AN ABSTRACT OF THE THESIS OF

Zhu, Feihong for the degree of Master of Science in Electrical & Computer Engineering presented on Mar. 9, 1987.

Title: Two Dimensional Recursive Fast Fourier Transform

Abstract approved: Redacted for Privacy

Wojciech J. Kolodziej

The purpose of this thesis is to extend one dimensional recursive fast Fourier transform (1-D RFFT) into two dimensions, by which the two dimensional signal can be processed recursively and the desired spectrum can be obtained.

Compared with FFT, RFFT has the following advantages: (1) No requirement that the number of input data should be equal to the number of frequencies. (2) Fourier transform can start before all the data are obtained. The latter makes this algorithm suitable for on-line spectral analysis.

In this thesis, a two dimensional RFFT algorithm is proved mathematically, and an application to signal detection is demonstrated. Also, this algorithm can be applied to image processing and other two dimensional signal processing problems.

TWO DIMENSIONAL RECURSIVE FAST FOURIER TRANSFORM

by

ZHU, FEIHONG

A THESIS

submitted to

Oregon State University

in partial fulfilment of

the requirements for the

degree of

Master of Science

Completed March 9, 1987

Commencement June, 1987

APPROVED:

Redacted for Privacy

Dr. W. J. Kolodziej, Department of Electrical & Computer Engineering, in charge of major

Redacted for Privacy

Dr. S. J. T. Owen, Chairman of Department of Electrical & Computer Engineering

Redacted for Privacy

Dean of Graduate School

Date thesis is presented March 9, 1987

ACKNOWLEDGEMENT

Special gratitude is due to my major professor Dr. Wojciech J. Kolodziej for his great help and support during the period of doing my thesis.

I would also like to thank the other members of my committee, Dr. Herzog J. H., Dr. Mohler R.R. and Dr. Fredericks William J., for their interest and helpful suggestions.

I especially thank my loving wife, Xu Yan, your constant support and sacrifice made this endeavor a reality. I am very proud and fortunate to have a wife like you.

A big thanks goes to Mr. Fred M. Quick and Mrs. Evelyn A. Quick whose support and never-ending encouragement made this thesis possible.

My mother, Mi Xurong, my father Zhu Peiji and my aunt, Zhu Wanying, deserve special thanks for your support, encouragement and love you have given to me throughout my life.

Also I wish to express deep gratitude to my brother, Zhu Changhong, and other members of my family for their continuous understanding and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TWO DIMENSIONAL RECURSIVE FAST FOURIER TRANSFORM

CHAPTER I

INTRODUCTION

Since the appearance of fast Fourier transform (FFT) (Cooley/Tukey, 1965) (Cooley/Cochran, 1967), many researchers have worked in this field and invented a variety of FFT algorithm implementations. FFT is an efficient method of computing discrete Fourier transform (DFT), but it requires that all the data sequence be obtained before the start of transformation. This limitation makes FFT difficult to be used for on-line applications because sometimes the duration of signal is unknown. Therefore the search of recursive algorithms of Fourier transform is attractive.

For one dimensional DFT, Ahmed (Ahmed/Natarajian/Rao, 1973) presented a recursive algorithm which was derived from the " mirror image " of the signal. By processing the input data one by one, this method can be used for on-line spectral identification. Hostetter (Hostetter, 1980) constructed another recursive algorithm based on the observer concept of control theory.

His first paper prompts more efforts in this area (Hostetter, 1983) (Bitmead, 1982). These approaches have recursive property, but the transform speed is low because the data are processed one by one. Therefore, a question is raised: how to find a method which has both fast and recursive properties. For this problem, a moderate method, recursive fast Fourier transform (RFFT), was presented (Zhu, 1985) (Zhu/Wang, 1986). This mehtod is developed from Ahmed's method, but the input data are processed segment by segment recursively, as opposed to one by one, and a revised fast algorithm is used in processing every segment.

Compared with FFT, this method has the following advantages:

(1) Flexibility: The number of input data in every segment may not equal the number of frequencies.

(2) Recursive property: Due to no limitation on recursive steps, the desired spectrum can be obtained as the input data increase.

Compared with other recursive methods mentioned above, this RFFT approach has higher transform speed because a revised fast algorithm is used in every recursive computation. The main idea of the algorithm is to transform the new data segment to frequencies by means of revised FFT algorithm and then to "repair" the old spectrum which is obtained by previous data. Although the total computational operations of RFFT are more than those of FFT, its transform speed may not be low because the part of computation can be finished between the sampling intervals.

For two dimensional DFT, the common methods used so far are still FFT type. The purpose of this thesis is to extend one dimensional RFFT to two dimensional case because recursive algorithm is also attractive for two dimensional signal processing.

The organization of this thesis is as follows. In chapter II, a literature review is given, which introduces the development of fast Fourier transform, recursive Fourier transform and other related algorithms for the computation of DFT. Chapter III presents the algorithm of two dimensional recursive fast Fourier transform. This algorithm consists of two parts: the recursive algorithm and the revised fast algorithm. Chapter IV describes numerical experiment results, including a simulation of signal detection by RFFT. In chapter V, the conclusion and discussion of possible applications of RFFT algorithm are given.

CHAPTER II

LITERATURE REVIEW

1. DEVELOPMENT OF FFT

Fourier transform is an essential analysis method in many scientific and engineering fields. One of the reasons that Fourier transform have such wide-ranging applications is because of the existence of powerful digital computers and efficient algorithms for computing discrete Fourier transform.

The definition of DFT is given by

$$F(k)=\sum_{n=0}^{N-1} x(n)W^{kn} , \qquad k=0,1,...,N-1, \qquad (2-1)$$

where $W=\exp(-j2\pi/N)$ and $x(n)$, $n=0,1,2,...,N-1$, is the sample sequence of a time signal.

Inverse discrete Fourier transform (IDFT) is given by

$$x(n)=(1/N)\sum_{n=0}^{N-1} F(k)W^{kn}, \qquad n=0,1,...,N-1. \qquad (2\text{-}2)$$

To indicate the importance of efficient computational procedures, it is necessary to consider the direct calculation of DFT equations. Since x(n) may be complex, it can be written as follows:

$$F(k)=\sum_{n=0}^{N-1} \{(Re[x(n)]Re[W^{kn}]-Im[x(n)]Im[W^{kn}])$$

$$+j(Re[x(n)]Im[W^{kn}]+Im[x(n)]Re[W^{kn}])\},$$

$$k=0,1,...,N-1. \qquad (2\text{-}3)$$

From Eq.(2-3), it can be noticed that for each value of k, the direct computation of F(k) requires 4N real multiplications and 4N-2 real additions. Since F(k) must be computed for N different values of k, the direct computation of DFT for a sequence x(n) requires $4N^2$ real multiplications and (4N-2)N real additions. It is evident that the number of arithmetic operations required to compute DFT by the direct method becomes very large for the large values of N because the amount of the computation is approximately proportional to N^2. For this reason, computational procedures that reduce the number of the arithmetic operations are of considerable interest.

Most algorithms which improve the efficiency of DFT computation depend on one or both of the following special properties of the quantity W:

1. $W^{k(N-n)} = (W^{kn})^*$,

2. $W^{kn} = W^{k(n+N)} = W^{(k+N)n}$,

where $(W)^*$ is the complex conjugate of W.

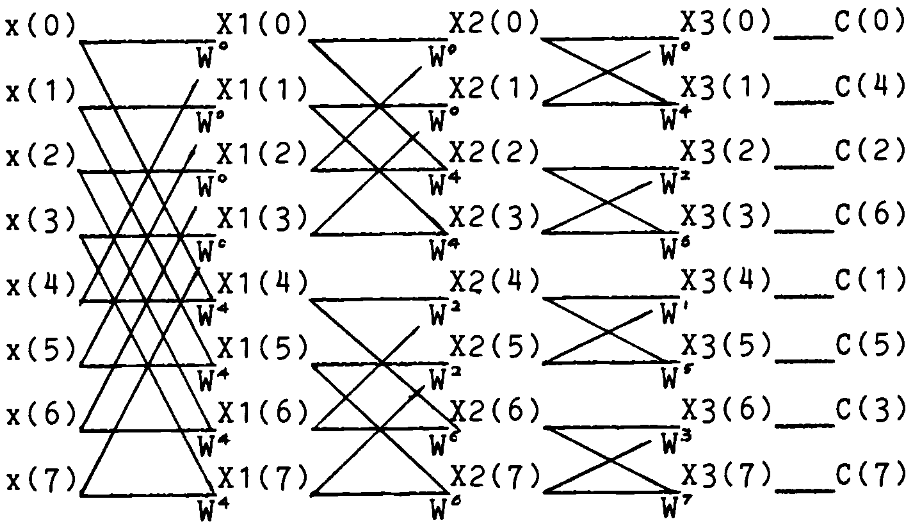
Computational algorithms that use both (1) the symmetry and (2) the periodicity of the sequence W were known long before the appearance of high-speed digital computation for Fourier transform (Colley/Cochcran, 1967). However this is not of great importance for the small values of N that are feasible for hand computations. The possibility of greatly reduced computation was generally overlooked until about 1965, when Cooley and Tukey (Cooley/Tukey, 1965) published the famous algorithm of fast Fourier transform. This algorithm requires much less computational effort and can be applied when N is a composite number (i.e., N is a product of two or more integers). The publication of this paper makes it possible to apply discrete Fourier transform to signal processing and results in the discovery of a variety of FFT algorithms.

The fundamental principle that FFT algorithms are based on is that of decomposing the computations of DFT sequence of length N into successively smaller DFT's. The manner in which the principle is implemented leads to many different algorithms, all with some improvements in computational speed. For the sake of comparison with RFFT, two basic classes of FFT are introduced: decimation-in-time algorithm and decimation-in-frequency algorithm.

To achieve the dramatic increase in efficiency which has been mentioned, it is necessary to decompose DFT computation into successively smaller DFT computations. In this process both the properties of the symmetry and the periodicity of complex exponential $W^{kn}=\exp(-j(2\pi/N)kn)$ are used. Algorithms where the decomposition is based on decomposing the sequence x(n), into successively smaller subsequences, are called decimation-in-time algorithms. They are originally discovered by Cooley and Tukey (Cooley/Tukey, 1965).

Alternatively, the computational methods can be considered dividing the output sequence F(k) into smaller and smaller subsequences in the same manner. The class of FFT algorithms based on this procedure is commonly referred to as decimation-in-frequency (Oppenheim/Schafer, 1975).

The following is a flow chart of the decimation-in-frequency algorithm (N=8).



8-point Butterfly Diagram of General FFT

Fig. 1

In Fig.1, DFT is computed by first forming the sequence X1(n) with X1(n)=X(n)+X(n+N/2) and X1(n+N/2)=X(n)-X(n+N/2), n=0,1,2,3, which results in two N/2-point DFTs, then computing X2(n) in the same manner, resulting in four N/4-point DFTs which reduce the computation to the two-point DFTs and can be implemented by addition and subtraction. The computation of FFT requires $N/2Log_2N$ complex multiplications and $NLog_2N$ complex additions. Thus the total computation is much less than that of direct computation of DFT for a large N.

Based on these principles, several algorithms have been developed to speed up the transform even further. The algorithm proposed in 1982 (Preuss, 1982) reduces the number of multiplications and shows that the DFT of a complex sequence can be computed from the DFT of four real sequences which satisfies the appropriate symmetry conditions. Another algorithm is based on the fact that a radix-2 (i.e., N is a power of 2) algorithm diagram can be transformed into a radix-4 algorithm diagram. This process requires fewer multiplications and additions (Duhamel/Hollmann, 1984). Also, an approach is proposed for the implementation of radix-2 FFT algorithm in multiprocessors to increase the efficiency of the transform (Bhuyan/Agrawal, 1983). The methods discussed above are radix-2 algorithms, the decomposition of which leads to a highly efficient computation of DFT. However, in some cases it may not be possible to choose N to be a power of 2. Thus it is necessary to consider the application in the case where N is a product of factors that are not all necessarily equal to 2 (Gentleman, 1966) (Singleton, 1969).

For the Fourier transform at arbitrary frequencies, an algorithm proposed is based on the fact that the Fourier transform at an arbitrary frequency can be expressed as a weighted sum of its DFT coefficients. The method retains the computational order of FFT but allows the flexibility of choosing arbitrary frequencies for a uniformly sampled signal (Sudhakar, 1981).

Besides FFT, two additional methods have been developed recently for the computation of DFT: the polynomial transforms (Pei/Wu, 1981) (Wu/Pei, 1984) (Nussbaumer, 1979) and estimator methods (Charles, 1984) (Charles, 1982). The underlying idea of the estimator methods is to approximate the designed estimator by iteratively solving the corresponding matrix equation, which makes it computationally feasible for the large system equations.

For two dimensional DFT, general method is by using one dimensional FFT repeatedly. The computational procedures consist of three steps: First, each row in the array is transformed with one dimensional FFT. Second, the array is transposed. Finally, the row transform process is repeated. This three-step procedure yields the transposition of the two dimensional transform array (Mitra, 1978). In addition, the contribution of Harris (Harris et al, 1977) described a calculation approach avoiding matrix transposition for the two dimensional FFT, by means of decomposing the two dimensional transform into successively smaller two dimensional transforms. Mersereau proposed yet another transformation procedure (Mersereau, 1974) to change two dimensional sequences into one dimensional ones. Using the mapping method, several two dimensional problems can be solved by one dimensional techniques.

In contrast to one dimensional DFT, the computations required by two dimensional DFT may place severe demands on even the largest computers. This is because of the large size data arrays usually encountered in practical applications. For this problem, Eklundh presented a fast computer method for matrix transposition. This method enables applications of FFT to the matrices, the size of which exceeds available main storage (Eklundh, 1972). A similar approach was proposed later for the transposition of non-square matrices (Schumann, 1973). Another algorithm presented by Twogood and Ekstrom is a more efficient extension of Eklundh's basic method (only two rows of the matrix are resident in the primary storage at any one time) (Twogood/Ekstrom, 1976). Also, the algorithms developed by Onoe (Onoe, 1975), Dellotto and Dotti (Dellotto/Dotti, 1975), Hinton and Saleh (Hinton/Saleh, 1984) are worth being mentioned here.

2. RECURSIVE ALGORITHM OF DFT

Although FFT is an efficient method of Fourier transformation, it requires that the number of time sequence elements be equal to the number of frequencies and the computation begins only after all the time sequence is obtained. In order to process signals on-line, the recursive computation of DFT has been also developed in recent years. Ahmed, Natarajan and Rao presented (Ahmed/Natarjan/Rao, 1973) an algorithm derived from the discrete Fourier transform formula by using the "mirror image" of time-sample series. The main recursive formula of the algorithm is

$$Z(w,s)=L(w)Z(w,s-1)+BX(s), \quad s=0,1,2,....,N-1, \qquad (2-4)$$

where $X(s)$, $s=0,1,...,N-1$, is the sample sequence of a time signal, w is the frequency, and

$$Z(w,-1)=\begin{bmatrix}0\\0\end{bmatrix}, \qquad B=\begin{bmatrix}1\\0\end{bmatrix}, \qquad L(w)=\begin{bmatrix}\cos w\Delta t & -\sin w\Delta t\\ \sin w\Delta t & \cos w\Delta t\end{bmatrix},$$

Δt is the sample interval.

