University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations | Theses, Dissertations, and Major Papers

1984

Image filtering using the NTT convolver.

V. Raja Ravi Varma
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

Varma, V. Raja Ravi, "Image filtering using the NTT convolver." (1984). *Electronic Theses and Dissertations*. 2378.
https://scholar.uwindsor.ca/etd/2378

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

National Library of Canada
Canadian Theses Service
Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Services des thèses canadiennes

CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL 339 (r. 86/01)

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(39)

0-315-24725-8

National Library of Canada | Bibliothèque nationale du Canada

CANADIAN THESES ON MICROFICHE | *THÈSES CANADIENNES SUR MICROFICHE*

NAME OF AUTHOR/*NOM DE L'AUTEUR* V. Raja Ravi Varma

TITLE OF THESIS/*TITRE DE LA THÈSE* Image filtering using the NTT convolver

UNIVERSITY/*UNIVERSITÉ* University of Windsor, Windsor, Ont.

DEGREE FOR WHICH THESIS WAS PRESENTED/ *GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE* M.A.Sc.

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE* Fall 1984

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* Dr. G. A. Jullien

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/*DATÉ* 19th June '84 SIGNED/*SIGNÉ* V. Raja Ravi Varma

PERMANENT ADDRESS/*RÉSIDENCE FIXÉ*

NL-91 (3-74)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

IMAGE FILTERING USING THE NTT CONVOLVER

by

V.Raja Ravi Varma

A thesis
presented to the University of Windsor
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in
Electrical Engineering

Windsor , Ontario, 1984

(c) V.Raja Ravi Varma, 1984

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Approved by:

Dr. G.A. Jullien

Dr. M. Ahmadi

Dr. M. Shridhar

Dr. W. C. Miller

806793

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ABSTRACT

This thesis addresses the problem of designing 2-D linear phase, Finite Impulse Response (FIR) digital filters and its applications. The task has been carried out in two parts.

The first part deals with the well established techniques of transformation of 1-D filters and windowing. Examples of filters designed using these approaches are provided.

The second part of the thesis examines application of these filters in image processing. The implementaion of the filter is carried out using the high speed NTT convolver, which constitutes part of the image processing system built around the SEL 32/27 minicomputer. The filters designed perform smoothing, sharpening and enhancement of images.

The discussion also reviews various FIR digital filter design techniques and the functional description of the image processing system.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my Supervisor Dr.G.A.Jullien for his valuable suggestions and guidance during the course of this research. Thanks are also due to Dr.W.C.Miller and Dr.M.Shridhar for their helpful suggestions and comments on this work.

Last but not least, I extend my sincerest thanks to my beloved parents and friends, without whom this work would not have been accomplished.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

TABLE OF CONTENTS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

LIST OF FIGURES

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter I

INTRODUCTION

The use of digital filters for signal processing applications is becoming widespread. As the cost of digital components decreases further and the technology improves, this trend will continue, making digital signal processing even more attractive. Among the applications of digital filters are speech processing, image processing and seismic processing.

Linear time-invariant digital filters can be divided into two classes depending on the duration of the filter's impulse response : infinite duration impulse response (IIR) filters (also known as recursive filters) and finite duration impulse response (FIR) filters (also called nonrecursive or transversal filters). In this thesis only FIR linear phase filters will be considered, since the FIR filters have some properties which are more attractive for signal processing applications. Among these features are the possibility of having exact linear phase, the lack of any stability problem, and the absence of limit cycles in the actual finite word length implementation of the filter. The latter two considerations are crucial in the syntnesis of IIR filters.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Another classification of digital filters can be made in terms of the dimensions of the filter. For example, a filter which processes samples of a time duration is a one-dimensional (1-D) filter because the data is a function of one independent variable, time. On the other hand, a filter which processes data which is a function of two independent variables is called a two-dimensional (2-D) filter. For example, image processing is a 2-D filtering operation in which the two independent variables are the two spatial co-ordinates of the image; whereas, seismic processing is another 2-D filtering operation where the independent variables are time and spatial location of the geophone sensors.

This thesis will develop methods for the design of optimal Chebychev (or min-max) FIR linear phase filters and the design of windowing technique which can be used with the Image processing system built around the SEL 32/27 computer in this Department for Digital Image Processing applications. In this context, the word "design" means calculation of the impulse response coefficients in order to approximate a desired ideal frequency response.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.1 HISTORY OF THE APPROXIMATION PROBLEM

The methods for the 1-D and 2-D digital filter design are presently in an advanced state of development. After tracing the development of the 1-D problem, it is easy to see that 2-D design follows a parallel course.

1.1.1 1-D FIR digital filter design techniques

Initial methods for filter design attempted to derive the coefficients of the 1-D digital filter to approximate an ideal desired frequency response in a Fourier series and to truncate the series to the desired filter length. The resulting filter design minimized the least-squares error between the desired response and the filter response. However, the Chebychev error (the maximum absolute value of the error) from this approach is too large, due to the Gibbs phenomenon which occurs because of truncation of a previously designed impulse response.

The technique of windowing [1],[2] seeks to reduce the Gibbs phenomenon by modifying the coefficients of the Fourier series. This technique is based on the fact that multiplication in the time domain corresponds to convolution in the frequency domain. Among the more popular windows are the Kaiser window, the Hamming window and the Hanning

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

window. Windowing is an analytical technique, whereas the following optimization techniques are iterative in nature.

In 1969 Gold and Jordan [3] suggested a technique wherein the design is carriedout in the frequency domain by prescribing the frequency response at equally spaced points to be one in the passband and zero in the stopband for lowpass filters. One, two or three points are left free in the region between the passband and stopband. Then the stopband deviation is minimized by the choice of the transtion points. This is a linear programming problem with very few independent variables, but a large number of constraints. Rabiner, Gold and McGonegal [4] describe the method in detail and provide a table of filters. It should be noted that since only the stopband deviation is being minimized, it is not possible to control the passband deviation.

A year later Herrmann [5] observed that if one wanted a Chebychev like approximation, then the filter's frequency response must have an equiripple behaviour. After assumming symmetry of the impulse response and fixing the passband and stopband deviations and the number of ripples in each band, it is possible to write down a set of nonlinear equations which completely describes the filter. Herrmann proceeded to solve these equations directly using an iterative descent method.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

In view of later results [6], it is possible to show that the filters of Herrmann are a restricted subset of optimal min-max filters, and hence are called extraripple filters. One drawback of this approach is that it is not possible to know a priori the locations of the passband and stopband cutoff frequencies.

Parks and McClellan [6] formulated the low pass approximation problem as one of the weighted Chebychev approximation of the desired response on two disjoint intervals, the passband and the stopband with a transition band left unspecified. Following Herrmann, the impulse response was assumed symmetric, giving a linear phase filter and also reducing the approximation problem to a real approximation problem. Necessary and sufficient conditions were obtained from the alternation theroem [7], and the Remez exchange algorithm [2],[8] was demonstrated as an effective tool for the computation of these optimal filters.

In 1972 Rabiner [9] illustrated that the linear programming offered an alternative method for computing the best Chebychev approximation. Although linear programming is flexible and can be used to approximate a wide variety of desired filter shapes, it is comparitively slow and hence, the length of the filter it can design is limited.

In 1979, Steiglitz [10] has shown the possibility of adding extra constraints in the frequency domain and apply the technique to the filter design of FIR filters whose

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

response in passbands is constrained to be monotonically decreasing or increasing.

Kodek [11] presented the results to design optimal finite word length FIR filters using general purpose integer programming techniques. In [12] FIR digital filters with discrete coefficient values selected from the power-of-two space are designed using the methods of integer programming. The above history is that of the 1-D design problem.

1.1.2 2-D FIR digital filter design techniques

Similar to 1-D FIR filter design, techniques have been explored in the case of 2-D filters also. Huang [13] has extended the windowing technique to two dimensions for circularly symmetric windows.

Another two-dimensional FIR filter design method is the frequency sampling method developed by Hu and Rabiner [14]. When a filter is designed by frequency sampling the DFT frequency domain coefficients corresponding to pass-bands and stop-bands, are fixed. The error of the approximation is minimized with respect to the transition band DFT coefficients. The impulse response coefficients are then calculated by taking the inverse DFT. An advantage of this filter design method is that it can be used to approximate any frequency response. This method consumes lot of computation time, but not as much time as a minimax method.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Stockham, Cannon and Ingebretsen [15] found how to identify the degrading signal, when it is not known, and how to remove it through a second convolution, which is known as a deconvolution. Lewis and Sakrison [16] have used Wiener filters to reduce the noise and optical distortion in electron micrographs.

Mahry and Aggarwal [17] suggested a new technique for the design of 2-D FIR filters with non rectangular arrays. Here a high-order impulse response array of the desired filter is optimally trunctated to give a low-order impulse response array.

Rajan and Swamy [18] made a modification to the McClellan transformation method used for the design of 2-D circularly symmetric FIR digital filters from 1-D filters is described. This modification entails the embedding of variable parameters in the different transformations applied to the different factors of the 1-D reference function.

1.2 PROBLEM STATEMENT

The Signals And Systems Laboratory in the Electrical Engineering department at the university of Windsor has developed an Image Processing System. It consists of the main computational element (SEL 32/27) a high speed convolution filter, a video camera and a digitizer, colour

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

image display system with touch screen input, colour printer plotter, 32 M Byte hard disk and a 1.2 M Byte floppy diskette, EPROM programmer and system console.

The work was accomplished as a group effort. Each member of the group was assigned a unique job: such as designing the video digitizer, interfacing the colour printer to the SEL computer, interfacing image display system, design of the convolution filter etc. Finally the individual units were integrated to form the total image processing system. More details about this image processing system are given in the next chapter.

1.2.1 Contribution of this Research

As a member of the above mentioned group the work assigned to me can be stated in two parts.

1.2.1.1 Part I

1. Investigation of various two dimensional FIR filter design techniques.
2. Compute filter coefficients for our specific requirements using design techniques that are investigated.
3. Develop algorithms for scaling the filter coefficients and to compute their Number Theoretic Transform.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4. Develop algorithms for transferring the NTT of filter coefficients to the high speed NTT convolutional filter.

5. Develop algorithms for transferring image data to and from the NTT convolver.

1-2-1-2 Part II

1. Testing of the above algorithms on different images.
2. To investigate the effects on images of the various types of filters as follows:
   a) Based on the filtering procedure obtain 2-D lowpass filter, 2-D highpass filter and 2-D bandpass filter.
   b) Obtain the three filtered images using the above coefficients and the NTT convolver.
   c) Observe image smoothing, image sharpening and image enhancement effects by filtering operation.

1-2-2 Thesis organization

Chapter II explains individual units in the Image processing system built around the SEL 32/27 computer.

Different FIR filter design techniques that are suitable for our application are discussed briefly in Chapter III.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter IV discusses about the NTT convolutional filter as a part of the image processing system and the application of the 2-D FIR filter design using our image processing system.

The results of the above work are discussed in Chapter V.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter II

IMAGE PROCESSING SYSTEM

2.1 INTRODUCTION

In this chapter the details of the image processing system, developed by the Signals and Systems laboratory at The University of Windsor, are given. The aim of this system is to develop the solutions to quality control image processing problems. The central element of the system is a SEL 32/27 processor, which was chosen because of its large direct memory access, high speed processing power and in particular its high throughput rate.

2.2 SYSTEM ORGANIZATION

Fig.2.1. shows the organization of the intelligent image processing system. It consists of the main computational element (SEL 32/27), high speed NTT convolutional filter, a video camera and illumination control, colour image display system with touch screen input, colour printer/plotter etc . The SEL 32/27 is the primary computational element of the image processing system and the high speed convolution filter is used for pre-processing the images .

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

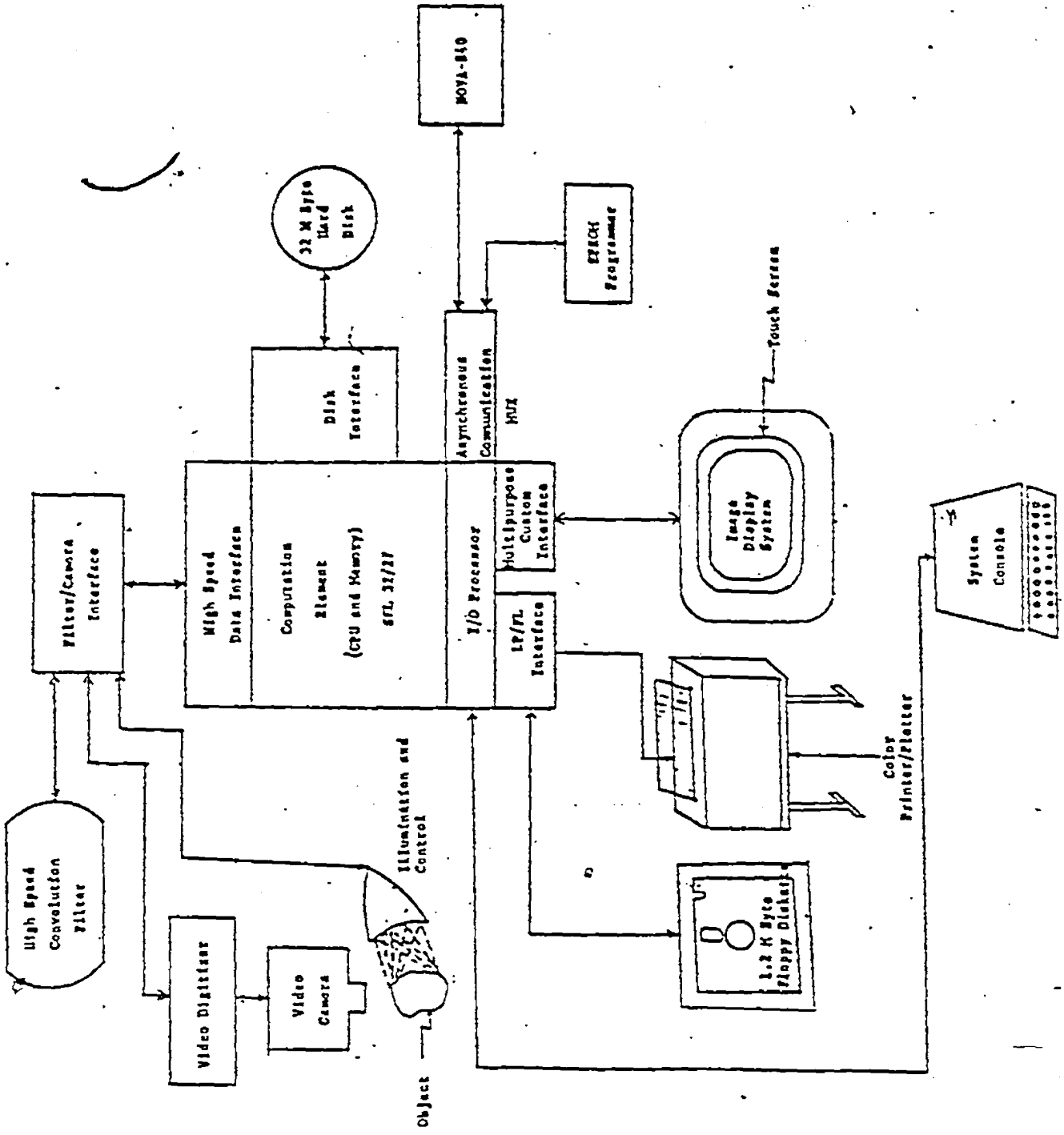


Fig. 2.1 Organization of the intelligent image processing system.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The operation of the image processing system is controlled through the system console. The touch screen input can also be used for specifying various functions such as digitizing, filtering, display etc. By activating the digitize function from the touch screen or the system console, the image of the part under inspection is digitized. The NTT convolutional filter is used to pre-process the image and is also used for template matching during the flaw detection process. The purpose of the image display system is to convert an image stored in a section of the SEL 32/27 memory into either a black and white composite video for a grey level display or to generate R , G and B signals for a pseudo colour display . The display of the processed images or the parts under inspection is of great importance for the developement and verification of various image processing algorithms. An AYDIN 5216 Image display system has been acquired for this purpose. It has got the capability of displaying a 512 X 512 image. A touch screen input device has been mounted on the display system which allows even an untrained person to use the image processing system .

The plotter is used to generate hard copies of the images of different parts . A TRILOG 100 colour plotter/printer has been acquired for this purpose . The plotter/printer system can plot images in colour on ordinary paper and also can be used for general purpose printing .

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The previous signal processing system, Data General NOVA 840 computer, is interfaced with the SEL machine to facilitate the transferring of image processing algorithms that were already developed. The EPROM Programmer is used to program the EPROMs required for implementation of residue arithmetic operations within the convoution filter.

2.3 SEL 32/27 MINI-COMPUTER

The SEL 32/27 is the basic computational element of the Image Processing System. It consists of a 32-bit processor, 512 K Bytes of MOS Memory, an input/output Processor, 32 M Bytes hard disc drive, 1.2 M Bytes floppy diskette drive, a high speed data interface, interfaces for the Line Printer/ Floppy Disc, Image Display System and a general purpose communication mutiplexer. To facilitate the developement of different image processing algorithms a Multitasking Executive (MPX-32), Fortran 77 compiler and an Assembler have also been incorporated into the image processing system.

The SEL 32/27 components are interconnected via two buses , the SEL Bus and the MP Bus. The SEL bus is a high speed 32-bit synchronous bus that can transfer data at the rate of 26.67 M bytes per sec. and each module connected to the SEL bus is assigned a unique priority. The Multipurpose

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(MP) bus is a medium speed 16-bit asynchronous bus that can transfer I/O data at the rate of 1.5 M Bytes per sec .

2.4 HIGH SPEED NTT CONVOLUTIONAL FILTER

The high speed NTT convolutional filter is the 2nd computational element of the image processing system. It is used to pre-process the image of the part under inspection and is also used for template matching during the flaw detection procedures. The design of the convolution filter is based on the Signals and Systems group researches into Number Theoretic hardware techniques and is implemented via a 2-D , radix-2 NTT computational algorithm. The entire filter has been constructed with EPROM's, adders, registers, two 128 X 128 word memory buffers and a 128 X 128 word coefficent memory. The filtering operation is performed by taking the NTT of the image, multiplying this by the NTT of the filter impulse response and then taking the inverse NTT of the product.

The NTT convolutional filter hardware is organized to compute the circular convolution of 128 X 128 image with 128 X 128 spatial filter kernel. The linear convolution of the different sizes of the images and the filter kernels have been implemented via software by using the overlap-save technique of sectioned convolutions.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The convolution filter has been interfaced to the SEL 32/27 via the High Speed Data Interface (HSD). Taking into account the I/O rates of of the HSD (3.2 M Bytes per sec.) the filter can process an image section of 128 X 128 Bytes in 83.5 ms. This time does not include the overhead of the HSD software handler. Taking this overhead into acccunt and the software overhead of the overlap save algorithm a filtering time of 0.8 sec is required to filter a 256 x 256 image by a 17 x 17 spatial filter kernel .

2.5 VIDEO DIGITIZER

The basic function of the video digitizer is to digitize the image or the part under inspection. A Hamamatsu C1000 special purpose camera is used for gathering the image. The camera is specially designed for use with digital computers and image processing hardware. The number of scanning lines in a field is in powers of two i.e. 256, 512 or 1024. The incoming signal from the camera which represents the brightness of the picture at different points, is sampled and quantized. The operation of sampling and quantizing is performed by the A/D converter within the video digitizer. The output of the A/D converter is 8 bit digitized data .

The design of the video digitizer is based on the basic principle that sampling is done only during the periods when

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the actual video signal is present, and it is stopped during the blanking periods. The circuitry consists of counters, comparators, decoders and memory buffers etc. Digitized data is transferred to the SEL 32/27 computer through the HSD interface. The convolution filter and the digitizer share the same HSD interface. This is of little concern since the filtering and the image gathering operations can be multiplexed without any loss in the computational speed of the SEL 32/27 .

The conversion rate of the A/D converter depends on the maximum rate at which digitized data can be transferred to SEL 32/27 computer memory. This rate is 3.3 MHz. To keep the output digitized data rate below this value, the sampling period has been set to 300 ns. At this sampling frequency, a 256 X 256 image can be digitized in two vidicon scans of the camera at about 33 ms.

2.6 DISPLAY UNIT - AYDIN 5216

The AYDIN 5216 is quite sophisticated and is built around the INTEL 8086 Microprocessor (16 Bit). The firmware provided has an instruction set of alphanumeric and graphic instructions and it does not provide its own operating system. The standard firmware for AYDIN is provided on one card and by including another card for the operating system the AYDIN 5216 can be used as a stand alone computer. As an

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

alternative to this, the second card may be replaced by the operating system supplied by a host computer. In this configuration the host communicates with the display unit as if it were a peripheral device of the host.

The AYDIN 5216 display at this Department is interfaced to the SEL 32/27 through a Multipurpose Custom Interface. The standard firmware provided accepts commands from the SEL computer and executes the necessary codes to generate alphanumeric or graphic data. The 5216 has a resolution of 512 X 512 pixels and upto 256 colours can be displayed on the monitor simultaneously. Apart from accepting commands from the SEL computer, the AYDIN monitor is also capable of executing commands entered through its keyboard.

The serial interface for the monitor enables 9600 BPS data transfer for loading the pixels in the refresh memory of the processor. Individual Red, Green and Blue composite outputs having 8 level intensity provide a total of 256 unique colour combinations. The pseudo colour capability and grey level translation is made possible by a RAM lookup table.

2.7 COLOUR PRINTER

The TRILOG COLORPLOT is a unique colour printer capable of plotting in multicolour or Black & White on a plain paper without any toners. It uses a colour ribbon made up of three

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

colour zones Magenta, Cyan and Yellow. The various colours are achieved by interspersing the colour dots and controlling the dot density by software. A typical example would be the interspering of the yellow and blue dots to produce a Green tone. The shade of Green (light or dark) that can be produced depends on the dot density of the two primary colours.

The combination of three colours is achieved by overwritting with the three colours. The reverse feed option of the printer is used for this purpose.

2.8 TOUCH SCREEN

The Touch Screen is a unique system providing an easy interaction between an untrained and unskilled person with a sophisticated Computer System. Access to different commands and data files is possible by merely touching the corrosponding position on the screen.

The digitizer of the touch screen measures the touch position by measuring the voltage distribution across the conducting film. Two thin transparent films are mounted in front of the CRT but are kept separated by an insulating layer at the edges. When this sandwich is touched, one conducting layer is forced into the other yielding an output voltage proportional to the touch position. The voltage is then converted to a binary number, combined with similar

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

data from other axis, filtered and formatted into ASCII characters. This data is then transmitted as a serial RS232C message.

The touch screen, when operating in the continuous mode, outputs position data continuously as long as the screen is touched. The output is updated approximately 60 times per second. In the initial touch mode the screen outputs data only on the initial touch. The screen must be released before any other output is produced. This feature is exploited for MENU selection as continuous mode is likely to transmit useless data to the host computer. The touch screen can also be used to operate in a fixed array of touch pads. The message format of the data that is sent to the host is dependent on the mode selected by the user.

In the configuration existing at this department the touch screen plays a very important role by providing access to the various peripherals and techniques. The touch screen here is trained to simulate the Executive for the Image Processing work station. It is interfaced to the SEL computer through a dedicated RS232C port.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter III

FIR FILTER DESIGN

3.1 INTRODUCTION

Two-dimensional digital filters are widely used for enhancement or low quality of images. In conjunction with this several approaches to the design of nonrecursive (FIR) and recursive (IIR) two-dimensional digital filters have been advanced in recent years. It had been reported [19] that these filters can be of better value in image processing applications if they have linear phase characteristics.

FIR filters have the advantages that they are free of stability problems, and that linear phase can easily be achieved. Hence an investigation has been made into various two-dimensional FIR digital filter techniques. The available design techniques are the window technique [2,13], linear programming [14], the maximally-flat criterion [20], the Remez exchange algorithm [21], or by transformation of 1-D digital filters [22,23].

Two methods which can be used to design lowpass, highpass and bandpass two-dimensional FIR linear phase digital filters have been described in this chapter. The

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

first method uses a 1-D Remez exchange algorithm technique described in Reference 2. This method yields one-dimensional linear phase digital filters. Then the transformation technique suggested by McClellan [22] has been used to obtain lowpass, highpass and bandpass two-dimensional linear phase FIR digital filters. The second method, uses a two-dimensional windowing technique in order to obtain lowpass, highpass and bandpass two-dimensional linear phase FIR digital filters. The next chapter explains how to use these filter design techniques and the high speed NTT convolver for image processing applications.

3.2 DESIGN OF 1-D LINEAR PHASE FIR DIGITAL FILTERS

The four types of linear phase FIR digital filters are

1. Positive symmetry, even length
2. Positive symmetry, odd length
3. Negative symmetry, even length and
4. Negative symmetry, odd length.

All these four types of filters can be rewritten in a common form so that a single central computation routine (based on the Remez exchange method) can be used to calculate the best approximation in each of the four cases. An algorithm can now be designed to calculate the filter impulse response.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3-2-1 Description of the design algorithm

As seen from Fig. 3.1, the design algorithm consists of the following steps:

1. An input section in which the desired frequency response, the weighting function and the filter duration are specified.
2. A formulation of the appropriate equivalent approximation problem.
3. Solution of the approximation problem using the Remez exchange algorithm.
4. Calculate the impulse response.
5. Calculate the frequency response.

Step 1 is the interface between the designer and the filter design algorithm. The user communicates the type of the filter to be designed and the particular specifications that must be met. In the next step the optimal linear phase FIR filter is formulated into a Chebychev approximation problem.

In step 3, the Remez exchange algorithm is used in order to find the best approximation to the desired frequency response. Here a dense grid of frequency points is used to find a set of extremal frequencies required by the alternation theorem [2]. Then this theorem has been applied to find the necessary and sufficient conditions in order to get the unique best approximation to the desired frequency response. Afterwards the impulse response and the frequency response are obtained.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

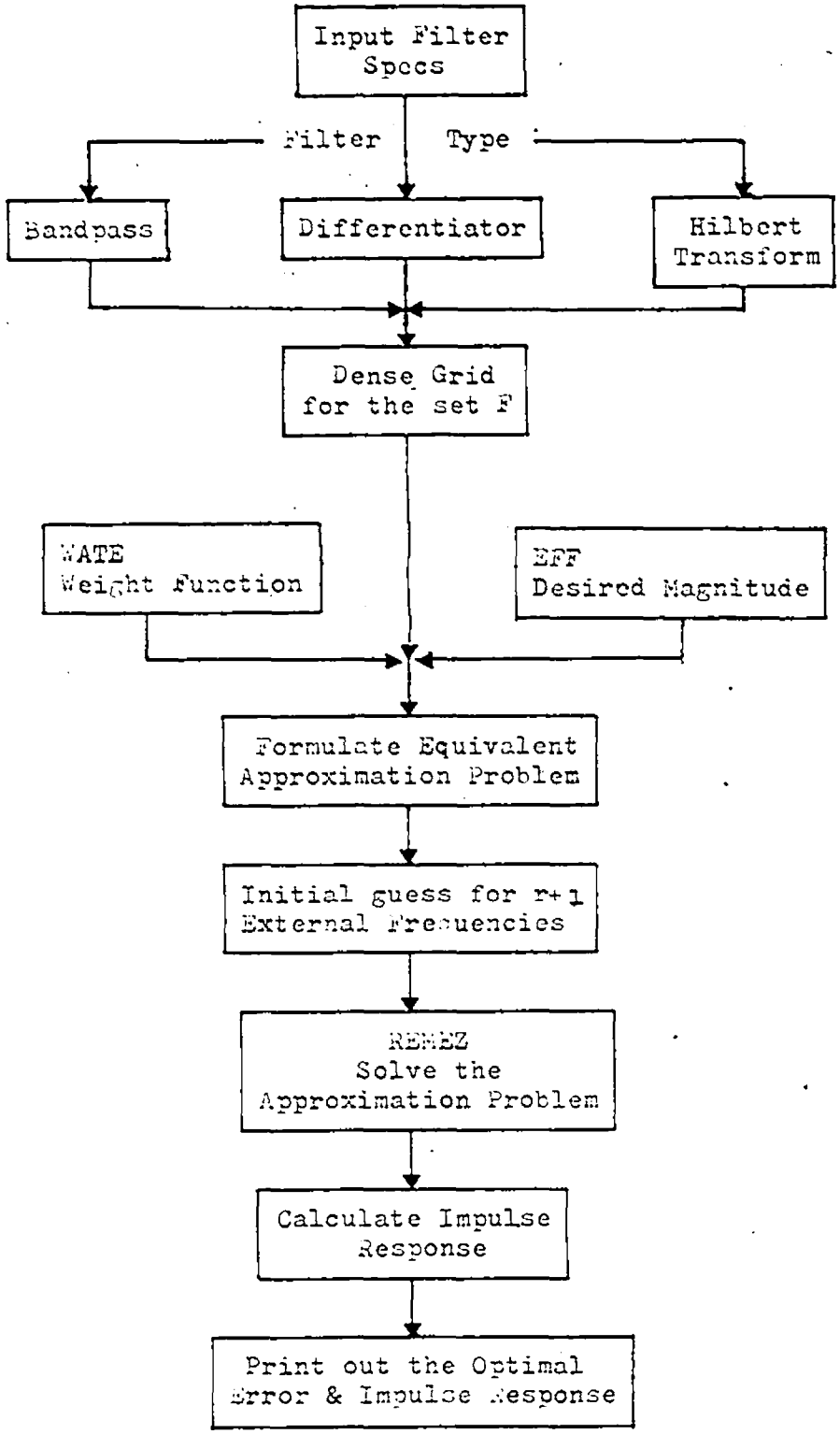


Fig. 3.1 Block diagram of the general design algorithm.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

A detailed program listing of the generalized design program using this algoritam is given in

3.2.2 Examples of design

Figures 3.2-3.7 show the specific examples of use of the design program for several typical filters of interest. For each of these filters, one figure shows the computer output listing of the impulse response and the other figure shows a plot of the frequency response.

