National Library
of Canada

Canadian Theses Division

Ottawa, Canada
K1A 0N4

Bibliothèque nationale
du Canada

Division des thèses canadiennes

0-315-23217-X

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Ecrire en lettres moulees ou dactylographier

Full Name of Author — Nom complet de l'auteur

[illegible]

Date of Birth — Date de naissance

[illegible]

Country of Birth — Lieu de naissance

Permanent Address — Residence fixe

[illegible]

Title of Thesis — Titre de la these

[illegible]

University — Universite

[illegible]

Degree for which thesis was presented — Grade pour lequel cette these fut presentee

[illegible]

Year this degree conferred — Année d'obtention de ce grade

[illegible]

Name of Supervisor — Nom du directeur de these

[illegible]

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.

Date

[illegible]

Signature

THE UNIVERSITY OF ALBERTA

Image Processing for Transillumination Images

by

Frank Verhagen

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

Department of Electrical Engineering

EDMONTON, ALBERTA

Spring 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Frank Verhagen

TITLE OF THESIS Image Processing for Transillumination Images

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED Spring 1985

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private scholarly or scientific research purposes only.

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.

(SIGNED) ....

PERMANENT ADDRESS

....

....

....

DATED 19

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Image Processing for Transillumination Images submitted by Frank Verhagen in partial fulfilment of the requirements for the degree of Master of Science.

....
Supervisor

....

....

....

Date ....

Abstract

Transillumination imaging is a medical imaging technique used in the detection of breast cancer. The work of this thesis consisted of developing the equipment capable of imaging a transilluminated breast and writing image processing programs to try and enhance these images. The resulting system consisted of a video camera, a digital computer, video processing boards and an RGB color monitor. This equipment was capable of performing real time digitization and display of transillumination images as well as being able to process these images to try and enhance the images that were obtained.

The equipment that was developed now requires to be tested in a clinical environment to determine its capabilities and also to suggest improvements in the current design.

Table of Contents

LIST OF FIGURES

Figure Page

LIST OF PLATES

I. Introduction

A. Transillumination Imaging

Transillumination imaging is an experimental medical examination technique for the detection of breast cancer. It is performed in a darkened room while a breast is illuminated from below or from the side with a bright light source. The illuminated breast is imaged using a photographic camera or a vidicon. A doctor then analyses the image looking for any abnormalities that may be a sign of cancer [15]

Diaphanography or transillumination imaging was first proposed as a method of diagnosing breast cancer by Cutler in 1929 [6]. Cutler's attempts to image the female breast met with failure. To capture or store his image, he had to use a high intensity light source which caused local heating of the breast. In 1972, Gros, Quenneville, and Hummel suggested some improvements [7] and in 1980, Ohlsson, Gunderson and Nilsson published results describing the effectiveness of transillumination imaging in the detection of breast cancer [15]

The aim of this thesis is to develop equipment capable of capturing, storing and displaying the image of a transilluminated breast with the use of a digital computer. The digitized images will be processed using various digital algorithms to try to improve the subjective quality of the image with the hope of improving the diagnostic capability of the images produced by transillumination imaging

B. Image Enhancement and Restoration

When one speaks of enhancing an image, the improvement in image quality is usually in regards to the human visual system. That is, the original image has been modified in some way so that analysis of the image by a human observer is made easier or more accurate. An image may be modified so that

1) the new image is more pleasing to the human eye.
2) relevant information is more easily extracted from the image.
3) noise is less noticeable or less bothersome.

Knowledge of the characteristics of the human visual system (HVS) is of use when designing image processing systems. It allows the designer to attempt to compensate for deficiencies in the spectral and spatial response of the human visual system by digitally processing the image   Some idiosyncrasies of the HVS also suggest processing methods which may improve the subjective quality of an image

A spot or object in a monochrome picture is represented as a change in grey tone or brightness (ΔI) in relation to the background intensity   The ability to perceive this ΔI depends on several factors including the size of the object the magnitude of ΔI and the relative brightness of the background   Weber s law [18] states that over a wide range of intensities the ΔI required to perceive an object of fixed size increases proportionally with an increase in background intensity . This suggests that objects in an image are more easily detected when the background intensity is of a lower value assuming the ΔI remains constant

