Reprint

W. Waung, J.M. Costa and A.N. Venetsanopoulos, "Two-dimensional digital filtering: Basic considerations and subroutines for implementation", in *Proceedings of the Second International Conference on Information Sciences and Systems, (INFO 2),* Vol. 2: Advances in Communications, D.J. Lainiotis and N.Z. Tszannes (Eds.), University of Patras, Greece, 9-13 July 1979, pp. 173-182.

TWO-DIMENSIONAL DIGITAL FILTERING. BASIC CONSIDERATIONS AND SUBROUTINES FOR IMPLEMENTATION*

W.Waung and A.N.Venetsanopoulos
University of Toronto
Toronto, Ontario, Canada

J.M. Costa
Bell Northern Research
Ottawa,Ontario,Canada

ABSTRACT

This paper considers filtering of two-dimensional data. The mathematics governing these filters are outlined. Different mathematical transforms useful in simplifying calculations and techniques to simplify computation are discussed. A subrountine (written in FORTRAN) to perform two-dimensional fast Fourier transforms of a real-valued array is briefly explained. An example of two-dimensional data processing is also given.

I. INTRODUCTION

During the past few decades, the rapid development of digital computers has made such devices available to the general public. Advances in this field have also greatly increased the capabilities of such machines [1], especially in the areas of storage capacity and speed of computation. These improvements have resulted in improvements in the retrieval of data in digital form. However with data stored digitally, it becomes more convenient to perform data processing in a digital fashion. This creates new problems, but at the same time offers degrees of freedom not available before [2-4].

In many cases, the data represent multidimensional information. Examples of such include contour maps, two-dimensional images, position versus time data in seismology etc. Of these data sets, two-dimensional information is usually most easily

* This work was supported by the Natural Sciences and Engineering Research Council of Canada, under grant No. A-7397.

D. G. Lainiotis and N. S. Tzannes (eds.), Advances in Communications, 173–182.

understood and thus generally most useful. Data sets of three or more dimensions are harder to visualize.

Some of the applications of two dimensional data processing include restoration of Images, enchancement of images and pattern recognition.

There are many other uses and examples [2] but those mentioned give a good indication of the importance and scope of such processing.

This paper is divided into two major section. The first section deals with an introduction to digital filtering and the mathematics involved. Brief mention is made of number theoretic transforms and discrete Fourier transforms. In the second, a subroutine to implment two-dimensional fast Fourier transforms is presented and briefly explained. Finally, an example of usage of the mentioned subroutine is given.

II. DIGITAL FILTERING

As stated in the introduction, digital data processing is particularly useful due to the availability of inexpensive digital computers. However the idea of processing data gives rise to notions of complicated algorithms. Viewed in a different manner this processing becomes easy to understand. One can usually model the process in terms of a digital filter. Then the output of such a filter is simply the processed result of the input to the filter [7]. The problem is then converted from that of the design of algorithms, to that of the design of a filter which will achieve the desired processing. Such a problem requires a different solution for each particular situation and is not discussed in this paper. There are many available references describing how to design digital filters for restoration, enhancement etc. [5-6].

The resulting filters can have either infinite (IIR) or finite impulse response (FIR). However, IIR filters possess certain properties that complicate their implementation. One problem is that of stability. It is generally, difficult to ensure bounded-input; bounded-output (BIBO) stability in multidimensional IIR filters. Yet FIR filters are always stable in the BIBO sense. (See Appendix A). This paper discusses FIR filters only.

It is well known that the output of a filter is the convolution of the filter impulse response and the input sequence. In the case of a finite input sequence, filtered by a FIR filter, we have

$$y(n_1,n_2) = \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} x(m_1,m_2)h(n_1-m_1,n_2-m_2)$$

where $x(n_1,n_2)$ is the input sequence of length M_1M_2, i.e.

$$x(n_1,n_2) = 0 \qquad \begin{matrix} n_1 < 0 \text{ or } n_1 > M_1 \\ n_2 < 0 \text{ or } n_2 > M_2 \end{matrix}$$

$h(n_1,n_2)$ is the impulse response of length $M_1'M_2'$
and $y(n_1,n_2)$ is the output sequence of length N_1N_2
where $N_1 = M_1'-1+M_1$
$N_2 = M_2'-1+M_2$

However, doing this convolution directly requires large number of operations. For each output sequence of length N_1,N_2, approximately $N_1^2N_2^2$ multiplications and additions are needed. There are faster and more efficient methods of achieving the same result. These are usually accomplished by the use of some mathematical transforms, so that a convolution can be replaced by simple multiplication.

