

Substituting eqn. 19 into eqn. 17 we obtain

$$\begin{aligned} X(k) = &\sum_{l_m=0}^{n_m-1} \omega_m(l_m, k_1, k_2, \ldots, k_m) \\ &\sum_{l_{m-1}=0}^{n_{m-1}-1} \omega_{m-1}(l_{m-1}, k_1, k_2, \ldots, k_{m-1}) \ldots \\ &\sum_{l_1=0}^{n_1-1} x(l)\omega_1(l_1, k_1) \end{aligned} \tag{20}$$

where

$$\begin{aligned} \omega_1(l_1, k_1) &= W_{n_1}^{l_1 k_1} = t_1(l_1, k_1) \\ \omega_2(l_2, k_1, k_2) &= W_{n_2 n_1}^{l_2(k_2 n_1 + k_1)} = W_{n_2}^{l_2 k_2 l_2} W_{n_2 n_1}^{k_1} \\ &= t_2(l_2, k_2) r_1(l_2, k_1) \\ &\vdots \\ \omega_m(l_m, k_1, k_2, \ldots, k_m) &= W_N^{l_m(k_m n_{m-1} n_{m-2} \ldots n_1 + k_{m-1} n_{m-2} \ldots n_1 + \ldots + k_2 n_1 + k_1)} \\ &= t_m(l_m, k_m) r_{m-1}(l_m, k_1, k_2, \ldots, k_{m-1}) \end{aligned} \tag{21}$$

where $t_i(l_i, k_i)$ are the elements of the ith transform with kernel W_{n_i}, and $r_i(l_{i+1}, k_1, k_2, \ldots, k_i)$ denotes the corresponding twiddle factors with kernel $W_{n_1 n_2 \ldots n_{i+1}}$.

Using this factorisation and applying the matrix decomposition [7] for A results in

$$A = P_1 \hat{P}_2 \ldots \hat{P}_{m-1} \hat{F}_m \hat{R}_{m-1} \hat{T}_{m-1} \ldots R_1 T_1 \tag{22}$$

where

$$\hat{P}_i = \text{diag}(P_i, P_i, \ldots, P_i)$$

the P_i being permutation matrices containing only zeros and ones as their elements,

$$\hat{R}_i = \text{diag}(R_i, R_i, \ldots, R_i)$$

the R_i being diagonal matrices containing the twiddle factors r_i,

$$\hat{T}_i = \text{diag}(T_i, T_i, \ldots, T_i)$$

the T_i being matrices containing only n_i nonzero elements (t_i) in each row, and

$$\hat{F}_m = \text{diag}(F_m, F_m, \ldots, F_m)$$

where F_m is the matrix denoting the mth transform with kernel W_{n_m}.

Knowing that matrix factorisation will speed up the operations, let us see how many arithmetic operations we need to accomplish the computations indicated by eqn. 22. The effect of permutation matrices $P_1 \hat{P}_2 \ldots \hat{P}_{m-1}$ is equivalent to interchaning rows, which can be done without any arithmetic operations. Therefore we only need to count the operations for $\hat{R}_i \hat{T}_i$. Let us put

$$F_i = R_i T_i \qquad 1 \leqslant i \leqslant m$$

and

$$F_1 x \equiv x^{(1)} \qquad x \triangleq x^{(0)} \tag{23}$$

In general, $\hat{F}_i x^{(i)} \equiv x^{(i+1)}$, $\hat{F}_i = \text{diag}(F_i, F_i, \ldots, F_i)$, $i \neq 1$. Then, using the same technique as introduced by Kahaner [7], for the transition $x^{(i)} \rightarrow x^{(i+1)}$, we count only the total number of operations for multiplying the F_i block by a portion of $x^{(i)}$, and then multiply this by the number of blocks. It is easy to show that the number of blocks for the ith ($i > 1$) transition is equal to $\prod_{l=1}^{i-1} n_l$. The number of blocks for the first step is one. The dimensions of each block for the ith transition are $N/\prod_{l=1}^{i-1} n_l \times N/\prod_{l=1}^{i-1} n_l$. On the other hand, to multiply each row requires $(n_i - 1)$ complex multiplications; therefore the number of complex multiplications for each block will be equal to $\left(N/\prod_{l=1}^{i-1} n_l\right)(n_i - 1)$, and the total number of multiplications for the ith transition is then equal to $N(n_i - 1)$. The final step $\hat{F}_m$ is free of twiddle factors, and thus its number of multiplications will be reduced to

$$(n_m - 1)(n_m - 1)n_{m-1} \ldots n_1 = (n_m - 1)N - N + N/n_m$$

because each block requires $(n_m - 1)(n_m - 1)$ multiplications and the number of blocks for the mth step is $n_{m-1} n_{m-2} \ldots n_1$.