This method provides a simple means of generating frequency amplitude plots of Fourier power and phase spectra recursively, which can be used in the spectral analysis of the signals whose duration is unknown. It was emphasized that FFT would be the most efficient way to compute spectra if the following constraints were satisfied:

i) The number of data N equals the number of frequencies A, i.e., N=A, and $N=2^n$, where n is a positive integer.

ii) $w=2\pi k/\Delta tA$, $k=0,1,2,......,(A-1)$.

However, using Eq.(2-4), these constraints could be removed. Another recursive DFT algorithm proposed by Hostetter (Hostetter, 1980) is based on the concept of the observer of control theory. The paper is developed from the fact that a linear, time invariant homogeneous differential equation with the characteristic polynomial

$$s(s^2+(2\pi/T)^2)(s^2+(4\pi/T)^2)......(s^2+(2N\pi/T)^2),$$

has a general solution of the form

$$y(t)=d_0+\sum_{n=1}^{N}(a_n\cos(2\pi nt/T)+b_n\sin(2\pi nt/T)), \qquad (2\text{-}5)$$

where d_0, a_n and b_n are the Fourier coefficients.

It is shown that the state variable representation for this homogeneous system is of the following form:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_{2N-1} \\ \dot{x}_{2N} \\ \dot{x}_{2N+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \;.\;.\;. & 0 & 0 \\ -(2\pi/T)^2 & 0 & 0 & 0 \;.\;.\;. & 0 & 0 \\ 0 & 0 & 0 & 1 \;.\;.\;. & 0 & 0 \\ 0 & 0 & -(4\pi/T)^2 & 0 \;.\;.\;. & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot\;\cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot\;\cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot\;\cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\;\cdot\;\cdot\;\cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -(2N\pi/T)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{2N-1} \\ x_{2N} \\ x_{2N+1} \end{bmatrix}$$

$$= AX, \qquad (2\text{-}6)$$

$$Y=[1\ 0\ 1\ 0\ \ldots\ldots 1\ 0\ 1]X=C^T X. \qquad (2\text{-}7)$$

From the viewpoint of this representation, deciding the coefficients of DFT is equivallent to the determination of the system initial conditions (i.e., to the classical deterministic observer problem).

Also, Hostetter proposed a recursive technique for Fourier spectral analysis, when the samples were unevenly spaced in time (Hostetter, 1983), and an approach to the inverse DFT (Hostetter, 1984), both of which are based on the observer of state variables. The latter gives a recursive solution for general discrete linear transform and shows that DFT is a special case of this algorithm.

In 1982, Bitmead (Bitmead, 1982) published a paper, the aim of which is to demonstrate that the approach by Hostetter yields the final results essentially equivalent to standard signal processing methods. He makes explicitly the connections between them and points out the possible extensions.

In addition, Stuller (Stuller, 1982) presented an approach for recursive discrete transform with respect to arbitrary transform bases. He relates the generalized transform to recursive discrete Fourier transform and points out that it is a special case of this generalized transformation.

As discussed above, every method has its own advantages, but slow computation speed is their common disadvantage although the recursive method is attractive to on-line applications. And for the Hostetter's method, practical realizations limit its applications due to a high dimension of observer needed for the large number of the coefficients of the Fourier transform. To overcome these disadvantages and keep the recursive property, an algorithm "Recursive Fast Fourier Transformation" (RFFT) of one dimension has been proposed (Zhu, 1985) (Zhu/Wang, 1986). This algorithm combines the advantages of FFT and the recursive computation of DFT. By processing all data segment by segment recursively and using the revised fast algorithm for every segment, this algorithm has high transform speed and can be used for on-line applications.

This thesis is focused on two dimensional recursive fast Fourier transform (2-D RFFT), which is developed from one dimensional RFFT. In addition, an application to signal detection is given.

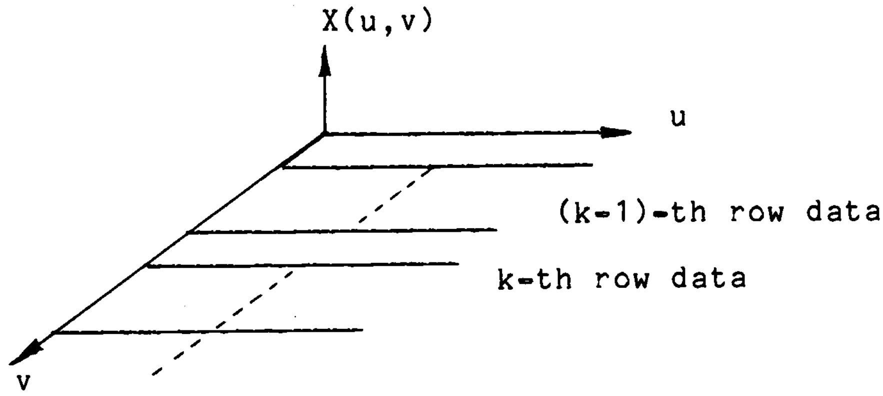
CHAPTER III

TWO DIMENSIONAL RFFT

1. RECURSIVE ALGORITHM OF RFFT

The main idea of the two dimensional recursive fast Fourier transform is to transfer the k-th row data into the one dimensional spectrum by means of one dimensional revised FFT or RFFT and then use the recursive algorithm to "repair" the old spectrum which is obtained from previous data (see Fig.2). This process is repeated until all the data are transformed or the desired spectrum is obtained.

Next, the construction of the recursive algorithm will be discussed according to this idea.



Two Dimensional Data Array

Fig.2

DEFINITION 1:

A two dimensional discrete signal sequence is defined as follows:

$$x^*(m1,m2)=\sum_{m2}\sum_{m1} X(u,v)\delta(u-m1\Delta u)\delta(v-m2\Delta v), \quad (3-1)$$

$$m1=0,1,2,...,N1-1, \quad m2=0,1,2,...,N2-1,$$

where, X(u,v) is a two dimensional signal function, Δu and Δv are the sample intervals along horizontal and vertical directions respectively and $\delta(\ )$ is the impulse function.

DEFINITION 2:

The mirror function of x*(m1,m2) is given by

$$x^{**}(m1,m2)=\sum_{m2}\sum_{m1} X(N1-1-u,N2-1-v)\delta(u-(N1-1-m1)\Delta u)\delta(v-(N2-1-m2)\Delta v). \quad (3-2)$$

LEMMA 1:

Let F(w1,w2) and F1(w1,w2) denote the Fourier transforms of X*(m1,m2) and X**(m1,m2) respectively:

$$F(w1,w2)=\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1}X(m1,m2)\exp(-jm1w1\Delta u)\exp(-jm2w2\Delta v), \quad (3-3)$$

$$F1(w1,w2)=\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1} X(N1-1-m1,\ N2-1-m2)\exp(-jm1w1\Delta u) \exp(-jm2w2\Delta v). \quad (3-4)$$

Then the relationships between the amplitude and phase spectra of F1(w1,w2) and F(w1, w2) are as follows

$$|F(w1,w2)|=|F1(w1,w2)|, \quad (3-5)$$

$$\Phi(w1,w2)=-(N1-1)w1\Delta u-(N2-1)w2\Delta v-\Phi1(w1,w2), \quad (3-6)$$

where, Φ(w1,w2) is the phase spectrum of F(w1,w2) and Φ1(w1,w2) is that of F1(w1,w2).

PROOF:

Eqs.(3-3)(3-4) can be proved directly from the definition of Fourier transform.

Let y1=N1-1-m1 and y2=N2-1-m2, from Eq.(3-4), it follows

$$F1(w1,w2)=\sum_{y2=0}^{N2-1}\sum_{y1=0}^{N1-1} X(y1,y2)\exp(-j(N1-1-y1)w1\Delta u) \exp(-j(N2-1-y2)w2\Delta v)$$

$$=\sum_{y2=0}^{N2-1}\sum_{y1=0}^{N1-1} X(y1,y2)\exp(jy1w1\Delta u)\exp(jy2w2\Delta v) \exp(-j(N1-1)w1\Delta u)\exp(-j(N2-1)w2\Delta v), \quad (3-7)$$

Compared Eq.(3-7) with Eq.(3-3), it follows that

$$F1(w1,w2)=F^*(w1,w2)\exp(-j(N1-1)w1\Delta u)\exp(-j(N2-1)w2\Delta v),$$

where $F^*(w1,w2)$ is the complex conjugate of $F(w1,w2)$.

Thus, Eqs.(3-5)(3-6) are proved.

QED

Represent $$F1(w1,w2)=R(w1,w2)-jI(w1,w2), \qquad (3\text{-}8)$$

where

$$R(w1,w2)=\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1} X(N1-1-m1,\ N2-1-m2)\cos(m1w1\Delta u+m2w2\Delta v),$$

$$I(w1,w2)=\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1} X(N1-1-m1,\ N2-1-m2)\sin(m1w1\Delta u+m2w2\Delta v).$$

Define a 2-dimensional vector

$$\mathbf{F1}(w1,w2)=\begin{bmatrix} R(w1,w2) \\ I(w1,w2) \end{bmatrix} \qquad (3\text{-}9)$$

Let

$$L1(w1)=\begin{bmatrix} \cos(w1\Delta u) & -\sin(w1\Delta u) \\ \sin(w1\Delta u) & \cos(w1\Delta u) \end{bmatrix}, \qquad (3\text{-}10)$$

$$L2(w2)=\begin{bmatrix} \cos(w2\Delta v) & -\sin(w2\Delta v) \\ \sin(w2\Delta v) & \cos(w2\Delta v) \end{bmatrix}. \qquad (3\text{-}11)$$

It can be shown (Ahmed & Rao, 1975) that L1(w1) and L2(w2) are orthgonal, and hence have the property

$$L1^{m1}(w1)=\begin{bmatrix} \cos(w1\Delta u) & -\sin(w1\Delta u) \\ \sin(w1\Delta u) & \cos(w1\Delta u) \end{bmatrix}^{m1}$$

$$=\begin{bmatrix} \cos(m1w1\Delta u) & -\sin(m1w1\Delta u) \\ \sin(m1w1\Delta u) & \cos(m1w1\Delta u) \end{bmatrix}, \qquad (3-12)$$

$$L2^{m2}(w2)=\begin{bmatrix} \cos(w2\Delta v) & -\sin(w2\Delta v) \\ \sin(w2\Delta v) & \cos(w2\Delta v) \end{bmatrix}^{m2}$$

$$=\begin{bmatrix} \cos(m2w2\Delta v) & -\sin(m2w2\Delta v) \\ \sin(m2w2\Delta v) & \cos(m2w2\Delta v) \end{bmatrix}. \qquad (3-13)$$

Hence

$$\mathbf{F1}(w1,w2)=\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1} L2^{m2}(w2)L1^{m1}(w1)eX(N1-1-m1,\ N2-1-m2), \qquad (3-14)$$

where

$$e=\begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Denote

$$\mathbf{F2}(w1,N2-1-m2)=\sum_{m1=0}^{N1-1} L1^{m1}(w1)eX(N1-1-m1,\ N2-1-m2). \qquad (3-15)$$

Then, (3-14) can be rewritten as

$$\mathbf{F1}(w1,w2)=\sum_{m2=0}^{N2-1} L2^{m2}(w2)\sum_{m1=0}^{N1-1} L1^{m1}(w1)eX(N1-1-m1,\ N2-1-m2)$$

$$=\sum_{m2=0}^{N2-1} L2^{m2}(w2)\mathbf{F2}(w1,N2-1-m2). \qquad (3-16)$$

<u>LEMMA 2</u>:

Define the recursive formula

$$Z2(w1,w2,S2)=L2(w2)Z2(w1,w2,S2-1)+\mathbf{F2}(w1,S2),$$

$$Z2(w1,w2,-1)=\begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad (3-17)$$

It follows that

$$Z2(w1,w2,N2-1)=\sum_{m2=0}^{N2-1} L2^{m2}(w2)\mathbf{F2}(w1,N2-1-m2)$$

$$=\mathbf{F1}(w1,w2). \qquad (3-18)$$

PROOF: (using mathematical induction)

1. For S2=0, $Z2(w1,w2,0)=\mathbf{F2}(w1,0)$.

2. Assume, $Z2(w1,w2,k)=\sum_{m2=0}^{k} L2^{m2}(w2)\mathbf{F2}(w1,k-m2)$ when S2=k.

3. Hence, when S2=k+1,

$$Z2(w1,w2,k+1)=L2(w2)Z2(w1,w2,k)+\mathbf{F2}(w1,k+1)$$

$$=\sum_{m2=0}^{k} L2^{m2+1}(w2)\mathbf{F2}(w1,k-m2)+\mathbf{F2}(w1,k+1)$$

$$=\sum_{d=0}^{k+1} L2^{d}(w2)\mathbf{F2}(w1,k+1-d),$$

where, m2+1=d.

Therefore, when S2=N2-1, using (3-16), we have

$$Z2(w1,w2,N2-1)=\sum_{m2=0}^{N2-1} L2^{m2}(w2)\mathbf{F2}(w1,N2-1-m2)$$

$$=\mathbf{F1}(w1,w2).$$

QED

NOTE:

For every S2, **F2**(w1,S2) is an one dimensional DFT, which can be computed by one dimensional FFT or RFFT.