Given an image with sinusoidal intensity variation as a function of angle the spatial frequency is defined as the number of cycles of intensity variation per degree of angle. The visual system has poor response at both high and low spatial frequencies as can be seen in Fig 1.   The poor low frequency response inhibits the visual system from detecting gradual changes in intensity especially in the presence of edges[18]   The poor high frequency response limits our ability to see fine detail since spatial frequency increases as detail becomes finer

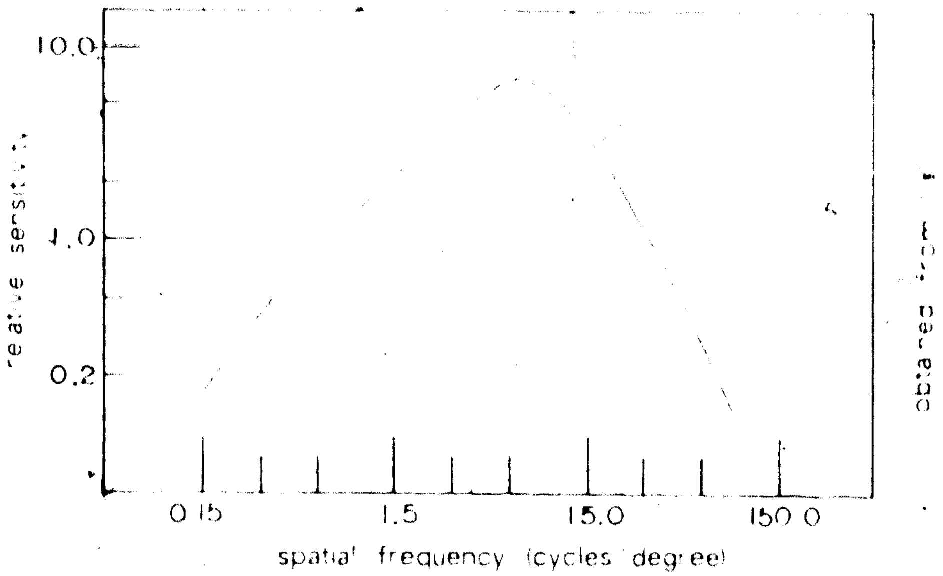
C. Image Enhancement

There are many ways to enhance the subjective quality of an image.   For example, an image with sharp edges is more pleasing to the human visual system than one with blurred edges and therefore edge sharpening improves the quality of an image.   If an image is of low contrast then the expansion of the image contrast produces quite dramatic improvements.   If an image is very noisy then a low pass filter program may produce a more pleasing image, even though the image would tend to become blurred.

D. Image Restoration

Image restoration can be defined as image enhancement using some *a priori* knowledge of the degradation in the imaging system.   If an imaging system is known to have poor high spatial frequency response, then restoration could simply be the

Figure 1



Spatial frequency response of the Human Visual System

processing of the degraded image with a filter that attempts to restore the high frequencies. If an image is known to be degraded by motion blur then restoration methods would try to reduce the effects caused by the motion.

Due to the availability of image processing hardware and its rapidly decreasing cost, there is a great amount of work being done in the field of image processing. This thesis will try to produce a number of image processing algorithms that will be of use in capturing, displaying and improving images of transilluminated breasts.

II. Experimental System

The system that was developed to allow the aquisition, digitization and processing of images consisted of the following:

1) IBM XT personal computer

2) KP-120 solid state video camera

3) IP-512 high speed video digitizer and frame buffer

4) Barco RGB monitor

5) Filter Wheel and Light Source

Plate 1 is a photograph of the experimental system. Shown in the photograph is the computer, the camera, the monitor and the card cage containing the image processing boards. A block diagram of the system is also shown in Plate 1.