III. MATHEMATICAL TRANSFORMS

The alternative to direct convolution is the use of mathematical transforms. The solution may require fewer operations in the transform domain, thus rendering it more practical.

Such an approach became practical when Cooly and Tudey first introduced a fast algorithm for performing the discrete Fourier transform and its inverse. A general DFT like transform in two-dimensions is

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1,n_2)\alpha_1^{n_1k_1}\alpha_2^{n_2k_2} \qquad (1)$$

where α_1 is a primitive N_1th root of unity
α_2 is a primitive N_2th root of unity.

The inverse transform is

$$x(n_1,n_2) = (N_1N_2)^{-1} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1,k_2)\alpha_1^{-n_1k_1}\alpha^{-n_2k_2} \qquad (2)$$

The expressions (1) and (2) can be shown to form a transform pair.

Note now that the transform of the output sequence is the point by point multiplication of the transforms of the two input sequences.

$$Y(k_1,k_2) = \sum_{n_1} \sum_{n_2} y(n_1,n_2)\alpha_1^{n_1k_1}\alpha_2^{n_2k_2}$$

$$= \sum_{n_1} \sum_{n_2} \sum_{m_1} \sum_{m_2} x(m_1,m_2)h(n_1-m_1,n_2,m_2)\alpha_1^{n_1k_1}\alpha^{n_2k_2}$$

$$= X(k_1,k_2)H(k_1,k_2) \qquad (3)$$

Therefore the output sequence can be obtained from the inverse transform of the point by point multiplication of the transforms of the input sequence. And

$$y(n_1,n_2) = (N_1,N_2)^{-1} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} X(k_1,k_2)H(k_1,k_2)\alpha_1^{-n_1k_1}\alpha_2^{-n_2k_2} \qquad (4)$$

By using fast transform algorithms, 1-dimensional transforms can be obtained using approximately $N\log_2 N$ multiplications and additions. Therefore the number of operations required to implement the convolution in the transform domain is approximately $3(N\log_2 N)+N$. For N large, this is much smaller than the N^2 operations required for direct convolution. Convolution by operations in the transform domain is usually preferred.

IV. NUMBER THEORETIC TRANSFORM (NTT)

In number theoretic transforms, all operations are done in modular arithmetic. That is, additions and multiplications are done modulo M (mod M) where M is an integer called the modulus. The allowable numbers are the set of integers 0,1,2,...M-2,M-1. This set is called a field or a ring. In modular arithmetic all numbers must be members of the above finite set. This is accomplished by a residue reduction operation such that

$$((x)) = X - rM \qquad (5)$$

where r is an integer and $0 \leq ((x)) \leq M-1$.

For a number in modular arithmetic to have multiplicative inverse, the modulus M must be a prime number. Therefore, the general NTT of a sequence of N_1N_2 integers

$$x(n_1,n_2);\ n_1 = 0,1,2,\ldots,\ N_1 - 1,\ n_2 = 0,1,\ldots,N_2 - 1$$

where modulus M is another sequence of N_1N_2 integers

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1,n_2)\alpha_1^{n_1k_1}\alpha_2^{n_2k_2} \quad (6)$$

where α_1 is relatively prime to M_1 and of order N_1 (i.e. $\alpha_1^{N_1} \equiv 1$)
α_2 is mutually prime to M_2 and of order N_2
and N^{-1} and N_2^{-1} (multiplicative inverses N_1 and N_2 respectively) exist,
$i\text{-}\alpha_1^i$, is mutually prime to M_1 for all $i = 1,2,\ldots,N-1$
$i\text{-}\alpha_2^i$, is mutually prime to M_2 for all $i = 1,2,\ldots,N-1$.

However, only NTT's with highly composite N_1 and N_2 can support an efficient FFT-like algorithm [8-9]. It is then necessary to find suitable α_1,α_2, M_1 and M_2 for such transforms. Unfortunately only a few classes of NTT's satisfy the above requirements. Two of these are the Fermat number transforms (FNT) and the Mersenne number transforms (MNT). FNT's use a Fermat number F_t as a modulus M, that is

$$F = F_t = 2^{2^t} + 1 \quad (7)$$

MNT's, on the other hand, use a modulus of a Mersenne number

$$M = M_p = 2^p - 1 \quad (8)$$

where P is a prime number.