LEMMA 3: For one dimensional RFFT, the recursive formula for **F2**(w1,S2) is

$$Z1(w1,hN-1,S2)=L1^{N}(w1)Z1(w1,(h-1)N-1,S2)$$

$$+L\mathbf{X}(hN-1,(h-1)N,S2), \qquad h=1,2,\ldots\ldots H,$$

$$Z1(w1,-1,S2)=[\ 0 \quad 0\ ]^{T}, \qquad (3\text{-}19)$$

where h counts the recursive steps along the u axis, N is the number of data for every recursive step (HN=N1) and

$$L=[L1^{0}(w1)e \quad L1^{1}(w1)e\ldots\ldots L1^{N-1}(w1)e], \qquad (3\text{-}20)$$

$$\mathbf{X}(hN-1,(h-1)N,S2)$$

$$=[X(hN-1,S2),X(hN-2,S2)\ldots X((h-1)N,S2)]^{T}. \qquad (3\text{-}21)$$

PROOF:

Using the same formula as in lemma 1 for **F2**(w1,S2), we have

$$Z1(w1,S1,S2)=L1(w1)Z1(w1,S1-1)+eX(S1,S2),$$

$$Z1(w1,-1,S2)=\begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad (3\text{-}22)$$

Therefore,

$$Z1(w1,N1-1,S2)=\sum_{m1=0}^{N1-1} L1^{m1}(w1)eX(N1-1-m1,S2)=\mathbf{F2}(w1,S2).$$

In Eq.(3-22), let S1=hN-1; it can be shown that

$$F2(w1,S2)=Z1(w1,hN-1,S2)=\sum_{m1=0}^{hN-1} L1^{m1}(w1)eX(hN-1-m1,S2). \tag{3-23}$$

Write Eq.(3-23) in the matrix form as follows:

For h=1

$$Z1(w1,N-1,S2)=[L1^{0}(w1)e\ L1^{1}(w1)e\ldots L1^{N-1}(w1)e]\begin{bmatrix} X(N-1,S2)\\ X(N-2,S2)\\ \cdot\\ \cdot\\ \cdot\\ X(1,S2)\\ X(0,S2)\end{bmatrix}. \tag{3-24}$$

For h=2

$$Z1(w1,2N-1,S2)=[L1^{0}(w1)e..L1^{N-1}(w1)e\quad L1^{N}(w1)e..L1^{2N-1}(w1)e]\begin{bmatrix} X(2N-1,S2)\\ \cdot\\ \cdot\\ X(N,S2)\\ X(N-1,S2)\\ \cdot\\ \cdot\\ \cdot\\ X(0,S2)\end{bmatrix}. \tag{3-25}$$

Eq.(3-25) can be rearranged as follows:

$$Z1(w1,2N-1,S2)=[L1^{0}(w1)e\ \ldots\ L1^{N-1}(w1)e]\begin{bmatrix} X(2N-1,S2)\\ X(2N-2,S2)\\ \cdot\\ \cdot\\ \cdot\\ X(N1,S2)\end{bmatrix}+$$

$$L1^{N}(w1)[L1^{0}(w1)e\ldots L1^{N-1}(w1)e]\begin{bmatrix} X(N-1,S2)\\ X(N-2,S2)\\ \cdot\\ \cdot\\ X(0,S2)\end{bmatrix}. \tag{3-26}$$

Considering Eqs.(3-20)(3-21)(3-24)(3-26), it follows that

$$Z1(w1,2N-1,S2)=LX(2N-1,N,S2)+L1^{N}(w1)Z1(w1,N-1,S2). \quad (3-27)$$

For h=3, we have

$$Z1(w1,3N-1,S2)=[L1^{0}(w1)e\ldots L1^{N-1}(w1)e\ L1^{N}(w1)e\ldots L1^{2N-1}(w1)e$$

$$L1^{2N}(w1)e\ldots L1^{3N-1}(w1)e]\begin{bmatrix} X(3N-1,S2) \\ \vdots \\ X(2N,S2) \\ X(2N-1,S2) \\ \vdots \\ X(N,S2) \\ X(N-1,S2) \\ \vdots \\ X(0,S2) \end{bmatrix}$$

$$=LX(3N-1,2N,S2)+L1^{N}(w1)Z1(w1,2N-1,S2). \quad (3-28)$$

Also, it can be shown (by using mathematical induction similar to that in the lemma 2) that

$$Z1(w1,hN-1,S2)=L1^{N}(w1)Z1(w1,(h-1)N-1,S2)$$

$$+LX(hN-1,(h-1)N,S2),$$

$$Z1(w1,-1,S2)=[0\ 0]^{T}. \quad (3-29)$$

<u>QED</u>

For convenience, all formulas necessary in the recursive calculation are listed below.

1) For **F2**(w1,S2)=Z1(w1,hN-1,S2),

$$Z1(w1,hN-1,S2)=L1^{N}(w1)Z1(w1,(h-1)N-1,S2)$$

$$+L\mathbf{X}(hN-1,(h-1)N,S2),$$

$$h=1,2,3,....,H,$$

$$Z1(w1,-1,S2)=\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \text{and} \quad e=\begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{3-30}$$

2) For **F1**(w1,w2)=Z2(w1,w2,N2-1),

$$Z2(w1,w2,S2)=L2(w2)Z2(w1,w2,S2-1)+Z1(w1,N1-1,S2),$$

$$S2=0,1,2,....,N2-1,$$

$$Z2(w1,w2,-1)=\begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{3-31}$$

3) $$|F(w1,w2)|=|Z2(w1,w2,N2-1)|, \tag{3-32}$$

$$\Phi(w1,w2)=-((N1-1)w1\Delta u+(N2-1)w2\Delta v$$

$$-\mathrm{arctg}(-I(w1,w2)/R(w1,w2))). \tag{3-33}$$

2. FAST ALGORITHM OF RFFT

Eq.(3-30) is a recursive formula for processing all data segment by segment (N sample data per segment). The inspection of Eq.(3-30) shows that the amount of calculations is mainly determined by the term of LX(hN-1,(h-1)N,S2). Hence, this term should be computed by revised FFT. The main idea in this section is to make use of the butterfly diagram of general FFT (see Fig.1). But since the number N of sample data is not equal to the number A1 of frequencies (this statement will be discussed below), a revised buttefly diagram is necessary to process these sample data. Next, the revised FFT algorithm is dicussed and a butterfly diagram (see Fig.3) similar to the butterfly diagram of general FFT is obtained.

Let

$$X(hN-1,(h-1)N,S2)=[Xh(0,S2)\ Xh(1,S2)...Xh(N-1,S2)]^T, \quad (3-34)$$

where, $[Xh(0,S2), Xh(1,S2) \ldots Xh(N-1,S2)]$ denotes the inverse sampled data sequence in the h-th segment and the S2-th row, i.e.,

$Xh(0,S2)=X(hN-1,S2)$,

$Xh(1,S2)=X(hN-2,S2)$,

.

.

.

$Xh(N-1,S2)=X((h-1)N,S2)$.

Then

$$LX(hN-1,(h-1)N,S2)=[L1^{0}(w1)e \ldots\ldots L1^{N-1}(w1)e]\begin{bmatrix} Xh(0,S2) \\ Xh(1,S2) \\ \vdots \\ Xh(N-1,S2) \end{bmatrix}$$

$$=\begin{bmatrix} \sum_{p=0}^{N-1} Xh(p,S2)\cos(pw1\Delta u) \\ \sum_{p=0}^{N-1} Xh(p,S2)\sin(pw1\Delta u) \end{bmatrix}. \qquad (3-35)$$

Assume that the bandwidth of the signal is limited. The maximum frequencies along w1 and w2 axes are f_{max1} and f_{max2} respectively. From sampling theorem, we should have $\Delta u <= 1/2f_{max1}$ and $\Delta v <= 1/2f_{max2}$. Select $\Delta u = 1/2f_{max1}$, $\Delta v = 1/2f_{max2}$, and suppose that A1 and A2 are the numbers of frequencies distributed between 0 and $2f_{max1}$, and 0 and $2f_{max2}$, respectively. Therefore,

$$w1=k1\Delta w1=k1 \times 2\pi f_{max1} \times 2/A1=k1 \times 2\pi/A1\Delta u, \qquad (3-36)$$

$$w2=k2 \times 2\pi/A2\Delta v. \qquad (3-37)$$

From Eqs.(3-35) (3-36), define

$$C(k1)=\sum_{p=0}^{N-1} Xh(p,S2)\exp(-j2\pi pk1/A1),$$
$$k1=0,1,2,......A1-1. \qquad (3-38)$$

Eq.(3-38) allows to obtain A1 frequencies from N sampled data. This is one of differences between RFFT and general FFT, since for FFT, the constraint of A1=N must be satisfied. The relation between A1 and N and the computational procedure for Eq.(3-38) will be shown in the following example.

<u>Notation</u>: Each decimal value of p, 0<=p<=N-1, is expressed in binary form

$$p=m_{n-1}2^{n-1}+m_{n-2}2^{n-2}+...+m_1 2^1+m_0 2^0,$$

where $m_i=0,1$, $i=0,1...n-1$, $n=\log_2 N$.

Similarly each decimal value of k1, 0<=k1<=A1-1, is expressed as

$$k1=k_{a-1}2^{a-1}+k_{a-2}2^{a-2}+...+k_1 2^1+k_0 2^0,$$

where $k_i=0,1$, $i=0,1...a-1$, $a=\log_2 A1$.

Denote the binary representation of Xh(p,S2) by

$$Xh(p,S2)=Dh(m_{n-1}m_{n-2}...m_1 m_0,S2). \qquad (3-39)$$

Eqs.(3-38) (3-39) lead to the relation

$$C(k1)=\sum_{p=0}^{N-1} Xh(p,S2)W^{k1p}$$
$$=\sum_{m_0}\sum_{m_1}...\sum_{m_{n-1}} Dh(m_{n-1}m_{n-2}...m_0,S2)W^{k1(m_{n-1}2^{n-1}+...+m_0 2^0)}, \qquad (3-40)$$

where $W=\exp(-j2\pi/A1)$.

Let N=4 A1=8, thus n=2, a=3, $W=\exp(-j2\pi/8)$. For this case

$$C(k1)=\sum_{m_0}\sum_{m_1} Dh(m_1,m_0,S2)W^{k1(m_1 2^1+m_0 2^0)}. \tag{3-41}$$

Assume,

$$M1=\sum_{m_1} Dh(m_1,m_0,S2)W^{k1m_1 2}=\sum_{m_1} Dh(m_1,m_0,S2)W^{2m_1(2^2K_2+2K_1+K_0)}.$$

Noting that $W^8=1$, we obtain

$$M1=\sum_{m_1} Dh(m_1,m_0,S2)W^{2m_1(2K_1+K_0)}. \tag{3-42}$$

The summation over m_1 in Eq.(3-42) results in a function of k_0, k_1 and m_0, which is denoted by $X1(k_0,k_1,m_0)$, i.e.,

$$M1=X1(k_0,k_1,m_0). \tag{3-43}$$

Replacing Eq.(3-43) into Eq.(3-41), we have

$$C(k1)=\sum_{m_0} X1(k_0,k_1,m_0)W^{k1m_0}=\sum_{m_0} X1(k_0,k_1,m_0)W^{m_0(2^2K_2+2K_1+K_0)}.$$

Similarly

$$C(k1)=X2(k_0,k_1,k_2)=M0. \tag{3-44}$$

With respect to $X1(k_0,k_1,m_0)$ (i.e., M1),

$$X1(k_0,k_1,m_0)=Dh(0,m_0,S2)+Dh(1,m_0,S2)W^{2(2K_1+K_0)}. \tag{3-45}$$

Case 1. $k_1 k_0$ =0 0,

$$X1(0,0,0)=Dh(0,0,S2)+Dh(1,0,S2)W^0,$$

$$X1(0,0,1)=Dh(0,1,S2)+Dh(1,1,S2)W^0.$$

Case 2. $k_1 k_0$ =0 1,

$X1(1,0,0)=Dh(0,0,S2)+Dh(1,0,S2)W^{2}$,

$X1(1,0,1)=Dh(0,1,S2)+Dh(1,1,S2)W^{2}$.

Case 3. $k_1 k_0 = 1\ 0$,

$X1(0,1,0)=Dh(0,0,S2)+Dh(1,0,S2)W^{4}$,

$X1(0,1,1)=Dh(0,1,S2)+Dh(1,1,S2)W^{4}$.

Case 4. $k_1 k_0 = 1\ 1$,

$X1(1,1,0)=Dh(0,0,S2)+Dh(1,0,S2)W^{6}$,

$X1(1,1,1)=Dh(0,1,S2)+Dh(1,1,S2)W^{6}$.

Similarly, with respect to $X2(k_0, k_1, k_2)$ (i.e. M0), it can be obtained

$$X2(k_0,k_1,k_2)=X1(k_0,k_1,0)+X1(k_0,k_1,1)W^{4k_2+2K_1+K_0}. \qquad (3\text{-}46)$$

Case 1. $k_2 k_1 k_0 = 0\ 0\ 0$,

$X2(0,0,0)=X1(0,0,0)+X1(0,0,1)W^{0}$.

Case 2. $k_2 k_1 k_0 = 0\ 0\ 1$,

$X2(1,0,0)=X1(1,0,0)+X1(1,0,1)W^{1}$.

Case 3. $k_2 k_1 k_0 = 0\ 1\ 0$,

$X2(0,1,0)=X1(0,1,0)+X1(0,1,1)W^{2}$.

Case 4. $k_2 k_1 k_0 = 0\ 1\ 1$,

$X2(1,1,0)=X1(1,1,0)+X1(1,1,1)W^{3}$.

Case 5. $k_2 k_1 k_0 = 1\ 0\ 0$,

$X2(0,0,1)=X1(0,0,0)+X1(0,0,1)W^{4}$.

Case 6. $k_2 k_1 k_0$ =1 0 1,

$X2(1,0,1)=X1(1,0,0)+X1(1,0,1)W^5$.

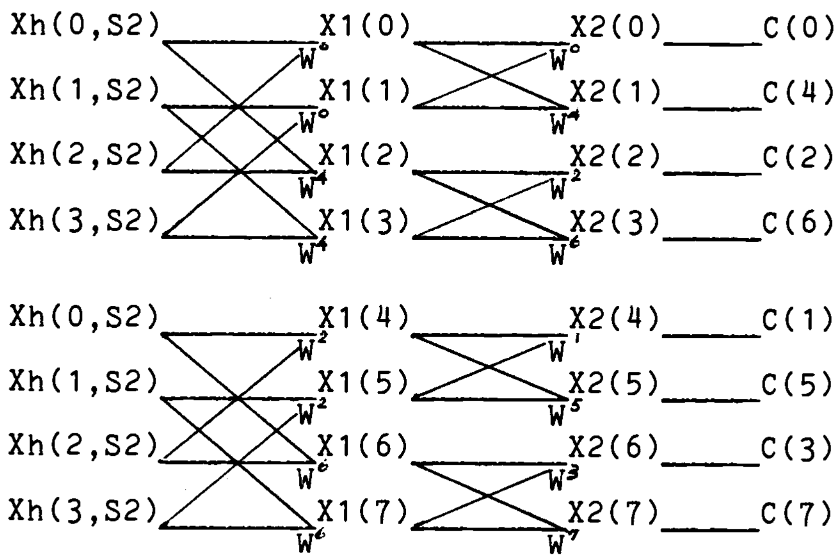
Case 7. $k_2 k_1 k_0$ =1 1 0,

$X2(0,1,1)=X1(0,1,0)+X1(0,1,1)W^6$.

Case 8. $k_2 k_1 k_0$ =1 1 1,

$X2(1,1,1)=X1(1,1,0)+X1(1,1,1)W^7$.

This sequence of arithmetic operations is shown in Fig.3.



8-point Butterfly Diagram of Revised FFT

Fig. 3

Here, the sample data $Dh(m_1, m_0, S2)$ expressed in binary form are replaced by the data Xh(p,S2) expressed in decimal form according to Eq.(3-39). From the complex conjugate property, we have

$$C(A1/2+L)=C^*(A1/2-L), \qquad L=1,2,......A1/2-1,$$

where, $C^*(\ )$ is the complex conjugate of $C(\ )$.