A. IBM XT Computer

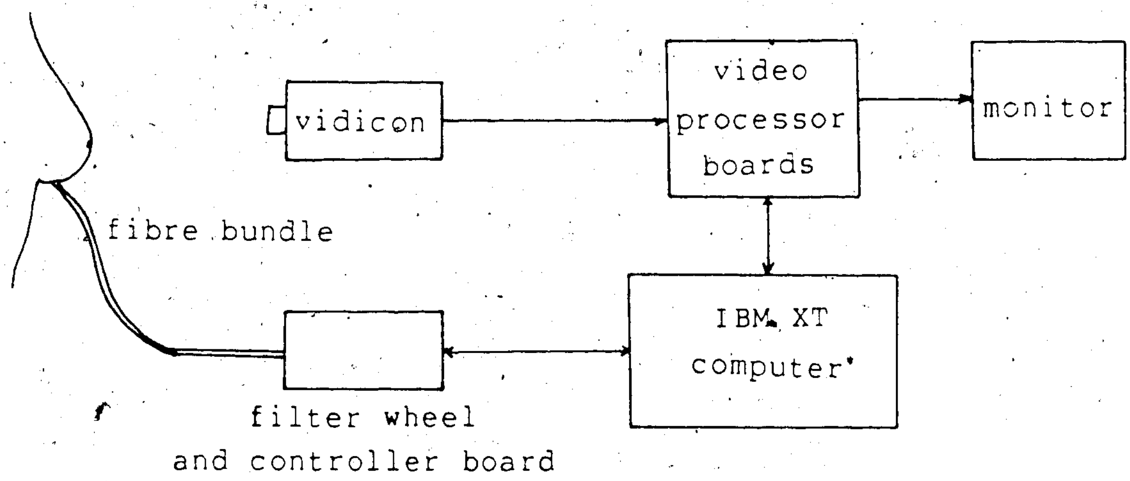
The IBM computer was installed with 512 kilobytes of memory to allow computations on large two dimensional arrays to be performed totally within the RAM memory of the computer without the need of time consuming I/O routines. An Intel 8087 math processor chip was also installed to allow floating point arithmetic calculations to be performed quickly.

The computer was also equipped with a 10 megabyte hard disk, eliminating the need of numerous floppy disks to store and process images. The hard disk also reads and writes data much faster than a floppy disk drive, greatly reducing the time required to store and retrieve images.

To achieve maximum processing speed, most of the image processing programs were written in 8088 Assembly language. Pascal language was available but it was found to have some drawbacks which eliminated many of its advantages over Assembly language programming. This project required programs to operate on very large arrays and their size could not be easily handled by,the version of Pascal that was available for the IBM computer. Use of the 8087 math chip by programs written in Pascal require Assembly language interfaces. As well as being time consuming, these interfaces eliminate many of the programming advantages of the high level language. I/O routines written in Pascal were also found to run much slower than an Assembly language counterpart. This was a

Plate 1

Photograph of experimental system



Block diagram of system

major disadvantage since many of the image processing programs are I/O intensive. The use of the 8087 also greatly simplified the development of programs in Assembly language because of its ability to perform floating point calculations and its ability to convert numbers from integer format into floating point format or vice versa. Only in a few cases was the use of Pascal found to be beneficial in these image processing applications.

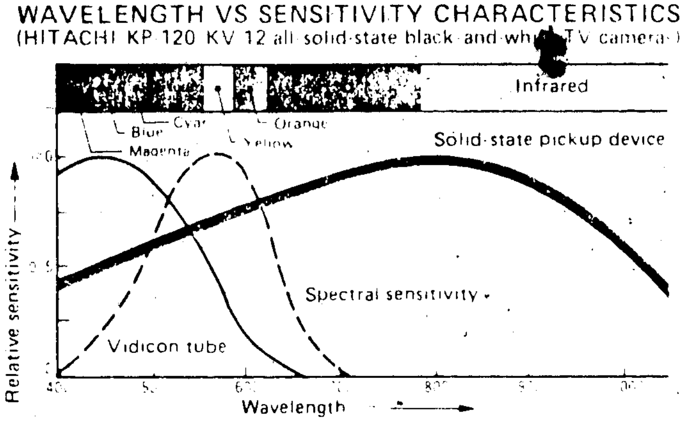
B. KP-120 camera

Most of the light emitted from a transilluminated breast is in the near infrared region of the light spectrum [15]. To capture these images, a video camera with good sensitivity at these wavelengths is required.

The video camera used in this research was an Hitachi KP-120 solid state camera. It is a black and white camera that employs a solid-state imaging device instead of a conventional vidicon tube. This camera was choosen because of its superior sensitivity versus wavelength characteristics. At near infrared wavelengths, standard vidicon tubes exhibit very poor response, whereas the sensitivity of the KP-120 peaks in this region (see Figure 2).

The KP-120 has a resolution of 240 lines by 190 lines which is adequate for diaphanography since transillumination images are of rather low resolution. The minimum illumination required by the camera is 5 lux when using an f 1.4 lens. The camera was fitted with a 20-80mm zoom lens with a minimum f-stop of 2.5. Greater sensitivity could be obtained by using a lens with a larger aperture.