Example 1- Lowpass Filter

Design a 17-point lowpass filter with passband cutoff frequency of 0.1 and stopband cutoff frequency of 0.2 and ripple ratio of 1.0.

The input parameters for this example are N=17, Fp=0.1, Fs=0.2, and K=1.0. Fig.3.2 shows the computer output listing of the impulse response (along with the specifications of the filter) and Fig.3.3 shows the frequency response of the resulting lowpass filter. The computational time for designing the desired lowpass filter is 1.1 sec. on the SEL computer.

Example 2- Highpass Filter

Design a 17-point highpass filter with stopband cutoff frequency of 0.1 and passband cutoff frequency of 0.2 and ripple ratio of 1.0.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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

                    FINITE IMPULSE RESPONSE(FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    LOWPASS FILTER

          FILTER LENGTH=  17

          ******** IMPULSE RESPONSE **********
               H(  1) =   .16674664E-01 =H(  17)
               H(  2) =   .49132071E-02 =H(  16)
               H(  3) =  -.19674193E-01 =H(  15)
               H(  4) =  -.46255231E-01 =H(  14)
               H(  5) =  -.39681047E-01 =H(  13)
               H(  6) =   .26796240E-01 =H(  12)
               H(  7) =   .14223689E+00 =H(  11)
               H(  8) =   .25432426E+00 =H(  10)
               H(  9) =   .30098733E+00 =H(   9)

                    BAND  1          BAND  2
LOWER BAND EDGE      .000000000      .199999988
UPPER BAND EDGE      .100000024      .500000000
DESIRED VALUE       1.000000000      .000000000
WEIGHTING           1.000000000     1.000000000
DEVIATION            .020443056      .020443056
DEVIATION IN DB   -33.789093018  -33.789093018

EXTERNAL FREQUENCIES
    .0000000    .0763888    .1000000    .2000000    .2208332
    .2659720    .3215273    .3805549    .4395824    .5000000

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

Fig. 3.2 1-D Impulse response for the design of a 17 point lowpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

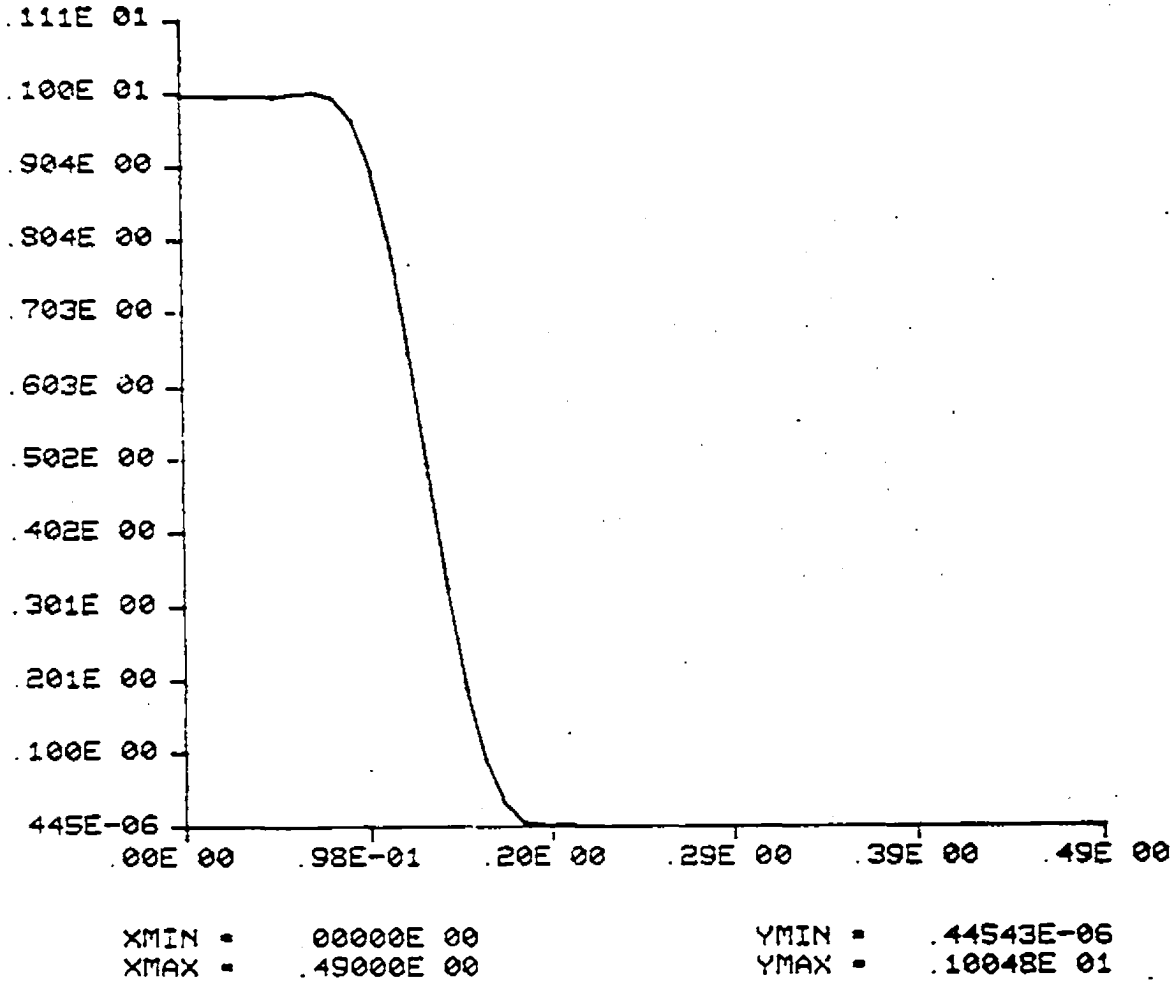


Fig. 1-1 Frequency response for the design of a 17-point lowpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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

                    FINITE IMPULSE RESPONSE(FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    HIGHPASS FILTER

          FILTER LENGTH=  17

          ******** IMPULSE RESPONSE ***********
               H(  1) =  -.16674738E-01 =H(  17)
               H(  2) =  -.49131438E-02 =H(  16)
               H(  3) =   .19674134E-01 =H(  15)
               H(  4) =   .46255291E-01 =H(  14)
               H(  5) =   .39680988E-01 =H(  13)
               H(  6) =  -.26796192E-01 =H(  12)
               H(  7) =  -.14223595E+00 =H(  11)
               H(  8) =  -.25432420E+00 =H(  10)
               H(  9) =   .69911271E+00 =H(   9)

                    BAND  1          BAND  2
LOWER BAND EDGE      .000000000       .199999988
UPPER BAND EDGE      .100000024       .500000000
DESIRED VALUE        .000000000      1.000000000
WEIGHTING           1.000000000      1.000000000
DEVIATION            .020443056       .020443056
DEVIATION IN DB   -33.789093018  -33.789093018

EXTERNAL FREQUENCIES
    .0000000    .0763688    .1000000    .2000000    .2208332
    .2659720    .3215273    .3805549    .4395824    .5000000

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

Fig. 3.4 1-D Impulse response for the design of a 17 point highpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

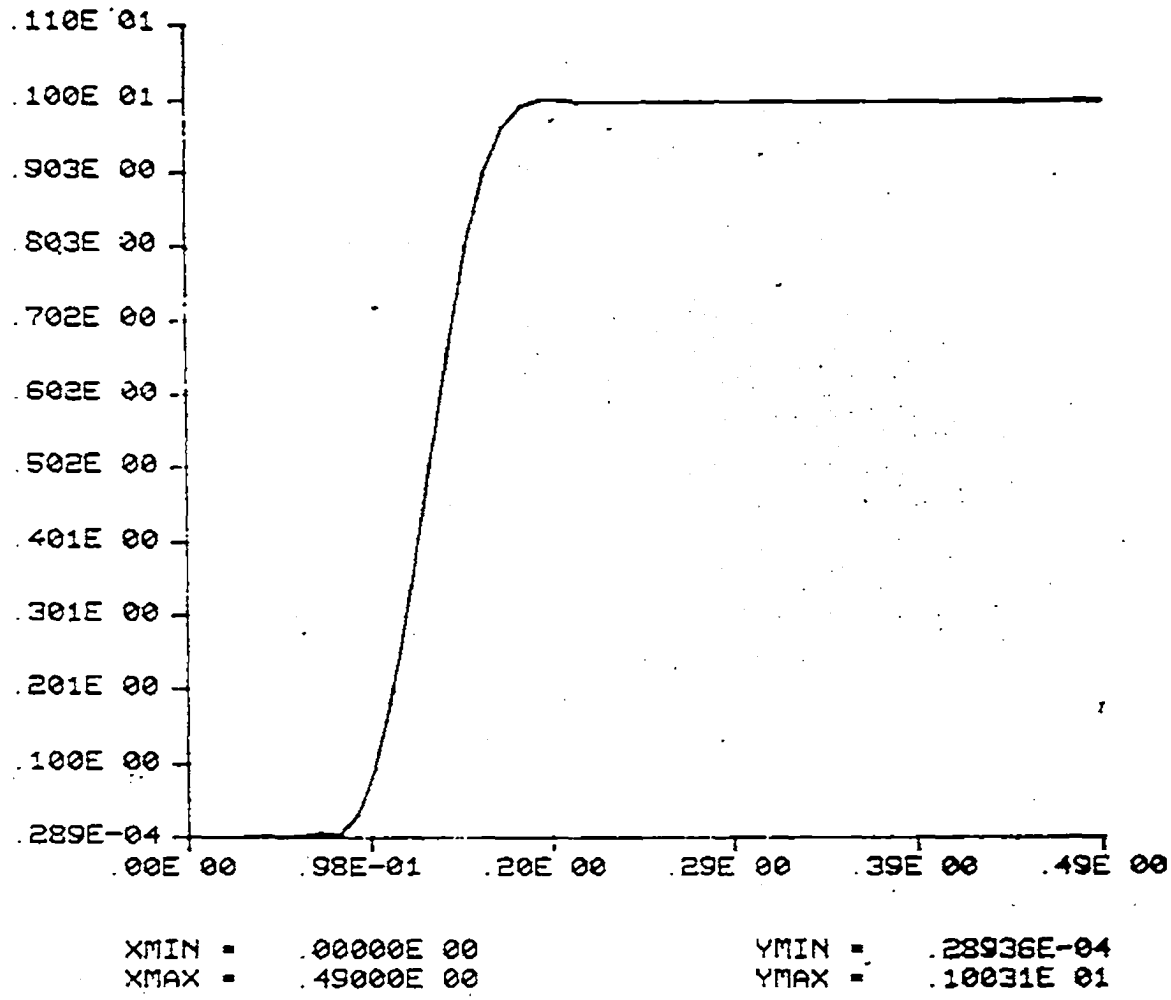


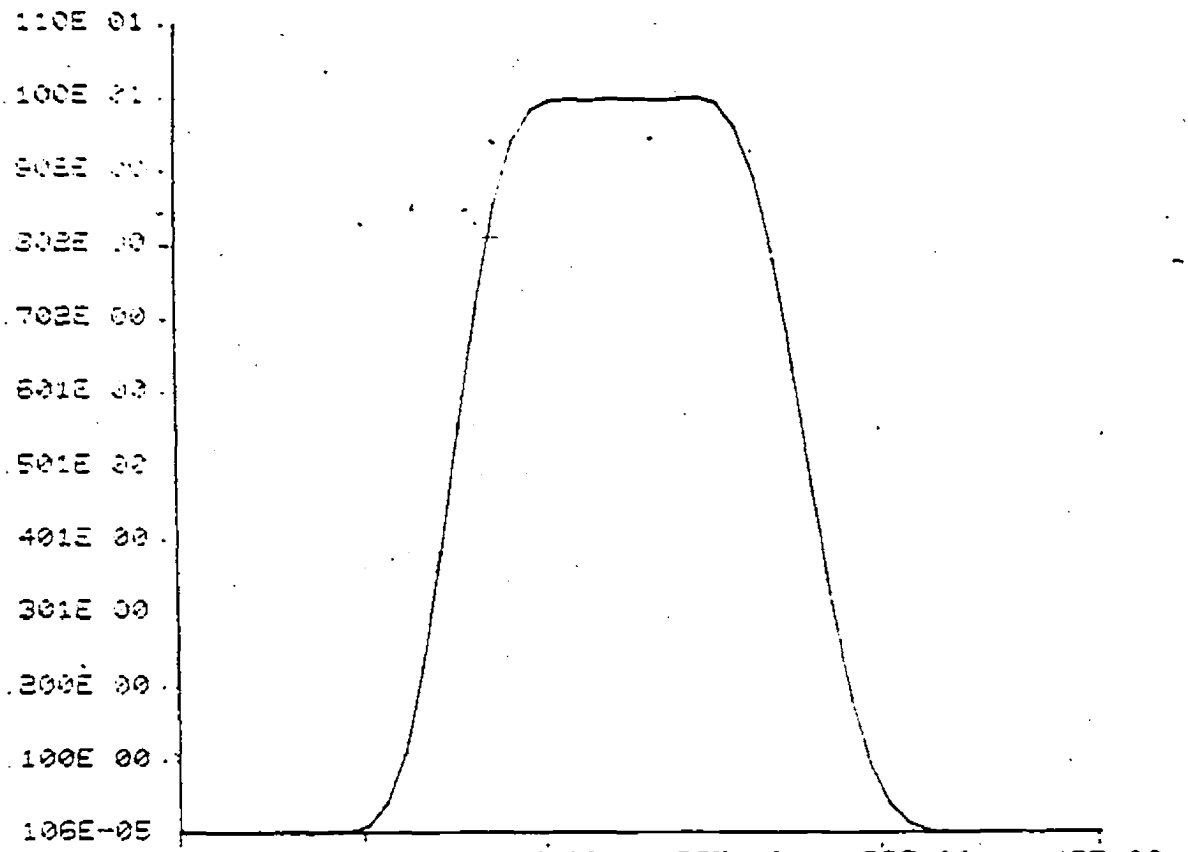
Fig.3.5 Frequency response for the design of a 17-point highpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

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

                    FINITE IMPULSE RESPONSE(FIR)
                    LINEAR PHASE DIGITAL FILTER DESIGN
                    REMEZ EXCHANGE ALGORITHM

                    BANDPASS FILTER

          FILTER LENGTH=  17

          ******** IMPULSE RESPONSE ***********
               H(  1) =  -.32642782E-01 =H(  17)
               H(  2) =  -.10302386E-03 =H(  16)
               H(  3) =   .38230658E-01 =H(  15)
               H(  4) =  -.95644502E-04 =H(  14)
               H(  5) =   .78324497E-01 =H(  13)
               H(  6) =  -.88665955E-04 =H(  12)
               H(  7) =  -.28762698E+00 =H(  11)
               H(  8) =  -.19169913E-03 =H(  10)
               H(  9) =   .39814264E+00 =H(   9)

                    BAND  1         BAND  2         BAND  3
LOWER BAND EDGE      .000000000      .199999988      .399999976
UPPER BAND EDGE      .100000024      .300000012      .500000000
DESIRED VALUE        .000000000     1.000000000      .000000000
WEIGHTING           1.000000000     1.000000000     1.000000000
DEVIATION            .011614893      .011614893      .011614893
DEVIATION IN DB   -38.699691772  -38.699691772  -38.699691772

EXTERNAL FREQUENCIES
     .0833333     .1000000     .2000000     .2173610     .2486109
     .2833330     .3000000     .4000000     .4173610     .4555553

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

Fig. 3.6 1-D Impulse response for the design of a 17-point bandpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.



XMIN = 00000E 00 YMIN = 10590E-05
XMAX = 49000E 00 YMAX = 10024E 01

Fig. 7.7 Frequency response for the design of a 17-point bandpass filter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The input parameters for this example are N=17, Fs=0.1, Fp=0.2 and K=1.0. Fig.3.4 shows the computer output listing of the impulse response (along with the specifications of the filter) and Fig.3.5 shows the frequency response response of the resulting highpass filter. The computational time for designing the desired highpass filter is 1.1 sec. on SEL computer.

Example 3- Bandpass Filter

Design a 17-point bandpass filter with stopband cutoff frequencies of 0.1 and 0.4 and passband cutoff frequencies of 0.2 and 0.3 and with ripple weights of 1 in the stopbands and passband.

The input parameters for this example are N=17, Fs1=0.1, Fs2=0.4, Fp1=0.3, Fp2=0.4 and K=1.0. Fig.3.6 shows the computer output listing of the impulse response (along with the specifications of the filter) and Fig.3.7 shows the frequency response of the resulting bandpass filter. The computational time for designing the desired bandpass filter is 1.4 sec. on the SEL computer.

In the next section a procedure has been explained how to obtain two-dimensional digital filters from this one-dimensional digital filter using McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.3 MCCLELLAN TRANSFORMATION FOR 2-D FIR DIGITAL FILTER DESIGN

The approximation problem for the 2-D digital filter is a much more difficult problem than the corresponding 1-D design problem. Some of the 1-D design techniques have been extended to two dimension [13,14] ; but for other techniques such a generalization appears unlikely. In particular, there is no efficient design procedure available in two dimensions for min-max filters. Even if the method could be extended, the size of the problem is a handicap. For example, the design of an optimum 31 X 31 FIR linear phase filter involves optimization over 16 X 16 = 256 parameters.

In view of these difficulties, it seems appropriate to look for methods of design which don't necessarily yield optimal filters, but which are computationally efficient. Within such a class of methods are the windowing [13] and the frequency sampling technique [14]. The method to be described here is based on McClellan transformation of 1-D filters [22].

The technique consists of transforming a 1-D filter into a 2-D filter by a change of varibles. The class of filter which can be approximated by this method is some what limited, but included in the class are many important filters, such as those filters whose magnitude is constant over regions of the 2-D frequency plane (e.g., lowpass and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

bandpass circularly symmetric filters). The primary advantage of this technique is that it is very fast, because the computing time is devoted almost entirely to the 1-D design which is known to be efficient [6].

3.3.1 The transformation

The McClellan transformation is a transformation of a 1-D linear phase FIR filter into a 2-D linear phase FIR filter by a substitution of variables. So this tranformation can be applied to the 1-D linear phase FIR filters designed in the previous section.

The frequency response of the 1-D digital filter of odd length, positive symmetry with impulse response h(n) can be written as

$$H(e^{jw}) = h(0) + \sum_{n=1}^{N} h(n)\left[e^{-jwn} + e^{jwn}\right]$$

$$= h(0) + \sum_{n=1}^{N} 2h(n)\cos wn. \quad \text{----- (3.1)}$$

and the frequency response of the 2-D filters with impulse response h(m,n) can be written as

$$H(e^{jw1}, e^{jw2}) = \sum_{m=0}^{M1}\sum_{n=0}^{M2} a(m,n)\cos mw1 \cos nw2 \text{ --- (3.2)}$$

The generalized McClellan transformation converts 1-D filters of the form (3.1) into 2-D filters of the form (3.2) by means of the substitution

$$\cos w = A\cos w1 \cos w2 + B\cos w1 + C\cos w2 + D$$

----- (3.3)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The relation between h(n) and h(m,n) can be seen by rewriting (3.1) in the form

$$H(e^{jw}) = \sum_{n=0}^{N} b(n)[\cos w]^n \quad \text{----- (3.4)}$$

and performing the substitution indicated in (3.3)

$$H(e^{jw1}, e^{jw2}) = \sum_{n=0}^{N} b(n)[A \cos w1 \cos w2 + B \cos w1 + C \cos w2 + D] \quad \text{------- (3.5)}$$

By exploiting the recurrence formula for Chebychev polynomials, this can be written in the form of (3.2). The 2-D filter which results from the transformation is of size (2N+1 X 2N+1).

3.3.2 The Design Procedure

1. Obtain the one-dimensional linear phase FIR digital filter with impulse response h(n) and the frequency response $H(e^{jw}) = h(0) + \sum_{n=1}^{N} 2\, h(n) \cos wn$
2. Rewrite the frequency response as $H(e^{jw}) = \sum_{n=0}^{N} b(n) [\cos w]^n$
3. Apply the generalized McClellan transformation which converts 1-D filters into 2-D filters by means of the substitution

   $\cos w = A \cos w1 \cos w2 + B \cos w1 + C \cos w2 + D.$

   a) For circularly symmetric filters the constants A,B,C and D are choosen as A = B = C = -D = 0.5.

   b) For fan filters the constants A,B,C and D are choosen as A = -B = 0.5 and C = D = 0.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.3.3 Results of the 2-D design technique

Example 1- Lowpass filter

Design a lowpass circularly symmetric filter, (with A=B=C=-D =0.5) which is useful for image processing problems, of spatial filter kernal 17 X 17 with the following specifications:

passband: $0.0 \leq r \leq 0.2\pi$ $r=(w1^2+w2^2)^{1/2}$

stopband: $0.4\pi \leq r \leq \pi$

desired value in passband: 1.0

desired value in stopband: 0.0

maximum allowed deviation = 0.1

As discussed in section 3.2 a 17 point 1-D FIR filter was designed. Then the McClellan transformation has been used to covert this 1-D filter into 2-D filter. Fig.3.8 shows the frequency response of the lowpass 2-D FIR digital filters. The computational time to design this filter is 0.5 sec. on the SEL computer.

Example 2- Highpass filter

Design a highpass circularly symmetric filter, (with A=B=C=-D =0.5) which is useful for image processing problems, of spatial filter kernal 17 X 17 with the following specifications:

passband: $0.4\pi \leq r \leq \pi$ $r=(w1^2+w2^2)^{1/2}$

stopband: $0.0 \leq r \leq 0.2\pi$

desired value in passband: 1.0

desired value in stopband: 0.0

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

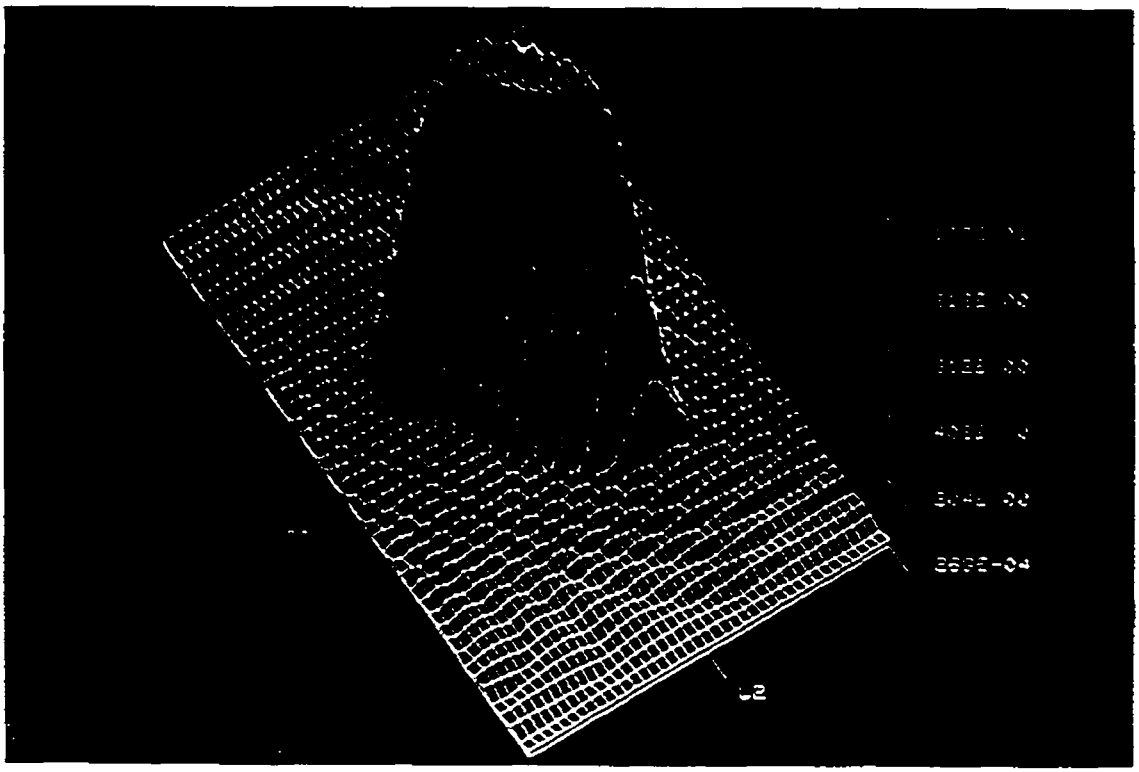


Fig.3.8 Frequency response of a lowpass 2-D FIR digital filter using McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

maximum allowed deviation = 0.1

As discussed in section 3.2 a 17 point 1-D FIR filter was designed. Then the McClellan transformation has been used to covert this 1-D filter into 2-D filter. Fig.3.9 shows the frequency response of the highpass 2-D FIR digital filters. The computational time to design this filter is 0.5 sec. on the SEL computer.