The $C(k1)$, $k1=0,1,2,3,4$, is calculated and others can be obtained by the conjugate property of DFT.

This completes the construction of revised FFT algorithm. Representing $C(k1)$ as real and imaginary parts

$$C(k1)=R1(k1)-jI1(k1), \tag{3-47}$$

and using Eq.(3-38), we have

$$R1(k1)=\sum_{p=0}^{N-1} Xh(p,S2)\cos(2\pi pk1/A1), \tag{3-48}$$

$$I1(k1)=\sum_{p=0}^{N-1} Xh(p,S2)\sin(2\pi pk1/A1). \tag{3-49}$$

Considering Eqs.(3-36)(3-37) and substituting Eqs.(3-48)(3-49) into Eq.(3-35), $LX(hN-1,(h-1)N,S2)$ can be obtained. By comparing Fig.3 with Fig.1, it is seen that the butterfly diagram of the revised FFT is similar to that of general FFT. The difference between them is only in the starting of iteration. Because of $a \geq n$ (see page 30), the first iteration for Eq.(3-38) agrees with the $(a-n+1)$-th iteration of general FFT. The sample data are used repeatedly, and the number of repetitions is $r=A1/N$.

CHAPTER IV

EXPERIMENTAL RESULTS

In order to illustrate the possible applications of 2-D RFFT algorithm, three experiments are presented in this chapter. In the first experiment, a 2-D exponential function is transformed by RFFT and the results are compared with DFT, FFT, and ideal spectrum. In the second experiment, a more complicated function is shown, and in the third one, a signal detection using the RFFT is simulated.

The following is a calculational diagram of the algorithm: when the new data appear, one dimensional RFFT or revised FFT is first used to transform these data to **F2**(w1,S2), and then **F1**(w1,w2) is computed by the recursive formula; this process continues until all data are used.

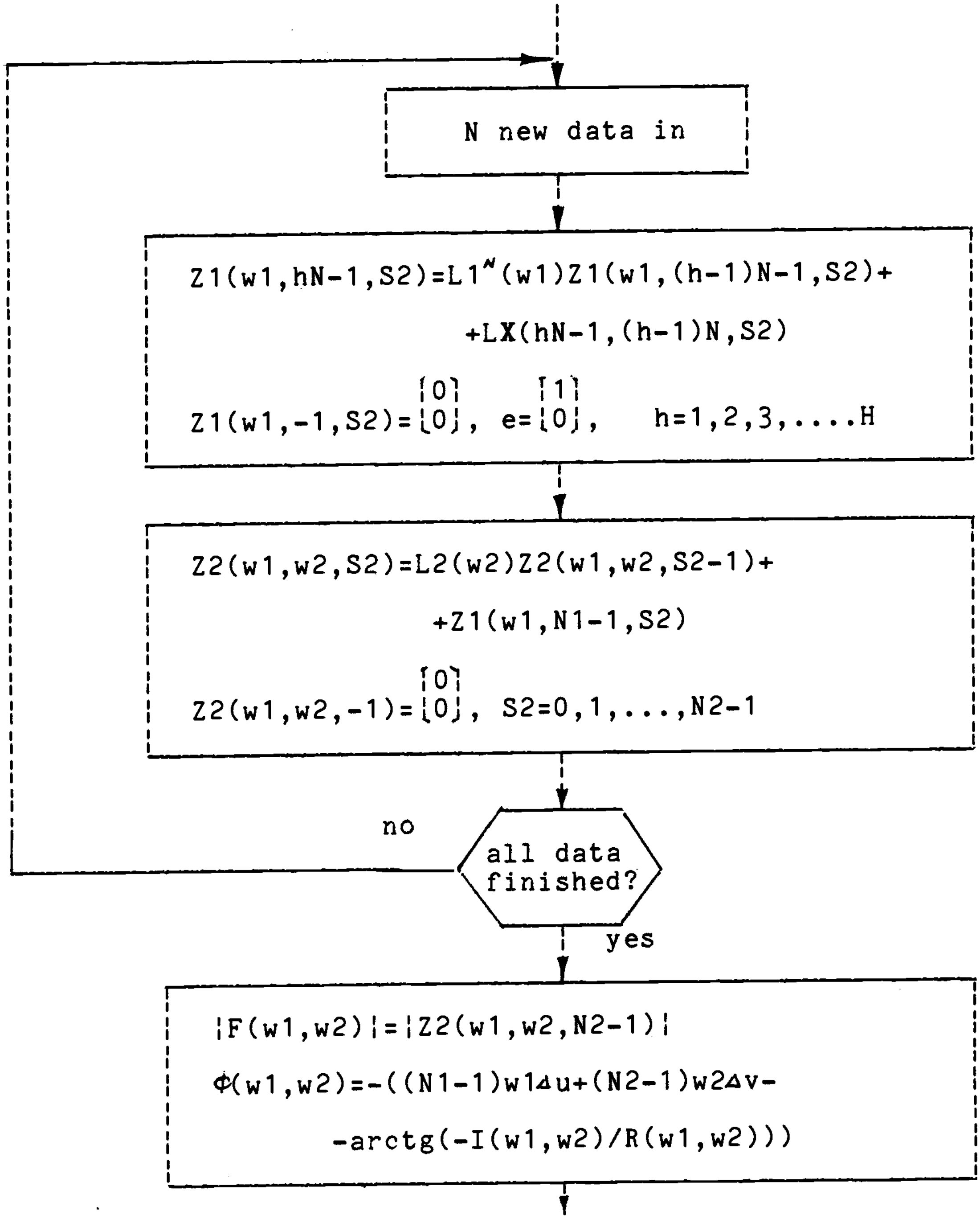


Diagram of RFFT Calculation

Fig. 4

Experiment 1:

For a two dimensional signal:

$$X(u,v)=\exp(-(u+v)) \quad u>=0,\ v>=0, \qquad (4\text{-}1)$$

its Fourier transform, i.e. the ideal spectrum, is

$$F(w1,w2)=(1/(jw1+1))(1/(jw2+1)), \qquad (4\text{-}2)$$

and the amplitude and phase spectra are

$$|F(w1,w2)|=1/[(w1^2+1)(w2^2+1)]^{1/2}, \qquad (4\text{-}3)$$

$$\phi(w1,w2)=-[\mathrm{arctg}(w1)+\mathrm{arctg}(w2)]. \qquad (4\text{-}4)$$

Table 1 shows the ideal spectrum. Fig.5, Fig.6, Table 3 and Table 5 to 7 are the spectra by using 2-D RFFT. From these experimental results, it is noticed that as the data increase, the spectra tend to the ideal spectrum monotonically.

Comparing Table 2, which represents the spectrum obtained by DFT, with Table 3, it can be noticed that the results of RFFT and DFT are identical. RFFT has the same advantage as DFT that the number of sample data may not be equal to the number of frequencies. The transform speed of RFFT is faster than that of DFT because the fast algorithm is used. Table 8 presents the spectrum obtained by RFFT with half sample interval of Table 2. Compared Table 8 with Table 3, it is seen that reducing sample interval makes the spectrum closer to the ideal spectrum, because reducing sample interval implies reducing the overlap of the spectrum.

Comparing Table 4, which represents the FFT algorithm results, with Table 6, it is seen that RFFT and FFT yield the same result when the number of sample data equals that of frequencies.

Experiment 2:

In this experiment, a complicated two dimensional signal to be transformed comes from the model of an ideal low-pass filter and is revised here for the purpose of practical realization.

Definition:

$$X(u,v)=\begin{cases} \exp(-(0.1v+u^2/4v))/v^{\frac{1}{2}} & u>=1, v>=0 \\ 0 & u<1, v<0 \end{cases} \quad . \qquad (4\text{-}5)$$

The Fourier Transform of X(u,v) is

$$F\{X(u,v)\}=\int_{1}^{\infty}\int_{0}^{\infty}[\exp(-(0.1v+u^2/4v))/v^{\frac{1}{2}}]\exp(-(jw2v+jw1u))dvdu$$

$$=\int_{1}^{\infty}[\pi/(0.1+jw2)]^{\frac{1}{2}}\exp[-u((0.1+jw2)^{\frac{1}{2}}+jw1)]du$$

$$=[\pi \exp(-(jw1+(0.1+jw2)^{1/2})]/(0.1+jw2)^{1/2}(jw1+(0.1+jw2)^{1/2}). \quad (4\text{-}6)$$

The amplitude spectrum of F{X(u,v)} can be obtained as

$$|F\{X(u,v)\}|=\pi \exp(-(0.1^2+w2^2)^{1/4}\cos(\theta/2))/\{[w1+(0.1^2+w2^2)^{1/4}(\sin(\theta/2))^2+(0.1^2+w2^2)^{1/2}\cos^2(\theta/2)]\ ((0.1^2+w2^2)^{1/4}\}, \quad (4\text{-}7)$$

where, θ=arctg(10w2).

The Fig.7 to Fig.13 show the convergence of the spectra to the ideal spectrum by using 2-D RFFT. Fig.13 shows a good agreement between the ideal spectrum and the result obtained from 2-D RFFT.

Experiment 3:

This is a simulation of signal detection by using RFFT. The signal g(u,v) is mixed with additive noise n(u,v) and the time of the signal appearance (a,b) is unknown. The input signal is

$$X(u,v)=g(u-a,v-b)U(u-a,v-b)+n(u,v)U(u,v), \quad a>=0,\ b>=0, \quad (4\text{-}8)$$

where U(u,v) is a 2-D unit step function.

Assume that the noise n(u,v) is Gaussian, white noise with zero mean and unit variance and has stationary and ergodic properties. Assume also that g(u,v) and n(u,v) are uncorrelated. Resulting from these assumptions, the autocorrelation function of X(u,v) is

$$Rx(z1,z2)=Rg(z1,z2)+Rn(z1,z2). \quad (4\text{-}9)$$

And the power spectrum of X(u,v) is

$$Sx(w1,w2)=Sg(w1,w2)+Sn(w1,w2), \quad (4\text{-}10)$$

where

$$Rx(z1,z2)=\lim_{\substack{T1\to\infty\\T2\to\infty}}\left[\int_0^{T2}\int_0^{T1} X(u,v)X(u-z1,v-z2)dudv\right]/T1T2,$$

$$Rg(z1,z2)=\lim_{\substack{T1\to\infty\\T2\to\infty}}\left[\int_0^{T2}\int_0^{T1} g(u,v)g(u-z1,v-z2)dudv\right]/T1T2,$$

$$Rn(z1,z2)=\lim_{\substack{T1\to\infty\\T2\to\infty}}\left[\int_0^{T2}\int_0^{T1} n(u,v)n(u-z1,v-z2)dudv\right]/T1T2,$$

$$Sx(w1,w2)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} Rx(z1,z2)\exp(-(jw1z1+jw2z2))dz1dz2,$$

$$Sg(w1,w2)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} Rg(z1,z2)\exp(-(jw1z1+jw2z2))dz1dz2,$$

$$Sn(w1,w2)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} Rn(z1,z2)\exp(-(jw1z1+jw2z2))dz1dz2.$$

If no signal is present, i.e., g(u,v)=0, then

$$Sx(w1,w2)=Sn(w1,w2)=1.$$

The key point here is to obtain Sx(w1,w2) from the random signal X(u,v) directly, because the calculation of correlation Rx(z1,z2) requires all sample data and therefore does not allow the recursive method to be used for on-line applications.

Assuming that X(u,v) is ergodic, the following formula can be derived

$$Sx(w1,w2)=\lim_{\substack{T_1\to\infty \\ T_2\to\infty}} \left| \int_0^{T_1}\int_0^{T_2} X(u,v)\exp(-(jw1u+jw2v))dudv \right|^2 /T1T2. \tag{4-11}$$

Because the detection of sinusoidal/narrow-band signals has received a great deal of attention (Kumaresan & Tufts, 1983) (Glover, 1977), the sinusoidal signal is selected as a signal to be detected in this experiment, i.e.,

$$g(u,v)=\sin(\pi u-\pi v). \tag{4-12}$$

The discrete form of Eq.(4-11) is

$$Sx^*(k1,k2)=\left|\sum_{m2=0}^{N2-1}\sum_{m1=0}^{N1-1} X(m1,m2)\exp(-(2\ k1m1/A1+ 2\ k2m2/A2)\right|^2 /T1T2. \tag{4-13}$$

Fig.14 to Fig.17 illustrate the experiment results. At the beginning, the result is only the noise power spectrum because the collected data does not include the signal. When the signal appears, the spectrum is changed and the signal is detected better and better with increase of recursive steps (Fig.14 to Fig.17). From the experimental results it is indicated that, in the presence of the additive white noise, RFFT provides an efficient signal detection and accurate estimate of the signal frequency despite a small signal-to-noise ratio (see Fig.18).

CHAPTER V

CONCLUSION AND POSSIBLE APPLICATIONS

The algorithm developed in this thesis provides a recursive fast Fourier transform method, by which the (A1xA2) values of frequencies are computed recursively from the (N1xN2) sample data sequence. Because of no constraints on the length of signal sequence and recursive steps, the results of transform can approach monotonically the real spectrum as the data sequence-length increases. RFFT is a recursive version of FFT, and theoretically its output is equal to that of general FFT for the same signal sequence. Hence, the convergence of RFFT to the real spectrum can be discussed similarly to that of FFT.

Studying the formulas in Chapter III, we notice that the amount of computational operations of RFFT is larger than that of FFT when the number of sample data is the same as that of frequencies. But from the viewpoint of on-line applications, the transform speed of RFFT might be not lower than that of FFT because part or most computational operations may be finished during the sample intervals of one segment. This is true if the computer efficiency is relatively high when compared with the duration of one data segment.

The above feature is very suitable for microcomputer or microprocessor implementations. The processes of sampling and calculating may be carried out by several microcomputers or microprocessors seperately. This may fully utilize the computer time and speed up the transformation even further, making it very useful for on-line spectral analysis and identification of two dimensional systems.

RFFT may also have other applications. In many cases, the analog-digital transform is necessary. For example, digital Butterworth filter, digital Chebyshev filter (Rader/Gold, 1967) and Elliptic filter (Gold/Rader, 1969) (Guillemin, 1957) (Storer, 1957) all present the design methods for analog-digital transform according to the desired frequency response specifications. Steiglitz (Steiglitz, 1970) (Rabiner/Steiglitz, 1970) has proposed an IIR (infinite impulse response) design procedure based on minimization of mean-square error in frequency domain. All of these methods require a given frequency response as design bases. Thus, if the desired frequency specification comes from a real analog system, all of the methods require the solution to analog-digital transform problem. With RFFT, all the above procedures might be realized on-line through the identifications of those unknown analog systems in frequency domain. For one dimensional system, the paper (Zhu & Wang, 1986) gives an application of such concept to system identification problem.