Number theoretic transforms offer the advantages of integer arithmetic. All operations and intermediate steps result in integers which can be represented exactly in digital machines. Thus all roundoff and truncation errors are limited to those resulting from initial quantization of the input sequences. However, this advantage is offset by the restrictions on size of the modulus N and α. These difficulties make NTT's a subject suitable to individuals with advanced background in modular arithmetic and transforms. For those interested, the topic is discussed in [8-9].

V. DISCRETE FOURIER TRANSFORMS

This discrete Fourier transform (DFT) is an extension of the Fourier Transform. Following the general form of mathematical transforms, we have

$$DFT(x(n_1,n_2)) = X(k_1,k_2) = \sum_{n_2=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1,n_2)\exp(-j\frac{2\pi}{N_1} n_1k_1) \exp(-j\frac{2\pi}{N_2} n_2k_2) \quad (9)$$

$$IDFT(X(k_1,k_2))=x(n_1,n_2)=(N_1N_2)^{-1}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}X(k_1,k_2)\exp(j\frac{2\pi}{N_1}n_1k_1)\exp(j\frac{2\pi}{N_2}n_2k_2) \quad (10)$$

This transform assumes a periodicity such that

$$(n_1+rN_1,n_2+\ell N_2) = x(n_1,n_2) \qquad r,\ell = \ldots -1,0,1,2\ldots \quad (11)$$

The convolution property applies for DFT also such that

$$y(n_1,n_2) = IDFT\{DFT(x(n_1,n_2))\times DFT(h(n_1,n_2))\} \quad (12)$$

However to avoid erroneous results from the assumed periodicity, the DFT's should be performed with

$$N_1 = M_1 + M_1' - 1 \quad (13)$$
$$N_2 = M_2 + M_2' - 1$$

where $x(n_1,n_2)$ is a sequence of length $M_1 \times M_2$
$h(n_1,n_2)$ is a FIR of length $M_1' \times M_2'$
$y(n_1,n_2)$ is a sequence of length $N_1 \times N_2$

To be able to use the fast Fourier transform algorithm developed by Cooley and Tukey [10-11], N_1 and N_2 must be powers of 2 (radix 2). This restriction is much easier to satisfy than those on NTT's. Further work on these algorithms has also made it possible to have a radix 4 implementation. This method has been also extended to radix 3 FFT algorithms [12] and other extensions [13], which allow more flexibility in the choice of transform size.

VI. SOFTWARE AVAILABLE

We shall now present the various subprograms that are currently available at the University of Toronto's Computer Centre (UTCC). These are subroutines written mainly by J.Costa a Ph.D. student in Electrical Engineering.

One often used two-dimensional FFT subroutine is discussed in some detail. This is the program FFT2R and it operates on two dimensional data that is purely real. This type of data is encountered most often in the real world therefore making this subroutine of special interest.

1. FFT2R

In general, a two-dimensional sequence of data is presented by a set of complex numbers. Using the Fortran programming language, this would mean $2N^2$ storage locations are required to

record an NxN matrix of points. That is, each complex number would have been stored as a number pair of real and imaginary parts of magnitude and argument. However, much of the data collected in actual situations involve only real numbers. This means that only N^2 memory locations are required. In general, the transform of such a set of real data, is a set of complex numbers, which requires $2N^2$ storage points. For an algorithm that produces the transform in place, it would seem that storage requirement is not reduced even if the input vector is purely real. Further examination reveals that the transform of a two-dimensional real matrix is characterized by certain symmetry properties. The two-dimensional transform is defined as

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1,n_2) e^{-j\frac{2\pi}{N_1} k_1 n_1} e^{-j\frac{2\pi}{N_2} k_2 n_2}$$

where n_1 and n_2 are indices of the matrix with dimension N_1 and N_2 respectively