Example 3- Bandpass filter

Design a bandpass circularly symmetric filter, (with A=B=C=-D =0.5)

which is useful for image processing problems, of spatial filter 17 X 17 with the following specifications:

passband: $0.4\pi \leq r \leq 0.6\pi$ $r=(w_1^2+w_2^2)^{1/2}$

stopband: $0.0 \leq r \leq 0.2\pi$

$0.8\pi \leq r \leq \pi$

desired value in passband: 1.0

desired value in stopbands: 0.0

maximum allowed deviation = 0.1

As discussed in section 3.2 a 17 point 1-D FIR filter was designed. Then the McClellan transformation has been used to covert this 1-D filter into 2-D filter. Fig.3.10 shows the frequency response cross-section of the bandpass 2-D FIR digital filters. The computational time to design this filter is 0.5 sec. on SEL computer.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

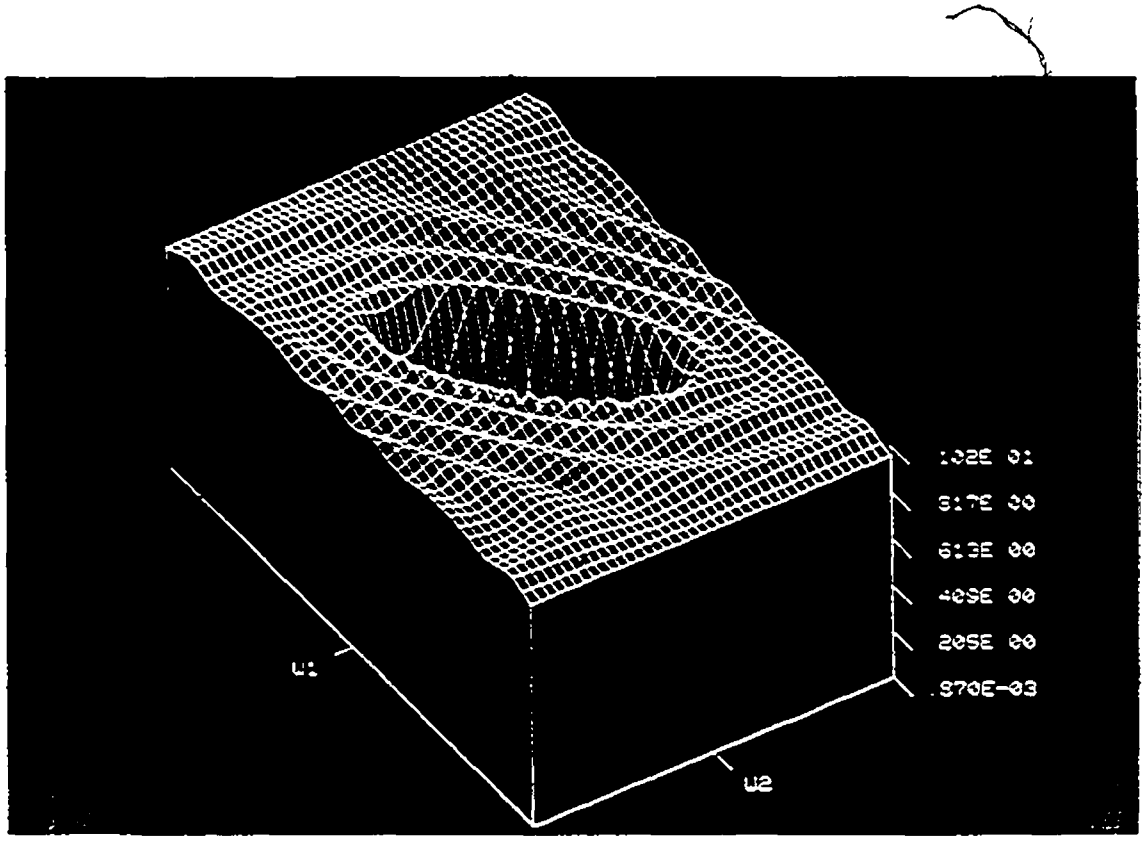


Fig.3.9 Frequency response of a highpass 2-D FIR digital filter using McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

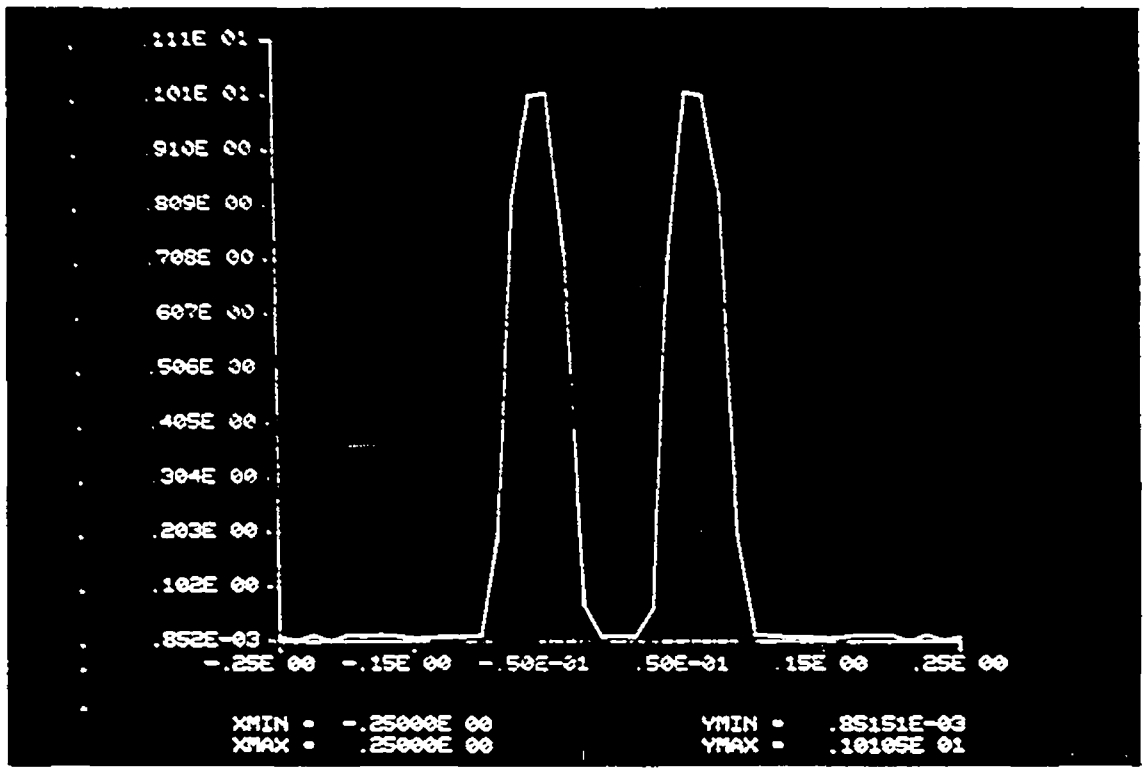


Fig. 3.10 Cross section of the frequency response of a bandpass 2-D FIR digital filter using McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4 FIR FILTER DESIGN USING WINDOWING TECHNIQUE

3.4.1 Window characteristics

One possible way of obtaining an FIR filter that approximates the frequency response of any digital filter would be to truncate the infinite Fourier series $h(n_1, n_2)$. Direct truncation of the series leads to the well known Gibbs phenomenon. A more successful way of obtaining an FIR filter is to use a finite weighting window sequence $\hat{w}(n_1, n_2)$ to modify the Fourier series coefficients. The task for windowing is to choose an appropriate window function $\hat{w}(n_1, n_2)$. The desired properties of the transform of the window are

1. $\hat{w}(e^{jw_1}, e^{jw_2})$ should approximate a circularly symmetric function.
2. The volume under the main lobe of $\hat{w}(e^{jw_1}, e^{jw_2})$ should be large.
3. The volume under the side lobes should be small.

It has been shown by Huang [13] that good two-dimensional windows satisfying the above properties can be obtained from good one-dimensional windows via the relation

$$\hat{w}(n_1, n_2) = w(\sqrt{n_1^2 + n_2^2})$$

where w is an appropriate one-dimensional window sampled at the appropriate values.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.2 Design method

If we denote the N-point windows as $w(n)$, for $0 \leq n \leq N-1$ and the circularly symmetric two-dimensional window $\hat{w}(n_1, n_2)$ can be obtained via the relation $\hat{w}(n_1, n_2) = w(\sqrt{n_1^2 + n_2^2})$, and we denote the impulse response of the digital filter (obtained as the inverse Fourier transform of the ideal frequency response of the filter) as $\hat{h}(n_1, n_2)$, then the windowed digital filter is given as

$$h(n_1, n_2) = \hat{w}(n_1, n_2)\hat{h}(n_1, n_2) \qquad 0 \leq n_1 \leq N1-1$$
$$0 \leq n_2 \leq N2-1$$
$$= 0 \quad \text{otherwise} \qquad \text{------(3.6)}$$

In order to design windowed filter requires specification of

1. Type of ideal filter- i.e., lowpass, highpass, bandpass or bandstop filter.
2. Filter cutoff frequencies.
3. Type of window- i.e., rectangular, triangular, Hamming, Hanning, Chebychev or Kaiser.
4. Window duration in samples.

From the filter specifications the sequences $\hat{w}(n_1, n_2)$ and $\hat{h}(n_1, n_2)$ are computed and the windowed filter is obtained as the final output.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.3 Form of Windows

The seven windows used here are

1. Rectangular Window
2. Triangular Window
3. Hamming Window
4. Hanning Window
5. Generalized Hamming Window
6. Kaiser Window and
7. Dolph-Chebychev Window.

3.4.3.1 Rectangular Window

For N odd, $w(n)=1 \quad -(N-1)/2 \leq n \leq (N-1)/2$ -------(3.7a)

For N even, $w(n)=1 \quad -(N/2) \leq n \leq (N/2-1)$ -------(3.7b)

3.4.3.2 Triangular Window

For N odd $w(n)=1-|2n|/(N+1) \quad -(N-1)/2 \leq n \leq (N-1)/2$

------(3.8a)

For N even $w(n)=1-|2n+1|/N \quad -(N/2) \leq n \leq (N/2-1)$

------(3.8b)

3.4.3.3 Hamming Window

For N odd,

$w(n)=0.54+0.46\cos[2\pi n/(N-1)] \quad -(N-1)/2 \leq n \leq (N-1)/2$

------(3.9a)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For N even,

$$w(n) = 0.54 + 0.46 \cos[2\pi(2n+1)/2(N-1)] \quad -(N/2) \leq n \leq (N/2-1)$$

------(3.9b)

3.4.3.4 Hanning Window

For N odd,

$$w(n) = 0.5 + 0.5 \cos[2\pi n/(N+1)] \quad -(N-1)/2 \leq n \leq (N-1)/2$$

------(3.10a)

For N even,

$$w(n) = 0.5 + 0.5 \cos[2\pi(2n+1)/2(N+1)] \quad -(N/2) \leq n \leq (N/2-1)$$

------(3.10b)

3.4.3.5 Generalized Hamming Window

For N odd,

$$w(n) = \alpha + (1-\alpha)\cos[2\pi n/(N-1)] \quad -(N-1)/2 \leq n \leq (N-1)/2$$

------(3.11a)

For N even,

$$w(n) = \alpha + (1-\alpha)\cos[2\pi(2n+1)/2(N-1)] \quad -(N/2) \leq n \leq (N/2-1)$$

------(3.11b)

where α is variable parameter.

3.4.3.6 Kaiser Window

For N odd,

$$w(n) = \frac{I_o\left[\beta\sqrt{1-\frac{4n^2}{(N-1)^2}}\right]}{I_o(\beta)} \quad -(N-1)/2 \leq n \leq (N-1)/2$$

-------(3.12a)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

For N even,

$$w(n) = \frac{I_o\left[\beta\sqrt{1-\frac{4(n+1/2)^2}{(N-1)^2}}\right]}{I_o(\beta)} \quad -(N/2) \leq n \leq (N/2-1)$$

-------(3.12b)

where β is window parameter related to minimum stopband attenuation.

3.4.3.7 Dolph-Chebychev Window

w(n) is obtained as the inverse DFT of the Chebychev polynomial, evaluated at N equally spaced frequencies around the unit circle. The parameter of the Dolph-Chebychev window are the ripple, the filter length and the normalized transition width. Only 2 of the 3 parameters can be indepentently specified [2]. If ripple and transition width are specified, then the filter length N is determined by the equation

$$N \geq 1 + \left[\frac{\cosh^{-1}\left[\frac{1+\delta p}{\delta p}\right]}{\cosh^{-1}\left[\frac{1}{\cos(\pi \Delta F)}\right]}\right] \quad ----(3.13)$$

If the filter length and ripple are specified, then the transition width is determined as

$$\Delta F = \frac{1}{\pi}\cos^{-1}\left[\frac{1}{\cosh\left[\frac{\cosh^{-1}\left[\frac{1+\delta p}{\delta p}\right]}{N-1}\right]}\right] \quad ----(3.14)$$

Finally if the parameters filter length and transition width are specified, then the ripple δp is determined as

$$\delta p = \frac{1}{\cosh\left[(N-1)\cosh^{-1}\left[\frac{1}{\cos(\pi \Delta F)}\right]\right]-1} \quad -----(3.15)$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

3.4.4 Results of the 2-D window design technique

As in the case of McClellan transformation design technique results here in the case of windowing also similar types filters like lowpass, highpass and bandpass filters are considered. The impulse response of all these filters can be obtained for different kinds of windows discussed in the previous section.

Example 1- Lowpass filter

Design a lowpass circularly symmetric filter, which is useful for image processing problems, of spatial filter kernal 17 X 17 with the following specifications:

passband: $0.0 \leq r \leq 0.2\pi$ $r=(w_1^2+w_2^2)^{1/2}$

stopband: $0.4\pi \leq r \leq \pi$

desired value in passband: 1.0

desired value in stopband: 0.0

maximum allowed deviation = 0.1

As discussed a 2-D FIR filter has been designed using Hamming window. Fig.3.11 shows the impulse response of the lowpass 2-D FIR digital filters. The computational time to design this filter is 4.8 sec. on SEL computer.

Example 2- Highpass filter

Design a highpass circularly symmetric filter, which is useful for image processing problems, of spatial filter kernal 17 X 17 with the following specifications:

passband: $0.4\pi \leq r \leq \pi$ $r=(w_1^2+w_2^2)^{1/2}$

stopband: $0.0 \leq r \leq 0.2\pi$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

FINITE IMPULSE RESPONSE (FIR)
DIGITAL FILTER DESIGN USING
WINDOWING TECHNIQUE
LOWPASS FILTER

FILTER KERNEL SIZE = 17 X 17

```
H( 1 1)=H( 1 17)=.0000E+00=H( 17 1)=H( 17 17)
H( 1 2)=H( 1 16)=.0000E+00=H( 17 2)=H( 17 16)
H( 1 3)=H( 1 15)=.0000E+00=H( 17 3)=H( 17 15)
H( 1 4)=H( 1 14)=.0000E+00=H( 17 4)=H( 17 14)
H( 1 5)=H( 1 13)=.0000E+00=H( 17 5)=H( 17 13)
H( 1 6)=H( 1 12)=.9877E-03=H( 17 6)=H( 17 12)
H( 1 7)=H( 1 11)=.9877E-03=H( 17 7)=H( 17 11)
H( 1 8)=H( 1 10)=.9877E-03=H( 17 8)=H( 17 10)
H( 1 9)=H( 1 9)=.9877E-03=H( 17 9)=H( 17 9)
H( 2 1)=H( 2 17)=.0000E+00=H( 16 1)=H( 16 17)
H( 2 2)=H( 2 16)=.0000E+00=H( 16 2)=H( 16 16)
H( 2 3)=H( 2 15)=.0000E+00=H( 16 3)=H( 16 15)
H( 2 4)=H( 2 14)=.9877E-03=H( 16 4)=H( 16 14)
H( 2 5)=H( 2 13)=.9877E-03=H( 16 5)=H( 16 13)
H( 2 6)=H( 2 12)=.1420E-02=H( 16 6)=H( 16 12)
H( 2 7)=H( 2 11)=.1420E-02=H( 16 7)=H( 16 11)
H( 2 8)=H( 2 10)=.1420E-02=H( 16 8)=H( 16 10)
H( 2 9)=H( 2 9)=.1420E-02=H( 16 9)=H( 16 9)
H( 3 1)=H( 3 17)=.0000E+00=H( 15 1)=H( 15 17)
H( 3 2)=H( 3 16)=.0000E+00=H( 15 2)=H( 15 16)
H( 3 3)=H( 3 15)=.9877E-03=H( 15 3)=H( 15 15)
H( 3 4)=H( 3 14)=.9877E-03=H( 15 4)=H( 15 14)
H( 3 5)=H( 3 13)=.1420E-02=H( 15 5)=H( 15 13)
H( 3 6)=H( 3 12)=.1420E-02=H( 15 6)=H( 15 12)
H( 3 7)=H( 3 11)=.2651E-02=H( 15 7)=H( 15 11)
H( 3 8)=H( 3 10)=.2651E-02=H( 15 8)=H( 15 10)
H( 3 9)=H( 3 9)=.2651E-02=H( 15 9)=H( 15 9)
H( 4 1)=H( 4 17)=.0000E+00=H( 14 1)=H( 14 17)
H( 4 2)=H( 4 16)=.9877E-03=H( 14 2)=H( 14 16)
H( 4 3)=H( 4 15)=.9877E-03=H( 14 3)=H( 14 15)
H( 4 4)=H( 4 14)=.1420E-02=H( 14 4)=H( 14 14)
H( 4 5)=H( 4 13)=.2651E-02=H( 14 5)=H( 14 13)
H( 4 6)=H( 4 12)=.2651E-02=H( 14 6)=H( 14 12)
H( 4 7)=H( 4 11)=.4493E-02=H( 14 7)=H( 14 11)
H( 4 8)=H( 4 10)=.4493E-02=H( 14 8)=H( 14 10)
H( 4 9)=H( 4 9)=.4493E-02=H( 14 9)=H( 14 9)
H( 5 1)=H( 5 17)=.0000E+00=H( 13 1)=H( 13 17)
H( 5 2)=H( 5 16)=.9877E-03=H( 13 2)=H( 13 16)
H( 5 3)=H( 5 15)=.1420E-02=H( 13 3)=H( 13 15)
H( 5 4)=H( 5 14)=.2651E-02=H( 13 4)=H( 13 14)
H( 5 5)=H( 5 13)=.2651E-02=H( 13 5)=H( 13 13)
H( 5 6)=H( 5 12)=.4493E-02=H( 13 6)=H( 13 12)
H( 5 7)=H( 5 11)=.6667E-02=H( 13 7)=H( 13 11)
H( 5 8)=H( 5 10)=.6667E-02=H( 13 8)=H( 13 10)
H( 5 9)=H( 5 9)=.6667E-02=H( 13 9)=H( 13 9)
H( 6 1)=H( 6 17)=.9877E-03=H( 12 1)=H( 12 17)
H( 6 2)=H( 6 16)=.1420E-02=H( 12 2)=H( 12 16)
H( 6 3)=H( 6 15)=.1420E-02=H( 12 3)=H( 12 15)
H( 6 4)=H( 6 14)=.2651E-02=H( 12 4)=H( 12 14)
H( 6 5)=H( 6 13)=.4493E-02=H( 12 5)=H( 12 13)
H( 6 6)=H( 6 12)=.6667E-02=H( 12 6)=H( 12 12)
H( 6 7)=H( 6 11)=.6667E-02=H( 12 7)=H( 12 11)
H( 6 8)=H( 6 10)=.8840E-02=H( 12 8)=H( 12 10)
H( 6 9)=H( 6 9)=.8840E-02=H( 12 9)=H( 12 9)
H( 7 1)=H( 7 17)=.9877E-03=H( 11 1)=H( 11 17)
H( 7 2)=H( 7 16)=.1420E-02=H( 11 2)=H( 11 16)
H( 7 3)=H( 7 15)=.2651E-02=H( 11 3)=H( 11 15)
H( 7 4)=H( 7 14)=.4493E-02=H( 11 4)=H( 11 14)
H( 7 5)=H( 7 13)=.6667E-02=H( 11 5)=H( 11 13)
H( 7 6)=H( 7 12)=.6667E-02=H( 11 6)=H( 11 12)
H( 7 7)=H( 7 11)=.8840E-02=H( 11 7)=H( 11 11)
H( 7 8)=H( 7 10)=.1068E-01=H( 11 8)=H( 11 10)
H( 7 9)=H( 7 9)=.1068E-01=H( 11 9)=H( 11 9)
H( 8 1)=H( 8 17)=.9877E-03=H( 10 1)=H( 10 17)
H( 8 2)=H( 8 16)=.1420E-02=H( 10 2)=H( 10 16)
H( 8 3)=H( 8 15)=.2651E-02=H( 10 3)=H( 10 15)
H( 8 4)=H( 8 14)=.4493E-02=H( 10 4)=H( 10 14)
H( 8 5)=H( 8 13)=.6667E-02=H( 10 5)=H( 10 13)
H( 8 6)=H( 8 12)=.8840E-02=H( 10 6)=H( 10 12)
H( 8 7)=H( 8 11)=.1068E-01=H( 10 7)=H( 10 11)
H( 8 8)=H( 8 10)=.1191E-01=H( 10 8)=H( 10 10)
H( 8 9)=H( 8 9)=.1191E-01=H( 10 9)=H( 10 9)
H( 9 1)=H( 9 17)=.9877E-03=H( 9 1)=H( 9 17)
H( 9 2)=H( 9 16)=.1420E-02=H( 9 2)=H( 9 16)
H( 9 3)=H( 9 15)=.2651E-02=H( 9 3)=H( 9 15)
H( 9 4)=H( 9 14)=.4493E-02=H( 9 4)=H( 9 14)
H( 9 5)=H( 9 13)=.6667E-02=H( 9 5)=H( 9 13)
H( 9 6)=H( 9 12)=.8840E-02=H( 9 6)=H( 9 12)
H( 9 7)=H( 9 11)=.1068E-01=H( 9 7)=H( 9 11)
H( 9 8)=H( 9 10)=.1191E-01=H( 9 8)=H( 9 10)
H( 9 9)=H( 9 9)=.1235E-01=H( 9 9)=H( 9 9)
```

Fig. 3.11 Impulse response of a lowpass 2-D FIR digital filter using windowing technique (Hamming window).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

desired value in passband: 1.0

desired value in stopband: 0.0

maximum allowed deviation = 0.1

As discussed a 2-D FIR filter has been designed using Hamming window. Fig.3.12 shows the impulse response of the highpass 2-D FIR digital filters. The computational time to design this filter is 4.8 sec. on SEL computer.

Example 3- Bandpass filter

Design a bandpass circularly symmetric filter, which is useful for image processing problems, of spatial filter kernal 17 X 17 with the following specifications:

passband: $0.4\pi \leq r \leq 0.6\pi$ $\quad r=(w_1^2+w_2^2)^{1/2}$

stopband: $0.0 \leq r \leq 0.2\pi$

$0.8\pi \leq r \leq \pi$

desired value in passband: 1.0

desired value in stopbands: 0.0

maximum allowed deviation = 0.1

As discussed a 2-D FIR filter has been designed using Hamming window. Fig.3.13 shows the impulse response of the bandpass 2-D FIR digital filters. The computational time to design this filter is 4.8 sec. on SEL computer.

A single program has been written to calculate the spatial filter kernel of any size using all seven windows is listed in Appendix.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

FINITE IMPULSE RESPONSE (FIR)
DIGITAL FILTER DESIGN USING
WINDOWING TECHNIQUE
HIGHPASS FILTER

FILTER KERNEL SIZE = 17 X 17

```
H(  1  1)=H(  1  17)=.0000E+00=H(  17   1)=H(  17  17)
H(  1  2)=H(  1  16)=.0000E+00=H(  17   2)=H(  17  16)
H(  1  3)=H(  1  15)=.0000E+00=H(  17   3)=H(  17  15)
H(  1  4)=H(  1  14)=.0000E+00=H(  17   4)=H(  17  14)
H(  1  5)=H(  1  13)=.0000E+00=H(  17   5)=H(  17  13)
H(  1  6)=H(  1  12)=.9877E-03=H(  17   6)=H(  17  12)
H(  1  7)=H(  1  11)=.9877E-03=H(  17   7)=H(  17  11)
H(  1  8)=H(  1  10)=.9877E-03=H(  17   8)=H(  17  10)
H(  1  9)=H(  1   9)=.9872E-03=H(  17   9)=H(  17   9)
H(  2  1)=H(  2  17)=.0000E+00=H(  16   1)=H(  16  17)
H(  2  2)=H(  2  16)=.0000E+00=H(  16   2)=H(  16  16)
H(  2  3)=H(  2  15)=.0000E+00=H(  16   3)=H(  16  15)
H(  2  4)=H(  2  14)=.9877E-03=H(  16   4)=H(  16  14)
H(  2  5)=H(  2  13)=.9877E-03=H(  16   5)=H(  16  13)
H(  2  6)=H(  2  12)=.1420E-02=H(  16   6)=H(  16  12)
H(  2  7)=H(  2  11)=.1420E-02=H(  16   7)=H(  16  11)
H(  2  8)=H(  2  10)=.1420E-02=H(  16   8)=H(  16  10)
H(  2  9)=H(  2   9)=.1420E-02=H(  16   9)=H(  16   9)
H(  3  1)=H(  3  17)=.0000E+00=H(  15   1)=H(  15  17)
H(  3  2)=H(  3  16)=.0000E+00=H(  15   2)=H(  15  16)
H(  3  3)=H(  3  15)=.9877E-03=H(  15   3)=H(  15  15)
H(  3  4)=H(  3  14)=.9877E-03=H(  15   4)=H(  15  14)
H(  3  5)=H(  3  13)=.1420E-02=H(  15   5)=H(  15  13)
H(  3  6)=H(  3  12)=.1420E-02=H(  15   6)=H(  15  12)
H(  3  7)=H(  3  11)=.2651E-02=H(  15   7)=H(  15  11)
H(  3  8)=H(  3  10)=.2651E-02=H(  15   8)=H(  15  10)
H(  3  9)=H(  3   9)=.2651E-02=H(  15   9)=H(  15   9)
H(  4  1)=H(  4  17)=.0000E+00=H(  14   1)=H(  14  17)
H(  4  2)=H(  4  16)=.9877E-03=H(  14   2)=H(  14  16)
H(  4  3)=H(  4  15)=.9877E-03=H(  14   3)=H(  14  15)
H(  4  4)=H(  4  14)=.1420E-02=H(  14   4)=H(  14  14)
H(  4  5)=H(  4  13)=.2651E-02=H(  14   5)=H(  14  13)
H(  4  6)=H(  4  12)=.2651E-02=H(  14   6)=H(  14  12)
H(  4  7)=H(  4  11)=.4493E-02=H(  14   7)=H(  14  11)
H(  4  8)=H(  4  10)=.4493E-02=H(  14   8)=H(  14  10)
H(  4  9)=H(  4   9)=.4493E-02=H(  14   9)=H(  14   9)
H(  5  1)=H(  5  17)=.0000E+00=H(  13   1)=H(  13  17)
H(  5  2)=H(  5  16)=.9877E-03=H(  13   2)=H(  13  16)
H(  5  3)=H(  5  15)=.1420E-02=H(  13   3)=H(  13  15)
H(  5  4)=H(  5  14)=.2651E-02=H(  13   4)=H(  13  14)
H(  5  5)=H(  5  13)=.2651E-02=H(  13   5)=H(  13  13)
H(  5  6)=H(  5  12)=.4493E-02=H(  13   6)=H(  13  12)
H(  5  7)=H(  5  11)=.6667E-02=H(  13   7)=H(  13  11)
H(  5  8)=H(  5  10)=.6667E-02=H(  13   8)=H(  13  10)
H(  5  9)=H(  5   9)=.6667E-02=H(  13   9)=H(  13   9)
H(  6  1)=H(  6  17)=.9877E-03=H(  12   1)=H(  12  17)
H(  6  2)=H(  6  16)=.1420E-02=H(  12   2)=H(  12  16)
H(  6  3)=H(  6  15)=.1420E-02=H(  12   3)=H(  12  15)
H(  6  4)=H(  6  14)=.2651E-02=H(  12   4)=H(  12  14)
H(  6  5)=H(  6  13)=.4493E-02=H(  12   5)=H(  12  13)
H(  6  6)=H(  6  12)=.6667E-02=H(  12   6)=H(  12  12)
H(  6  7)=H(  6  11)=.6667E-02=H(  12   7)=H(  12  11)
H(  6  8)=H(  6  10)=.8840E-02=H(  12   8)=H(  12  10)
H(  6  9)=H(  6   9)=.8840E-02=H(  12   9)=H(  12   9)
H(  7  1)=H(  7  17)=.9877E-03=H(  11   1)=H(  11  17)
H(  7  2)=H(  7  16)=.1420E-02=H(  11   2)=H(  11  16)
H(  7  3)=H(  7  15)=.2651E-02=H(  11   3)=H(  11  15)
H(  7  4)=H(  7  14)=.4493E-02=H(  11   4)=H(  11  14)
H(  7  5)=H(  7  13)=.6667E-02=H(  11   5)=H(  11  13)
H(  7  6)=H(  7  12)=.6667E-02=H(  11   6)=H(  11  12)
H(  7  7)=H(  7  11)=.8840E-02=H(  11   7)=H(  11  11)
H(  7  8)=H(  7  10)=.1068E-01=H(  11   8)=H(  11  10)
H(  7  9)=H(  7   9)=.1068E-01=H(  11   9)=H(  11   9)
H(  8  1)=H(  8  17)=.9877E-03=H(  10   1)=H(  10  17)
H(  8  2)=H(  8  16)=.1420E-02=H(  10   2)=H(  10  16)
H(  8  3)=H(  8  15)=.2651E-02=H(  10   3)=H(  10  15)
H(  8  4)=H(  8  14)=.4493E-02=H(  10   4)=H(  10  14)
H(  8  5)=H(  8  13)=.6667E-02=H(  10   5)=H(  10  13)
H(  8  6)=H(  8  12)=.8840E-02=H(  10   6)=H(  10  12)
H(  8  7)=H(  8  11)=.1068E-01=H(  10   7)=H(  10  11)
H(  8  8)=H(  8  10)=.1191E-01=H(  10   8)=H(  10  10)
H(  8  9)=H(  8   9)=.1191E-01=H(  10   9)=H(  10   9)
H(  9  1)=H(  9  17)=.9872E-03=H(   9   1)=H(   9  17)
H(  9  2)=H(  9  16)=.1420E-02=H(   9   2)=H(   9  16)
H(  9  3)=H(  9  15)=.2651E-02=H(   9   3)=H(   9  15)
H(  9  4)=H(  9  14)=.4493E-02=H(   9   4)=H(   9  14)
H(  9  5)=H(  9  13)=.6667E-02=H(   9   5)=H(   9  13)
H(  9  6)=H(  9  12)=.8840E-02=H(   9   6)=H(   9  12)
H(  9  7)=H(  9  11)=.1068E-01=H(   9   7)=H(   9  11)
H(  9  8)=H(  9  10)=.1191E-01=H(   9   8)=H(   9  10)
H(  9  9)=H(  9   9)=.9877E+00=H(   9   9)=H(   9   9)
```

Fig. 3.12 Impulse response of a highpass 2-D FIR digital filter using windowing technique (Hamming window).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