Another application of RFFT is to solve the problem of processing a signal sequence which otherwise requires a large computer memory. Especially, in 2-D applications, the large-size data arrays are commonly encountered. Although the signal sequence may be very long, the number of frequencies does not have to be too large for the computer memory to accept. This is because the frequencies in the question are usually distributed in a certain range only. For such a problem, it is difficult to use general FFT (requiring the same number of frequencies and sampling data) to process a large amount of sample data. While selecting the number of data that the computer could accept, the real spectrum will be distorted. RFFT can solve this problem. By selecting the necessary number of frequencies, the desired spectrum could be obtained as the data are processed step by step, regardless of the amount of samples.

Presented here, a "signal detection" experiment (Experiment 3) has shown another advantage of RFFT. Since the time when the signal appears is unknown, it is necessary to sample and detect the signal continuously, which results in an increasing-size 2-D datum array. If the signal appears after a long delay, then, the amount of sample data will become very large, most of them being the noise. This "useless" information will be transformed, and the number of frequencies produced by FFT will increase greatly because of the large amount of sample data. This may set a big burden on the computers. On the other hand, the signal detected by RFFT can produce the spectrum whose frequency number is usually much smaller than that of sample data in practical applications. Hence, the demand for the computer memory and calculations is relaxed. Moreover, because of the recursive property of RFFT, it is possible to stop the transform as soon as the signal is detected.

In the summary, the recursive property, speed and flexibility of RFFT algorithm make it attractive for the applications in a variety of signal processing and analysis problems.

THE EXPLANATION OF FIGURES AND TABLES

N1: the number of data along horizontal direction.

N: the number of data (per segment) along horizontal direction.

H: the recursive times (for segments) along horizontal direction.

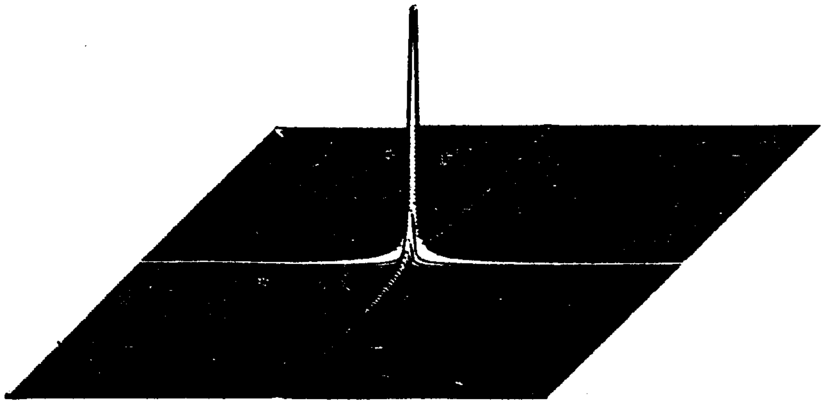
N2: the number of data along vertical direction.

Δu: the sample interval along horizontal direction.

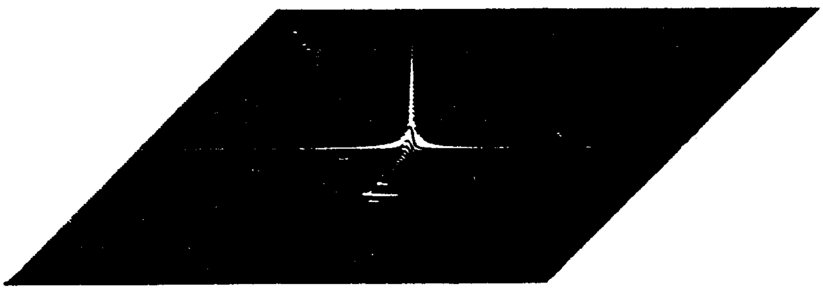
Δv: the sample interval along vertical direction.

A1: the number of frequencies along horizontal direction.

A2: the number of frequencies along vertical direction.



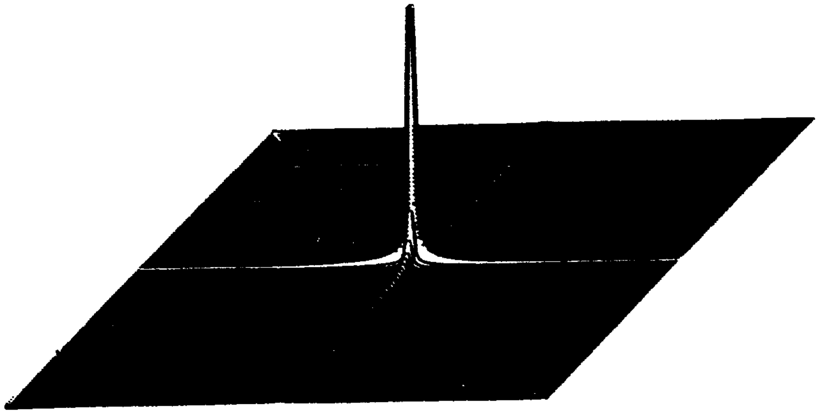
IDEAL SPECTRUM



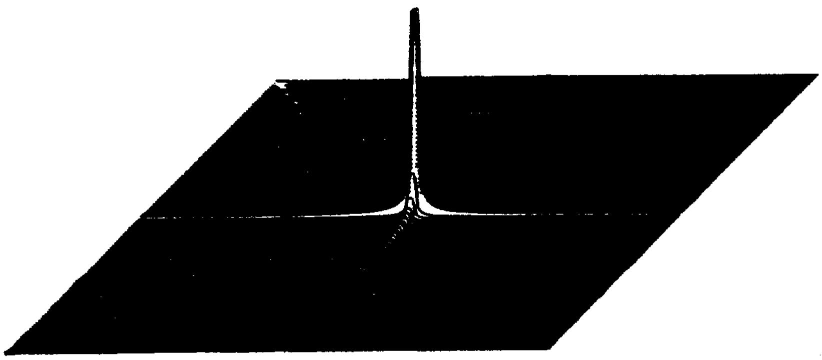
EXP. 1: X(u,v)=exp(-(u+v))   u>=0, v>=0

N1=128 (H=4, N=32)   N2=128
Δu=0.01   Δv=0.01
A1=128   A2=128

Fig. 5



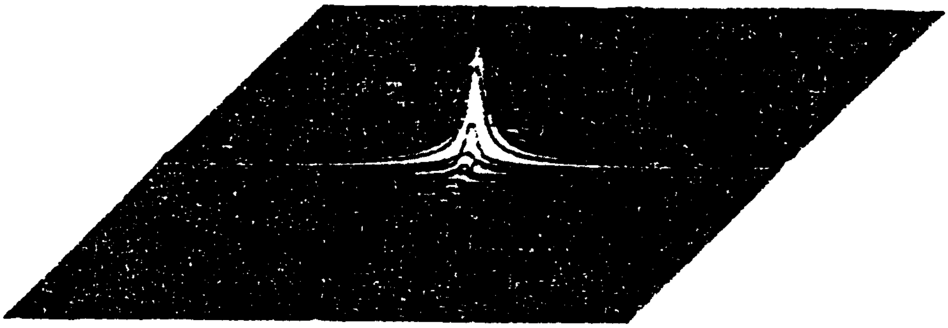
IDEAL SPECTRUM



EXP. 1: X(u,v)=exp(-(u+v))   u>=0, v>=0

N1=256 (H=8, N=32)   N2=250
Δu=0.01   Δv=0.01
A1=128   A2=128

Fig. 6

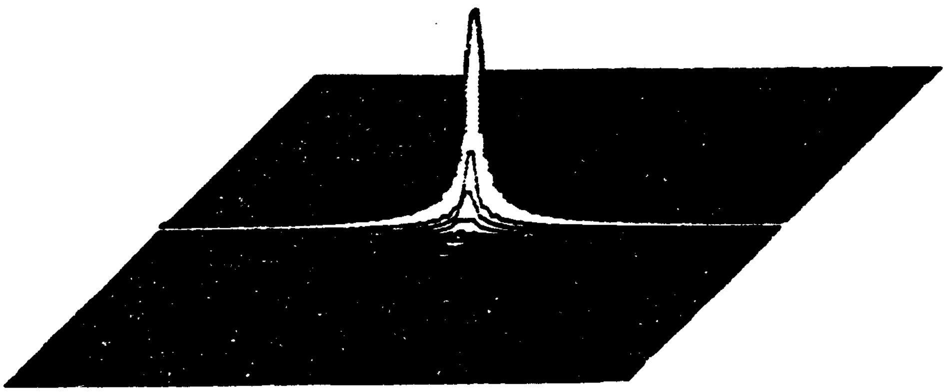


$$\text{EXP. 2:}\quad X(u,v)=\begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0 \\ 0 & u<1,\ v<0 \end{cases}$$

N1=64 (H=1, N=64)
Δu=0.1
A1=128

N2=50
Δv=0.1
A2=128

Fig. 7

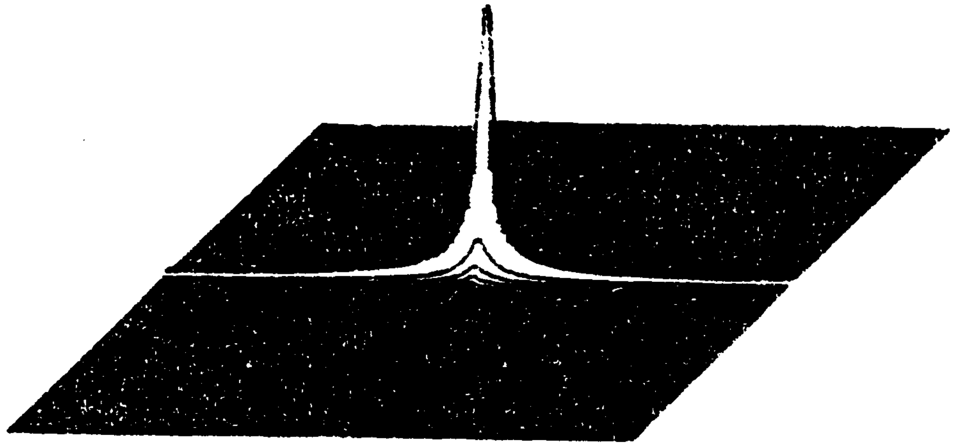


$$\text{EXP. 2:}\quad X(u,v)=\begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0 \\ 0 & u<1,\ v<0 \end{cases}$$

N1=64 (H=1, N=64)
Δu=0.1
A1=128

N2=100
Δv=0.1
A2=128

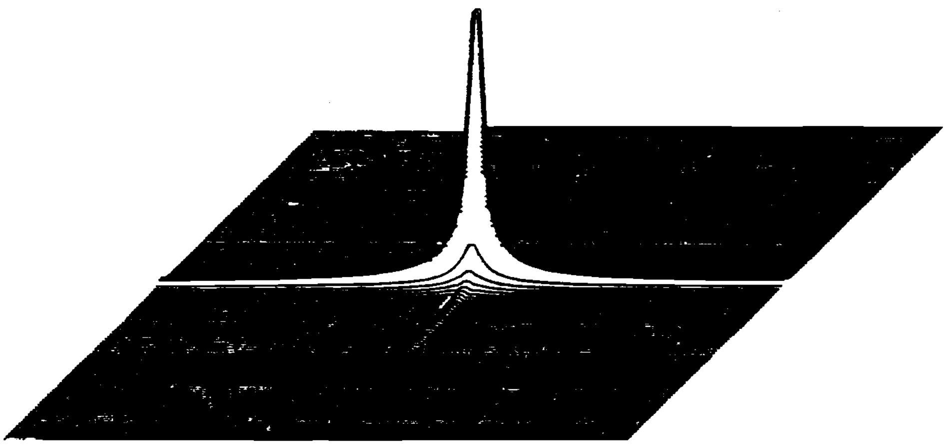
Fig. 8



EXP. 2: $X(u,v) = \begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0 \\ 0 & u<1,\ v<0 \end{cases}$

N1=128 (H=1, N=128) N2=128
Δu=0.1 Δv=0.1
A1=128 A2=128

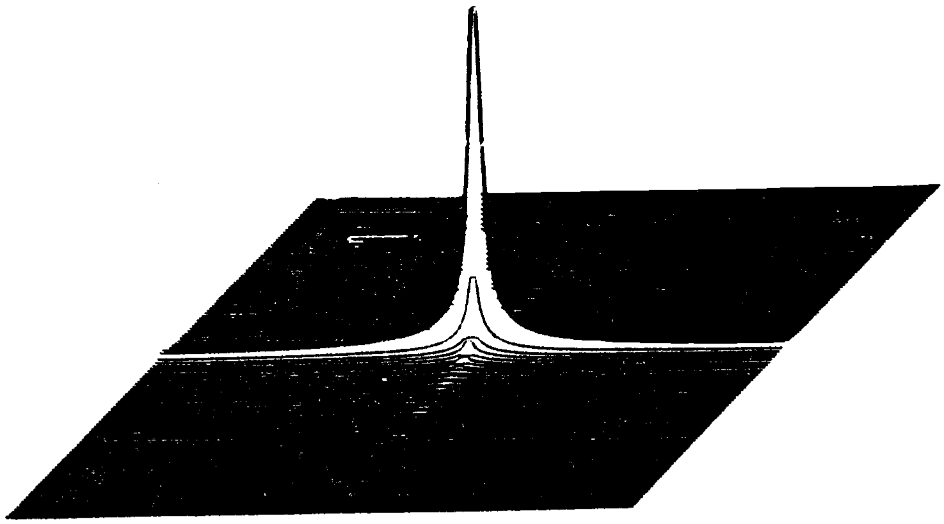
Fig. 9



EXP. 2: $X(u,v) = \begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0 \\ 0 & u<1,\ v<0 \end{cases}$

N1=128 (H=2, N=64) N2=128
Δu=0.1 Δv=0.1
A1=128 A2=128

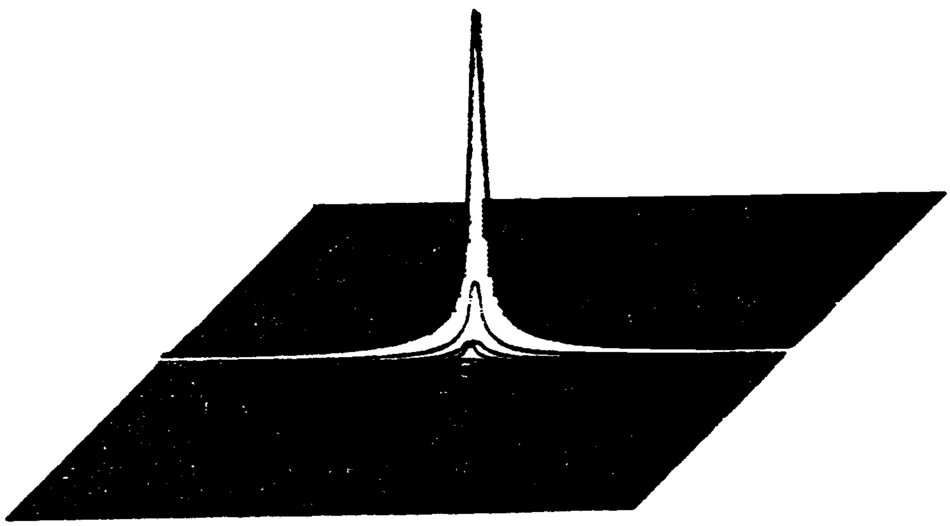
Fig. 10



$$\text{EXP. 2: } X(u,v)=\begin{cases}\exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0\\ 0 & u<1,\ v<0\end{cases}$$