Figure 2



Frequency response of the KP-120 video camera

C. Imaging Technology Video Processor Boards

The video processor boards used in this project are Imaging Technology FB-512 and AP-512 real time image processors. The AP-512 is the analog processing board. It receives the analog video signal from the video camera and converts each frame, or image, of this signal into digital numbers. This data is then sent to the FB-512 frame buffer for storage. The analog processor also simultaneously receives data from the frame buffer and converts this data into three analog video signals that are used to drive an RGB monitor.

The FB-512 is capable of storing a 512x512 image at 8 bits of data per pixel. When used for this project the AP-512 was programmed, via software, to digitize images into a 256x256 array since the KP-120 video camera has a resolution of only 190x240 lines. This allowed four images to be stored simultaneously in the FB-512.

During image acquisition, the frame buffer is continually updated with data from the analog processor. In this mode of operation, video image digitization and display are done at video rates (30 frames per second). When an image is to be captured, acquisition of new data is suspended by sending the appropriate command to the frame buffer. The data in the frame buffer is then transferred to the IBM computer for digital processing. A program that stores images in the file system of the computer was developed, allowing images to be processed and displayed at a later date.

Although the AP-512 can digitize only black and white images, it is capable of displaying images in color. The AP-512 has three D/A converters, each converter driving one of the three electron guns in an RGB monitor. From Figure 3 it can be seen that each of the three output channels on the AP-512 receives the same byte of data from the frame buffer. Between the frame buffer and each D/A converter there is a hardware look-up table. When a byte is received from the frame buffer, it addresses 1 of 256 locations in each of the three look-up tables. The data in each table, at the currently addressed memory location, is then fed into its corresponding D/A converter. This allows each gun to be sent different data even though all three channels receive the same byte from the frame buffer.

The data in each memory location of each look-up table is loaded by the computer during initialization of the video processor boards. In this manner, the AP-512 can be programmed to display a certain color, or hue, depending on the value of the byte received from the frame buffer. For example, if a full intensity blue is to be displayed on the RGB monitor when a byte of value 34 is received from the frame buffer, the following would be done

memory location 34 of red table loaded with number 00
memory location 34 of green table loaded with number 00
memory location 34 of blue table loaded with number 256

If a half intensity yellow is to be displayed when a byte of value 56 is received, then

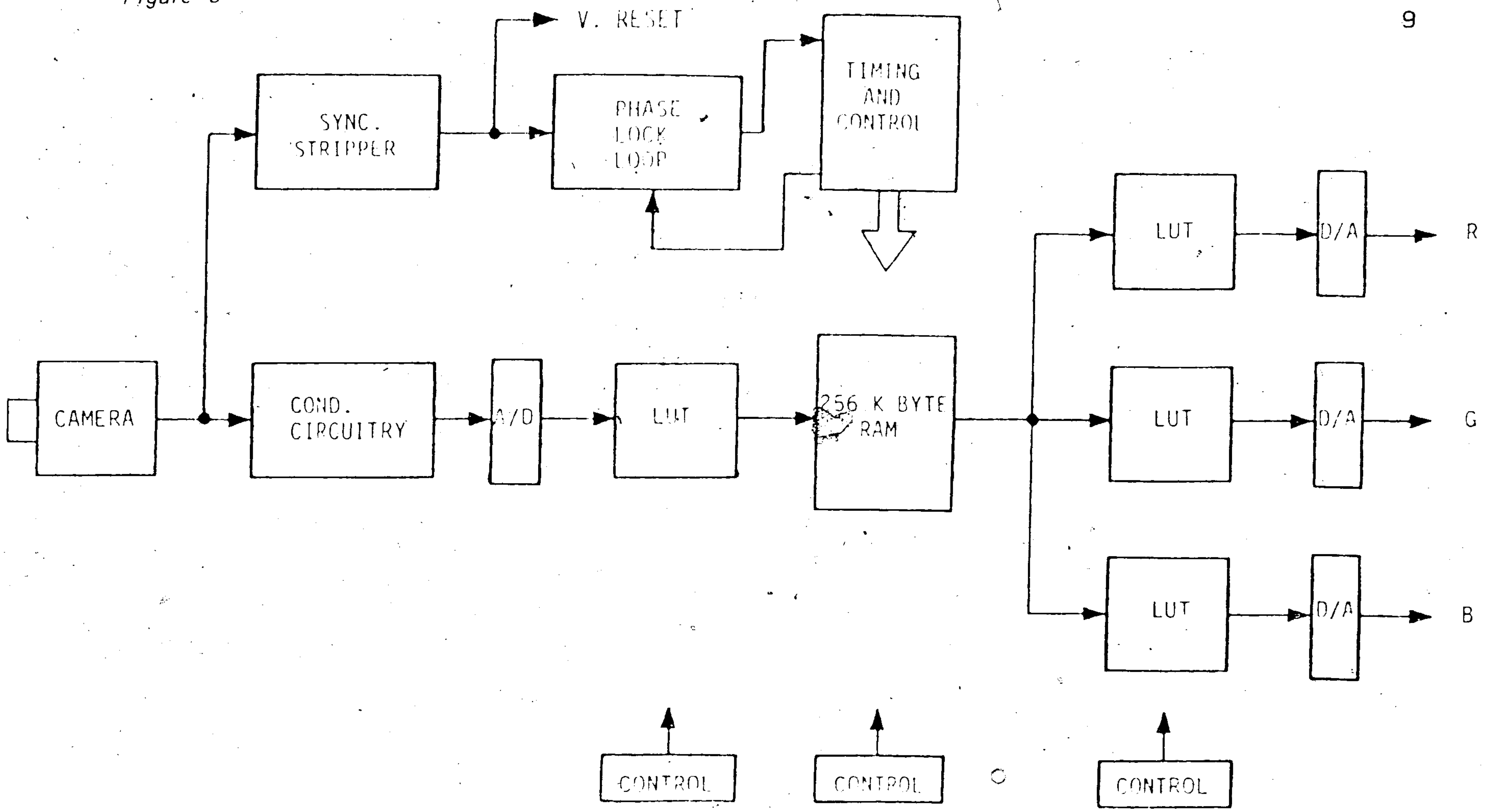
memory location 56 of red table loaded with number 128
memory location 56 of green talbe loaded with number 128
memory location 56 of blue table loaded with number 00