Furthermore, by considering the fact that $r_{i-1}(l_i, k_1, k_2, \ldots, k_{i-1}) = 1$ when $l_i \equiv 0 \bmod (n_i)$, the total number of multiplications may be reduced further by

$$\begin{aligned} &(n_1 - 1) + n_1(n_2 - 1) + n_1 n_2(n_3 - 1) + \ldots \\ &\quad + n_1 n_2 \ldots n_{m-1}(n_m - 1) = N - 1 \end{aligned}$$

Hence the total number of complex multiplications for the decomposition of eqn. 22 is

$$N_{multi} = N \sum_{i=1}^{m} n_i - (m + 2)N + \frac{N}{n_m} + 1 \tag{24}$$

7.2 Appendix 2

By considering the definition of $G_{i,j}$ in eqn. 11, eqn. 15 may

be expressed as

$$\|\hat{G}\|^2 = \frac{Q}{L}\|\hat{G}_0\|^2 + \left(\frac{Q}{L}-1\right)\|\hat{G}_1\|^2 \tag{25}$$

where

$$\hat{G}_0 = G_0^{m+a,n+b} - G_0^{m,n}$$

$$\hat{G}_1 = G_1^{m+a,n+b} - G_1^{m,n}$$

Also

$$\|\hat{G}_0\|^2 = \frac{P}{K}\|\hat{S}_0\|^2 + \left(\frac{P}{K}-1\right)\|\hat{S}_1\|^2 \tag{26}$$

and similarly for $\|\hat{G}_1\|^2$. Matrices $\hat{S}_0$ and $\hat{S}_1$ can be defined as

$$\hat{S}_0 = S_0^{m+a,n+b} - S_0^{m,n} = (\hat{S}_{0i,j}) \qquad i = 1,2,\ldots,K \qquad j = 1,2,\ldots,K$$

where

$$\hat{S}_{0i,j} = \begin{cases} \hat{H}_p & p = i-j \quad 0 \leqslant p \leqslant m+a \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{S}_1 = S_1^{m+a,n+b} - S_1^{m,n} = (\hat{S}_{1i,j}) \qquad i = 1,2,\ldots,K \qquad j = 1,2,\ldots,K$$

where

$$\hat{S}_{1i,j} = \begin{cases} \hat{H}_p & p = K-(j-i) \quad 1 \leqslant p \leqslant m+a \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

and similarly for $\hat{S}_0'$ and $\hat{S}_1'$, where the constituent block matrices are $\hat{H}_m'$. Matrices $\hat{H}_m$ and $\hat{H}_m'$ are of dimensions $L \times L$ and can be defined as for $0 \leqslant p \leqslant m$:

$$\hat{H}_p = H_p^{n+b} - H_p^n = (\alpha_{i,j}) \qquad i = 1,2,\ldots,L \qquad j = 1,2,\ldots,L$$

where

$$\alpha_{i,j} = \begin{cases} h_{p,q} & \text{when } q = i-j \quad n+1 \leqslant q \leqslant n+b \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{H}_p' = H_p'^{n+b} - H_p'^{n} = (\beta_{i,j}) \qquad i = 1,2,\ldots,L \qquad j = 1,2,\ldots,L$$

where

$$\beta_{i,j} = \begin{cases} h_{p,q} & \text{when } q = L-(j-i) \quad n+1 \leqslant q \leqslant n+b \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

For $m+1 \leqslant p \leqslant m+a$, we have

$$\hat{H}_p = H_p^{n+b} - H_p^n = (\gamma_{i,j}) \qquad i = 1,2,\ldots,L \qquad j = 1,2,\ldots,L$$

where

$$\gamma_{i,j} = \begin{cases} h_{p,q} & q = i-j \quad 0 \leqslant q \leqslant n+b \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{H}_p' = H_p'^{n+b} - H_p'^{n} = (\delta_{i,j}) \qquad i = 1,2,\ldots,L \qquad j = 1,2,\ldots,L$$

where

$$\delta_{i,j} = \begin{cases} h_{p,q} & q = L-(j-i) \quad 1 \leqslant q \leqslant n+b \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

Using these definitions, one can easily deduce that

$$\|\hat{S}_0\|^2 = \sum_{p=0}^{m+a} (K-p)\|\hat{H}_p\|^2 \qquad \|\hat{S}_1\|^2 = \sum_{p=0}^{m+a} p\|\hat{H}_p\|^2 \tag{30}$$

and similarly for $\hat{S}_0'$ and $\hat{S}_1'$. In addition,

$$\left.\begin{aligned} \|\hat{H}_p'\|^2 &= \sum_{q=n+1}^{n+b} (L-q)h_{p,q}^2 \\ \|\hat{H}_p'\|^2 &= \sum_{q=n+1}^{n+b} q h_{p,q}^2 \end{aligned}\right\} \text{for } 0 \leqslant p \leqslant m \tag{31}$$

$$\left.\begin{aligned} \|\hat{H}_p\|^2 &= \sum_{q=0}^{n+b} (L-q)h_{p,q}^2 \\ \|H_p'\|^2 &= \sum_{q=0}^{n+b} q h_{p,q}^2 \end{aligned}\right\} \text{for } m+1 \leqslant p \leqslant m+a \tag{32}$$

Therefore eqn. 15 may be reduced to

$$\|\hat{G}\|^2 = \sum_{p=0}^{m} (P-p) \sum_{q=n+1}^{n+b} (Q-q)h_{p,q}^2 + \sum_{p=m+1}^{m+a} (P-p) \sum_{q=0}^{n+b} (Q-q)h_{p,q}^2 \tag{33}$$