N1=192 (H=3, N=64) N2=200
Δu=0.1 Δv=0.1
A1=128 A2=128

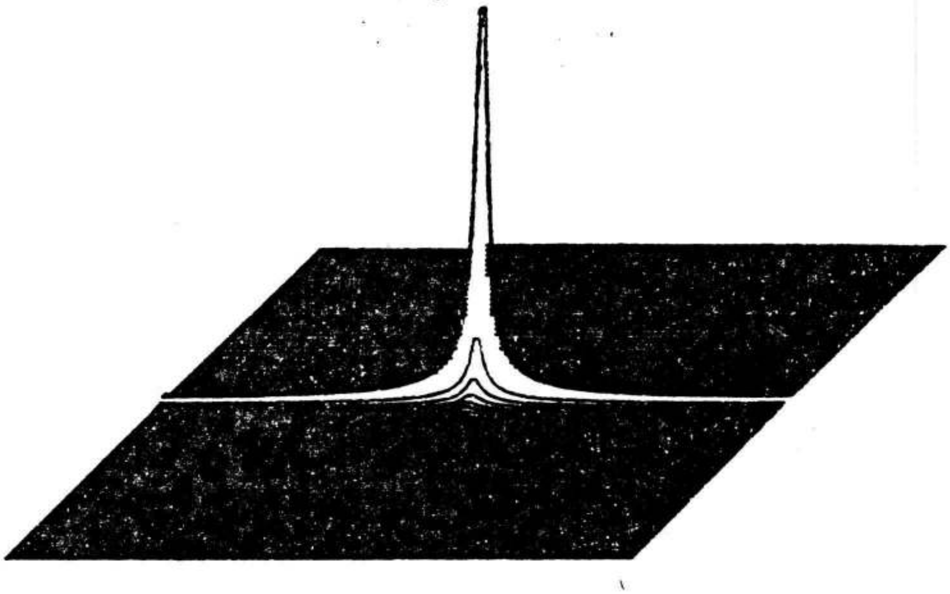
Fig. 11



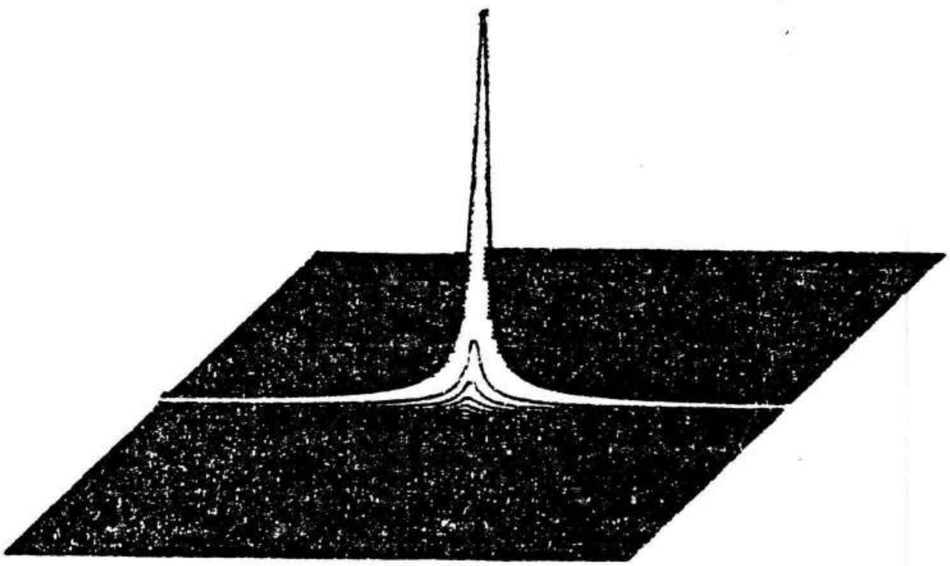
$$\text{EXP. 2: } X(u,v)=\begin{cases}\exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0\\ 0 & u<1,\ v<0\end{cases}$$

N1=256 (H=2, N=128) N2=200
Δu=0.1 Δv=0.1
A1=128 A2=128

Fig. 12



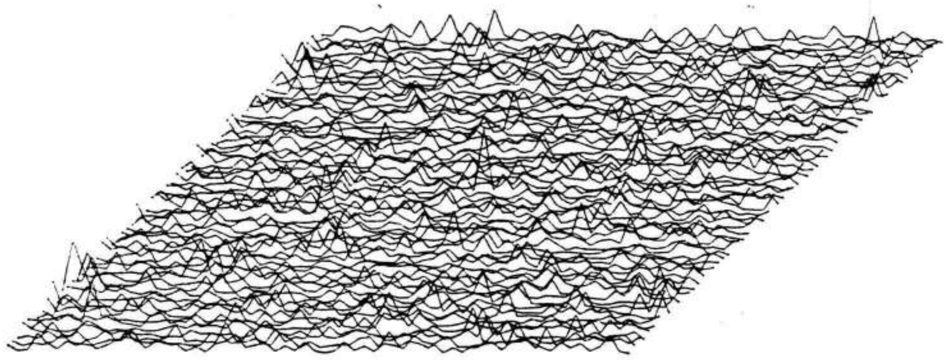
IDEAL SPECTRUM



$$\text{EXP. 2: } X(u,v)=\begin{cases} \exp(-(0.1v+u^2/4v))/v^{1/2} & u>=1, v>=0 \\ 0 & u<1,\ v<0 \end{cases}$$

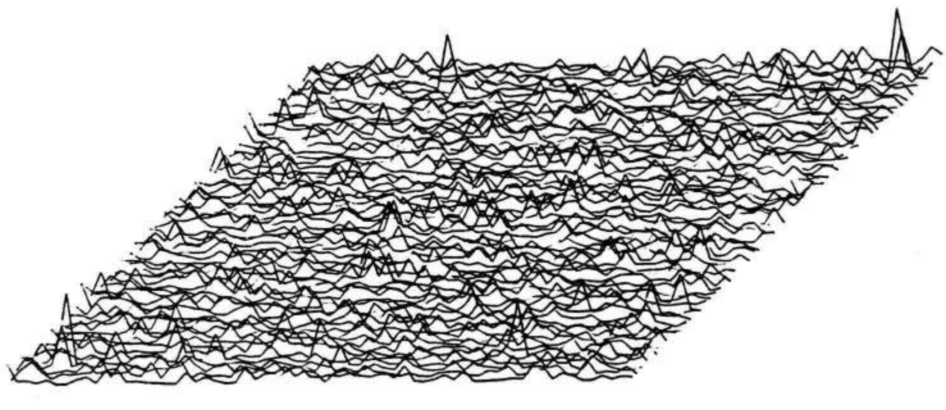
N1=384 (H=3, N=128) N2=350
$\Delta u=0.1$ $\Delta v=0.1$
A1=128 A2=128

Fig. 13



EXP. 3: Spectrum of Two Dimensional White Gaussian Noise n(u,v)

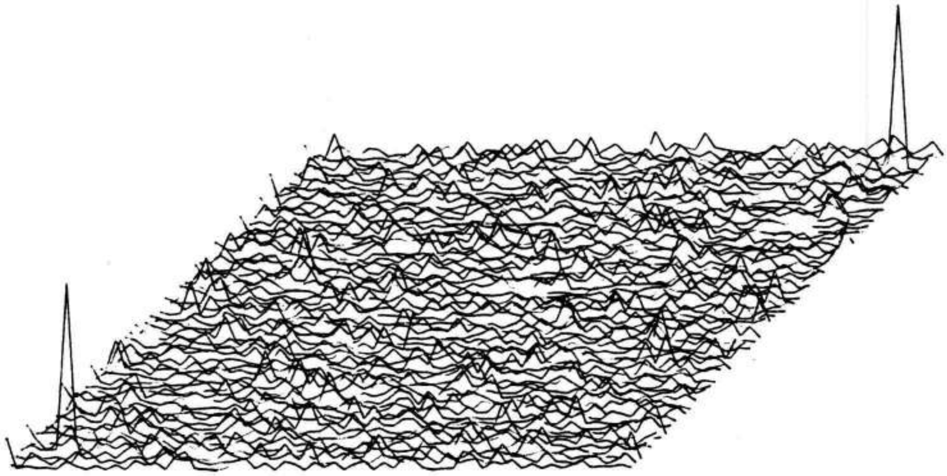
Fig. 14



EXP. 3: $X(u,v)=\mathrm{Sin}(\pi(u-4)-\pi(v-4))U(u-4,v-4)+n(u,v)U(u,v)$

N1=112 (H=7,N=16)	N2=112
$\Delta u=0.125$	$\Delta v=0.125$
A1=64	A2=64

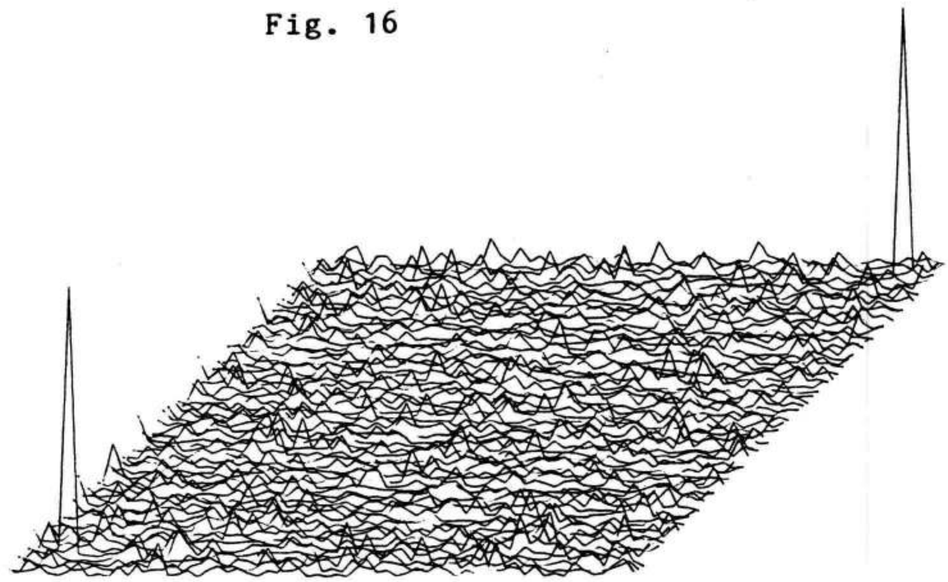
Fig. 15



EXP. 3: X(u,v)=Sin(π(u-4)-π(v-4))U(u-4,v-4)+n(u,v)U(u,v)

N1=128 (H=4,N=32)	N2=128
Δu=0.125	Δv=0.125
A1=64	A2=64

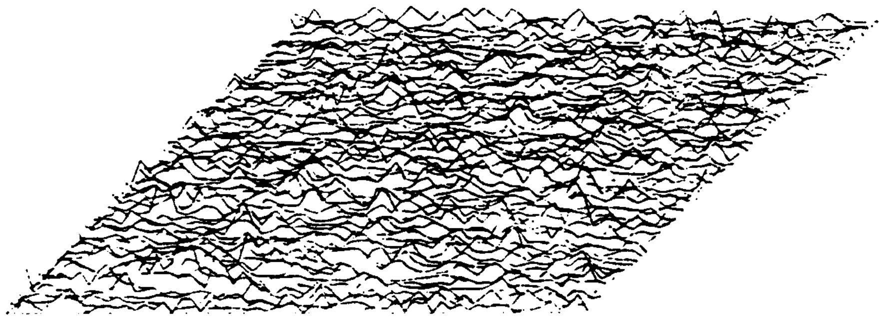
Fig. 16



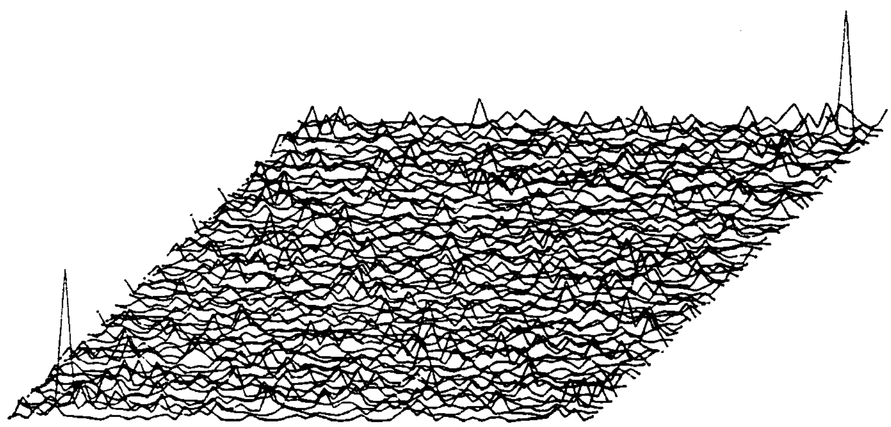
EXP. 3: X(u,v)=Sin(π(u-4)-π(v-4))U(u-4,v-4)+n(u,v)U(u,v)

N1=160 (H=5,N=32)	N2=150
Δu=0.125	Δv=0.125
A1=64	A2=64

Fig. 17



NOISE SPECTRUM



EXP. 3: $X(u,v)=0.1\mathrm{Sin}(\pi(u-4)-\pi(v-4))U(u-4,v-4)+n(u,v)U(u,v)$

N1=640 (H=10,N=64)
Δu=0.125
A1=64

N2=640
Δv=0.125
A2=64

Fig. 18

****

SPECTRUM ANALYSIS WITH TRANSFER FUNCTION

FUNCTION: f(x,y)=exp(-(x*c1+y*c2))
TRANSFER FUNCTION:F(S1,S2)=1/((S1+C1)*(S2+C2))
FACTOR: C1=1.0000 C2=1.0000