Obviously, only 256 different colors or hues can be displayed at one time since the data from the frame buffer is 8-bits wide.

D. Filter Wheel and Light Source

The acquisition of an image of a transilluminated breast requires a light source to illuminate the breast of a patient. A light guide or fibre optic bundle was chosen to deliver the light to the breast because it is light, flexible and fairly inexpensive. Large diameter fibre bundles were used to allow intense light to be coupled into the fibre. The light source that was used to illuminate the fibre bundle was a 150W projector lamp with a built-in parabolic reflector. This reflector concentrates most of the available light into a

Figure 3



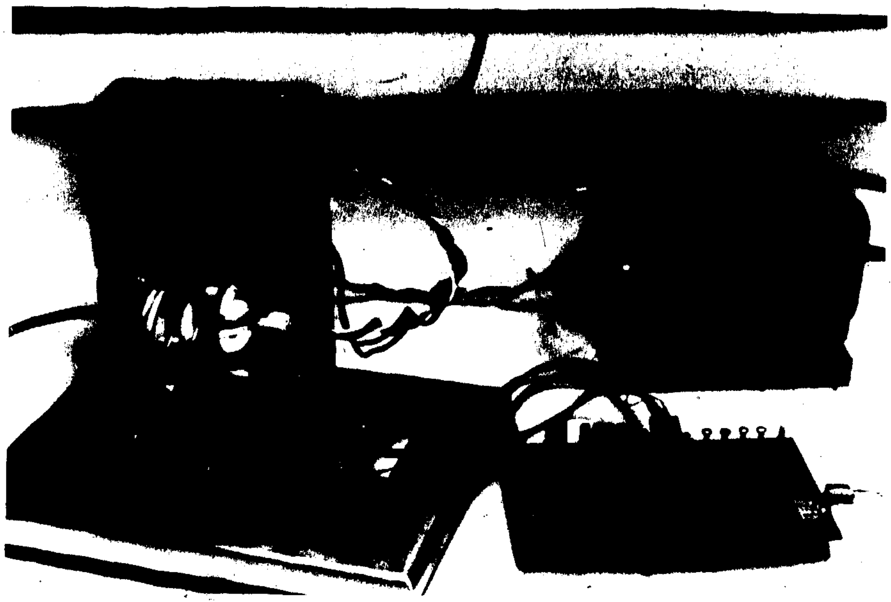
Block Diagram of Image Processing Boards circular region having a diameter of about 1 centimeter, allowing approximately 10% of the available light to be imaged onto a 0.3cm. fibre. The actual amount of light coupled into the fibre is much less due to the acceptance angle of the fibre and the fact that part of the cross-sectional area of the fibre bundle is due to cladding. This cladding does not transmit light and any light imaged onto it is simply lost. Despite the high coupling loss, an intense light that was adequate for initial experiments emerged from the opposite end of the fibre bundle.