$x(n_1,n_2)$ is the two-dimensional input sequence and $X(k_1,k_2)$ is the two-dimensional transformed sequence. Now examine

$$X(N_1-k_1,N_2-k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1 n_2) e^{-j\frac{2\pi}{N_1}(N_1-k_1)n_1} e^{-j\frac{2\pi}{N_2}(N_2-k_2)n_2}$$

$$= \left[\sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x^*(n_1,n_2) e^{-j\frac{2\pi}{N_1} k_1 n_1} e^{j\frac{2\pi}{N_2} k_2 n_2} \right]$$

$$= X^*(k_1,k_2)$$

where * indicates the complex conjugate of a number. The above symmetry property, together with the assumed periodicity property means that the transform of a NxM real sequence will result in $(\frac{NM}{2} + 2)$ distinct numbers, four of which are real numbers. These numbers can therefore be stored in the same memory space used to store NM real numbers. This observation is used in the subroutine FFT2R to conserve storage requirements. However, slightly more than NM storage locations were used to reduce the processing complexity.

The actual memory allocation techniques are best explained by way of an illustration. Suppose we wish to obtain the DFT of a real two-dimensional sequence of data, a(x,y). That is we want

$$A(x,y) = DFT[a,(x,y)]. \qquad (14)$$

The pionts a(x,y) are stored in a FORTRAN array as follows:

$$X(I,J)\ ;\quad 1 \le I \le M, \qquad 1 \le J \le N \tag{15}$$

Therefore X(1,1) is the value of a (0,0) - the value at the origin. To illustrate this M and N is chosen to equal 8, i.e. M=N=8. Generalization to the other powers of 2 is simple.

The real function $a(x,y)$, $-\frac{M}{2} < x \le \frac{M}{2}$, $-\frac{N}{2} < y \le \frac{N}{2}$ is stored as shown in Figure 1.

$$\begin{bmatrix}
a(0,0) \to X(1,1) & a(0,1) \to X(1,2) & a(0,2) \to X(1,3) \ldots a(0,-1) \to X(1,8) \\
a(1,0) \to X(2,1) & a(1,1) \to X(2,2) & a(1,2) \to X(2,3) \ldots a(1,-1) \to X(2,8) \\
a(2,0) \to X(3,1) & a(2,1) \to X(3,2) & a(2,2) \to X(3,3) \ldots a(2,-1) \to X(3,8) \\
a(3,0) \to X(4,1) & a(3,1) \to X(4,2) & a(3,2) \to X(4,3) \ldots a(3,-1) \to X(4,8) \\
a(4,0) \to X(5,1) & a(4\ 1) \to X(5,2) & a(4,2) \to X(5,3) \ldots a(4,\ 1) \to X(5,8) \\
a(-3,0) \to X(6,1) & a(-3,1) \to X(6,2) & a(-3,2) \to X(6,3) \ldots a(-3,-1) \to X(6,8) \\
a(-2,0) \to X(7,1) & a(-2,1) \to X(7,2) & a(-2,2) \to X(7,3) \ldots a(-2,-1) \to X(7,8) \\
a(-1,0) \to X(8,1) & a(-1,1) \to X(8,2) & a(0,2) \to X(8,3) \ldots a(-1,-1) \to X(8,8)
\end{bmatrix}$$

Figure 1. Storege allocation of (8,8) data sequence

VII. EXAMPLE

In Figure 2 to 4 an example achieving two dimensional filtering of a noisy input is presented. Below are a set of print-outs showing pictorially the input sequence, filter characteristic and the filtered output.

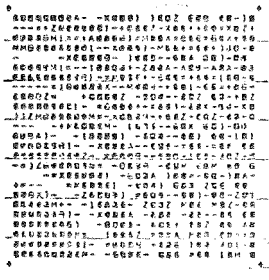

Figure 2
Input sequence

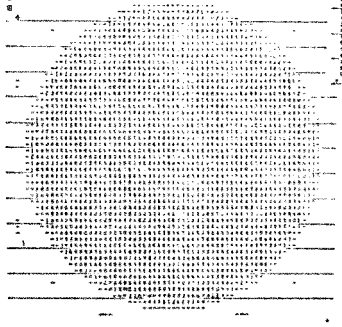

Figure 3 Filter characteristic

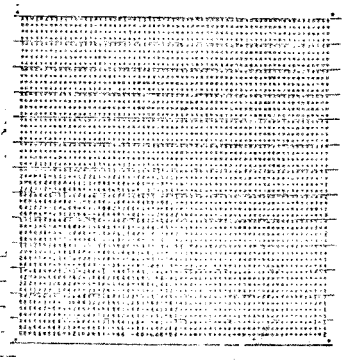

Figure 4
Filtered output

VIII. CONCLUSIONS

This paper has dealt with digital filtering which is a very powerful tool in data processing. The emphasis has been placed on two-dimensional data, as it is both useful and conceptually easy to visualize. Details on how such filtering can be achieved by software was presented, with special discussion on a two-dimensional FFT program available at UTCC. It is hoped that the reader will find out more about these subroutines and make use of them. The usefulness and power of digital filtering can only be fully appreciated by actual experience with such methods.