FINITE IMPULSE RESPONSE (FIR)
DIGITAL FILTER DESIGN USING
WINDOWING TECHNIQUE
BANDPASS FILTER

FILTER KERNEL SIZE = 17 X 17

```
H(  1  1)=H(  1  17)=.0000E+00=H(  17   1)=H(  17  17)
H(  1  2)=H(  1  16)=.0000E+00=H(  17   2)=H(  17  16)
H(  1  3)=H(  1  15)=.0000E+00=H(  17   3)=H(  17  15)
H(  1  4)=H(  1  14)=.0000E+00=H(  17   4)=H(  17  14)
H(  1  5)=H(  1  13)=.0000E+00=H(  17   5)=H(  17  13)
H(  1  6)=H(  1  12)=.1177E-09=H(  17   6)=H(  17  12)
H(  1  7)=H(  1  11)=.2841E-08=H(  17   7)=H(  17  11)
H(  1  8)=H(  1  10)=.3271E-08=H(  17   8)=H(  17  10)
H(  1  9)=H(  1   9)=.4932E-06=H(  17   9)=H(  17   9)
H(  2  1)=H(  2  17)=.0000E+00=H(  16   1)=H(  16  17)
H(  2  2)=H(  2  16)=.0000E+00=H(  16   2)=H(  16  16)
H(  2  3)=H(  2  15)=.0000E+00=H(  16   3)=H(  16  15)
H(  2  4)=H(  2  14)=.3769E-09=H(  16   4)=H(  16  14)
H(  2  5)=H(  2  13)=.2943E-09=H(  16   5)=H(  16  13)
H(  2  6)=H(  2  12)=.0000E+00=H(  16   6)=H(  16  12)
H(  2  7)=H(  2  11)=.4431E-08=H(  16   7)=H(  16  11)
H(  2  8)=H(  2  10)=.4291E-08=H(  16   8)=H(  16  10)
H(  2  9)=H(  2   9)=.3104E-06=H(  16   9)=H(  16   9)
H(  3  1)=H(  3  17)=.0000E+00=H(  15   1)=H(  15  17)
H(  3  2)=H(  3  16)=.0000E+00=H(  15   2)=H(  15  16)
H(  3  3)=H(  3  15)=.2538E-08=H(  15   3)=H(  15  15)
H(  3  4)=H(  3  14)=.2355E-09=H(  15   4)=H(  15  14)
H(  3  5)=H(  3  13)=.8464E-10=H(  15   5)=H(  15  13)
H(  3  6)=H(  3  12)=.4935E-09=H(  15   6)=H(  15  12)
H(  3  7)=H(  3  11)=.2078E-08=H(  15   7)=H(  15  11)
H(  3  8)=H(  3  10)=.3769E-08=H(  15   8)=H(  15  10)
H(  3  9)=H(  3   9)=.4474E-06=H(  15   9)=H(  15   9)
H(  4  1)=H(  4  17)=.0000E+00=H(  14   1)=H(  14  17)
H(  4  2)=H(  4  16)=.1208E-08=H(  14   2)=H(  14  16)
H(  4  3)=H(  4  15)=.2633E-09=H(  14   3)=H(  14  15)
H(  4  4)=H(  4  14)=.8464E-10=H(  14   4)=H(  14  14)
H(  4  5)=H(  4  13)=.7901E-09=H(  14   5)=H(  14  13)
H(  4  6)=H(  4  12)=.2375E-08=H(  14   6)=H(  14  12)
H(  4  7)=H(  4  11)=.4490E-08=H(  14   7)=H(  14  11)
H(  4  8)=H(  4  10)=.2224E-07=H(  14   8)=H(  14  10)
H(  4  9)=H(  4   9)=.3474E-06=H(  14   9)=H(  14   9)
H(  5  1)=H(  5  17)=.0000E+00=H(  13   1)=H(  13  17)
H(  5  2)=H(  5  16)=.2427E-09=H(  13   2)=H(  13  16)
H(  5  3)=H(  5  15)=.8464E-09=H(  13   3)=H(  13  15)
H(  5  4)=H(  5  14)=.7901E-09=H(  13   4)=H(  13  14)
H(  5  5)=H(  5  13)=.1611E-08=H(  13   5)=H(  13  13)
H(  5  6)=H(  5  12)=.1104E-08=H(  13   6)=H(  13  12)
H(  5  7)=H(  5  11)=.1669E-07=H(  13   7)=H(  13  11)
H(  5  8)=H(  5  10)=.2427E-07=H(  13   8)=H(  13  10)
H(  5  9)=H(  5   9)=.1410E-06=H(  13   9)=H(  13   9)
H(  6  1)=H(  6  17)=.1242E-08=H(  12   1)=H(  12  17)
H(  6  2)=H(  6  16)=.2934E-08=H(  12   2)=H(  12  16)
H(  6  3)=H(  6  15)=.3664E-08=H(  12   3)=H(  12  15)
H(  6  4)=H(  6  14)=.4997E-09=H(  12   4)=H(  12  14)
H(  6  5)=H(  6  13)=.1339E-06=H(  12   5)=H(  12  13)
H(  6  6)=H(  6  12)=.7391E-08=H(  12   6)=H(  12  12)
H(  6  7)=H(  6  11)=.2223E-07=H(  12   7)=H(  12  11)
H(  6  8)=H(  6  10)=.5278E-07=H(  12   8)=H(  12  10)
H(  6  9)=H(  6   9)=.2630E-06=H(  12   9)=H(  12   9)
H(  7  1)=H(  7  17)=.2476E-08=H(  11   1)=H(  11  17)
H(  7  2)=H(  7  16)=.1449E-08=H(  11   2)=H(  11  16)
H(  7  3)=H(  7  15)=.9481E-09=H(  11   3)=H(  11  15)
H(  7  4)=H(  7  14)=.1515E-08=H(  11   4)=H(  11  14)
H(  7  5)=H(  7  13)=.2810E-08=H(  11   5)=H(  11  13)
H(  7  6)=H(  7  12)=.2544E-08=H(  11   6)=H(  11  12)
H(  7  7)=H(  7  11)=.1419E-07=H(  11   7)=H(  11  11)
H(  7  8)=H(  7  10)=.3559E-07=H(  11   8)=H(  11  10)
H(  7  9)=H(  7   9)=.6481E-06=H(  11   9)=H(  11   9)
H(  8  1)=H(  8  17)=.3485E-08=H(  10   1)=H(  10  17)
H(  8  2)=H(  8  16)=.2710E-08=H(  10   2)=H(  10  16)
H(  8  3)=H(  8  15)=.5697E-09=H(  10   3)=H(  10  15)
H(  8  4)=H(  8  14)=.1339E-08=H(  10   4)=H(  10  14)
H(  8  5)=H(  8  13)=.1638E-08=H(  10   5)=H(  10  13)
H(  8  6)=H(  8  12)=.8496E-08=H(  10   6)=H(  10  12)
H(  8  7)=H(  8  11)=.2620E-07=H(  10   7)=H(  10  11)
H(  8  8)=H(  8  10)=.7687E-07=H(  10   8)=H(  10  10)
H(  8  9)=H(  8   9)=.1772E-05=H(  10   9)=H(  10   9)
H(  9  1)=H(  9  17)=.4698E-06=H(   9   1)=H(   9  17)
H(  9  2)=H(  9  16)=.2868E-06=H(   9   2)=H(   9  16)
H(  9  3)=H(  9  15)=.4418E-06=H(   9   3)=H(   9  15)
H(  9  4)=H(  9  14)=.3281E-06=H(   9   4)=H(   9  14)
H(  9  5)=H(  9  13)=.1551E-06=H(   9   5)=H(   9  13)
H(  9  6)=H(  9  12)=.2462E-06=H(   9   6)=H(   9  12)
H(  9  7)=H(  9  11)=.6108E-06=H(   9   7)=H(   9  11)
H(  9  8)=H(  9  10)=.8200E-06=H(   9   8)=H(   9  10)
H(  9  9)=H(  9   9)=.1000E+01=H(   9   9)=H(   9   9)
```

Fig. 3.13 Impulse response of a bandpass 2-D FIR digital filter using windowing technique (Hamming window).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter IV

APPLICATION OF THE 2-D FIR FILTER DESIGN USING NTT CONVOLVER

4.1 WORKING PRINCIPLE OF NTT CONVOLVER

The design of the convolutional filter is based on the Number Theoretic techniques and is implemented via 2-D radix-2 NTT algorithm. The filtering operation is performed by taking the NTT of the image, multiplying this by the NTT of the filter impulse response and then taking the inverse of the product. The convolutional filter hardware is organized to compute the circular convolution of an 128 X 128 image with 128 X 128 spatial filter kernel.

As shown in Fig.2.1 the NTT convolver forms a part of the image processing system. Since the main purpose of the convolver is to process the image at very high speed, it is necessary to interface this device to the CPU through a high speed data interface (HSD).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.2 HIGH SPEED DATA INTERFACE

The high speed data interface (HSD) is an optional feature for the SYSTEMS 32 SERIES computer. It provides a full 32-bit parallel interface to a customer designed device (NTT convolver) at rates upto 834 K transfers per second.

The handler design is based on the notion that the HSD hardware acts as a controller : it performs the handshaking with the CPU and performs all SELBUS operations needed to fetch and store either the data or status relating to the required operations. Therefore, references to the HSD simply imply controller functions which are generic to I/O operations in general. The device attached to the HSD is a user addition. The general assusmption about the device is that it can source and/or synchronize data with the possibility of presenting device specific status on request.

Some important points about HSD :

1. High speed data transfers upto 834Kwords per second. Each word is 32 bits. So effective data transfer rates of 834 * 4Kbytes/sec.
2. Upto 64 K transfers per block.
3. Simple handshake protocol between HSD and customer designed equipment (NTT convolver).
4. External mode capability which allows the customer device to provide both memory address and data for each transfer.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

5. Automatic status posting.

6. Intercomputer link capability.

4.3 HSD AND CONVOLUTIONAL FILTER INTERFACE

Fig.4.1 shows the block diagram of the interface between the High Speed Data interface (HSD) and the NTT convolution filter. This interface logic is controlled by the signals from the HSD. The filter function register within the interface can be loaded from the HSD to specify the next filtering function such as load image data, load NTT of the filter coefficient, send filtered image, send camera data etc. Once the filter function is specified the data between the HSD and the NTT convolver is transferred asynchronously at the rate of 1.2 us per 32-bit word.

The NTT stage/butterfly counter has been designed to serve dual purpose. During the transfer of data between the convolver and the HSD it is used to count the number of pixels and during the convolution operation it serves as a stage and butterfly counter. The counter control generates the necessary control signals for filling and flushing the butterfly pipeline so that a proper butterfly count can be obtained at the start of an NTT stage.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

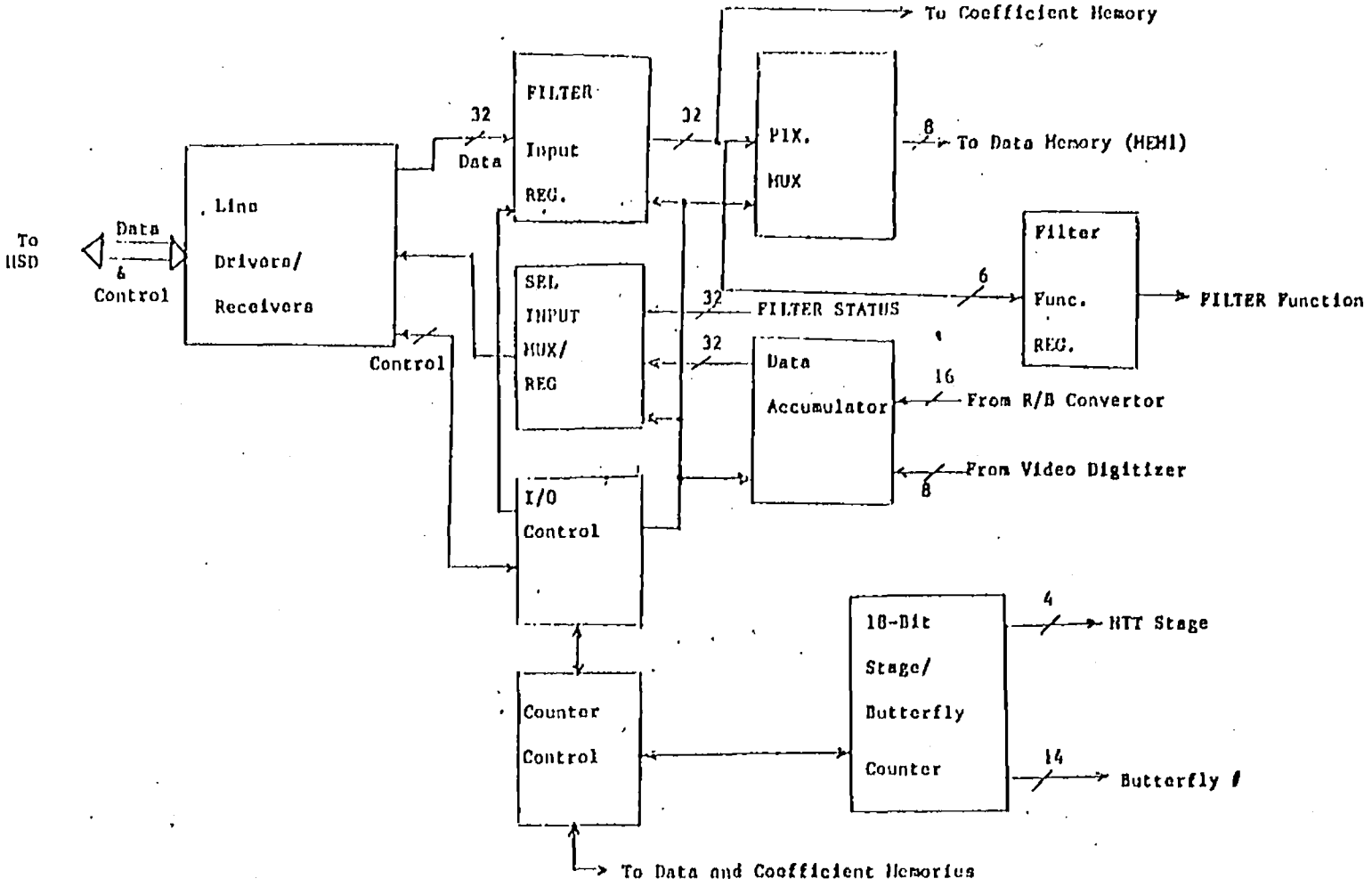


Fig. 4.1 Block diagram of Image Processing System and Convolution Filter Interface.

4.4 DATA TRANSFER BETWEEN SEL 32/27 AND NTT CONVOLVER

Before going into the procedure of data transfer between the convolver and the SEL 32/27 let us see all the possible transfers that can occur

1. Transfer of NTT of filter coefficients to the convolver.
2. Transfer of image data to be convolved.
3. Transfer of command to filter such as clear main counter, start filter, read status etc.
4. Transfer of filtered image from the convolver to SEL.

We shall now see how these various types of transfers are performed. All the programs have been written in assembly language to improve the computational time.

4.4.1 Transfer of NTT of filter coefficients

The assembly language subroutine used for this purpose is LDCOF. This subroutine can be called from a FORTRAN program as

CALL LDCOF(IDAT,IER1,IER2,IOCH)

where

IDAT ..... Data array containing NTT of filter coefficients to be sent to the convolver.

IER1 ..... Status of handler posted in the File Control Board (FCB).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

IER2 ..... Status of device as posted by the handler in Input/Output Command List (IOCL).

IOCM ..... Parameter to indicate that the I/O operation is complete.

The IOCL used for this purpose is given below.

```
IOCL    GEN 8/X'A2',8/X'30',16/0
        GEN 32/W(DATA)
        DATAD 0
        GEN 8/X'02',8/X'20',16/X'4000'
        GEN 32/0
        DATAD 0
        GEN/X'A8',8/X'00',16/00
        GEN/32/W(DATA)
        DATAD 0
```

where HSD command X'A2' means Input transfer, Device status request and command chain,

User Device Dependent (UDD) command X'30' means Clear main counter of filter,

GEN 32/W(DATA) is to generate dummy data address, HSD command X'02' means Output transfer and command chain, UDD command X'20' means Load coefficients into filter, 16/X'4000' means the transfer count is 16 K words, HSD command X'A8' means Input transfer, Device status request and interrupt after completed processing Input/Output Control Block (IOCB) and UDD command X'00' means NOP

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.4.2 Transfer of 128 X 128 image to the NTT convolver

The assembly language subroutine used for this purpose is LDIMG and can be called from a FORTRAN main program as

CALL LDIMG(IDAT,IER1,IER2,IOCM) The parameters given are similar to that of LDCOF. The IOCL for this purpose is given below.

```
IOCL  GEN 8/X'A2',8/X'30',16/0
      GEN 32/W(DATA)
      DATAD 0
      GEN 8/X'02',8/X'10',16/X'1000'
      GEN 32/0
      DATAD 0
      GEN 8/X'A8',8/X'00',16/0
      GEN 32/W(DATA)
      DATAD 0
```

where all the HSD and UDD commands are same as for LDCOF case. the only two differences are UDD command X'10' which means Load image and X'1000' indicates that the transfer count is 4 K words.

4.4.3 Transfer of filtered image from convolver to SEL

The assembly language subroutine used for this purpose is STFIMG, and can be called from a FORTRAN main program as

CALL STFIMG(IDAT,IER1,IER2,IOCM)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where all the parameters mean the same as for the two previous cases. The IOCL used in this subroutine is

```
IOCL  GEN 8/X'A2',8/X'30',16/0
      GEN 32/W(DATA)
      DATAD 0
      GEN 8/X'82',8/X'10',16/X'2000'
      GEN 32/0
      DATAD 0
      GEN 8/X'A8',8/X'00',16/0
      GEN 32/W(DATA)
      DATAD 0
```

Here again the HSD and UDD commands are the same as for the previous two cases. The only differences are the HSD command X'82' which means Read data and command chain, the UDD command X'10' now means Read filtered image back from convolver to SEL and X'2000' indicates that the transfer count is 8 K words. (It should be noted that the filtered data is 16 bits whereas the input data is only 8 bits. Hence the transfer count is doubled)

4.4.4 Transfer of 256 X 256 image from SEL to convolver

To transfer an image of size 256 X 256 we segment the image into nine sections each of size 128 X 128 or less. Then the section is transferred into a dummy 128 X 128 array which is then transferred by the procedure explained above.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The transfer to the dummy array is made faster by means of an assembly language program DMAT.

4.4.5 Transfer of commands and status

This is made possible by an assembly language subroutine CLSAT which can be used for clearing the main counter of NTT convolver, for starting the convolver and for reading the status of the convolver.

4.5 APPLICATION OF NTT CONVOLVER IN IMAGE PROCESSING

Let us consider the various steps involved in using the convolver :

1. Scaling filter coefficients:- The NTT convolver has been designed to give a 16 bit output. Hence the output data can be scaled such that the output can fully utilize this range. However, the image data can be assumed to have mean value which is a constant. Hence the filter coefficients can be scaled such that the output is limited to 16 bits.
2. NTT of scaled filter coefficients:- Since it is intended to use the filter to pre-process images in a real time environment it can be assumed that the same filter can be convolved with different images. Hence the convolver is designed to store the NTT of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the scaled filter coefficients rather than computing it each time by means of hardware. Hence it is necessary to compute the NTT of filter coefficients by means of software.

3. Transfer the NTT of the scaled filter coefficients to the convolver as mentioned in section 4.4.1.
4. Transfer image data to NTT convolver as mentioned in section 4.4.2.
5. Perform convolution using NTT convolver.
6. Transfer filtered image from NTT convolver to SEL 32/27 as explained in section 4.4.3.

4.6 PSEUDO-COLOUR IMAGE PROCESSING

A recent and potentially powerful area of digital image processing is the use of pseudo colour for image display and enhancement. The motivation for using colour in image processing is provided by the fact that the human eye can discern thousands of colour shades and intensities. This is in sharp contrast with the eye's relatively poor performance with gray levels. This type of processing is important in detecting a small fault in a manufactured part or growth of cancer in a tissue. The objective in this technique is to assign a colour to each pixel based, for example, on its intensity.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.7 FILTERING APPROACH FOR IMAGE PROCESSING PROBLEMS

Fig.4.2 shows a filtering approach scheme for image processing problems that is based on the high speed NTT convolutional filter. The main idea depicted here is that the Number Theoretic Transform of an image is modified independently by the filter functions to produce the filtered images that can be fed into the R, G and B inputs of the AYDIN colour monitor.

In image processing, the filters required for image smoothing applications have the magnitude characteristics that are generally lowpass in nature and linear phase characteristics over most of the frequency domain. Similarly the filters required for image sharpening applications have magnitude characteristics that are generally highpass in nature and linear phase characteristics over most of the frequency domain.

The image enhancement effect can be achieved by bandpass filtering the images. As indicated earlier in chapter III, the suitable filters for all these problems can be obtained using the transformation of 1-D digital filters and windowing techiques.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

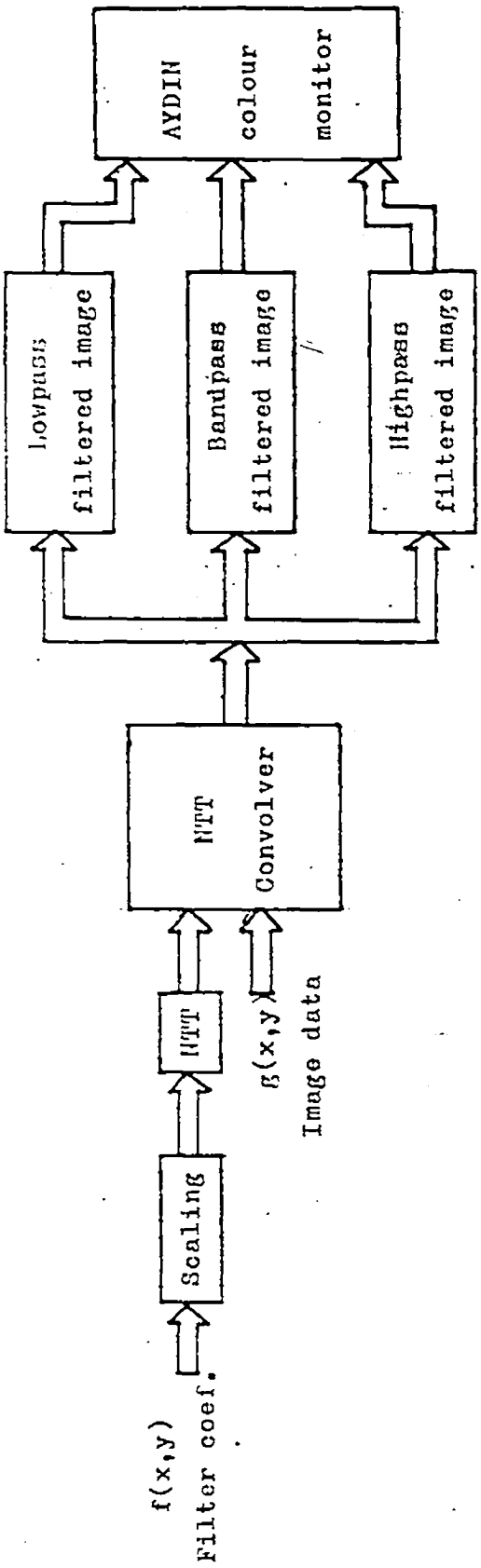


Fig. 4.2 A filtering approach scheme for image processing problems based on NTT convolver.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

4.8 FILTER DESIGN AS APPLIED IN THE SYSTEM

As mentioned before these filter design techniques are to be used in the image processing system as a pre-processing step. Here the procedure has been simplified such that it can be used even by the untrained personnel. The following menu pads can be used for this purpose.

Fig.4.3 shows the menu pad of the general system. The filtering operation can then be started by selecting the "Filter Image" in the menu pad. This step invokes the filtering algorithm using the convolver. Now it is necessary to select the type of the technique that is to be employed for filtering the image. This selection can be done by touching either "Transformation" or "Window" as shown in Fig.4.4. Immediately the next step to be followed will appear on the touch screen. In this case the McClellan transformation technique Fig.4.6 will appear which indicates the type of filtering operations to be done. In the case of the windowing technique first the type of window to be selected will appear as shown in Fig.4.5, then it will be followed by the Fig.4.6.

4-8-1 Image smoothing

One way of achieving a smoothing image is to pass the image through a lowpass filter and thus emphasize the low frequency components of the image. Consider the image of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

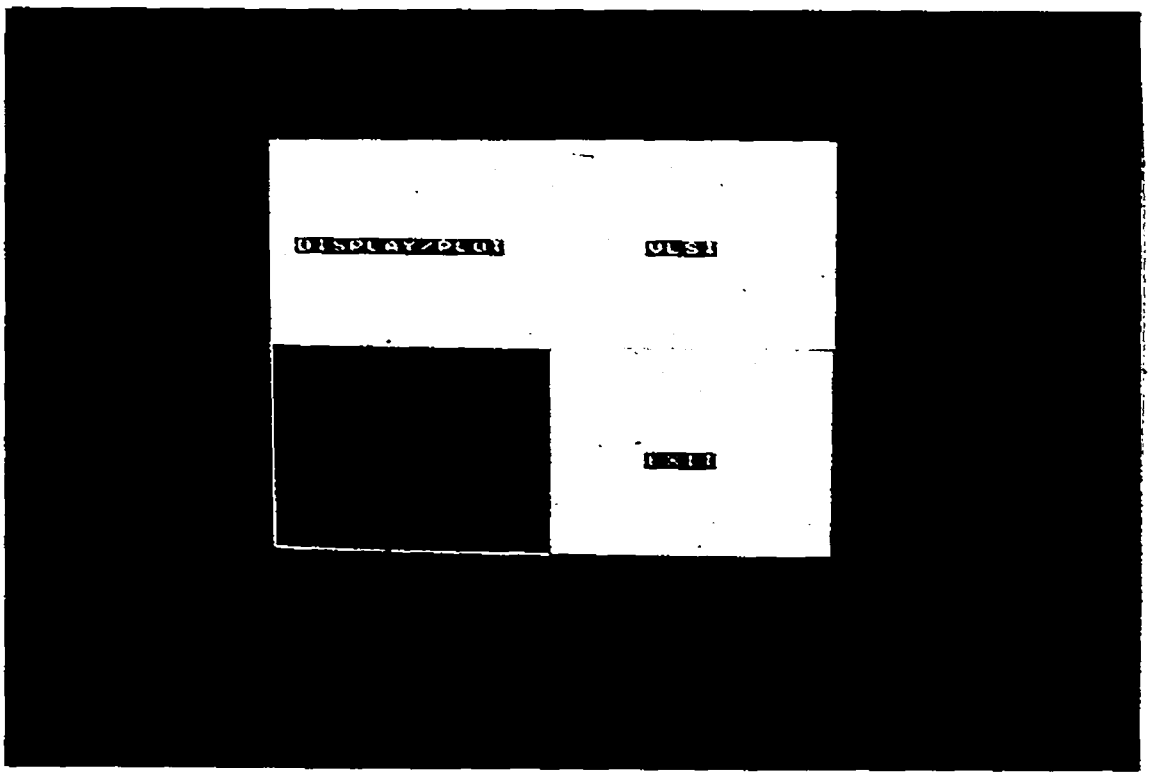


Fig. 4.3 Menu pad of the general image processing system.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

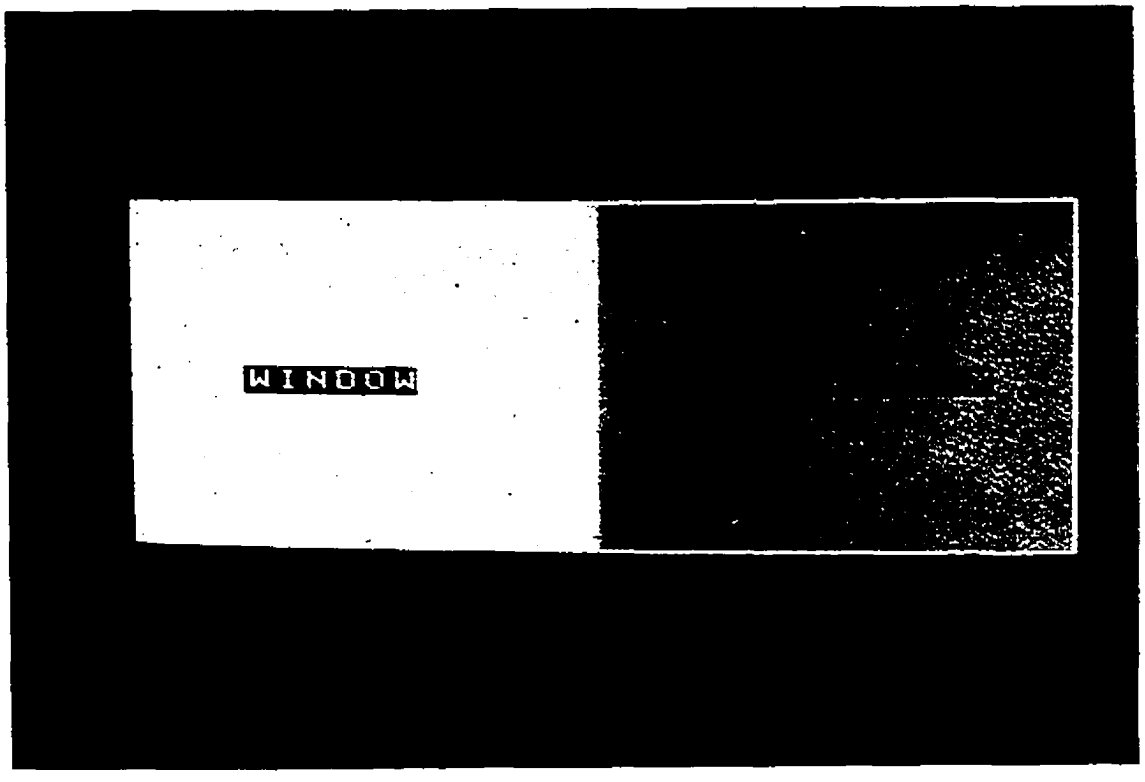


Fig. 4.4 Menu for selection of the design technique.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