Greater light intensities can be obtained with the use of a light source having a small and very intense source region (an arc lamp). This small source could then be imaged onto the fibre bundle using an elliptical reflector to allow a higher percentage of the available light to be captured.

The use of an arc lamp as a light source was not attempted because it would have required a major effort to install. Arc lamps emit ozone and must be ventilated when they are used. The elliptical reflector would have to be cooled extremely well since a great amount of heat would be generated by the lamp. Arc lamps also require a more sophisticated power supply than the projector lamp, which merely requires a 21V transformer.

To obtain near infrared color images, the ability to illuminate the breast with light of different wavelengths, or color, is required. To accomplish this, a filter wheel was placed between the projector lamp and the fibre bundle. Three bandpass optical filters, with center frequencies in the near infrared spectrum, were mounted in the wheel and the filter wheel was rotated using a small electric motor. As each filter comes into position between the lamp and the fibre bundle, the breast is illuminated by light with a spectrum that depends on the properties of the filter. As the filter wheel spins, the breast is successively illuminated by three different spectrums of light, allowing three images of the same breast to be obtained quickly. These images can then be digitally combined to produce one infrared color image.

Plate 2



Photograph of filter wheel and light source.

III. Fourier Transforms

The two-dimensional Fourier Transform of an image can be used to perform many useful image processing algorithms. Filtering in the Fourier domain is merely the multiplication of the Fourier Transform of an image with the desired frequency response and then obtaining the inverse transform of the result. The convolution of two images can be calculated by inverse transforming the product of their Fourier Transforms. Deconvolution is also theoretically quite simple in the Fourier domain.

The calculation of the Fourier Transform of an image requires a large number of complex floating point calculations. Until the development of the Fast Fourier Transform (FFT) in 1965 by Cooley and Tukey [5], Fourier transforms of large arrays were not practical to implement on a computer, much less on a microcomputer (if they had existed). Even using FFT methods, the calculation of two-dimensional Fourier transforms of large arrays requires a great number of calculations.

The discrete Fourier Transform of a one dimensional array of numbers is defined as

$$X(k)=\sum_{n=0}^{N-1} x(n)\exp\{-j(2\pi/N)nk\} \qquad k=0,1,\ldots,N-1 \qquad (1)$$

The Fast Fourier transform is obtained by successively breaking down equation (1) into smaller parts. This technique is described in many texts and will not be mentioned here. A one-dimensional FFT algorithm, taken from [17] is listed in Table 1. A two-dimensional FFT is performed by first calculating the one-dimensional FFT of each row in the image. The one-dimensional FFT of each column is then calculated, resulting in the two-dimensional Fourier transform of the image.

A two-dimensional FFT program, written in Assembly language, was developed for the IBM XT computer to process images obtained from the video processor boards. This FFT program requires approximately 65 seconds to calculate the Fourier transform of a 128x128 image. During the development of the Assembly language FFT program, some modifications were made to the FORTRAN language FFT program shown in Table 1. These changes are listed below.

1. To obtain a result with the data in the correct order, an FFT program must shuffle either the input data or the output data in a bit reversed order. For example, if an

FFT is being done on 128 points of data, data at binary location 0101000 (location 40) must be exchanged with the data at location 0001010 (the bit reversed location of the data). This must be done for all the data in the array. The program in Table 1 accomplishes this using a fairly complex algorithm. The Assembly language FFT program written on the IBM shuffles data using a look-up table. This table is used to determine where each pixel of data must be moved, resulting in a much faster shuffle speed.

2. The calculation of the sine and cosine of a number using the 8087 is rather slow and involves a number of program steps. To bypass the use of the 8087 for the calculation of the sine and cosine function, the FFT program uses a sine and cosine look-up table. This is practical since only 14 different sine and cosine values are required to perform a 128 point one-dimensional FFT using the algorithm shown in Table 1. Since this 128 point FFT program is called 256 times during the calculation of a 128x128 two-dimensional FFT, a saving in execution time is realized. The sine and cosine values stored in the look-up table are used to obtain additional sine and cosine values by repeated multiplications (see Table 1) therefore the sine and cosine look-up table uses a double precision format.