****

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2))	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= 1.000000	IM= 0.000000	MOD= 1.000000	ANL= 0.000000
W1= 1	W2= 0	RE= .060897	IM= -.239141	MOD= .246772	ANL=284.286620
W1= 2	W2= 0	RE= .015357	IM= -.125293	MOD= .126304	ANL=277.256100
W1= 3	W2= 0	RE= .007154	IM= -.084275	MOD= .084578	ANL=274.851810
W1= 4	W2= 0	RE= .004036	IM= -.063405	MOD= .063533	ANL=273.642640
W1= 5	W2= 0	RE= .002587	IM= -.050798	MOD= .050864	ANL=272.915530
W1= 6	W2= 0	RE= .001798	IM= -.042365	MOD= .042403	ANL=272.430240
W1= 7	W2= 0	RE= .001322	IM= -.036330	MOD= .036354	ANL=272.083370
W1= 8	W2= 0	RE= .001012	IM= -.031799	MOD= .031815	ANL=271.823180
W1= 9	W2= 0	RE= .000800	IM= -.028272	MOD= .028283	ANL=271.620730
W1= 10	W2= 0	RE= .000648	IM= -.025448	MOD= .025457	ANL=271.458740
W1= 11	W2= 0	RE= .000536	IM= -.023137	MOD= .023144	ANL=271.326170
W1= 12	W2= 0	RE= .000450	IM= -.021211	MOD= .021216	ANL=271.215700
W1= 13	W2= 0	RE= .000384	IM= -.019581	MOD= .019585	ANL=271.122190
W1= 14	W2= 0	RE= .000331	IM= -.018183	MOD= .018186	ANL=271.042050
W1= 15	W2= 0	RE= .000288	IM= -.016972	MOD= .016974	ANL=270.972600
W1= 0	W2= 1	RE= .060897	IM= -.239141	MOD= .246772	ANL=284.286620
W1= 1	W2= 1	RE= -.053480	IM= -.029126	MOD= .060897	ANL=208.573210
W1= 2	W2= 1	RE= -.028991	IM= -.011445	MOD= .031168	ANL=201.542690
W1= 3	W2= 1	RE= -.019718	IM= -.006843	MOD= .020872	ANL=199.138400
W1= 4	W2= 1	RE= -.014917	IM= -.004826	MOD= .015678	ANL=197.929230
W1= 5	W2= 1	RE= -.011990	IM= -.003712	MOD= .012552	ANL=197.202120

TABLE 1 Spectrum Analysis with Transfer Function: F(w1,w2)=1/((jw1+1)(jw2+1))

```
***************************************************

      SPECTRUM ANALYSIS WITH DFT

INPUT FUNCTION:       f(x,y)=exp(-(x*c1+y*c2))
        FACTOR:       C1=1.0000      C2=1.0000
NUMBER OF INPUT:      NX= 28         NY= 30
NUMBER OF SPECTRUM:NFX= 16          NFY= 16
INPUT SAMPLE STEP:  T= .1000

***************************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)).	IM=IMAG(F(W1,W2))	MOD=MODEL(F(W1,W2))	ANL=ANGLE(F(W1.W2))
W1= 0	W2= 0	RE= .985468	IM= 0.000000	MOD= .985468	ANL= 0.000000
W1= 1	W2= 0	RE= .097249	IM= -.[illegible]	MOD= .261084	ANL=291.[illegible]
W1= 2	W2= 0	RE= .070770	IM= -.125713	MOD= .144264	ANL=299.377260
W1= 3	W2= 0	RE= .062469	IM= -.070594	MOD= .094265	ANL=311.505860
W1= 4	W2= 0	RE= .051563	IM= -.046656	MOD= .069538	ANL=317.859990
W1= 5	W2= 0	RE= .051508	IM= -.036494	MOD= .063126	ANL=324.682070
W1= 6	W2= 0	RE= .056060	IM= -.021873	MOD= .060177	ANL=338.685550
W1= 7	W2= 0	RE= .053121	IM= -.006712	MOD= .053543	ANL=352.799070
W1= 8	W2= 0	RE= .049232	IM= -.000000	MOD= .049232	ANL=359.999880
W1= 9	W2= 0	RE= .053121	IM= .006711	MOD= .053543	ANL= 7.200704
W1= 10	W2= 0	RE= .056060	IM= .021873	MOD= .060176	ANL= 21.314533
W1= 11	W2= 0	RE= .051508	IM= .036494	MOD= .063126	ANL= 35.317070
W1= 12	W2= 0	RE= .051563	IM= .046656	MOD= .069538	ANL= 42.140053
W1= 13	W2= 0	RE= .062469	IM= .070594	MOD= .094265	ANL= 48.494301
W1= 14	W2= 0	RE= .070770	IM= .125713	MOD= .144264	ANL= 60.622719
W1= 15	W2= 0	RE= .097249	IM= .242296	MOD= .261084	ANL= 68.131165
W1= 0	W2= 1	RE= .098199	IM= -.228467	MOD= .248677	ANL=293.258910
W1= 1	W2= 1	RE= -.046482	IM= -.046690	MOD= .065883	ANL=225.127620
W1= 2	W2= 1	RE= -.022093	IM= -.028934	MOD= .036404	ANL=232.636200
W1= 3	W2= 1	RE= -.010141	IM= -.021517	MOD= .023787	ANL=244.764680
W1= 4	W2= 1	RE= -.005679	IM= -.016603	MOD= .017548	ANL=251.118930
W1= 5	W2= 1	RE= -.003328	IM= -.015578	MOD= .015930	ANL=257.940920

TABLE 2 Spectrum Analysis with DFT (N1=28, N2=30)

```
*******************************************

        SPECTRUM ANALYSIS WITH RFFT

INPUT FUNCTION:      f(x,y)=exp(-(x*c1+y*c2))
          FACTOR:    c1=1.0000      c2=1.0000
NUMBER OF INPUT:     NX=  4         NY= 30
NUMBER OF SPECTRUM:NFY= 16          NFY= 16
RECURSIVE TIMES:     H=  7
INPUT SAMPLE STEP:  T= .1000

*******************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2)),	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .985467	IM= 0.000000	MOD= .985467	ANL= 0.000000
W1= 1	W2= 0	RE= .097249	IM= -.242296	MOD= .261084	ANL=291.868710
W1= 2	W2= 0	RE= .070770	IM= -.125713	MOD= .144264	ANL=299.377140
W1= 3	W2= 0	RE= .062469	IM= -.070594	MOD= .094265	ANL=311.505740
W1= 4	W2= 0	RE= .051563	IM= -.046656	MOD= .069538	ANL=317.859920
W1= 5	W2= 0	RE= .051508	IM= -.036494	MOD= .063126	ANL=324.682250
W1= 6	W2= 0	RE= .056060	IM= -.021873	MOD= .060176	ANL=338.606040
W1= 7	W2= 0	RE= .053121	IM= -.006711	MOD= .053543	ANL=352.799190
W1= 8	W2= 0	RE= .049232	IM= .000000	MOD= .049232	ANL= .000546
W1= 9	W2= 0	RE= .053121	IM= .006712	MOD= .053543	ANL= 7.201861
W1= 10	W2= 0	RE= .056060	IM= .021875	MOD= .060176	ANL= 21.315987
W1= 11	W2= 0	RE= .051507	IM= .036496	MOD= .063126	ANL= 35.319954
W1= 12	W2= 0	RE= .051561	IM= .046659	MOD= .069538	ANL= 42.142700
W1= 13	W2= 0	RE= .062468	IM= .070595	MOD= .094265	ANL= 48.495110
W1= 14	W2= 0	RE= .070773	IM= .125711	MOD= .144264	ANL= 60.621155
W1= 15	W2= 0	RE= .097257	IM= .242292	MOD= .261083	ANL= 68.129333
W1= 0	W2= 1	RE= .098198	IM= -.228467	MOD= .248676	ANL=293.258670
W1= 1	W2= 1	RE= -.046483	IM= -.046689	MOD= .065883	ANL=225.127080
W1= 2	W2= 1	RE= -.022093	IM= -.028934	MOD= .036404	ANL=232.635680
W1= 3	W2= 1	RE= -.010141	IM= -.021517	MOD= .023787	ANL=244.764370
W1= 4	W2= 1	RE= -.005678	IM= -.016603	MOD= .017548	ANL=251.118960
W1= 5	W2= 1	RE= -.003326	IM= -.015578	MOD= .015930	ANL=257.940860

TABLE 3 Spectrum Analysis with RFFT (N1=28, N2=30)

****

SPECTRUM ANALYSIS WITH FFT

INPUT FUNCTION: f(x,y)=exp(-(x*c1+y*c2))
FACTOR: c1=1.0000 c2=1.0000
NUMBER OF SPECTRUM: NFX= 16 NFY= 16
INPUT SAMPLE STEP: T= .1000

****

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2)),	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .703374	IM= 0.000000	MOD= .703374	ANL= 0.000000
W1= 1	W2= 0	RE= .074790	IM= -.157873	MOD= .174693	ANL=295.348630
W1= 2	W2= 0	RE= .044721	IM= -.079440	MOD= .091163	ANL=299.377260
W1= 3	W2= 0	RE= .038854	IM= -.049685	MOD= .063073	ANL=308.025940
W1= 4	W2= 0	RE= .036803	IM= -.033301	MOD= .049633	ANL=317.860050
W1= 5	W2= 0	RE= .035883	IM= -.022282	MOD= .042238	ANL=328.161870
W1= 6	W2= 0	RE= .035425	IM= -.013822	MOD= .038026	ANL=338.685490
W1= 7	W2= 0	RE= .035205	IM= -.006640	MOD= .035826	ANL=349.319340
W1= 8	W2= 0	RE= .035139	IM= 0.000000	MOD= .035139	ANL= 0.000000
W1= 9	W2= 0	RE= .035205	IM= .006640	MOD= .035826	ANL= 10.680649
W1= 10	W2= 0	RE= .035425	IM= .013822	MOD= .038026	ANL= 21.314526
W1= 11	W2= 0	RE= .035883	IM= .022282	MOD= .042238	ANL= 31.838135
W1= 12	W2= 0	RE= .036803	IM= .033301	MOD= .049633	ANL= 42.139961
W1= 13	W2= 0	RE= .038854	IM= .049685	MOD= .063073	ANL= 51.974068
W1= 14	W2= 0	RE= .044721	IM= .079440	MOD= .091163	ANL= 60.622765
W1= 15	W2= 0	RE= .074791	IM= .157873	MOD= .174693	ANL= 64.651352
W1= 0	W2= 1	RE= .074790	IM= -.157873	MOD= .174693	ANL=295.348630
W1= 1	W2= 1	RE= -.027482	IM= -.033574	MOD= .043388	ANL=230.697330
W1= 2	W2= 1	RE= -.013075	IM= -.018485	MOD= .022642	ANL=234.725920
W1= 3	W2= 1	RE= -.007020	IM= -.014004	MOD= .015665	ANL=243.374600
W1= 4	W2= 1	RE= -.003561	IM= -.011801	MOD= .012327	ANL=253.208740
W1= 5	W2= 1	RE= -.001186	IM= -.010423	MOD= .010491	ANL=263.510560

TABLE 4 Spectrum Analysis with FFT (N1=16, N2=16)

```
************************************************

        SPECTRUM ANALYSIS WITH RFFT

INPUT FUNCTION:      f(x,y)=exp(-(x*c1+y*c2))
         FACTOR:     c1=1.0000     c2=1.0000
NUMBER OF INPUT:     NX=  4        NY= 10
NUMBER OF SPECTRUM:  NFX= 16       NFY= 16
RECURSIVE TIMES:     H=  2
INPUT SAMPLE STEP:   T= .1000

************************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2)),	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .384379	IM= 0.000000	MOD= .384379	ANL= 0.000000
W1= 1	W2= 0	RE= .107571	IM= -.227069	MOD= .251260	ANL=295.348630
W1= 2	W2= 0	RE= .024439	IM= -.043412	MOD= .049819	ANL=299.377200
W1= 3	W2= 0	RE= .055884	IM= -.071461	MOD= .090718	ANL=308.026000
W1= 4	W2= 0	RE= .020112	IM= -.018198	MOD= .027123	ANL=317.860050
W1= 5	W2= 0	RE= .051611	IM= -.032048	MOD= .060751	ANL=328.161800
W1= 6	W2= 0	RE= .019359	IM= -.007553	MOD= .020781	ANL=338.685550
W1= 7	W2= 0	RE= .050636	IM= -.009550	MOD= .051528	ANL=349.319210
W1= 8	W2= 0	RE= .019203	IM= -.000000	MOD= .019203	ANL=359.999880
W1= 9	W2= 0	RE= .050636	IM= .009550	MOD= .051528	ANL= 10.680559
W1= 10	W2= 0	RE= .019359	IM= .007553	MOD= .020781	ANL= 21.314236
W1= 11	W2= 0	RE= .051611	IM= .032047	MOD= .060751	ANL= 31.837975
W1= 12	W2= 0	RE= .020112	IM= .018198	MOD= .027123	ANL= 42.140083
W1= 13	W2= 0	RE= .055884	IM= .071462	MOD= .090718	ANL= 51.974243
W1= 14	W2= 0	RE= .024439	IM= .043412	MOD= .049819	ANL= 60.622467
W1= 15	W2= 0	RE= .107570	IM= .227069	MOD= .251260	ANL= 64.651505
W1= 0	W2= 1	RE= .045973	IM= -.188808	MOD= .194324	ANL=283.684750
W1= 1	W2= 1	RE= -.098671	IM= -.079997	MOD= .127025	ANL=219.033480
W1= 2	W2= 1	RE= -.018401	IM= -.017197	MOD= .025186	ANL=223.061950
W1= 3	W2= 1	RE= -.028418	IM= -.035997	MOD= .045863	ANL=231.710720
W1= 4	W2= 1	RE= -.006534	IM= -.012056	MOD= .013712	ANL=241.544800
W1= 5	W2= 1	RE= -.009569	IM= -.029184	MOD= .030713	ANL=251.846590

TABLE 5 Spectrum Analysis with RFFT (N1=8, N2=10)

```
*******************************************
        SPECTRUM ANALYSIS WITH RFFT

INPUT FUNCTION:      f(x,y)=exp(-(x*c1+y*c2))
         FACTOR:     c1=1.0000     c2=1.0000
NUMBER OF INPUT:     NX=  4        NY= 16
NUMBER OF SPECTRUM:NFX= 16         NFY= 16
RECURSIVE TIMES:     H=  4
INPUT SAMPLE STEP:   T= .1000