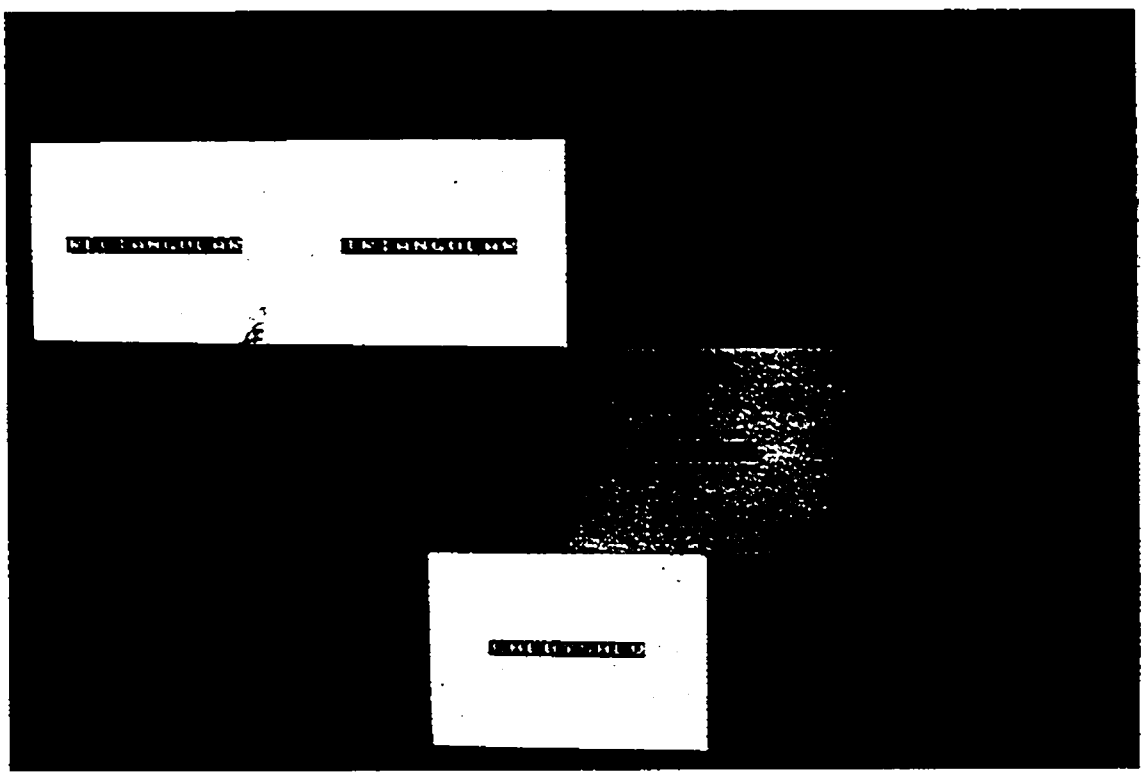


Fig. 4.5 Menu for selection of the type of window to be used.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

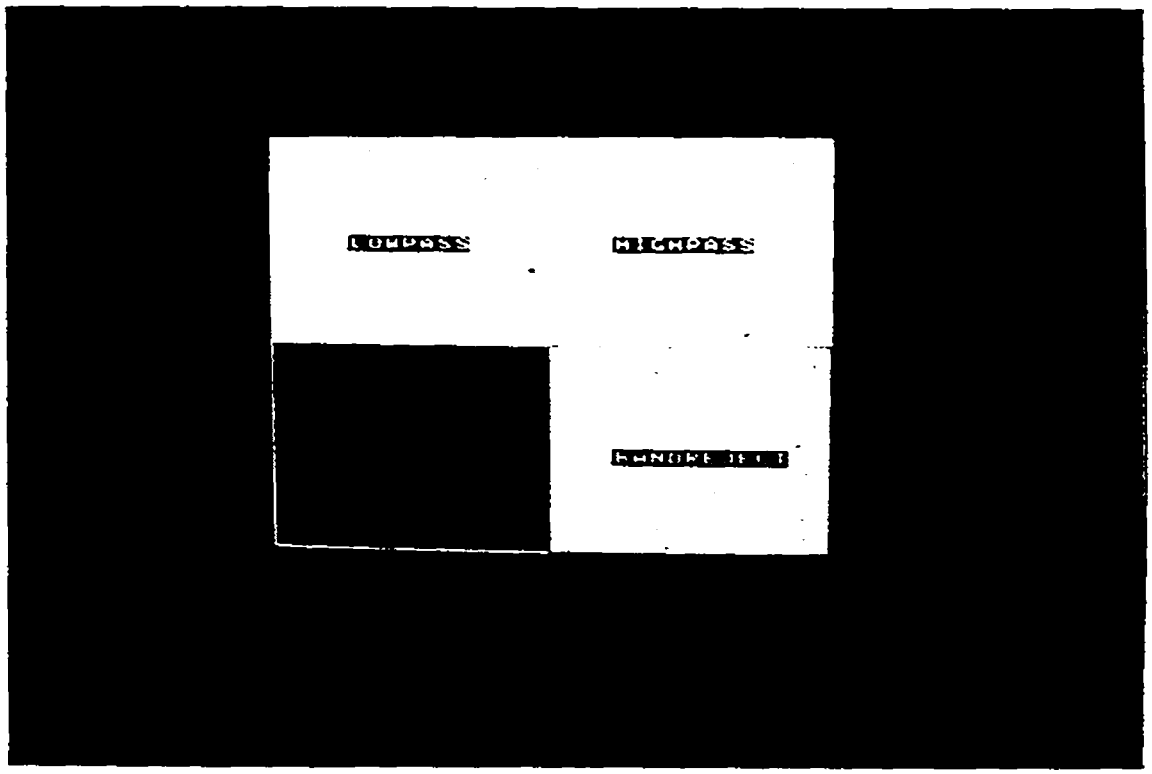


Fig. 4.6 Menu for selection of the type of filtering operation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Fig.4.7 which is the original image of the piston head. This image can be smoothened by convolving with lowpass filters which attenuates the high frequency components as shown in Fig.4.8 and 4.9. Fig.4.8 is the smoothened image using transformation of 1-D digital filter technique whereas Fig.4.9 is the smoothened image using windowing technique.

4.8.2 Image sharpening

Often it is required to make an image sharper. One way of achieving this objective is to pass the image through a highpass filter and thus emphasize the high frequency components of the image. Consider the image of Fig.4.7 which is the original image of the piston head. This image can be sharpened by convolving with highpass filters which attenuates the low frequency components as shown in Fig.4.10 and 4.11. Fig.4.10 is the sharpened image using transformation of 1-D digital filter technique whereas Fig.4.11 is the sharpened image using windowing technique.

4.8.3 Image enhancement

Some cases it is necessary to obtain an enchanced image. The principal objective of enhancement technique is to process a given image so that the result is more suitable than the original image. In order to achieve this, consider the original piston head image of Fig.4.7. This image can be

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

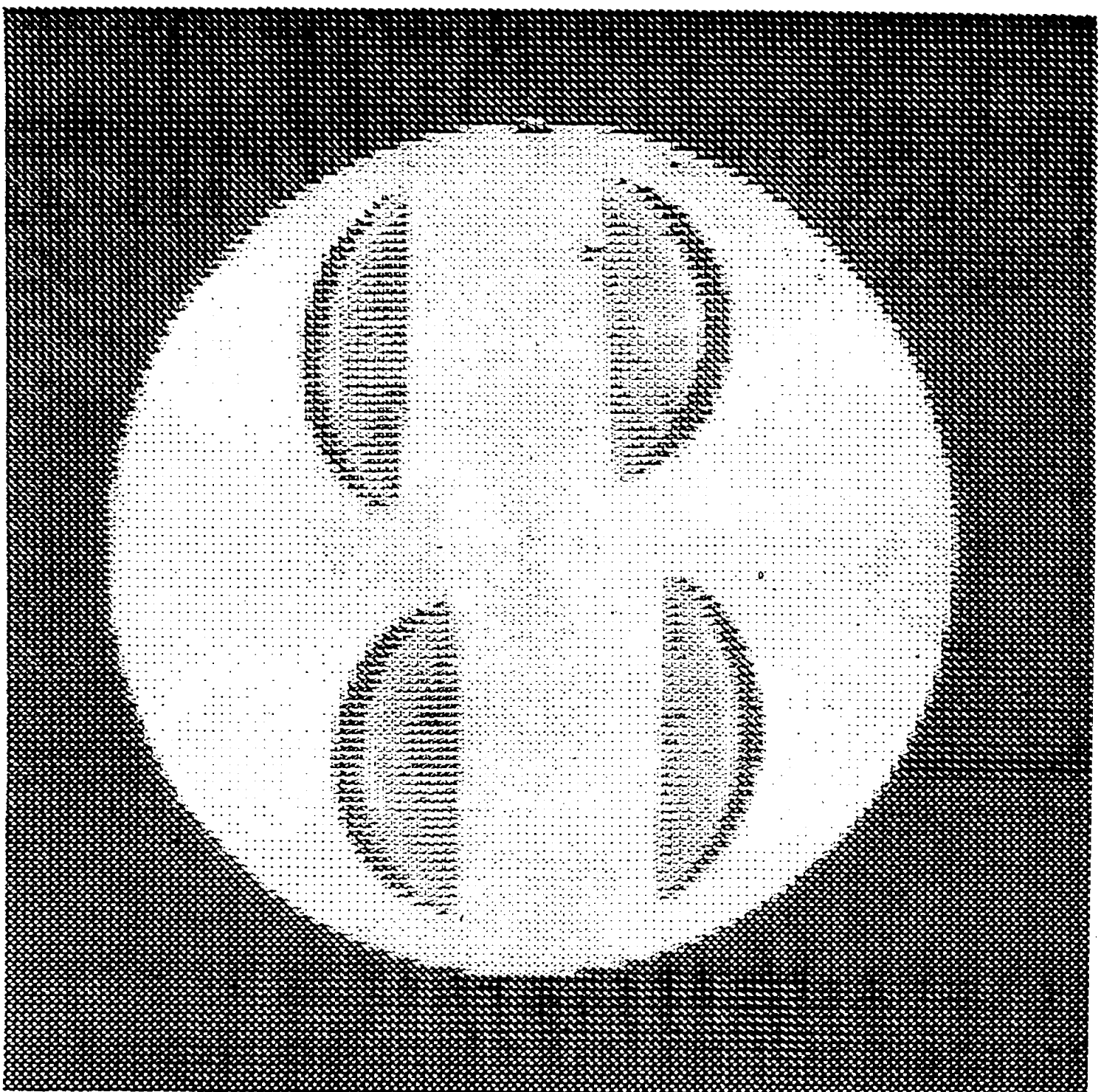


Fig.4.7 Original image of the Piston head.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

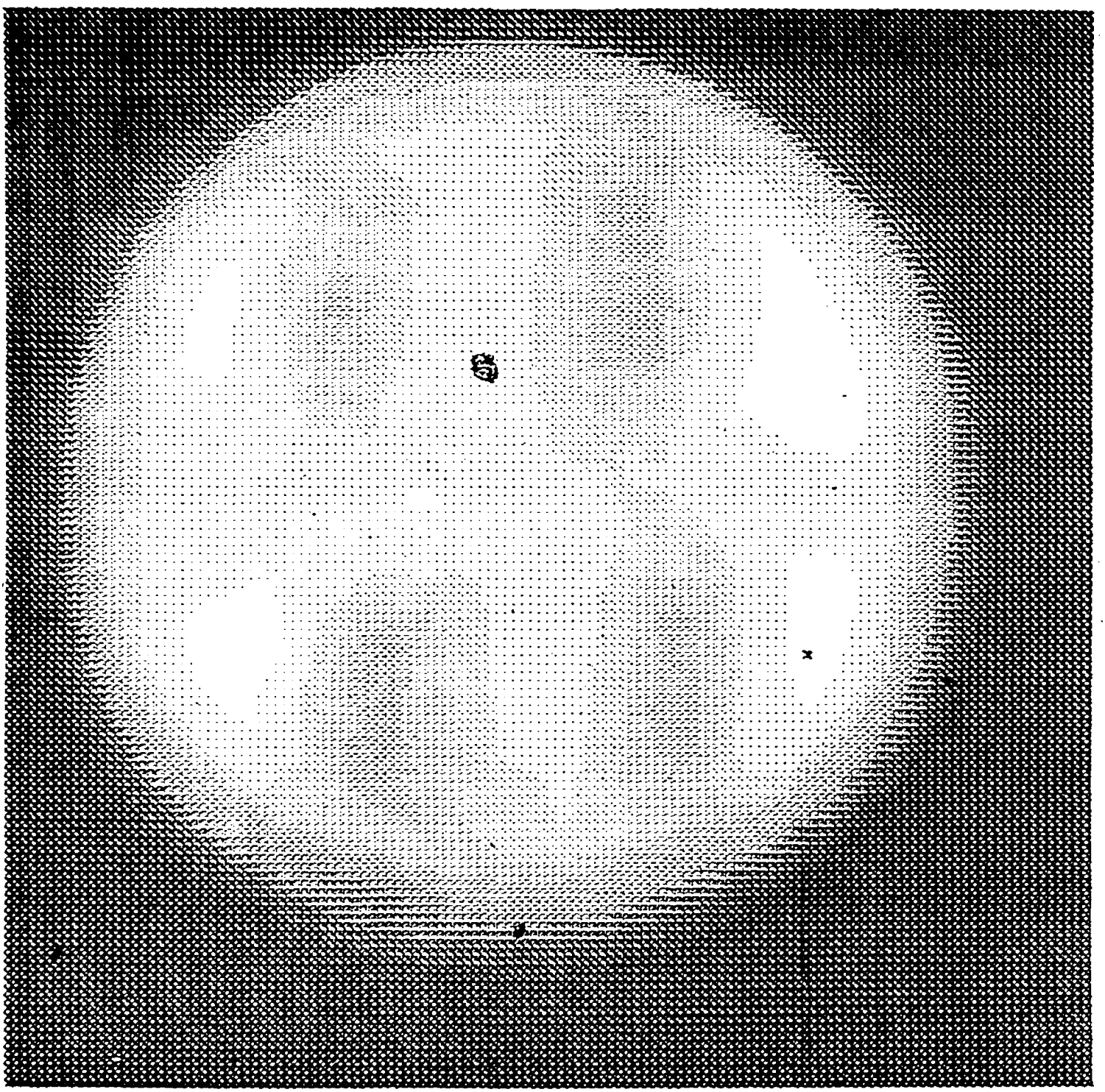


Fig.4.8 Lowpass filtered image using

McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

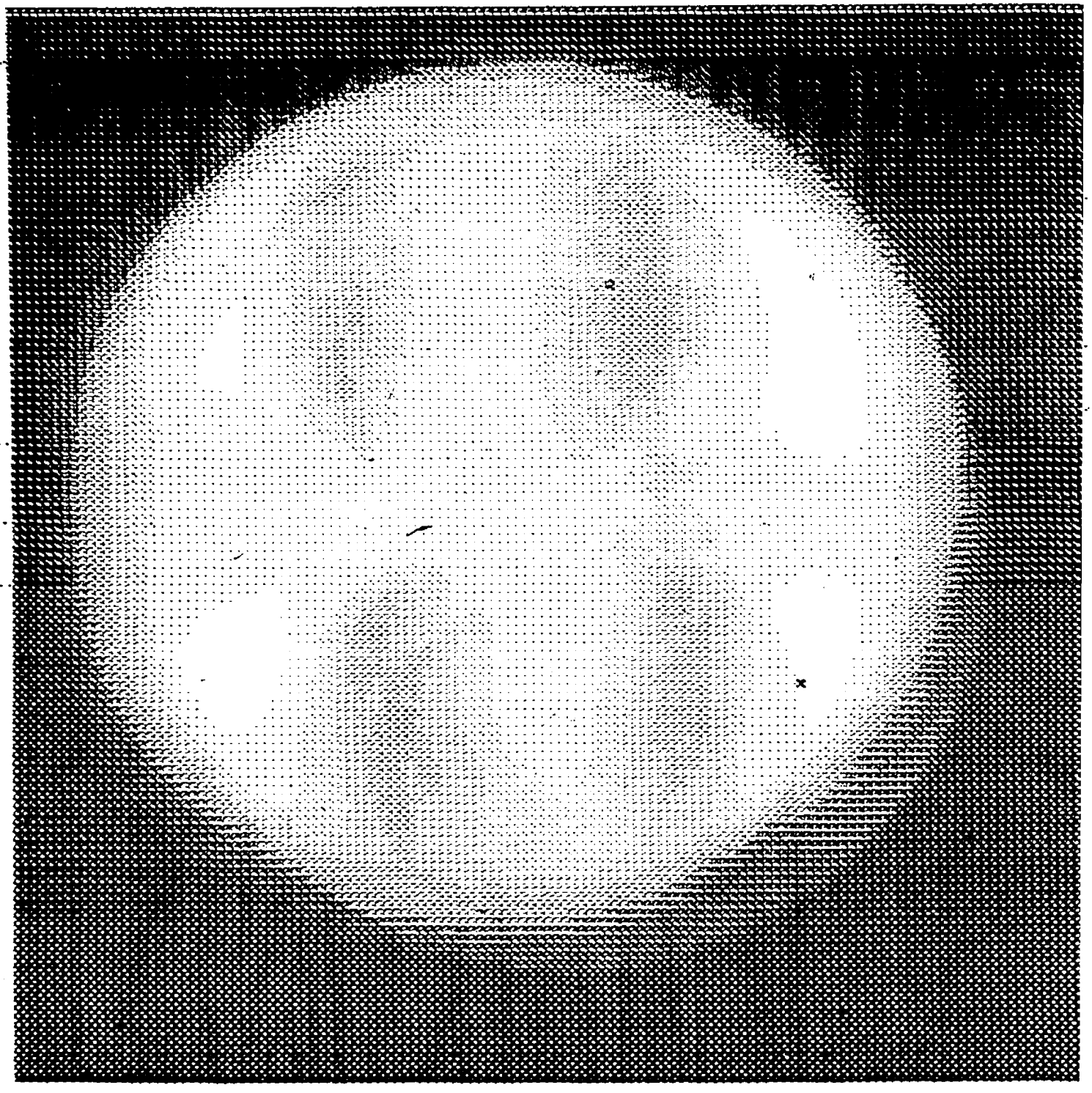


Fig.4.9 Lowpass filtered image using

Windowing technique.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

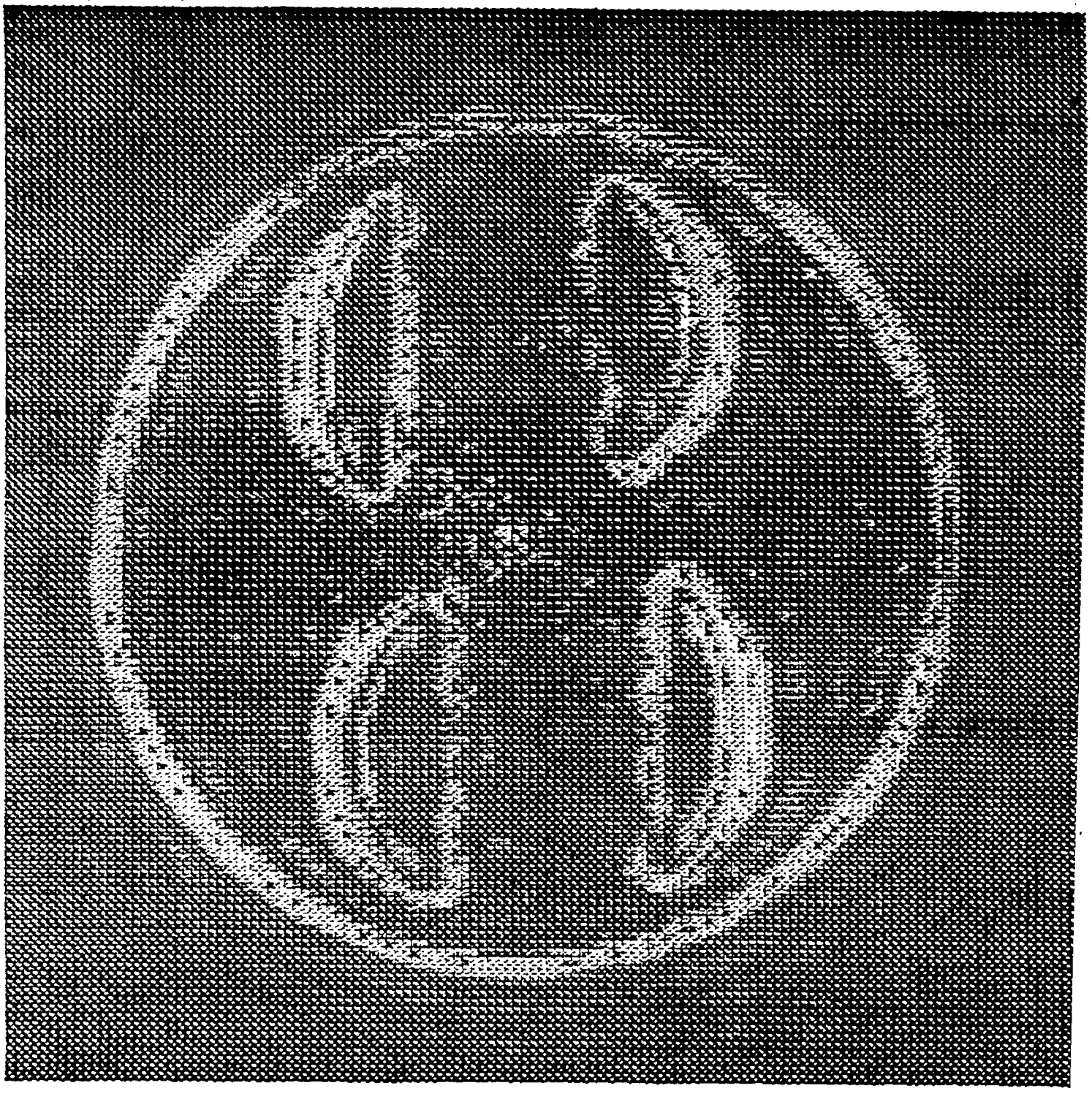


Fig.4.10 Highpass filtered image using

McClellan transformation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

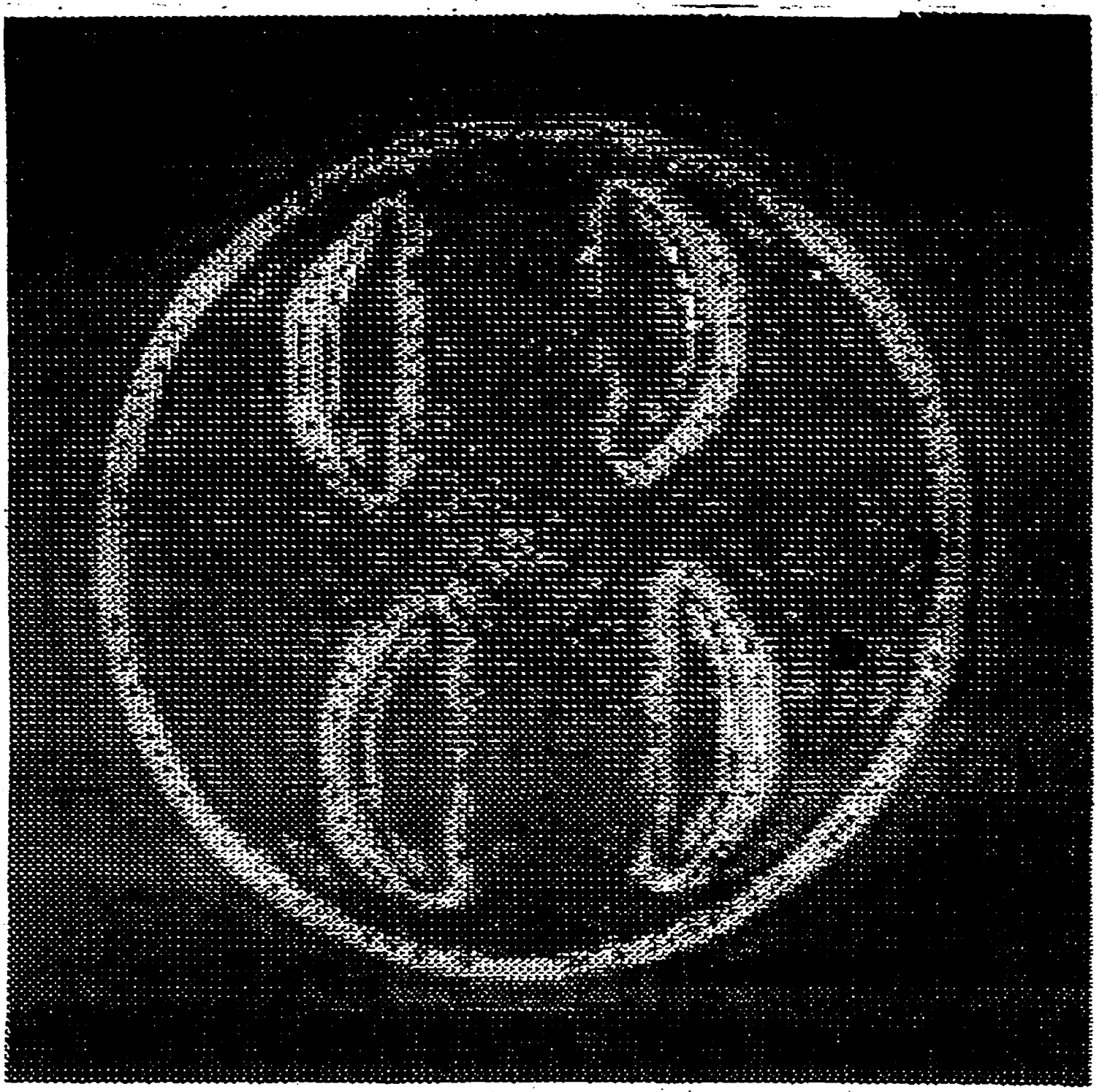


Fig.4.11 Highpass filtered image using

Windowing technique

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

enchanced by bandpass filtering the image. Fig.4.12 shows the enchanced image obtained by using McClellan transformation technique whereas Fig.4.13 shows the enhanced image obtained by windowing technique.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

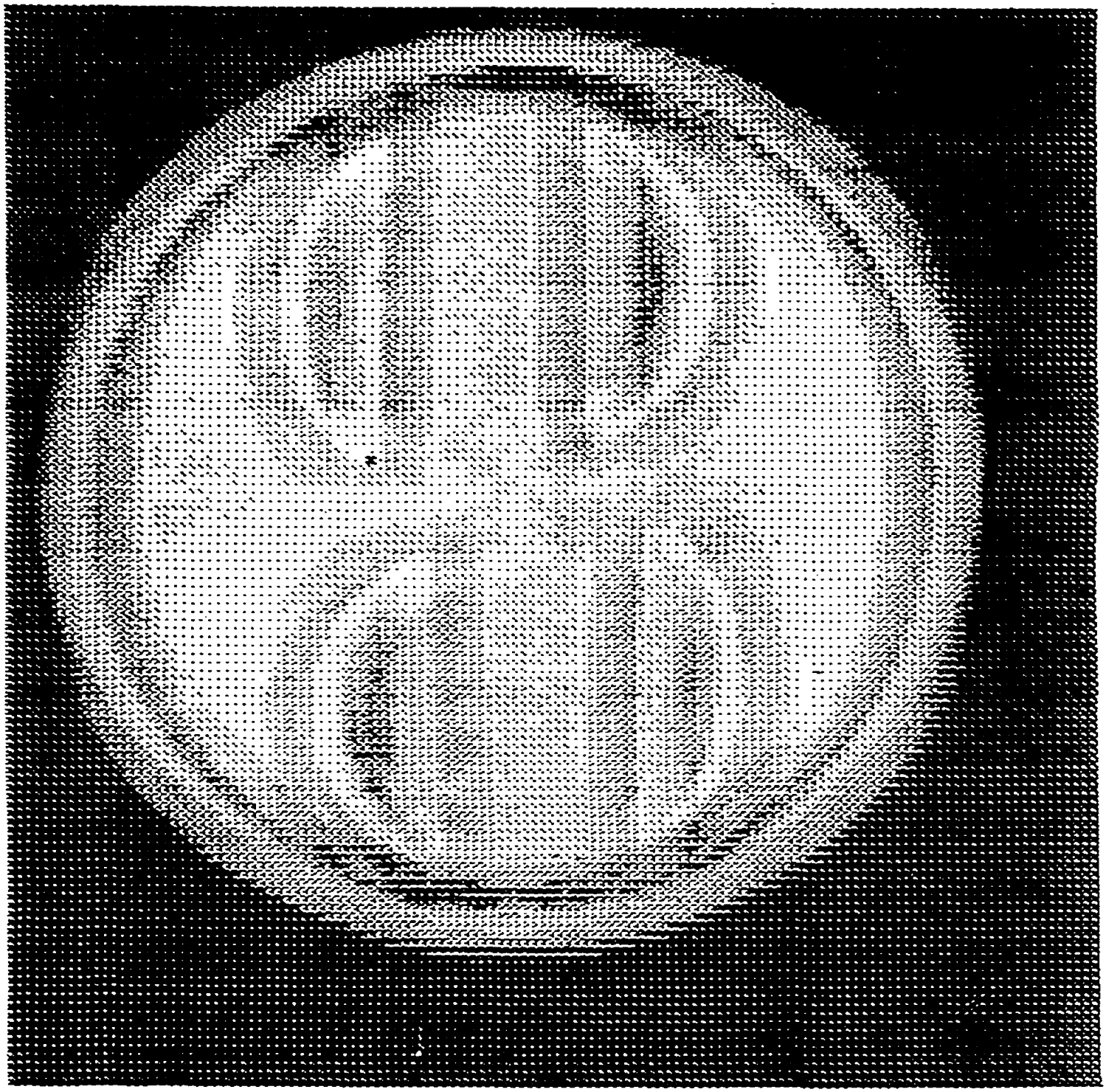


Fig.4.12 Enhanced image using McClellan transformation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