A complex number requires two real numbers to define both its real and imaginary parts. The FFT program written for the IBM, stores real numbers in single precision floating point format, requiring 4 bytes of memory per number. Therefore eight bytes of memory are required to store a complex number. The Fourier transform of a 256x256 image contains over 65,000 complex numbers and would require over 500k bytes of memory storage. The IBM computer cannot easily meet this memory requirement and as a result images are reduced to a size of 128x128 before being Fourier transformed. This image size requires a total of 128k bytes of memory storage per Fourier transform.

For the implementation of FFT programs on the IBM microcomputer, an Intel 8087 math processor chip was used to increase the execution speed. This chip is designed to perform floating point arithmetic calculations. Consequently it contains data registers capable of handling single and double precision floating points numbers. It can calculate most of the standard arithmetic functions in one instruction step and executes its calculations much faster than floating point calculations performed by the general purpose 8088 microprocessor.

The math processor also increases the speed of the FFT program by converting between number system formats internally. Integers are converted to real numbers and real numbers to integers with the use of the 8087 math chip. Images obtained from the video processor boards are in the form of 8-bit integers. These 8-bit integers are converted into 16-bit integers using the 8088 and subsequently stored in memory. The conversion of these integers into a real number format is accomplished by loading an integer to the math processor using a *load integer* command and then storing the number back into memory using a *store real number* command. The conversion of the number from one format to another is done automatically by the 8087 eliminating the need of software conversion routines.

A one-dimensional FFT calculation on an array of length N requires approximately $(N/2)\log_2(N)$ complex multiplications and $N\log_2(N)$ complex additions and subtractions. A two-dimensional FFT calculation on an array of dimension NxN requires a one-dimensional FFT on each row and column in the array, resulting in approximately $N^2\log_2(N)$ complex multiplications and $2N^2\log_2(N)$ complex additions and subtractions. This does not include the additional complex calculations required to update the sine and cosine values in each stage of the FFT.

Using the previous equations, performing an FFT on an array of 128x128 points requires over 100,000 complex floating point multiplications and over 200,000 complex floating point additions and subtractions. Each complex multiplication requires 4 real number multiplications and 2 real number additions while each complex addition or subtraction requires 2 real number additions or subtractions. This results in a total of over 400,000 real floating point multiplications and over 600,000 real floating point additions and subtractions. Additional instructions are also required to perform the other requirements of the FFT calculation (i.e. loop control, data transfer etc.).

The Assembly language FFT program written for the IBM performs a 128x128 two-dimensional Fourier transform in approximately 65 seconds. Considering the number of calculations required, this is quite a respectable time. Included in this performance figure is the time required to transfer the 256x256 image from the hard disk into the system memory, the reduction of the array from 256x256 to 128x128, conversion of the integer array into a complex floating point format, calculation of the 128x128 Fast Fourier transform and finally, the storage of the resulting complex floating point array back onto the hard disk of the computer.

Additional programs were also written that perform various functions on these complex floating point arrays. The inversion of each point in an array, multiplication of corresponding points in two arrays and the calculation of the log magnitude of each complex number in the FFT array are some of the functions that were implemented. The combination of multiplication and inversion enabled deconvolution of an image to be attempted. Calculation of the log magnitude of the Fourier transform of an image allows the frequency spectrum of the image to be displayed on a video monitor useful in observing the effects image processing routines have on the frequency content of an image. Filtering in the Fourier domain is performed by multiplying the Fourier transform of an image by a suitable function. The result is then inverse transformed to obtain the filtered image.

The inverse Fourier transform of a one dimensional array is defined as

$$x(n)=(1/N)\sum_{k=0}^{N-1} X(k)\exp\{-j(2\pi/N)nk\}$$

and can be computed using FFT algorithms [17]. This is accomplished by first complex conjugating the array to be inverse Fourier transformed and then performing an FFT algorithm on it. The result is then complex conjugated once more to obtain the inverse Fourier transform of the original array. This is also valid for two-dimensional arrays.