APPENDIX A - BIBO stability of two-dimensional FIR filters

A two-dimensional finite impulse response (FIR) filter is defined by an impulse response sequence h(m,n) such that

$$h(m,n) = 0 \text{ for } m < 0 \text{ or } m > N_1 \text{ or } n < 0 \text{ or } n > N_2 \qquad (a.1)$$

gives an input sequence x(m,n), the output of the filter is

$$y(k,\ell) = \sum_{m=0}^{N_1} \sum_{n=0}^{N_2} h(m,n)\, x(k-m,\ \ell-n) \qquad (a.2)$$

The magnitude of the output sequence is therefore

$$|y(k,\ell)| = \left| \sum_{m=0}^{N_1} \sum_{n=0}^{N_2} h(m,n)\, x(k-m,\ell-n) \right|$$

$$\leq \sum_{m=0}^{N_1} \sum_{n=0}^{N_2} |h(m,n)|\,|x(k-m,\ell-n)| \qquad (a.3)$$

If the input sequence is bounded such that

$$|x(m,n)| \leq B \text{ for all } m,n \qquad (a.4)$$

then it follows that

$$|y(k,\ell)| \leq B \sum_{m=0}^{N_1} \sum_{n=0}^{N_2} |h(m,n)| < \infty \qquad (a.5)$$

Therefore, for a bounded input, the output of a two-dimensional FIR filter is always bounded. Thus bounded-input-bounded-output (BIBO) stability is guaranteed for all FIR filters.

REFERENCES

1. Stanley, W.D.: 1975, Digital Signal Processing, Reston Publishing Co.
2. Ackroyd, M.H.: 1973, Digital Filters, Butterworths, London.
3. Bogner, R.E. and Constantinides, A.G. (editor): 1975, Introduction to Digital Filtering, Wiley.

4. Rabiner, L.R. and Gold, B.: 1975, Theory and Application of Digital Signal Processing, Prentice Hall.
5. Hunt, B.R. and Andrews, H.C.: 1977, Digital Image Restoration, Prentice Hall.
6. Rosenfeld, A. and Kak, A.C.: 1969, Digital Picture Processing, Academic Press, N.Y.
7. Oppenheim, A.V. (editor): 1969, Papers on Digital Signal Processing, MIT Press.
8. Agarwal, R.C. and Burrus, C.S.: 1974, "Fast Convolution Using Fermat Number Transforms with Application to Digital Filtering", IEEE Transactions on Acoustic Speech and Signal Processing, ASSP-22, pp. 87-97.
9. Dubois, E. and Venetsanopoulos, A.N.: 1978, "The Discrete Fourier Transform Over Finite Rings with Application to Fast Convolution", IEEE Trans. on Computers, C-27, 7, pp. 586-59[illegible]
10. Brigham, E.O.: 1974, The Fast Fourier Transform, Prentice-Hall.
11. Cooley, J.W. and Tukey, J.W.: 1965, "An Algorithm for Machine Calculation of Complex Fourier Series", Math. Computation,19, pp. 297-301.
12. Venetsanopoulos, A.N. and Dubois, E.: 1968, "A Radix 3 FFT with No Multiplications in the 3 Point DFT's", Trans. on Acoustics Speech and Signal Processing, ASSP-26, 3.
13. Rabiner, L.R. and Rader, C.M.: 1973, Digital Signal Processing, IEEE Press.
14. Hamming, R.W.: 1977, Digital Filters, Prentice-Hall, N.J.
15. Waung, W., Costa, J.M. and Venetsanopoulos, A.N.: 1979, "Two Dimensional Digital Signal Processing - Introductory Theory and Subroutines for Implementation", Communications Tech. Report No. 79-1, University of Toronto, Toronto, Canada