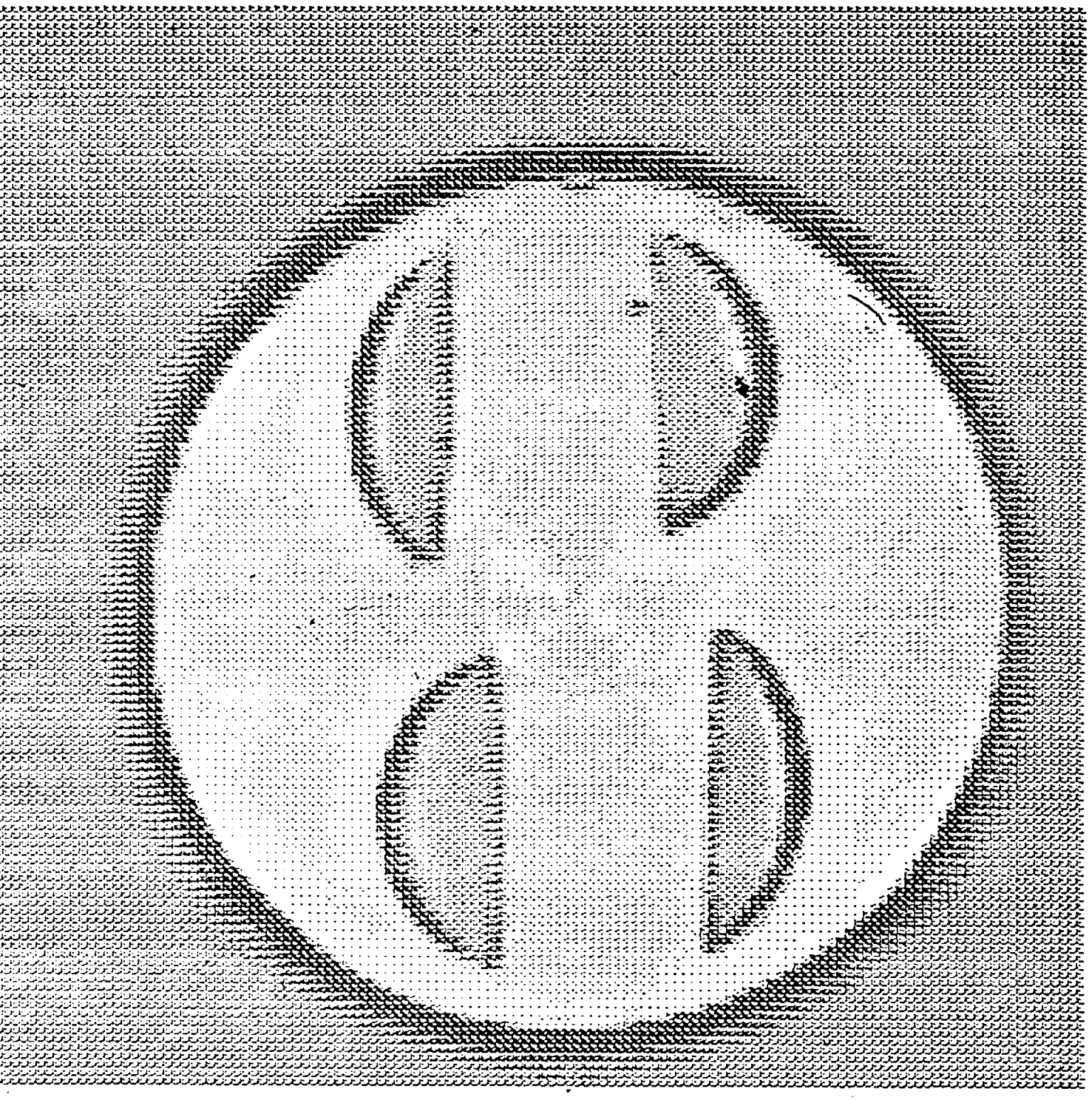


Fig.4.13 Enhanced image using Windowing technique.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter V

SUMMARY AND CONCLUSIONS

5.1 SUMMARY OF THE WORKDONE

The problem considered in this work is: " Image filtering using the NTT convolver ". An outline of this work is shown in Fig.4.2. In this direction the following work has been accomplished.

1. Investigation has been made into various two-dimensional FIR filter design techniques.
2. Two-dimensional frequency response and impulse response which can be used with the NTT convolver have been obtained using the McClellan transformation. Similarly the 2-D impulse response has been obtained using the Windowing technique.
3. Algorithms have been developed for scaling the filter coefficients and computing their Number Theoretic Transform.
4. Algorithms for transferring the filter coefficients have been generalized.
5. Algorithms for transferring the image data to and from the convolver have been tested.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

6. The above algorithms have been tested on various images.
7. Image smoothing, image sharpening and image enhancement effects have been investigated by lowpass, highpass and bandpass filtering the image.

5.2 CONCLUSION

The work described in this thesis demonstrates the successful implementation of 2-D FIR digital filters in image processing applications. Specifically these filters can be used in pre-processing to smooth, sharpen or enhance the images which will greatly aid in automatic machine identification of features in the image.

However, care should be taken in the design of these filters to select appropriate cut-off frequencies, scaling of filter coefficients to get best results. Since the viewer is the best judge for the performance of the filter, the display system and its hardware plays very important role in illustrating the results. It is suggested that a smooth variation in colours over the dynamic range of grey levels provides a better viewer perspective.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Appendix A