*******************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2)),	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .703374	IM= 0.000000	MOD= .703374	ANL= 0.000000
W1= 1	W2= 0	RE= .074790	IM= -.157874	MOD= .174693	ANL=295.348630
W1= 2	W2= 0	RE= .044721	IM= -.079440	MOD= .091163	ANL=299.377200
W1= 3	W2= 0	RE= .038854	IM= -.049685	MOD= .063073	ANL=308.025760
W1= 4	W2= 0	RE= .036803	IM= -.033301	MOD= .049633	ANL=317.859920
W1= 5	W2= 0	RE= .035883	IM= -.022282	MOD= .042238	ANL=328.161320
W1= 6	W2= 0	RE= .035425	IM= -.013822	MOD= .038026	ANL=338.685060
W1= 7	W2= 0	RE= .035205	IM= -.006640	MOD= .035826	ANL=349.319210
W1= 8	W2= 0	RE= .035139	IM= .000000	MOD= .035139	ANL= .000109
W1= 9	W2= 0	RE= .035205	IM= .006640	MOD= .035826	ANL= 10.680996
W1= 10	W2= 0	RE= .035425	IM= .013822	MOD= .038026	ANL= 21.314674
W1= 11	W2= 0	RE= .035883	IM= .022282	MOD= .042238	ANL= 31.838631
W1= 12	W2= 0	RE= .036803	IM= .033301	MOD= .049633	ANL= 42.140083
W1= 13	W2= 0	RE= .038853	IM= .049686	MOD= .063073	ANL= 51.975555
W1= 14	W2= 0	RE= .044719	IM= .079441	MOD= .091163	ANL= 60.623779
W1= 15	W2= 0	RE= .074788	IM= .157875	MOD= .174693	ANL= 64.652390
W1= 0	W2= 1	RE= .074790	IM= -.157874	MOD= .174693	ANL=295.348510
W1= 1	W2= 1	RE= -.027483	IM= -.033574	MOD= .043388	ANL=230.697110
W1= 2	W2= 1	RE= -.013075	IM= -.018485	MOD= .022642	ANL=234.725650
W1= 3	W2= 1	RE= -.007021	IM= -.014004	MOD= .015665	ANL=243.374080
W1= 4	W2= 1	RE= -.003561	IM= -.011801	MOD= .012327	ANL=253.208470
W1= 5	W2= 1	RE= -.001186	IM= -.010423	MOD= .010490	ANL=263.509950

TABLE 6 Spectrum Analysis with RFFT (N1=16, N2=16)

```
******************************************************

          SPECTRUM ANALYSIS WITH RFFT

INPUT FUNCTION:       f(x,y)=exp(-(x*c1+y*c2))
        FACTOR:       c1=1.0000      c2=1.0000
NUMBER OF INPUT:      NX=  4         NY= 25
NUMBER OF SPECTRUM:NFX= 16           NFY= 16
RECURSIVE TIMES:      H=  5
INPUT SAMPLE STEP:  T= .1000

******************************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)).	IM=IMAG(F(W1,W2)).	MOD=MODEL(F(W1,W2)).	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .876431	IM= 0.000000	MOD= .876431	ANL= 0.000000
W1= 1	W2= 0	RE= .138567	IM= -.212920	MOD= .254039	ANL=303.055910
W1= 2	W2= 0	RE= .073167	IM= -.129972	MOD= .149151	ANL=299.377140
W1= 3	W2= 0	RE= .046301	IM= -.079177	MOD= .091721	ANL=300.318480
W1= 4	W2= 0	RE= .045858	IM= -.041494	MOD= .061844	ANL=317.859920
W1= 5	W2= 0	RE= .056055	IM= -.025112	MOD= .061423	ANL=335.868710
W1= 6	W2= 0	RE= .057959	IM= -.022614	MOD= .062215	ANL=338.685550
W1= 7	W2= 0	RE= .049438	IM= -.016434	MOD= .052098	ANL=341.612120
W1= 8	W2= 0	RE= .043785	IM= .000000	MOD= .043785	ANL= .000546
W1= 9	W2= 0	RE= .049438	IM= .016435	MOD= .052098	ANL= 18.388947
W1= 10	W2= 0	RE= .057959	IM= .022615	MOD= .062215	ANL= 21.315113
W1= 11	W2= 0	RE= .056055	IM= .025112	MOD= .061423	ANL= 24.131992
W1= 12	W2= 0	RE= .045857	IM= .041495	MOD= .061844	ANL= 42.140953
W1= 13	W2= 0	RE= .046298	IM= .079179	MOD= .091721	ANL= 59.683945
W1= 14	W2= 0	RE= .073165	IM= .129973	MOD= .149151	ANL= 60.623779
W1= 15	W2= 0	RE= .138561	IM= .212924	MOD= .254039	ANL= 56.945747
W1= 0	W2= 1	RE= .102492	IM= -.233750	MOD= .255232	ANL=293.676030
W1= 1	W2= 1	RE= -.040583	IM= -.061856	MOD= .073981	ANL=236.731840
W1= 2	W2= 1	RE= -.026108	IM= -.034713	MOD= .043435	ANL=233.053160
W1= 3	W2= 1	RE= -.015702	IM= -.021608	MOD= .026711	ANL=233.994290
W1= 4	W2= 1	RE= -.005704	IM= -.017083	MOD= .018010	ANL=251.536010
W1= 5	W2= 1	RE= -.000142	IM= -.017887	MOD= .017888	ANL=269.544980

TABLE 7 Spectrum Analysis with RFFT (N1=20, N2=25)

```
*********************************************

      SPECTRUM ANALYSIS WITH RFFT

INPUT FUNCTION:     f(x,y)=exp(-(x*c1+y*c2))
         FACTOR:    c1=1.0000      c2=1.0000
NUMBER OF INPUT:    NX=  8         NY= 60
NUMBER OF SPECTRUM:NFX= 32         NFY= 32
RECURSIVE TIMES:    H=  7
INPUT SAMPLE STEP:  T= .0500

*********************************************
```

SPECTRUM F(W1,W2):		RE=REAL(F(W1,W2)),	IM=IMAG(F(W1,W2)),	MOD=MODEL(F(W1,W2)),	ANL=ANGLE(F(W1,W2))
W1= 0	W2= 0	RE= .937991	IM= 0.000000	MOD= .937991	ANL= 0.000000
W1= 1	W2= 0	RE= .069763	IM= -.237266	MOD= .247310	ANL=286.384770
W1= 2	W2= 0	RE= .042537	IM= -.127783	MOD= .134677	ANL=288.411870
W1= 3	W2= 0	RE= .036375	IM= -.077777	MOD= .085862	ANL=295.064760
W1= 4	W2= 0	RE= .026763	IM= -.054986	MOD= .061153	ANL=295.953000
W1= 5	W2= 0	RE= .024325	IM= -.047083	MOD= .052995	ANL=297.322570
W1= 6	W2= 0	RE= .027924	IM= -.038587	MOD= .047631	ANL=305.892580
W1= 7	W2= 0	RE= .027667	IM= -.028057	MOD= .039404	ANL=314.599300
W1= 8	W2= 0	RE= .024016	IM= -.022844	MOD= .033146	ANL=316.432680
W1= 9	W2= 0	RE= .024159	IM= -.021508	MOD= .032346	ANL=318.321960
W1= 10	W2= 0	RE= .026769	IM= -.017246	MOD= .031844	ANL=327.207760
W1= 11	W2= 0	RE= .025928	IM= -.011478	MOD= .028355	ANL=336.121700
W1= 12	W2= 0	RE= .023543	IM= -.009467	MOD= .025375	ANL=338.094060
W1= 13	W2= 0	RE= .024570	IM= -.008904	MOD= .026134	ANL=340.080380
W1= 14	W2= 0	RE= .026506	IM= -.005137	MOD= .026999	ANL=349.032900
W1= 15	W2= 0	RE= .025115	IM= -.000880	MOD= .025130	ANL=357.994140
W1= 16	W2= 0	RE= .023445	IM= -.000002	MOD= .023445	ANL=359.996150
W1= 17	W2= 0	RE= .025115	IM= .000876	MOD= .025130	ANL= 1.998240
W1= 18	W2= 0	RE= .026507	IM= .005133	MOD= .026999	ANL= 10.959450
W1= 19	W2= 0	RE= .024571	IM= .008900	MOD= .026134	ANL= 19.911915
W1= 20	W2= 0	RE= .023544	IM= .009464	MOD= .025375	ANL= 21.898247
W1= 21	W2= 0	RE= .025929	IM= .011474	MOD= .028355	ANL= 23.370586
W1= 22	W2= 0	RE= .026772	IM= .017242	MOD= .031844	ANL= 32.782837

TABLE 8 Spectrum Analysis with RFFT (N1=56, N2=60)

CHAPTER VI

BIBLIOGRAPHY

Ahmed,N., Natarajan,T. & Rao,K.R., "An algorithm for the on-line computation of Fourier spectra", International Journal of Computer Mathematics, Vol.3, No.4, pp 361-370, 1973.

Bhuyan,L.N. & Agrawal,P.P., "Performance analysis of FFT algorithm on multiprocessor systems", IEEE Trans. on Software Eng., Vol.9, No.4, pp 512-521, July 1983.

Bitmead, R.R., "On recursive discrete Fourier transformation", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.30, No.2, pp 319-322, Apr. 1982.

Byrne,C.L. & Fitzgerald,R.M., "Linear and nonlinear estimators for one- and two-dimensional Fourier transforms", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.32, No.4, pp 914-916, Aug. 1984.

Byrne,C.L. & Fitzgerald,R.M., "Reconstruction from Partial Information with Applictions to Topography", SIAM. J. Appl. Math., Vol.42, pp 933-940, Aug. 1982.

Cooley,J.W. & Cochran,W.T., "What is the fast Fourier transform?", IEEE Trans. on Audio Electroacoust, Vol.15, pp 45-55, June 1967.

Cooley,J.W. & Tukey,J.W., "An algorithm for the machine calculation of complex Fourier series", Math. of Comput., Vol.19, pp 297-301, Apr. 1965.

Delotto and Dotti,D., "Two dimensional transforms by minicomputers without matrix transposing", Computer Graphics and Image Processing, Vol.4, No.3, pp 271-276, Sept. 1975.

Duhamel,P. & Hollmann,H., "'Split radix' FFT algorithm", Electronics Letters, Vol.20, No.1, pp 14-16, Jan. 1984.

Eklundh,J.O., "A fast computer method for matrix transposing", IEEE Trans. on Computers, Vol.21, No.7, pp 801-803, July 1972.

Gentleman,W.M. & Sande,G., "Fast Fourier transform-for fun and profit", D.C., Spartan 1966, pp 563-578.

Glover,J., "Adaptive noise cancelling applied to sinusoidal interferences", IEEE Assp. Vol.25, pp 484-491, Dec, 1977.

Gold,B. & Rader,C.M., "Digital processing of signals", McGraw-Hill Book Company, New York, 1969.

Guillemin,E.A., "Synthesis of passive networks", John wiley and Sons, Inc., New York, 1957.

Harris,D.B., McClellan,J.H., Chan,D.S.K. & Schuessler,H.W., "Vector radix fast Fourier transform", IEEE Internat. Conf. on Acoust, Speech, Signal Processing, pp548-551, 1977.

Hinton,O.R. & Saleh,R.A., "Two-dimensional discrete Fourier transform with small multiplicative complexity using number theoretic transforms", IEE Proceedings, Vol.131, pt.G, No.6, pp 234-236, Dec. 1984

Hostetter,G.H., "Recursive discrete functional expansion", IEEE Trans. on Automatic Control, Vol.29, No.7, pp 654-656, July 1984.

Hostetter,G.H., "Recursive discrete Fourier transform with unevenly spaced data", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.31, No.1, pp 206-209, Feb. 1983.

Hostetter,G.H., "Recursive discrete Fourier transformation", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.28, pp 184-190, Apr. 1980.

Hoyer,E.A. & Berry,W.R., "An algorithm for the two-dimensional FFT", Proc. of IEEE Internat. Conf. on Acoust, Speech, Signal Processing, pp 552-555, 1977.

Kumaresan,R. & Tufts,D.W., "Estimating the parameters of exponentially dampted sinusoids and pole-zero modeling in noise", IE Assp. Vol.30, pp 833-840, 1983.

Mersereau,R.M., & Dudgeon,D.E., "The representation of two dimensional sequences as one dimensional sequences", IEEE Trans. on Acoust, Speech, Signal Processing. Vol.22, No.5, pp 320-325, Oct. 1974.

Mitra,S.K. & Ekstrom,M.P., "Two-dimensional digital signal processing", Dowden Hutchinson and Ross, Inc., 1978.

Nussbaumer,H.J. & Quandalle,P., "Fast computation of discrete Fourier transforms using polynomial transforms", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.27, No.2, pp 169-181, Apr. 1979.

Onoe,M., "A method for computing large-scale two-dimensional transform without transposing data matrix", IEEE Proc. Vol.63, No.1, pp 196-197, 1975.

Oppenheim,A.V. & Schafer,R.W., "Digital signal processing", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

Pei,Soo-Chang & Wu,Ja-Ling, "Fast biased polynomial transforms for two-dimesional convolutions", Electronics Letters, Vol.17, No.14, pp 547-548, July 1981.

Preuss,R.D., "Very fast computation of the radix-2 DFT discrete Fourier transform", IEEE Trans. on Asouct, Speech, Signal Processing, Vol.30, No.4, pp 595-607, Aug. 1982.

Rabiner,L.R. & Steiglitz,K., "The design of wide-band recursive and nonrecursive digital differentiators", IEEE Trans. on Audio Electroacoust, Vol.18, No.2, pp 204-209, June 1970.

Rader,C.M. & Gold,B., "Digital filter design techniques in the frequency domain", IEEE Proc., Vol.55, pp 149-171, Feb. 1967.

Schumann,U., "Comments on ' A fast computer method for matrix transposing ' and application to the solution of Poisson's equation", IEEE Trans. on Computers, Vol.22, No.5, pp 524-542, May 1973.

Singleton,R.C., "An algorithm for computing the mixed radix fast Fourier transform", IEEE Trans. on Audio Electroacoust, Vol.17, pp 93-103, June 1969.

Steiglitz,K., "Computer-aided design of recursive digital filters", IEEE Trans. on Audio Electroacoust., Vol.18, No.2, pp 123-129, June 1970.

Storer,J.E., "Passive network synthesis", McGraw-Hill Book Company, New York, 1957.

Stuller,J.A., "Generalized running discrete transform", IEEE Trans. on Acoust, Speech, Signal Processing, Vol.30, No.1, pp 60-68, Feb. 1982.

Sudhakar,R., Agarwal,R.C. & Dutta Roy,S.C., "Fast computation of Fourier transforn at arbitrary frequencies", IEEE Trans. on Circuits and Systems, Vol.28, No.10, pp 972-980, Oct. 1981.

Twogood,R.E. & Ekstrom,M.P., "An extension of Eklundh's matrix transposition algorithm and its applications in digital image processing", IEEE Trans. on Computer, Vol.25, No.9, pp 950-952, Sept. 1976.

Wu,Ja-Ling & Pei,Soo-Chang, "Fast biased polynomial transforms for 2D prime length discrete Fourier transforms", Electronics Letters, Vol.20, No.22, pp 932-934, Oct. 1984.

Zhu,Feihong, "Recursive fast Fourier transform", ACTA Automatica Sinicana, Vol.11, Suppl. No.2, pp 221-225, Dec. 1986.

Zhu,Feihong and Wang Xiedong, "System spectrum analysis with recursive fast Fourier transformation", Journal of Chinese Society of Automation, Information and Control, Vol.15, No.1, pp6-13, Feb. 1986.