LIST OF PROGRAMS

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB MAIN VARMA,RAVI SLOF=DUMMY
$NOTE ***************************************************
$NOTE FILE NAME :- MAIN
$NOTE THIS IS THE MAIN PROGRAM FOR DEMONSTRATING
$NOTE THE FILTERING OPERATION USING THE TOUCH SCREEN
$NOTE ***************************************************
$NOTE LOOK - LOAD MODULE TO GENERATE THE COLOUR LOOK-UP TABLE
$NOTE        OBTAINED FROM THE FILE LOOKCOL
$NOTE PROG1.SV - LOAD MODULE TO DISPLAY TWO TYPES OF
$NOTE           DESIGN TECHNIQUES ON THE TOUCH SCREEN
$NOTE           WHICH IS OBTAINED FROM THE FILE PROG1
$NOTE PROG2.SV - LOAD MODULE TO DISPLAY SEVEN TYPES OF
$NOTE           WINDOWS ON THE TOUCH SCREEN WHICH IS
$NOTE           FROM THE FILE PROG2
$NOTE PROG3.SV - LOAD MODULE TO DISPLAY FILTER TYPE ON
$NOTE          THE TOUCH SCREEN WHICH IS OBTAINED FROM
$NOTE          THE FILE PROG3
$NOTE ---------------------------------------------------
$NOTE ONE.SV - LOAD MODULE TO DESIGN THE 1-D FIR DIGITAL
$NOTE         FILTER  WHICH IS OBTAINED FROM THE REFERENCE 2.
$NOTE TWO.SV - LOAD MODULE TO DESIGN 2-D FIR FILTER AND
$NOTE          SCALING THE FILTER COEFFICIENTS WHICH IS OBTAINED
$NOTE          FROM THE FILE TWO
$NOTE THREE.SV - LOAD MODULE TO DESIGN 2-D FIR FILTER USING
$NOTE           WINDOWING TECHNIQUE WHICH IS OBTAINED FROM
$NOTE          THE FILE THREE
$NOTE SCALESV - LOAD MODULE TO SCALE THE FILTER COEFFICIENTS
$NOTE            WHICH IS OBTAINED FROM THE FILE SCALE
$NOTE---------------------------------------------------
$NOTE FILRTDSV.- LOAD MODULE TO FILTER THE IMAGE USING THE
$NOTE           SOFTWARE SIMULATION OF THE NTT CONVOLVER WHICH
$NOTE           IS IS OBTAINED FROM THE FILE FILRTD
$NOTE                          OR
$NOTE CONFILSV - LOAD MODULE TO FILTER THE IMAGE USING THE
$NOTE             HARDWARE NTT CONVOLVER WHICH IS OBTAINED
$NOTE             THE FILE CONFIL
$NOTE---------------------------------------------------
$PAGE 0
$RUN LOOK
$RUN PROG1.SV
$IFT FILE WINDFIL WINDOW
$IFT FILE TRANFIL TRANSFORM
$NOTE *** ERROR *** NO FILE HAS BEEN OPENED
$GOTO END
%WINDOW
$RUN PROG2.SV
$RUN PROG3.SV
$RUN THREE.SV
$RUN SCALESV
$RUN FILRTDSV
$DELETE WINDFIL
$GOTO END
%TRANSFORM
$RUN PROG3.SV
$RUN ONE.SV
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$RUN TWO.SV
$RUN FILRTDSV
$DELETE TRANFIL
%END
$RUN @SYSTEM(SYSTEM)DISPLCOL
A1 1=DUMMYIMG
$RUN DISP128
$PAGE 23
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB PROG1 VARMA,RAVI SLOF=DUMMY
$OPTION   2 3 4 5 17
$EXECUTE FORTRAN
C*****************************************************************************
C         FILENAME :- PROG1
C         TO DISPLAY TWO TYPES OF DESIGN TECHNIQUES ON THE TOUCH SCREEN
C*****************************************************************************
          IMPLICIT INTEGER*2 (A-Z)
          INTEGER*4 FCB(16),DEVICE,TIME,SNSBUF
          DIMENSION X(5),Y(5),IX(5),IY(5)
          DIMENSION BANNER(20)
          INTEGER*1 ICHAR(8),IDCHAR(20)
          EXTERNAL IAND,CHAR
          COMMON /FCBS/ FCB,SNSBUF
          COMMON /MAXBLK/ RITMAX,BOTMAX,CHNMAX
          COMMON /VWPBLK/ RITVIW(15),LFTVIW(15),TOPVIW(15),BOTVIW(15)
          COMMON /CONLIM/ RITCON(15),LFTCON(15),TOPCON(15),BOTCON(15)
          COMMON /PARSET/ TABNUM
          DATA DEVICE/X'7EA0'/
          RITVIW(1)=BOTVIW(1)=511
          TOPVIW(1)=LFTVIW(1)=0
          RITMAX=511
          BOTMAX=511
          CHNMAX=16
          TABNUM=1
          CALL IOINIT(DEVICE,ERRSTAT,*100)
          CALL SELVIW(1)
          CALL PUP
          CALL VCCOFF(1)
          CALL MCWOUT(66)
2         READ(5,*)COL
          IF(COL.EQ.0) GO TO 10
          READ(5,*)(X(I),Y(I),I=1,5)
          CALL CURPOS(X(1),Y(1))
          DO 1 I=2,5
          CALL TTT(X(I),Y(I))
1         CONTINUE
          XX=X(1)+5.  ;  YY=Y(1)+5
          CALL CURPOS(XX,YY)
          CALL PIXFOR(COL)
          CALL FIL
          READ(5,*)LENGTH,XX,YY
          READ(5,22)(IDCHAR(I),I=1,LENGTH)
22        FORMAT(20A1)
          CALL CURPOS(XX,YY)
          DO 23 I=1,LENGTH
          BANNER(I)=IDCHAR(I)
23        CONTINUE
          CALL RCHAR(LENGTH,BANNER)
          GO TO 2
10        READ(1,3)(ICHAR(I),I=1,8)
3         FORMAT(8A1)
          DO 4 I=2,7
          ICHAR(I)=ICHAR(I)-X'30'
4         CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
          LCX=ICHAR(2)*100+ICHAR(3)*10+ICHAR(4)
          LCY=ICHAR(5)*100+ICHAR(6)*10+ICHAR(7)
          LCX=LCX*2 ; LCY=LCY*2
          IF(LCX.LT.256) THEN
          OPEN(UNIT=2,FILE='WINDFIL',BLOCKED=.FALSE.,FORM='UNFORMATTED
          CLOSE(UNIT=2)
          ELSE
          OPEN(UNIT=2,FILE='TRANFIL',BLOCKED=.FALSE.,FORM='UNFORMATTED
          CLOSE(UNIT=2)
          ENDIF
          CALL VCCON(1)
99        STOP
100       TYPE *,'ERROR OCCURRED IN INITIALISING'
          STOP
          END
$ASSIGN LIB TO @SYSTEM(SYSTEM)AYDLIB BLOC=N
$ASSIGN DIR TO @SYSTEM(SYSTEM)AYDDIR BLOC=N
$EXECUTE CATALOG
A3 1=CATEC7
A1 5=PROG1DAT
BUILD PROG1.SU,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB PROG1 VARMA,RAVI SLOF=DUMMY
$OPTION   2 3 4 5 17
$EXECUTE FORTRAN
C*****************************************************************
C         FILENAME :- PROG2
C         TO DISPLAY SEVEN TYPES OF WINDOWS ON THE TOUCH SCREEN
C*****************************************************************
          IMPLICIT INTEGER*2 (A-Z)
          INTEGER*4 FCB(16),DEVICE,TIME,SNSBUF
          DIMENSION X(5),Y(5),IX(5),IY(5)
          DIMENSION BANNER(20)
          INTEGER*1 ICHAR(8),IDCHAR(20)
          EXTERNAL IAND,CHAR
          COMMON /FCBS/ FCB,SNSBUF
          COMMON /MAXBLK/ RITMAX,BOTMAX,CHNMAX
          COMMON /UWPBLK/ RITUIW(15),LFTUIW(15),TOPUIW(15),BOTUIW(15)
          COMMON /CONLIM/ RITCON(15),LFTCON(15),TOPCON(15),BOTCON(15)
          COMMON /PARSET/ TABNUM
          DATA DEVICE/X'7EA0'/
          RITUIW(1)=BOTUIW(1)=511
          TOPUIW(1)=LFTUIW(1)=0
          RITMAX=511
          BOTMAX=511
          CHNMAX=16
          TABNUM=1
          CALL IOINIT(DEVICE,ERRSTAT,*100)
          CALL SELUIW(1)
          CALL PUP
          CALL UCCOFF(1)
          CALL MCWOUT(66)
2         READ(5,*)COL
          IF(COL.EQ.0) GO TO 10
          READ(5,*)(X(I),Y(I),I=1,5)
          CALL CURPOS(X(1),Y(1))
          DO 1 I=2,5
          CALL TTT(X(I),Y(I))
1         CONTINUE
          XX=X(1)+5  ;  YY=Y(1)+5
          CALL CURPOS(XX,YY)
          CALL PIXFOR(COL)
          CALL FIL
          READ(5,*)LENGTH,XX,YY
          READ(5,22)(IDCHAR(I),I=1,LENGTH)
22        FORMAT(20A1)
          CALL CURPOS(XX,YY)
          DO 23 I=1,LENGTH
          BANNER(I)=IDCHAR(I)
23        CONTINUE
          CALL RCHAR(LENGTH,BANNER)
          GO TO 2
10        READ(1,3)(ICHAR(I),I=1,8)
3         FORMAT(8A1)
          DO 4 I=2,7
          ICHAR(I)=ICHAR(I)-X'30'
4         CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      LCX=ICHAR(2)*100+ICHAR(3)*10+ICHAR(4)
      LCY=ICHAR(5)*100+ICHAR(6)*10+ICHAR(7)
      LCX=LCX*2 ; LCY=LCY*2
      IF(LCX.LT.128) THEN
        ITYPE=1
        GO TO 31
      ENDIF
      IF(LCX.GT.384) THEN
        ITYPE=4
        GO TO 31
      ENDIF
      IF(LCY.GT.384) THEN
        ITYPE=7
        GO TO 31
      ENDIF
      IF(LCY.GT.256) THEN
        IF(LCX.LT.256) THEN
          ITYPE=5
          GO TO 31
        ELSE
          ITYPE=6
          GO TO 31
        ENDIF
      ELSE
        IF(LCX.LT.256) THEN
          ITYPE=2
          GO TO 31
        ELSE
          ITYPE=3
          GO TO 31
        ENDIF
      ENDIF
31    OPEN(UNIT=2,FILE='DATA3',BLOCKED=.TRUE.,FORM='FORMATTED')
      WRITE(2,*)ITYPE
      CLOSE(UNIT=2)
      CALL UCCON(1)
99    STOP
100   TYPE *,'ERROR OCCURRED IN INITIALISING'
      STOP
      END
$ASSIGN LIB TO @SYSTEM(SYSTEM)AYDLIB BLOC=N
$ASSIGN DIR TO @SYSTEM(SYSTEM)AYDDIR BLOC=N
$EXECUTE CATALOG
A3 1=CA7EC7
A1 5=PROG2DAT
BUILD PROG2.SU,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB PROG1 VARMA,RAVI SLOF=DUMMY
$OPTION   2 3 4 5 17
$EXECUTE FORTRAN
C*****************************************************************************
C         FILENAME :- PROG3
C         TO DISPLAY TYPES OF FILTER TO BE USED ON THE TOUCH SCREEN
C*****************************************************************************
          IMPLICIT INTEGER*2 (A-Z)
          INTEGER*4 FCB(16),DEVICE,TIME,SNSBUF
          DIMENSION X(5),Y(5),IX(5),IY(5)
          DIMENSION BANNER(20)
          INTEGER*1 ICHAR(8),IDCHAR(20)
          EXTERNAL IAND,CHAR
          COMMON /FCBS/ FCB,SNSBUF
          COMMON /MAXBLK/ RITMAX,BOTMAX,CHNMAX
          COMMON /VWPBLK/ RITVIW(15),LFTVIW(15),TOPVIW(15),BOTVIW(15)
          COMMON /CONLIM/ RITCON(15),LFTCON(15),TOPCON(15),BOTCON(15)
          COMMON /PARSET/ TABNUM
          DATA DEVICE/X'7EA0'/
          RITVIW(1)=BOTVIW(1)=511
          TOPVIW(1)=LFTVIW(1)=0
          RITMAX=511
          BOTMAX=511
          CHNMAX=16
          TABNUM=1
          CALL IOINIT(DEVICE,ERRSTAT,*100)
          CALL SELVIW(1)
          CALL PUP
          CALL VCCOFF(1)
          CALL MCWOUT(66)
2         READ(5,*)COL
          IF(COL.EQ.0) GO TO 10
          READ(5,*)(X(I),Y(I),I=1,5)
          CALL CURPOS(X(1),Y(1))
          DO 1 I=2,5
          CALL TTT(X(I),Y(I))
1         CONTINUE
          XX=X(1)+5  ;  YY=Y(1)+5
          CALL CURPOS(XX,YY)
          CALL PIXFOR(COL)
          CALL FIL
          READ(5,*)LENGTH,XX,YY
          READ(5,22)(IDCHAR(I),I=1,LENGTH)
22        FORMAT(20A1)
          CALL CURPOS(XX,YY)
          DO 23 I=1,LENGTH
          BANNER(I)=IDCHAR(I)
23        CONTINUE
          CALL RCHAR(LENGTH,BANNER)
          GO TO 2
10        READ(1,3)(ICHAR(I),I=1,8)
3         FORMAT(8A1)
          DO 4 I=2,7
          ICHAR(I)=ICHAR(I)-X'30'
4         CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
          LCX=ICHAR(2)*100+ICHAR(3)*10+ICHAR(4)
          LCY=ICHAR(5)*100+ICHAR(6)*10+ICHAR(7)
          LCX=LCX*2 ; LCY=LCY*2
          IF(LCY.GT.256) THEN
            IF(LCX.LT.256) THEN
              ITYP=3
              NBANDS=3
              GO TO 31
            ELSE
              ITYP=4
              NBANDS=3
              GO TO 31
            ENDIF
          ELSE
            IF(LCX.LT.256) THEN
              ITYP=1
              NBANDS=2
              GO TO 31
            ELSE
              ITYP=2
              NBANDS=2
              GO TO 31
            ENDIF
          ENDIF
31        OPEN(UNIT=2,FILE='DATA1',BLOCKED=.TRUE.,FORM='FORMATTED')
          WRITE(2,*)ITYP,NBANDS
          CLOSE(UNIT=2)
          CALL UCCON(1)
          CALL PUP
99        STOP
100       TYPE *,'ERROR OCCURRED IN INITIALISING'
          STOP
          END
$ASSIGN LIB TO @SYSTEM(SYSTEM)AYDLIB BLOC=N
$ASSIGN DIR TO @SYSTEM(SYSTEM)AYDDIR BLOC=N
$EXECUTE CATALOG
A3 1=CA7EC7
A1 5=PROG3DAT
BUILD PROG3.SU,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB LOOKCOL SYSTEM,GOULD SLOF=DUMMY
$OPTION 2 3 4 5 17
$EXECUTE FORTRAN
C     FILENAME :- LOOKCOL
C
C     PROGRAM TO LOAD A LOOK-UP TABLE IN THE AYDIN MEMORY
C
      IMPLICIT INTEGER*2 (A-Z)
      REAL TEMP
      INTEGER*4 LOCBUF,FCB(16),DEVICE,TIME,SNSBUF
      DIMENSION BUFF1(258),BUFFER(256),RED(256),GREEN(256),BLUE(256)
      EQUIVALENCE (BUFF1(1),LOCBUF),(BUFF1(3),BUFFER(1))
      DATA FCB/X'00414141',0,X'02000000',13*0/
      DATA DEVICE/X'7EA0'/
      TIME=-10
      CALL AYOPEN(DEVICE,FCB,*800,IERR)
      CALL X:BRK(*1000,ERR,*200)
      CALL AYBRK(FCB)
      DO 10 I=1,16
      TEMP=(I-1)
      RED(I)=TEMP
      TEMP=(I-1)*3.0/15.0+0.5
      GREEN(I)=TEMP
      TEMP=(I-1)*1.0/15.0+0.5
      BLUE(I)=TEMP
10    CONTINUE
      DO 11 I=17,24
      TEMP=2.0
      RED(I)=TEMP
      TEMP=(24-I)*1.0/7.0+2.5
      GREEN(I)=TEMP
      TEMP=(I-17)*1.0/7.0+1.5
      BLUE(I)=TEMP
11    CONTINUE
      DO 14 I=25,32
      TEMP=(I-25)*1.0/7.0+2.5
      RED(I)=TEMP
      TEMP=2.0
      GREEN(I)=TEMP
      TEMP=2.0
      BLUE(I)=TEMP
14    CONTINUE
      DO 15 I=33,42
      TEMP=3.0
      RED(I)=TEMP
      TEMP=(I-33)*1.0/9.0+2.5
      GREEN(I)=TEMP
      TEMP=(42-I)*1.0/9.0+1.5
      BLUE(I)=TEMP
15    CONTINUE
      DO 16 I=43,52
      TEMP=(I-43)*1.0/9.0+3.5
      RED(I)=TEMP
      TEMP=3.0
      GREEN(I)=TEMP
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      TEMP=1.0
      BLUE(I)=TEMP
16    CONTINUE
      DO 17 I=53,64
      TEMP=4.0
      RED(I)=TEMP
      TEMP=3.0
      GREEN(I)=TEMP
      TEMP=(I-53)*1.0/11.0+1.5
      BLUE(I)=TEMP
17    CONTINUE
      DO 18 I=65,85
      RED(I)=4.0
      TEMP=(I-65)*2.0/20.0+3.5
      GREEN(I)=TEMP
      BLUE(I)=2.0
18    CONTINUE
      DO 19 I=86,106
      TEMP=(I-86)*1.0/20.0+4.5
      RED(I)=TEMP
      GREEN(I)=5.0
      TEMP=(106-I)*1.0/20.0+1.5
      BLUE(I)=TEMP
19    CONTINUE
      DO 20 I=107,128
      RED(I)=5.0
      TEMP=5.0
      GREEN(I)=TEMP
      TEMP=(I-107)*1.0/21.0+1.5
      BLUE(I)=TEMP
20    CONTINUE
      DO 21 I=129,149
      RED(I)=5.0
      TEMP=(I-129)*1.0/20.0+5.5
      GREEN(I)=TEMP
      BLUE(I)=2.0
21    CONTINUE
      DO 22 I=150,170
      TEMP=(I-150)*1.0/20.0+5.5
      RED(I)=TEMP
      GREEN(I)=6.0
      TEMP=2.0
      BLUE(I)=TEMP
22    CONTINUE
      DO 23 I=171,192
      TEMP=(I-171)*1.0/21.0+6.5
      RED(I)=TEMP
      GREEN(I)=6.0
      TEMP=2.0
      BLUE(I)=TEMP
23    CONTINUE
      DO 24 I=193,203
      TEMP=(203-I)*1.0/10.0+6.5
      RED(I)=TEMP
      GREEN(I)=6.0
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      TEMP=(I-193)*1.0/10.0+2.0
      BLUE(I)=TEMP
24    CONTINUE
      DO 25 I=204,214
      TEMP=(I-204)*1.0/10.0+6.5
      RED(I)=TEMP
      GREEN(I)=6.0
      TEMP=3.0
      BLUE(I)=TEMP
25    CONTINUE
      DO 26 I=215,224
      RED(I)=7.0
      TEMP=(I-215)*1.0/9.0+5.5
      GREEN(I)=TEMP
      TEMP=(224-I)*2.0/10.0+1.5
      BLUE(I)=TEMP
26    CONTINUE
      DO 27 I=225,235
      RED(I)=7
      TEMP=7.0
      GREEN(I)=TEMP
      TEMP=(I-225)*1.0/10.0+1.5
      BLUE(I)=TEMP
27    CONTINUE
      DO 28 I=236,246
      RED(I)=7
      TEMP=7.0
      GREEN(I)=TEMP
      BLUE(I)=2
28    CONTINUE
      DO 29 I=247,256
      RED(I)=7
      TEMP=7.0
      GREEN(I)=TEMP
      TEMP=(I-247)*1.0/9.0+2.5
      BLUE(I)=TEMP
29    CONTINUE
      DO 66 I=1,256
      BUFFER(I)=0
      CALL MVBITS(RED(I),0,3,BUFFER(I),0)
      CALL MVBITS(GREEN(I),0,3,BUFFER(I),3)
      CALL MVBITS(BLUE(I),0,2,BUFFER(I),6)
66    CONTINUE
      BUFF1(1)=X'CBD8'
      BUFF1(2)=X'00FF'
      CALL AYWRITE(FCB,LOCBUF,516,*100,SNSBUF)
      GO TO 999
1000  CALL X:BRKXIT
200   CONTINUE
      TYPE *,' BREAK ERROR =',ERR
      GO TO 999
100   TYPE *,' WRITE ERROR =',SNSBUF
      GO TO 999
800   TYPE *,'OPEN ERROR =',IERR
999   CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      STOP
      END
$AS LIB TO @SYSTEM(SYSTEM)AYDLIB BLOCKED=N
$AS DIR TO @SYSTEM(SYSTEM)AYDDIR BLOCKED=N
$EXECUTE CATALOG
A4 5=UT
BUILD LOOK
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB. RAVI EEDEPT SLOF=DUMMY
$OPTION 2 3 4 5 17 20
$EXECUTE FORTRAN
C-------------------------------------------------------------
C---  FILE NAME :- TWO : PROGRAM FOR THE DESIGN OF 2-D ---
C---  FIR DIGITAL FILTER USING TRANSFORMATION TECHNIQUE---
C---  AND SCALING THE FILTER COEFFICIENTS.              ---
C-------------------------------------------------------------
C      INPUT:- IMPULSE RESPONSE OBTAINED FROM 1-D FILTER
C      OUTPUT:- SCALED 2-D FILTER COEFFICIENTS
C      N = FILTER KERNEL SIZE
C-------------------------------------------------------------
       DIMENSION HW(50),T(50)
       REAL MAGR(25,25)
       INTEGER*1 ITIME(4,12)
       INTEGER*4  IMAGR(25,25)
       COMPLEX X(25,25),Y(25,25)
       COMPLEX CTERM,CDUM,SUM
       EQUIVALENCE (X(1,1),IMAGR(1,1))
       CDUM=CMPLX(0.0,1.0)
       PI2=8.0*ATAN(1.0)
       N=17
       OPEN(UNIT=1,BLOCKED=.FALSE.,FORM='UNFORMATTED')
       READ(1)(HW(I),I=1,N)
       CLOSE(UNIT=1)
X      CALL M:TDAY(ITIME(1,1))
       NUM=((N-1)/2)+1
       N2=N-1
       N3=NUM-1
       NUM1=NUM-2
       F1=0.0
       F2=-0.25
       DO 33 I=1,N
       DO 22 J=1,N
       W1=PI2*F1
       W2=PI2*F2
       TRANS=0.5*(-1.0+COS(W1*2)+COS(W2*2)+COS(W1*2)*COS(W2*2))
       SUM=HW(NUM)+2.0*HW(NUM-1)*TRANS+2.0*HW(NUM-2)*(2.0*TRANS**2-1.0)
       T(2)=2.0*TRANS**2-1.0
       T(1)=TRANS
       DO 11 K=3,N3
       T(K)=2.0*TRANS*T(K-1)-T(K-2)
       SUM=SUM+2.0*HW(NUM-K)*T(K)
11     CONTINUE
       X(I,J)=SUM
       SUM=SUM*CEXP(CDUM*W1*2*N3)
       SUM=SUM*CEXP(CDUM*W2*2*N3)
       F1=F1+(0.5/FLOAT(N2))
22     CONTINUE
       F2=F2+(0.5/FLOAT(N2))
33     CONTINUE
X      CALL M:TDAY(ITIME(1,2))
X      WRITE(6,576)((ITIME(I,J),I=1,4),J=1,2)
X576   FORMAT(4(5X,I3,5X),/)
       CALL DFT(N,N,X,Y,1.0)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      DO 44 I=1,N
      DO 44 J=1,N
      MAGR(I,J)=CABS(Y(I,J))
44    CONTINUE
      TOT=0.0
      DO 5 I=1,N
      DO 5 J=1,N
      TOT=TOT+MAGR(I,J)
5     CONTINUE
      TOT=TOT/(128.0*128.0)
      SCALE=0.025/TOT
      TYPE 6,TOT,SCALE
6     FORMAT(5X,'AVERAGE VALUE = ',E14.7,5X,'SCALE FACTOR = ',E14.7)
9     DO 10 I=1,N
      DO 10 J=1,N
      TEMP=MAGR(I,J)*SCALE
      IMAGR(I,J)=TEMP
10    CONTINUE
      OPEN(UNIT=2,BLOCKED=.FALSE.,FORM='UNFORMATTED',INCREMENT=4)
      WRITE(2)((IMAGR(I,J),J=1,N),I=1,N)
      CLOSE(UNIT=2)
      STOP
      END
      INCLUDE DFT
$EXECUTE CATALOG
A1 1=COEF
A1 2=IOENF17
A4 6=UT
BUILD TWO.SU
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB RAVI EEDEPT SLOF=DUMMY
$OPTION 2 3 4 5 17 20
$EXECUTE FORTRAN
C
C*************************************************************************
C*****            FILE NAME :- THREE                                  *****
C*****      DESIGN OF 2-D FIR DIGITAL FILTER USING                    *****
C*****            WINDOWING TECHNIQUES                                *****
C*************************************************************************
C
C     INPUT:-ITYPE - TYPE OF WINDOW; ITYP - TYPE OF FILTER
C            (BOTH ARE OBTAINED FROM THE TOUCH SCREEN
C              DEPENDING UPON THE REQUIREMENT)
C           NF = FILTER KERNEL SIZE.
C          FC, FL AND FH ARE THE CUTOFF FREQUENCIES
C     OUTPUT:- FILTER KERNEL OF SIZE NF X NF
C*************************************************************************
      DIMENSION W(9),WIN(9,9),H(9,9)
      REAL*4 H1(17,17)
      INTEGER*1 ITIME(4,12)
      COMPLEX X(9,9),Y(9,9),WN(9,9)
      PI=4.0*ATAN(1.0)
      TWOPI=2.0*PI
      NF=17
      OPEN(UNIT=1,FILE='DATA3',BLOCKED=.TRUE.,FORM='FORMATTED')
      READ(1,*)ITYPE
      CLOSE(UNIT=1)
      OPEN(UNIT=2,FILE='DATA1',BLOCKED=.TRUE.,FORM='FORMATTED')
      READ(2,*)ITYP,NBANDS
      CLOSE(UNIT=2)
X     CALL M:TDAY(ITIME(1,1))
      N=(NF+1)/2
      IF(ITYP.NE.1.AND.ITYP.NE.2) GO TO 50
40    FC=0.3
      GO TO 60
50    FL=0.2 ; FH=0.4
60    IF(ITYPE.NE.7) GO TO 70
      DPLOG=60.0
      DF=0.05
      DP=10.0**(-DPLOG/20.0)
      CALL CHEBC(NF,DP,DF,N,A0,AN)
70    IEO=MOD(NF,2)
      IF(IEO.EQ.1.OR.ITYP.EQ.1.OR.ITYP.EQ.3) GO TO 80
      NF=NF+1
      N=(1+NF)/2
      IEO=1
80    CONTINUE
      F1=0.0
      F2=0.0
      DO 21 I1=1,N
      DO 20 J1=1,N
      W1=TWOPI*F1
      W2=TWOPI*F2
      GO TO (1,2,3,4),ITYP
1     R=TWOPI*FC
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      IF((W1*W1+W2*W2).GT.R*R) GO TO 19
      X(I1,J1)=CMPLX(1.0,0.0)
      GO TO 22
2     R=TWOPI*FC
      IF((W1*W1+W2*W2).LT.R*R) GO TO 19
      X(I1,J1)=CMPLX(1.0,0.0)
      GO TO 22
3     R1=TWOPI*FL
      R2=TWOPI*FH
      IF(((W1*W1+W2*W2).LT.R1*R1).AND.(W1*W1+W2*W2).GT.R2*R2)GO TO 19
      X(I1,J1)=CMPLX(1.0,0.0)
      GO TO 22
4     R1=TWOPI*FL
      R2=TWOPI*FH
      IF(((W1*W1+W2*W2).GT.R1*R1).AND.(W1*W1+W2*W2).LT.R2*R2)
     *   GO TO 19
      X(I1,J1)=CMPLX(1.0,0.0)
      GO TO 22
19    X(I1,J1)=CMPLX(0.0,0.0)
22    F2=0.5*FLOAT(J1+1)/FLOAT(N)
20    CONTINUE
      F2=0.0
      F1=0.5*FLOAT(I1+1)/FLOAT(N)
21    CONTINUE
      CALL DFT(N,N,X,Y,1.0)
      DO 15 I=1,N
      DO 15 J=1,N
      H(I,J)=CABS(Y(I,J))
15    CONTINUE
      IF(ITYPE.EQ.1) WRITE(6,9989) NF
9989  FORMAT(23H RECTANGULAR WINDOW-NF=,I4)
      DO 100 I=1,N
      W(I)=1.0
100   CONTINUE
      GO TO (200,110,120,140,150,160,170), ITYPE
110   CALL TRIANG(NF,W,N,IEO)
      WRITE(6,9988) NF
9988  FORMAT(22H TRIANGULAR WINDOW-NF=,I4)
      GO TO 180
120   ALPHA=0.54
      WRITE(6,9987) NF
9987  FORMAT(19H HAMMING WINDOW-NF=,I4)
130   BETA=1.0-ALPHA
      CALL HAMMIN(NF,W,N,IEO,ALPHA,BETA)
      WRITE(6,9986) ALPHA
9986  FORMAT(7H ALPHA=,F14.7)
      GO TO 180
140   ALPHA=0.54
      WRITE(6,9984) NF
9984  FORMAT(31H GENERALIZED HAMMING WINDOW-NF=,I4)
      GO TO 130
150   ALPHA=0.5
      WRITE(6,9983) NF
9983  FORMAT(19H HANNING WINDOW-NF=,I4)
      NF=NF+2
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      N=N+1
      GO TO 130
160   ATT=60.0
      IF(ATT.GT.50.0)BETA=0.1102*(ATT-8.7)
      IF(ATT.GE.20.96.AND.ATT.LE.50.0)BETA=0.58417*(ATT-20.96)**
     1  0.4+(0.07886*(ATT-20.96))
      IF(ATT.LT.20.96)BETA=0.0
      CALL KAISER(NF,W,N,IEO,BETA)
      WRITE(6,9981) NF
9981  FORMAT(18H KAISER WINDOW-NF=,I4)
      WRITE(6,9980) ATT,BETA
9980  FORMAT(6H ATT=,F14.7,7H BETA=,F14.7)
      GO TO 180
170   CALL CHEBY(NF,W,N,IEO,DP,DF,XO,XN)
      WRITE(6,9979) NF
9979  FORMAT(21H CHEBYCHEV WINDOW-NF=,I4)
      WRITE(6,9978) DP,DF
9978  FORMAT(2X,'DP=',F14.7,2X,'DF=',F14.7)
180   IF(ITYPE.EQ.5)NF=NF-2
      IF(ITYPE.EQ.5)N=N-1
200   DO 33 I=1,N
      DO 33 J=1,N
      NN=SQRT(FLOAT(I*I+J*J))
      IF(NN.GT.N) WIN(I,J)=0.0
      IF(NN.LE.N) WIN(I,J)=W(NN)
      H(I,J)=WIN(I,J)*H(I,J)
33    CONTINUE
X     CALL MTDAY(ITIME(1,2))
X     WRITE(6,576)((ITIME(I,J),I=1,4),J=1,2)
X576  FORMAT(4(5X,I3,5X),/)
      WRITE(6,9975)
9975  FORMAT(14H WINDOW VALUES)
      DO 210 I=1,N
      DO 210 I1=1,N
      J=N+1-I
      K=NF+1-I
      J1=N+1-I1
      K1=NF+1-I1
      WRITE(6,9974) I,I1,I,K1,WIN(J,J1),K,I1,K,K1
9974  FORMAT(2X,'W(',I4,I4,')=W(',I4,I4,')=',E8.4,'=W(',I4,I4,')=W(',I4
     *I4,')')
210   CONTINUE
220   IF(ITYP.EQ.1)WRITE(6,9973)
9973  FORMAT(26H **LOWPASS FILTER DESIGN**)
      IF(ITYP.EQ.2)WRITE(6,9972)
9972  FORMAT(27H **HIGHPASS FILTER DESIGN**)
      IF(ITYP.EQ.3) WRITE(6,9971)
9971  FORMAT(27H **BANDPASS FILTER DESIGN**)
      IF(ITYP.EQ.4) WRITE(6,9970)
9970  FORMAT(27H **BANDPASS FILTER DESIGN**)
      IF(ITYP.EQ.1) WRITE(6,9969) FC
9969  FORMAT(22H IDEAL LOWPASS CUTOFF=,F14.7)
      IF(ITYP.EQ.2) WRITE(6,9968) FC
9968  FORMAT(23H IDEAL HIGHPASS CUTOFF=,F14.7)
      IF(ITYP.EQ.3.OR.ITYP.EQ.4) WRITE(6,9967) FL,FH
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
9967  FORMAT(26H IDEAL CUTOFF FREQUENCIES=,2F14.7)
      DO 250 I=1,N
      DO 250 I1=1,N
      J1=N+1-I1
      K1=NF+1-I1
      J=N+1-I
      K=NF+1-I
      H1(I,I1)=H(J,J1)
      H1(I,K1)=H(J,J1)
      H1(K,I1)=H(J,J1)
      H1(K,K1)=H(J,J1)
      WRITE(6,9966) I,I1,I,K1,H(J,J1),K,I1,K,K1
9966  FORMAT(2X,'H(',I4,I4,')=H(',I4,I4,')=',E9.4,'=H(',I4,I4,')=H(',I4
     *I4,')')
250   CONTINUE
      WRITE(6,9965)
9965  FORMAT(1H /1H /1H /1H )
      WRITE(6,9964)
9964  FORMAT(1H1)
      OPEN(UNIT=3,BLOCKED=.FALSE.,FORM='UNFORMATTED')
      WRITE(3)((H1(I,J),J=1,NF),I=1,NF)
      CLOSE(UNIT=3)
      STOP
      END
      SUBROUTINE TRIANG(NF,W,N,IEO)
      DIMENSION W(1)
      FN=N
      DO 10 I=1,N
      XI=I-1
      IF(IEO.EQ.0) XI=XI+0.5
      W(I)=1.0-XI/FN
10    CONTINUE
      RETURN
      END
      SUBROUTINE HAMMIN(NF,W,N,IEO,ALPHA,BETA)
      DIMENSION W(1)
      PI2=8.0*ATAN(1.0)
      FN=NF-1
      DO 10 I=1,N
      FI=I-1
      IF(IEO.EQ.0) FI=FI+0.5
      W(I)=ALPHA+BETA*COS((PI2*FI)/FN)
10    CONTINUE
      RETURN
      END
      SUBROUTINE KAISER(NF,W,N,IEO,BETA)
      DIMENSION W(1)
      REAL INO
      XIND=FLOAT(NF-1)*FLOAT(NF-1)
      DO 10 I=1,N
      XI=I-1
      IF(IEO.EQ.0) XI=XI+0.5
      XI=4.0*XI*XI
      W(I)=INO(BETA*SQRT(1.0-XI/XIND))
      W(I)=W(I)/BES
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
10      CONTINUE
        RETURN
        END
        REAL FUNCTION INO(X)
        Y=X/2.
        T=1.0E-08
        E=1.0
        DE=1.0
        DO 10 I=1,25
        XI=I
        DE=DE*Y/XI
        SDE=DE*DE
        E=E+SDE
        IF(E*T-SDE) 10,40,20
10      CONTINUE
20      INO=E
        RETURN
        END
        SUBROUTINE CHEBC(NF,DP,DF,N,XO,XN)
        PI=4.0*ATAN(1.0)
        IF(NF.NE.0) GO TO 10
        C1=COSHIN((1.0+DP)/DP)
        CO=COS(PI*DF)
        X=1.0+C1/COSHIN(1.0/CO)
        NF=X+1.0
        N=(NF+1)/2
        XN=NF-1
        GO TO 30
10      IF(DF.NE.0.0) GO TO 20
        XN=NF-1
        C1=COSHIN((1.0+DP)/DP)
        C2=COSH(C1/XN)
        DF=ARCCOS(1.0/C2)/PI
        GO TO 30
20      XN=NF-1
        CO=COS(PI*DF)
        C1=XN*COSHIN(1.0/CO)
        DP=1.0/(COSH(C1)-1.0)
30      XO=(3.0-COS(2.*PI*DF))/(1.+COS(2.*PI*DF))
        RETURN
        END
        SUBROUTINE CHEBY(NF,W,N,IEO,DP,DF,XO,XN)
        DIMENSION W(1)
        DIMENSION PR(256),PI(256)
        PIE=4.0*ATAN(1.0)
        XN=NF-1
        FNF=NF
        ALPHA=(XO+1.)/2.
        BETA=(XO-1.)/2.
        TWOPI=2.*PIE
        C2=XN/2.
        DO 40 I=1,NF
        XI=I-1
        F=XI/FNF
        X=ALPHA*COS(TWOPI*F)+BETA
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      IF(ABS(X)-1.) 10,10,20
10    P=DP*COS(C2*ARCCOS(X))
      GO TO 30
20    P=DP*COSH(C2*COSHIN(X))
30    PI(I)=0.0
      PR(I)=P
      IF(IEO.EQ.1) GO TO 40
      PR(I)=P*COS(PIE*F)
      PI(I)=-P*SIN(PIE*F)
      IF(I.GT.(NF/2+1)) PR(I)=-PR(I)
      IF(I.GT.(NF/2+1)) PI(I)=-PI(I)
40    CONTINUE
      TWN=TWOPI/FNF
      DO 60 I=1,N
      XI=I-1
      SUM=0.0
      DO 50 J=1,NF
      XJ=J-1
      SUM=SUM+PR(J)*COS(TWN*XJ*XI)+PI(J)*SIN(TWN*XJ*XI)
50    CONTINUE
      W(I)=SUM
60    CONTINUE
      C1=W(1)
      DO 70 I=1,N
      W(I)=W(I)/C1
70    CONTINUE
      RETURN
      END
      REAL FUNCTION COSHIN(X)
      COSHIN=ALOG(X+SQRT(X*X-1.0))
      RETURN
      END
      FUNCTION ARCCOS(X)
      IF(X)30,20,10
10    A=SQRT(1.0-X*X)/X
      ARCCOS=ATAN(A)
      RETURN
20    ARCCOS=2.0*ATAN(1.0)
      RETURN
30    A=SQRT(1.0-X*X)/X
      ARCCOS=ATAN(A)+4.0*ATAN(1.0)
      RETURN
      END
      REAL FUNCTION COSH(X)
      COSH=(EXP(X)+EXP(-X))/2.
      RETURN
      END
      SUBROUTINE DFT(N1,N2,X,Y,SIGN)
      COMPLEX X(N1,N2),Y(N1,N2),SUM,CDUM
      TWOPI=8.0*ATAN(1.0)
      CDUM=CMPLX(0.0,SIGN)*TWOPI
      DO 2 I=1,N1
      DO 2 J=1,N2
      SUM=0.0
      DO 1 I1=1,N1
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      DO 1 J1=1,N2
      SUM=SUM+(X(I1,J1)*CEXP(CDUM*(I1-1)*(I-1)/N1)*
     1  CEXP(CDUM*(J1-1)*(J-1)/N2))
1     CONTINUE
      IF(SIGN.EQ.1.0) SUM=SUM/(N1*N2)
      Y(I,J)=SUM
2     CONTINUE
      RETURN
      END
$EXECUTE CATALOG
A1 1=DATA3
A1 2=DATA1
A1 3=COEF
A4 6=UT
BUILD THREE.SU
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB ENNAR SYSTEM SLOF=DUMMY
$OPTION 2 3 4 5 17
$EXECUTE FORTRAN
C*************************************************************************
C******                FILE NAME:- SCALE                              ****
C******  SCLALING THE 2-D FILTER KERNEL IN ORDER TO USE THE  ****
C******                 NTT CONVOLVER                                 ****
C*************************************************************************
C INPUT :- 2-D FILTER KERNEL OF SIZE N X N
C          SCALE - SCALE FACTOR
C OUTPUT :- SCALED FILTER COEFFICIENTS
C*************************************************************************
      REAL*4 X(17,17)
      INTEGER*4 IX(17,17)
      N=17
      OPEN(UNIT=1,BLOCKED=.FALSE.,FORM='UNFORMATTED')
      READ(1)((X(I,J),J=1,N),I=1,N)
      CLOSE(UNIT=1)
      SUM=0.0
      DO 5 I=1,N
      DO 5 J=1,N
      SUM=SUM+X(I,J)
5     CONTINUE
      SUM=SUM/(128.0*128.0)
      SCALE=0.025/SUM
      TYPE 6,SUM,SCALE
6     FORMAT(5X,'AVERAGE VALUE = ',E14.7,5X,'SCALE FACTOR = ',E14.7)
9     DO 10 I=1,N
      DO 10 J=1,N
      TEMP=X(I,J)*SCALE
      IX(I,J)=TEMP
10    CONTINUE
      OPEN(UNIT=2,BLOCKED=.FALSE.,FORM='UNFORMATTED',INCREMENT=4)
      WRITE(2)((IX(I,J),J=1,N),I=1,N)
      CLOSE(UNIT=2)
      STOP
      END
$ALLOCATE 1F000
$EXECUTE CATALOG
A1 1=COEF
A1 2=IOENF17,,U
A4 6=UT
BUILD SCALESU
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB VARMA SYSTEM SLOF=DUMMY
$OPTION 2 3 4 5 17 20
$EXECUTE FORTRAN
C*****************************************************************
C      FILENAME :-FILRTD
C      SOFTWARE SIMULATION OF THE NTT CONVOLVER
C*****************************************************************
       IMPLICIT INTEGER (A-H,O-Z)
       REAL AM1,AM2,BIG,SMALL,TEMP1,TEMP
       INTEGER*4 IX(128,128),IC(17,17)
       INTEGER*2 IX641(128,128),IX769(128,128)
       INTEGER*2 IC641(128,128),IC769(128,128)
       INTEGER*2 XIO(128,128,2),TFTAB(128,2),TFTAB1(128,2)
       INTEGER*1 IDX(128,128),ITIME(4,12)
       EQUIVALENCE (XIO(1,1,1),IX(1,1))
       COMMON/NTT1/INB,IOB,NSTG
       COMMON /GLOBAL 00/IX,IX641,IX769,IC641,IC769
       CALL X_INCLD('@SYSTEM(SYSTEM)GLOBAL00','SYSTEM','RW',.FALSE.,2,LR
       IF(LR.NE.0) GO TO 100
       TYPE*,'FILTER OPERATION IS GOING ON '
       NFIL=17
       INB=1     ;IOB=2      ;NSTG=7
       N=128 ;M=7 ;MOD1=641 ;MOD2=769  ;IA1=335;IA2=5
       TFTAB(1,1)=1     ;TFTAB1(1,1)=1
       TFTAB(2,1)=IA1   ;TFTAB1(2,1)=IA2
       DO 220 I=3,128
       I1=TFTAB((I-1),1)*IA1      ;I2=TFTAB1((I-1),1)*IA2
       TFTAB(I,1)=MOD(I1,MOD1) ;TFTAB1(I,1)=MOD(I2,MOD2)
220    CONTINUE
       TFTAB(1,2)=1       ;TFTAB1(1,2)=1
       DO 330 I=2,128
       TFTAB(I,2)=TFTAB((130-I),1)   ;TFTAB1(I,2)=TFTAB1((130-I),1)
330    CONTINUE
       OPEN(UNIT=1,BLOCKED=.FALSE.,FORM='UNFORMATTED')
       READ(1)((IC(I,J),J=1,NFIL),I=1,NFIL)
       CLOSE(UNIT=1)
       OPEN(UNIT=2,BLOCKED=.FALSE.,FORM='UNFORMATTED')
       READ(2)((IDX(I,J),J=1,N),I=1,N)
       CLOSE(UNIT=2)
X      CALL M:TDAY(ITIME(1,1))
       DO 152 I=1,N
       DO 152 J=1,N
       IC641(I,J)=IC769(I,J)=0
152    CONTINUE
       DO 900 I=1,17
       DO 900 J=1,17
       IC641(I,J)=MOD(IC(I,J),MOD1)
       IC769(I,J)=MOD(IC(I,J),MOD2)
       IF(IC641(I,J).LT.0) IC641(I,J)=IC641(I,J)+MOD1
       IF(IC769(I,J).LT.0) IC769(I,J)=IC769(I,J)+MOD2
900    CONTINUE
       DO 901 I=1,N
       DO 901 J=1,N
       IX641(I,J)=IX769(I,J)=IDX(I,J)
901    CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      DO 2 I=1,N
      DO 2 J=1,N
2     XIO(I,J,1)=IX641(I,J)
X     CALL M:TDAY(ITIME(1,2))
      CALL NTT2D(XIO,TFTAB,0,N,MOD1)
X     CALL M:TDAY(ITIME(1,3))
      DO 3 I=1,N
      DO 3 J=1,N
3     IX641(I,J)=XIO(I,J,2)
      DO 22 I=1,N
      DO 22 J=1,N
22    XIO(I,J,1)=IX769(I,J)
      INB=1     ;IOB=2
      CALL NTT2D(XIO,TFTAB1,0,N,MOD2)
      DO 33 I=1,N
      DO 33 J=1,N
33    IX769(I,J)=XIO(I,J,2)
      DO 221 I=1,N
      DO 221 J=1,N
221   XIO(I,J,1)=IC641(I,J)
      INB=1     ;IOB=2
      CALL NTT2D(XIO,TFTAB,0,N,MOD1)
      DO 331 I=1,N
      DO 331 J=1,N
331   IC641(I,J)=XIO(I,J,2)
      DO 222 I=1,N
      DO 222 J=1,N
222   XIO(I,J,1)=IC769(I,J)
      INB=1     ;IOB=2
      CALL NTT2D(XIO,TFTAB1,0,N,MOD2)
      DO 333 I=1,N
      DO 333 J=1,N
333   IC769(I,J)=XIO(I,J,2)
      INV1=636 ;INV2=763
      DO 7 I=1,N
      DO 7 J=1,N
      I1=IC641(I,J)*INV1*INV1
      IC641(I,J)=MOD(I1,MOD1)
      I1=IC769(I,J)*INV2*INV2
      IC769(I,J)=MOD(I1,MOD2)
 7    CONTINUE
      DO 8 I=1,N
      DO 8 J=1,N
      ITEMP      =IX641(I,J)*IC641(I,J)
      IX641(I,J)=MOD(ITEMP,MOD1)
      ITEMP      =IX769(I,J)*IC769(I,J)
      IX769(I,J)=MOD(ITEMP,MOD2)
 8    CONTINUE
      DO 1000 I=1,N
      DO 1000 J=1,N
1000  XIO(I,J,1)=IX641(I,J)
      INB=1     ;IOB=2
      CALL NTT2D(XIO,TFTAB,1,N,MOD1)
      DO 1001 I=1,N
      DO 1001 J=1,N
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
1001  IX641(I,J)=XIO(I,J,2)
      DO 1002 I=1,N
      DO 1002 J=1,N
1002  XIO(I,J,1)=IX769(I,J)
      INB=1     ;IOB=2
      CALL NTT2D(XIO,TFTAB1,1,N,MOD2)
      DO 1003 I=1,N
      DO 1003 J=1,N
1003  IX769(I,J)=XIO(I,J,2)
      AM1=(MOD1*MOD2)/16.0+0.5
      AM2=AM1/2.0
      M1=AM1
      M1BY2=AM2
      BIG=0.0  ;SMALL=0.0
      DO 9 I=1,N
      DO 9 J=1,N
      IF(IX769(I,J).GT.384) IX769(I,J)=IX769(I,J)-MOD2
      IF(IX641(I,J).GT.320) IX641(I,J)=IX641(I,J)-MOD1
      ITEMP=IX769(I,J)-IX641(I,J)
      IF(IX641(I,J).LT.0) ITEMP=ITEMP+128
      ITEMP=ITEMP*6
      ITEMP=(MOD(ITEMP,MOD2))
      IF(ITEMP.LT.0) ITEMP=ITEMP+MOD2
      ITEMP=ITEMP*MOD1
      ITEMP1=IX641(I,J)
      IF(ITEMP1.LT.0) ITEMP1=ITEMP1+MOD1
      TEMP1=ITEMP1/16.0+0.5
      TEMP=ITEMP/16.0+0.5
      ITEMP=TEMP
      ITEMP1=TEMP1
      IX(I,J)=ITEMP+ITEMP1
      IF(IX(I,J).GT.M1BY2) IX(I,J)=IX(I,J)-M1
      TEMP1=IX(I,J)
      IF(SMALL.GT.TEMP1) SMALL=TEMP1
      IF(BIG.LT.TEMP1) BIG=TEMP1
9     CONTINUE
      BIG=ABS(BIG)+ABS(SMALL)
      DO 572 I=1,N
      DO 572 J=1,N
      TEMP1=IX(I,J)
      TEMP1=TEMP1+ABS(SMALL)
      TEMP=TEMP1/BIG*256
      IDX(I,J)=TEMP
572   CONTINUE
X     CALL M:TDAY(ITIME(1,4))
      OPEN(UNIT=3,BLOCKED=.FALSE.,FORM='UNFORMATTED')
      WRITE(3)((IDX(I,J),J=1,128),I=1,128)
      CLOSE(UNIT=3)
X     WRITE(6,576)((ITIME(I,J),I=1,4),J=1,4)
X576  FORMAT(4(5X,I3,5X),/)
      STOP
100   TYPE *,'ERROR IN INCLUDING COMMON AREA ...',LP
      STOP
      END
$AS LIB TO FILLIB BLOC=N
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$AS DIR TO FILDIR BLOC=N
$EXECUTE CATALOG
A1 1=IOENF17,,U
AS 2 TO @SYSTEM(IMAGE)PSTZ
A1 3=DUMMYIMG
A4 6=UT
BUILD FILRTDSV,P,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB VARMA SYSTEM SLOF=DUMMY
$OPTION 1 2 3 4 5 17
$EXECUTE FORTRAN
C*****************************************************************
C       FILENAME :- NTFLCF
C*****************************************************************
C       THIS PROGRAM FINDS THE NTT OF THE FILTER COEFFS
C       AND THEIR INDICES.THEN IT FORMS A 24 BIT WORD USING
C       FOUR 6 BIT NUMBERS AND STORES IT IN A FILE FOR LATER USE
C*****************************************************************
      IMPLICIT INTEGER(A-H,O-Z)
      INTEGER*2 IX641(128,128),IX769(128,128)
      INTEGER*1 IX162(128,128),IX163(128,128)
      INTEGER*1 IX262(128,128),IX263(128,128)
      INTEGER*1 IND62(769),IND63(769)
      INTEGER*4 IV(769)
      INTEGER*4 IX(128,128),IY(128,128)
      EQUIVALENCE (IY(1,1),IX641(1,1)),(IY(1,65),IX769(1,1))
      COMMON /GLOBAL OO/IX,IY,IX162,IX163,IX262
      CALL X_INCLD('@SYSTEM(SYSTEM)GLOBALOO','SYSTEM','RW',.FALSE.,
     1  2,IER)
      IF(IER.NE.0) GO TO 7
      TYPE *,'ENTER THE SIZE OF THE FILTER, NFIL------>'
      ACCEPT *,NFIL
      N=128 ;M=7 ;MOD1=641 ;MOD2=769 ;IA1=335 ;IA2=5
      G1=3 ;G2=11 ;INV1=636 ;INV2=763
      OPEN(UNIT=1,BLOCKED=.FALSE.,FORM='UNFORMATTED',INCREMENT=4)
      READ(1)((IX(I,J),J=1,NFIL),I=1,NFIL)
      CLOSE(UNIT=1)
      DO 1 I=1,N
      DO 1 J=1,N
      IF(I.LE.NFIL.AND.J.LE.NFIL) THEN
      IX641(I,J)=MOD(IX(I,J),MOD1)
      IX769(I,J)=MOD(IX(I,J),MOD2)
      IF(IX641(I,J).LT.0) IX641(I,J)=IX641(I,J)+MOD1
      IF(IX769(I,J).LT.0) IX769(I,J)=IX769(I,J)+MOD2
      ELSE
      IX641(I,J)=IX769(I,J)=0
      ENDIF
1     CONTINUE
      CALL TDNTT(IX641,M,N,MOD1,IA1,0)
      CALL TDNTT(IX769,M,N,MOD2,IA2,0)
      DO 2 I=1,N
      DO 2 J=1,N
      I1=IX641(I,J)*INV1*INV1
      IX641(I,J)=MOD(I1,MOD1)
      I1=IX769(I,J)*INV2*INV2
      IX769(I,J)=MOD(I1,MOD2)
2     CONTINUE
      CALL TABIND(MOD1,G1,IND62,IND63,IV)
      DO 3 I=1,N
      DO 3 J=1,N
      K1=IX641(I,J)
      IX162(I,J)=IND62(K1+1)
      IX163(I,J)=IND63(K1+1)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
3       CONTINUE
        CALL TABIND(MOD2,G2,IND62,IND63,IU)
        DO 4 I=1,N
        DO 4 J=1,N
        K1=IX769(I,J)
        IX262(I,J)=IND62(K1+1)
        IX263(I,J)=IND63(K1+1)
4       CONTINUE
C
C       NOW WE COMBINE THE FOUR  6 BITS INTO A 32 BIT VARIABLE.
C
        DO 5 I=1,N
        DO 5 J=1,N
        IX(I,J)=IX162(I,J)
        CALL MVBITS(IX163(I,J),0,6,IX(I,J),6)
        CALL MVBITS(IX262(I,J),0,6,IX(I,J),12)
        CALL MVBITS(IX263(I,J),0,6,IX(I,J),18)
5       CONTINUE
        I1=1 ;I2=65 ;J1=1 ;J2=65
        DO 6 I=1,N
        DO 6 J=1,N,4
        IY(I,J)=IX(I1,J1)
        IY(I,J+1)=IX(I2,J1)
        IY(I,J+2)=IX(I1,J2)
        IY(I,J+3)=IX(I2,J2)
        J1=J1+1
        IF(J1.EQ.65) THEN
          J1=1 ;J2=65 ;I1=I1+1 ;I2=I2+1
        ELSE
          J2=J2+1
        END IF
6       CONTINUE
        OPEN(UNIT=2,BLOCKED=.FALSE.,FORM='UNFORMATTED',INCREMENT=4)
        WRITE(2)((IY(I,J),I=1,N),J=1,N)
        CLOSE(UNIT=2)
        STOP
7       TYPE *,'ERROR IN INCLUDING THE COMMON AREA'
        STOP
        END
        SUBROUTINE TDNTT(IX,M,N,MOD1,IA1,IER)
        INTEGER*2 IX(N,N)
        INTEGER*4 IT(128)
        DO 9 I=1,N
        DO 5 J=1,N
5       IT(J)=IX(I,J)
        CALL NTT(IT,M,N,MOD1,IA1,IER)
        DO 6 J=1,N
6       IX(I,J)=IT(J)
9       CONTINUE
        DO 15 J=1,N
        DO 10 I=1,N
10      IT(I)=IX(I,J)
        CALL NTT(IT,M,N,MOD1,IA1,IER)
        DO 11 I=1,N
11      IX(I,J)=IT(I)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
15      CONTINUE
        RETURN
        END
        SUBROUTINE NTT(IX,M,N,MOD1,ALPHA,IFR)
        IMPLICIT INTEGER(A-H,O-Z)
        INTEGER IX(N),TWID(128)
        TWID(1)=1
        TWID(2)=ALPHA
        DO 1 I=3,128
        I1=TWID(I-1)*ALPHA
        TWID(I)=MOD(I1,MOD1)
1       CONTINUE
        NV2=N/2
        NM1=N-1
        J=1
        DO 7 I=1,NM1
        IF(I.GE.J) GO TO 5
        T=IX(I)
        IX(I)=IX(J)
        IX(J)=T
5       K=NV2
6       IF(K.GE.J) GO TO 7
        J=J-K
        K=K/2
        GO TO 6
7       J=J+K
        DO 20 L=1,M
        LE=2**L
        LE1=LE/2
        SCL=N/LE
        U=1
        DO 20 J=1,LE1
        TF=TWID(U)
        IF(IFR.NE.1) GO TO 18
        IF(U.EQ.1) TF=TWID(1)
        IF(U.NE.1) TF=TWID(128-U)
18      DO 10 I=J,N,LE
        IP=I+LE1
        T=IX(IP)*TF
        T=MOD(T,MOD1)
        IX(IP)=IX(I)-T
        IX(I)=IX(I)+T
        IX(IP)=MOD(IX(IP),MOD1)
        IX(I)=MOD(IX(I),MOD1)
        IF(IX(IP).LT.0) IX(IP)=IX(IP)+MOD1
        IF(IX(I).LT.0) IX(I)=IX(I)+MOD1
10      CONTINUE
20      U=U+SCL
        RETURN
        END
      SUBROUTINE TABIND(MMOD1,G1,IND62,IND63,IU)
      IMPLICIT INTEGER (A-H,O-Z)
      INTEGER*1 IND62(769),IND63(769)
      INTEGER*4 IND(769),IU(769)
      SM1=62
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      SM2=63
      IV(1)=1
      MM1=MMOD1-1
      DO 1 I=2,MM1
      R=IV(I-1)*G1
      R=MOD(R,MMOD1)
      IV(I)=R
      IND(R+1)=I-1
1     CONTINUE
      IV(MMOD1)=1
      IND(2)=0
      DO 2 I=2,MMOD1
      IND62(I)=MOD(IND(I),SM1)
      IND63(I)=MOD(IND(I),SM2)
2     CONTINUE
      IND62(1)=63
      IND63(1)=63
      RETURN
      END
$EXECUTE CATALOG
A1 1=IOENF17,,U
A1 2=NTIOENF,,U
BUILD NTFLCFSU,P,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
$JOB VARMA SYSTEM SLOF=DUMMY
$OPTION 1 2 3 4 5 17
$EXECUTE FORTRAN
C*****************************************************************
C         FILENAME :-CONFIL
C
C*****************************************************************
C
C         THIS PROGRAM SENDS THE NTT OF THE FILTER COEFFS AND THE
C         IMAGE TO THE NTT CONVOLVER AND GETS BACK THE FILTERED
C         IMAGE FROM THE CONVOLVER.
C
C         ASSIGN1 1=NTTOENF ---> FILENAME FOR NTT COEFFS
C         ASSIGN1 2=IMAGE FILENAME. (UNFORMATTED,UNBLOCKED)
C         ASSIGN1 3=OUTPUT IMAGE FILENAME. (--DO--)
C
C*****************************************************************
          INTEGER*4 IX(128,128),IP1(8192)
          INTEGER*1 IP(128,128),ITIME(4,12)
          EQUIVALENCE (IP(1,1),IP1(1))
          COMMON /GLOBAL OO/IX,IP1
          CALL X_INCLD('@SYSTEM(SYSTEM)GLOBALOO','SYSTEM','RW',.FALSE.,
     1    2,IER)
          IF(IER.NE.0) GO TO 100
          TYPE *,'THE FILTER PROGRAM HAS STARTED'
          MOD1=641   :MOD2=769
          N=128
          OPEN(UNIT=1,BLOCKED=.FALSE.,FORM='UNFORMATTED')
          READ(1)((IX(I,J),J=1,N),I=1,N)
          CLOSE(UNIT=1)
          OPEN(UNIT=2,BLOCKED=.FALSE.,FORM='UNFORMATTED')
          READ(2)((IP(I,J),J=1,N),I=1,N)
          CLOSE(UNIT=2)
          TYPE *,'WISH TO TRY HOMOMORPHIC FILTERING ? Y(1)/N(0)'
          ACCEPT *,IHOM
          IF(IHOM.NE.1) GO TO 110
          ALOG255=ALOG10(255.0)
          DO 1 I=1,N
          DO 1 J=1,N
          TEMP=FLOAT(IP(I,J)+1)
          TEMP=ALOG10(TEMP)*255.0/ALOG255
          IP(I,J)=IFIX(TEMP)
1         CONTINUE
110       IOCM=IER1=IER2=0
X         CALL M:TDAY(ITIME(1,1))
          TYPE *,'O.K.1'
          CALL LDCOEF(IX,IER1,IER2,IOCM)
          TYPE *,'O.K.2'
29        IF(IOCM.EQ.1) GO TO 30
          GO TO 29
30        CONTINUE
X         CALL M:TDAY(ITIME(1,2))
          WRITE(6,2)IER1,IER2,IOCM
          IOCM=IER1=IER2=0
X         CALL M:TDAY(ITIME(1,3))
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
      CALL LDIMG(IP1,IER1,IER2,IOCM)
39    IF(IOCM.EQ.1) GO TO 40
      GO TO 39
40    CONTINUE
X     CALL M:TDAY(ITIME(1,4))
      WRITE(6,2)IER1,IER2,IOCM
2     FORMAT(1X,'FCB STATUS=',Z8,5X,'CONVOLVER STATUS=',Z8,5X,'IOCM
     1  = ',I3,/)
      IOCM=IER1=IER2=0
X     CALL M:TDAY(ITIME(1,5))
      CALL CLSRST(IER1,IER2,1)
      IOCM=IER1=IER2=0
50    CALL CLSRST(IER1,IER2,2)
      ISAMP=8Z00000010
      ITEST=IAND(IER2,ISAMP)
      IF(ITEST.EQ.16) GO TO 50
X     CALL M:TDAY(ITIME(1,6))
      IOCM=IER1=IER2=0
      CALL STFIMG(IP1,IER1,IER2,IOCM)
59    IF(IOCM.EQ.1) GO TO 60
      GO TO 59
60    CONTINUE
X     CALL M:TDAY(ITIME(1,7))
      WRITE(6,2)IER1,IER2,IOCM
C     WRITE(4,3)(IP1(I),I=1,8192)
C3    FORMAT(16(' ',8(Z8,1X),/))
X     CALL M:TDAY(ITIME(1,8))
      K=1
      DO 70 I=1,N
      DO 70 J=1,N,2
      IX(J,I)=IBITS(IP1(K),16,16)
      IX(J+1,I)=IBITS(IP1(K),0,16)
      K=K+1
70    CONTINUE
X     CALL M:TDAY(ITIME(1,9))
      AM1=(MOD1*MOD2)/16.0+0.5
      AM2=AM1/2.0
      M1=AM1
      M1BY2=AM2
      BIG=0.0  ;SMALL=0.0
      DO 571 I=1,N
      DO 571 J=1,N
      IF(IX(I,J).GT.M1BY2) IX(I,J)=IX(I,J)-M1
      TEMP1=IX(I,J)
      IF(SMALL.GT.TEMP1) SMALL=TEMP1
      IF(BIG.LT.TEMP1) BIG=TEMP1
571   CONTINUE
      BIG=ABS(BIG)+ABS(SMALL)
      DO 572 I=1,N
      DO 572 J=1,N
      TEMP1=IX(I,J)
      TEMP1=TEMP1+ABS(SMALL)
      TEMP=TEMP1/BIG*256
      IP(I,J)=TEMP
572   CONTINUE
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
X         CALL M:TDAY(ITIME(1,10))
X         WRITE(6,576)((ITIME(I,J),I=1,4),J=1,10)
X576      FORMAT(4(5X,I3,5X),/)
          IF(IHOM.NE.1) GO TO 111
          DO 574 I=1,N
          DO 574 J=1,N
          TEMP=(FLOAT(IP(I,J))*ALOG255)/255.0
          IP(I,J)=(10.0**TEMP)-1
574       CONTINUE
111       OPEN(UNIT=3,BLOCKED=.FALSE.,FORM='UNFORMATTED')
          WRITE(3)((IX(I,J),J=1,N),I=1,N)
          CLOSE(UNIT=3)
          GO TO 15
100       TYPE *,'ERROR IN INCLUDING THE COMMON AREA'
15        STOP
          END
$OPTION 5 20
$EXECUTE ASSEMBLE
*         SUBROUTINE LDCOEF(IDAT,IER1,IER2,IOCM)
*         SUBROUTINE TO LOAD THE NTT OF FILTER COEFFS INTO THE CONVOLVER
*         IDAT....DATA ARRAY TO BE SENT TO THE CONVOLVER.
*         IER1....STATUS OF THE HANDLER POSTED IN THE FCB.
*         IER2....STATUS OF THE DEVICE POSTED IN THE IOCL
*         IOCM....A PARAMETER TO INDICATE THAT THE I/O IS COMPLETE.
*
          REL
          PROGRAM LDCOEF
          DEF LDCOEF
          LIST ON,NODATA,NOMAC,NOREP
          M.EQUS
          BOUND     4
          M.FCBEXP  FCB1,FIL,IOCL1,0,,NWI,,,,ERROR1,ERROR1
IOCL1     GEN       8/X'A2',8/X'30',16/00
          GEN       32/W(DATA)
          DATAD     0
          GEN       8/X'02',8/X'20',16/X'4000'
          GEN       32/0
          DATAD     0
          GEN       8/X'A8',8/X'00',16/00
          GEN       32/W(DATA)
          DATAD     0
LDCOEF    STF       R0,SAVREG       SAVE THE CONTENTS OF ALL REGS.
*         BL        *C.DEBUG        ENTER SYSTEM DEBUGGER TO TEST.
*         LF        R0,SAVREG       RESTORE THE CONTENTS OF REGS.
          TRR       R0,R1           TRANSFER PARAM LIST ADDR TO R1.
          LA        R2,*1W,R1       STORE DATA ADRR INTO R2
          STW       R2,DATADR       SAVE THIS ADDR IN A LOC.
          STW       R2,IOCL1+20     PLACE THIS VALUE IN THE IOCLS
          LA        R2,*2W,R1       GET THE ADDR OF IER1 INTO A LOC
          STW       R2,SAVIER1
          LA        R2,*3W,R1
          STW       R2,SAVIER2
          LA        R2,*4W,R1       THIS IS TO GET THE ADDR OF IOCM
          STW       R2,SAVIOCM      STORE IT IN A MEMORY LOCATION.
          ABR       R0,29           START CALCULATING RETURN ADDR
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
            ADMW    R0,0,R1
            STW     R0,SAVREG
            LA      1,FCB1
            SVC     1,X'25'
            LF      R0,SAVREG
            TRSW    R0                  RETURN BACK TO THE MAIN PROGRAM.
ERROR1      LW      R2,FCB1+12
            LW      R3,IOCL1+44
            STW     R2,*SAVIER1
            STW     R3,*SAVIER2
            LI      R3,1                TO INDICATE THAT I/O IS COMPLETE
            STW     R3,*SAVIOCM         LOAD THIS INFORMATION IN IOCM.
            SVC     1,X'2C'             RETURN BACK TO IOCS
SAVREG      RES     1F
DATA        RES     1W
SAVIER1     RES     1W
SAVIER2     RES     1W
DATADR      RES     1W
SAVIOCM     RES     1W
            END
*           SUBROUTINE LDIMG (IDAT,IER1,IER2,IOCM)
*           SUBROUTINE TO LOAD THE IMAGE TO BE FILTERED INTO THE CONVOLVER
*           IDAT....DATA ARRAY TO BE SENT TO THE CONVOLVER.
*           IER1....STATUS OF THE HANDLER POSTED IN THE FCB.
*           IER2....STATUS OF THE DEVICE POSTED IN THE IOCL
*           IOCM....A PARAMETER TO INDICATE THAT THE I/O IS COMPLETE.
*
            REL
            PROGRAM LDIMG
            DEF LDIMG
            LIST ON,NODATA,NOMAC,NOREP
            M.EQUS
            M.FCBEXP  FCB2,FIL,IOCL2,0,,NWI,,,,ERROR2,ERROR2
IOCL2       GEN       8/X'A2',8/X'30',16/00
            GEN       32/W(DATA)
            DATAD     0
            GEN       8/X'02',8/X'10',16/X'1000'
            GEN       32/0
            DATAD     0
            GEN       8/X'A8',8/X'00',16/00
            GEN       32/W(DATA)
            DATAD     0
LDIMG       STF       R0,SAVREG         SAVE THE CONTENTS OF ALL REGS.
            TRR       R0,R1              TRANSFER PARAM LIST ADDR TO R1.
            LA        R2,*1W,R1          STORE DATA ADRR INTO R2
            STW       R2,DATADR          SAVE THIS ADDR IN A LOC.
            STW       R2,IOCL2+20        PLACE THIS VALUE IN THE IOCLS
            LA        R2,*2W,R1          GET THE ADDR OF IER1 INTO A LOC
            STW       R2,SAVIER1
            LA        R2,*3W,R1
            STW       R2,SAVIER2
            LA        R2,*4W,R1          THIS IS TO GET THE ADDR OF IOCM
            STW       R2,SAVIOCM         STORE IT IN A MEMORY LOCATION.
            ABR       R0,29              START CALCULATING RETURN ADDR
            ADMW      R0,0,R1
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
            STW     R0,SAVREG
            LA      1,FCB2
            SVC     1,X'25'
            LF      R0,SAVREG
            TRSW    R0                  RETURN BACK TO THE MAIN PROGRAM.
ERROR2      LW      R2,FCB2+12
            LW      R3,IOCL2+44
            STW     R2,*SAVIER1
            STW     R3,*SAVIER2
            LI      R3,1                TO INDICATE THAT I/O IS COMPLETE
            STW     R3,*SAVIOCM         LOAD THIS INFORMATION IN IOCM.
            SVC     1,X'2C'             RETURN BACK TO IOCS
SAVREG      RES     1F
DATA        RES     1W
SAVIER1     RES     1W
SAVIER2     RES     1W
DATADR      RES     1W
SAVIOCM     RES     1W
            END
*           SUBROUTINE STFIMG(IDAT,IER1,IER2,IOCM)
*           SUBROUTINE TO READ BACK THE FILTERED IMAGE FROM  THE CONVOLVER
*           IDAT....DATA ARRAY TO BE SENT TO THE CONVOLVER.
*           IER1....STATUS OF THE HANDLER POSTED IN THE FCB.
*           IER2....STATUS OF THE DEVICE POSTED IN THE IOCL
*           IOCM....A PARAMETER TO INDICATE THAT THE I/O IS COMPLETE.
*
            REL
            PROGRAM STFIMG
            DEF STFIMG
            LIST ON,NODATA,NOMAC,NOREP
            M.EQUS
            M.FCBEXP  FCB3,FIL,IOCL3,0,,NWI,,,,ERROR3,ERROR3
IOCL3       GEN       8/X'A2',8/X'30',16/00
            GEN       32/W(DATA)
            DATAD     0
            GEN       8/X'82',8/X'10',16/X'2000'
            GEN       32/0
            DATAD     0
            GEN       8/X'A8',8/X'00',16/00
            GEN       32/W(DATA)
            DATAD     0
STFIMG      STF       R0,SAVREG         SAVE THE CONTENTS OF ALL REGS.
            TRR       R0,R1             TRANSFER PARAM LIST ADDR TO R1.
            LA        R2,*1W,R1         STORE DATA ADRR INTO R2
            STW       R2,DATADR         SAVE THIS ADDR IN A LOC.
            STW       R2,IOCL3+20       PLACE THIS VALUE IN THE IOCLS
            LA        R2,*2W,R1         GET THE ADDR OF IER1 INTO A LOC
            STW       R2,SAVIER1
            LA        R2,*3W,R1
            STW       R2,SAVIER2
            LA        R2,*4W,R1         THIS IS TO GET THE ADDR OF IOCM
            STW       R2,SAVIOCM        STOPE IT IN A MEMORY LOCATION.
            ABR       R0,29             START CALCULATING RETURN ADDR
            ADMW      R0,0,R1
            STW       R0,SAVREG
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
            LA          1,FCB3
            SVC         1,X'25'
            LF          R0,SAVREG
            TRSW        R0                  RETURN BACK TO THE MAIN PROGRAM.
ERROR3      LW          R2,FCB3+12
            LW          R3,IOCL3+44
            STW         R2,*SAVIER1
            STW         R3,*SAVIER2
            LI          R3,1                TO INDICATE THAT I/O IS COMPLETE
            STW         R3,*SAVIOCM         LOAD THIS INFORMATION IN IOCM.
            SVC         1,X'2C'             RETURN BACK TO IOCS
SAVREG      RES         1F
DATA        RES         1W
SAVIER1     RES         1W
SAVIER2     RES         1W
DATADR      RES         1W
SAVIOCM     RES         1W
            END
*           SUBROUTINE CLSRST(IER1,IER2,IS)
*           THIS SUBROUTINE IS USED FOR THREE PURPOSES.
*           1)..TO CLEAR MAIN COUNTER AND READ THE STATUS.
*           2)..TO JUST READ THE STATUS WITH A NOP.
*           3)..TO CLEAR THE MAIN COUNTER AND START THE FILTER.
*           IER1.....GIVES BACK THE HANDLER STATUS AS POSTED IN THE FCB
*           IER2.....GIVES THE STATUS OF THE FILTER AS POSTED IN IOCL
*           IS.......USED TO INDICATE THE FUNCTION AS FOLLOWS
*           0........CLEAR MAIN-COUNTER AND READ STATUS
*           1........CLEAR MAIN-COUNTER AND START FILTER.
*           2........READ STATUS WITH A NOP.
*
            REL
            PROGRAM CLSRST
            DEF CLSRST
            LIST ON,NODATA,MAC
            M.EQUS
IOCL4       GEN 8/X'A2',8/X'30',16/00
            GEN 32/W(DATA)
            DATAD 0
IOCL5       GEN 8/X'A8',8/X'00',16/0
            GEN 32/W(DATA)
            DATAD 0
            BOUND 4
            M.FCBEXP FCB4,FIL,IOCL4,0,,,,,,,ERROR1
            M.FCBEXP FCB5,FIL,IOCL5,0,,,,,,,ERROR2
CLSRST      STF R0,SAVREG
            TRR R0,R1
            LA R3,*1W,R1
            STW R3,SAVIER1
            LA R3,*2W,R1
            STW R3,SAVIER2
            ABR R0,29
            ADMW R0,0,R1
            STW R0,SAVREG
            TBM         30,*3W,R1
            BNS         NEXT1
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
          LA        1,FCB5          THIS IS JUST TO READ THE STATUS.
          SVC       1,X'25'
ERROR2    LW        R5,FCB5+12      GET THE HANDLER STATUS FROM FCB.
          BU        NEXT3
NEXT1     TBM 31,*3W,R1
          BNS NEXT2
          SBM 0,IOCL4+1
NEXT2     LA 1,FCB4
          SVC 1,X'25'
ERROR1    LW R5,FCB4+12
NEXT3     LW R4,IOCL4+28
          STW R4,*SAVIER2
          STW R5,*SAVIER1
          LF R0,SAVREG
          TRSW R0
SAVREG    RES 1F
SAVIER1   RES 1W
SAVIER2   RES 1W
DATA      DATAW 0
          END
$EXECUTE CATALOG
AS FIL=U2
A1 1=NTTQENF,,U
ASSIGN 2 TO @SYSTEM;IMAGE;PSEC
A1 3=DUMMYIMG
CATALOG CONFILSU,P,NOM
$EOJ
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

REFERENCES

1. J.F.Kaiser,"Digital Filter," in Systems Analysis by Digital Computer,F.F.Kuo and J.F.Kaiser,Eds. New York : Wiley,1966,Ch.7.

2. L.R.Rabiner and B.Gold, Theory and application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 1975.

3. B.Gold and K.L.Jordan,"A direct search procedure for designing FIR filters," IEEE Trans. Audio Electroacoustics,vol.AU-17,pp.33-36, March 1969.

4. L.R.Rabiner,B.Gold and C.A.McGonegal,"An approach to the approximation problem for nonrecursive filters," IEEE Trans. Audio Electroacoustics,vol.AU-18,pp.83-106,June 1970.

5. O.Herrmann,"Design of nonrecursive digital filters with linear phase," Electronics letters,pp.328-329,1970.

6. P.W.Parks,J.H.McClellan,"Chebychev approximation for nonrecursive digital filters with linear phase," IEEE Trans. Circuit Theory, vol.CT-19,pp.189-194,March 1972.

7. E.W.Cheney,Introduction to Approximation Theory,New York : McGraw Hill,1966,pp.66-100.

8. T.W.Parks and J.H.McClellan,"A program for the design of linear phase finite impulse response digital filters," IEEE Trans. Audio Electroacoustics,vol.AU-20,pp.195-199,August 1972.

9. L.R.Rabiner,"The design of finite impulse response digital filters using linear programming techniques," BSTJ,vol.51, pp.1177-1198,July-August 1972.

10. K.Steiglitz,"Optimal design of FIR filters with monotone passband response," IEEE Trans. Acoust., Speech, Signal processing, vol.ASSP-27,pp.643-649,Nov.1979.

11. D.Kodek,"Design of optimal finite word length FIR digital filters using integer programming techniques," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-28,pp.304-308,June 1980.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

12. Y.C.Lim, S.R.Parker and A.G.Constantinides,"Finite word length using integer programming over a discrete coefficient space," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-30,pp.661-668,Aug.1982.

13. T.S.Huang,"Two-dimensional windows," IEEE Trans. Audio Electroacoustics (corresp.),vol.AU-20,pp.88-89,March 1972.

14. J.V.Hu and L.R.Rabiner,"Design Techniques for two-dimensional digital filters," IEEE Trans. Audio Elecroacoustics,vol.AU-20, pp.249-257,October 1972.

15. T.G.Stockham,Jr.,T.M.Cannon and R.B.Ingebresten,"Blind deconvolution through Digital Signal Processing," Proc. IEEE,pp.678-692,April 1975.

16. B.L.Lewis and D.J.Sakrison,"Computer Enhancement of Scanning Electron Micrographs," IEEE Trans. Circuits and Systems,vol.CAS-22, pp.267-278,March 1975.

17. M.T.Manry and J.K.Aggarwal,"Design of two-dimensional FIR digital filters with nonrectangular arrays," IEEE Trans. Acoust., Speech, Signal Processing,vol.ASSP-26,pp.314-318,Aug.1978.

18. P.K.Rajan and M.N.S.Swamy,"Design of circularly symmetric two-dimensional FIR digital filters employing transfomations with variable parametres," IEEE Trans. Acoust., Speech, Signal processing, vol.ASSP-31,pp.637-642,June 1983.

19. T.S.Huang,J.W.Burnett and A.G.Deczky,"The importance of phase in image processing filters," IEEE Trans. Acoust., Speech, Signal Processing,vol.ASSP-23,pp.524-542,Dec.1975.

20. Y.Kamp and J.P.Thiran,"Maximally flat nonrecursive two-dimensional digital filters," IEEE Trans. Circuits and Systems, vol.CAS-21,pp.437-449,1974.

21. Y.Kamp and J.P.Thiran,"Chebychev approximation for two-dimensional nonrecursive digital filters," IEEE Trans. Circuits and Systems, vol.CAS-22,pp.208-218,1975.

22. R.Mersereau,W.F.G.Mecklenbrauker, and T.F.Quatieri,Jr.,"McClellan transformation for two-dimensional digital filtering: I-Design," IEEE Trans. Circuits Syst.,vol.CAS-23,pp.405-414,July 1976.

23. W.F.G.Mecklenbrauker and R.Mersereau,"McClellan transformation for two-dimensional digital filtering: II-Implementation," IEEE Trans. Circuits Syst.,vol.CAS-23,pp.414-422,July 1976.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

VITA ACTORIS

1958 Born on 20th of June in Kamayagoundan patti, Madurai Dist., India.

1974 Obtained Secondary School Leaving Certificate at S.B.K.High School, Kalloorani, India.

1975 Completed Pre-University Course at A.J.College, Sivakasi, India.

1980 Completed Bachelor of Engineering (Electronis & Communication) at P.S.G.College of Technology, Coimbatore, India.

1982 Completed Master of Engineering (Applied Electronics) at P.S.G.College of Technology, Coimbatore, India.

1984 Candidate for the Degree of Master of Applied Science at the University of Windsor, Canada.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.