The inverse Fourier transform program developed for this project stores the real and imaginary parts of the inverse Fourier transform in different files so that the real and imaginary parts can be displayed separately on a monitor. Since the original image consists of only real numbers, the inverse transform of a modified transform array should produce a real result. In practice, the imaginary part was usually very small and in most cases was ignored.

It is of interest to determine the amount of round-off error that results when an image is Fourier transformed and then inverse Fourier transformed. This was done on an image and the total squared error between the original image and the result was calculated using the equation

$$E = \sum_{x=1}^{128} \sum_{y=1}^{128} [\mathrm{orig}(x,y)-\mathrm{result}(x,y)]^2$$

The total squared error that resulted was of the magnitude of 2^{-20} or 10^{-6}. This is very small and demonstrates that this FFT program has negligible round off error.

Table 1 One dimensional FFT algorithm

```
      SUBROUTINE FFT(A,M,N)
      COMPLEX A(N),U,W,T
      N = 2**M
      NV2 = N/2
      J = 1
C
C     Due to the nature of the FFT algoritm either the data
C     used by the FFT or the data output by the FFT must be
C     shuffled in order to obtain the proper order in the output
C     data.  In this FFT algorithm the input data is shuffled
C     prior to the execution of the FFT
C
      DO 7 I=1,NM1
      IF(I.GE.J) GO TO 5
      T=A(J)
      A(J)=A(I)
      A(I)=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
C
C     The data has been shuffled and the FFT program now
C     follows
C
      PI=3.14159265358979З
      DO 20 L=1,M
      LE=2**L
      LE1=LE/2
      U=(1.0,0.0)
      W=CMPLX(COS(PI/LE1),SIN(PI/LE1))
      DO 20 J=1,LE1
      DO 10 1=J,N,LE
      IP=I+LE1
      T=A(IP)*U
      A(IP)=A(I)-T
10      A(I)=A(I)+T
20      U=U*W
      RETURN
      END
```

IV. Filtering

A. Filtering by Spatial Convolution

The convolution of two arrays in the spatial domain is defined by the algorithm

$$g(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n) \qquad (3)$$

If an array of size mxm is convolved with an array of size nxn the result of the convolution will be of size (m+n)x(m+n). A program was developed on the IBM XT to convolute a large array (an image) with a smaller array (an enhancement function). Instead of producing an output array of size (m+n)x(m+n), this program was designed to produce an output array of the same size as the image array, keeping it compatible with the other image processing programs. With an enhancement array of size nxn, n/2 rows of information-are lost along each edge of the enhanced image when using the convolution program.

In this project, spatial convolution usually consisted of an image array being convoluted with a smaller enhancement function. The size and contents of the enhancement function array is dependant on its purpose. For example, a high pass filter function usually requires a larger array than a low pass filter or edge detection function would need.

The spatial convolution of two large arrays requires an enormous number of calculations. If the function array is of size nxn then n^2 multiplications and additions are required to calculate each point in the convoluted array. Obviously, the size of the filter function must be kept small to obtain reasonable execution times.

The design of an enhancement function is most easily done in the Fourier domain. It is well known that the response of a system to an impulse is simply the transfer function of that system. A filter can therefore be designed to have a certain frequency response by setting its impulse response to the desired frequency response. This can be accomplished by designing in the Fourier domain. To start the design, a filter function having a gain of 1.0 at all frequencies will be built. This is simply an impulse function in the spatial domain. The Fourier transform of the impulse function is calculated and produces a filter function in the Fourier domain that passes all frequencies equally with a gain of unity. The array is then modified to exhibit the desired response of the filter and then transformed back into the spatial domain to yeild the required filter function.

A filter array designed in the Fourier domain is of the same dimension as the images to be filtered and is therefore much too large to spatially convolve with an image array. The filter function must be reduced in size by using a window function. A window function sets all the data that is more than a certain distance away from the origin of the filter array to zero. To smooth the data along the edges of the reduced array, a window function will usually scale the data near the cutoff region (see Fig 4).

A program to convolve an image array with a smaller filter array was developed and its speed was compared to convolution in the Fourier domain for various sizes of filter functions. When the convolution of two arrays is performed in the Fourier domain, the image array and the filter array are of the same size. In this project, images are reduced to 128x128 before they are transformed into the Fourier domain. At this image size, a complete convolution requires approximately 150 seconds. This consists of Fourier transforming an image, multiplying the transformed image with the filter function and then inverse Fourier transforming the result to produce the processed image. This convolution speed is constant for all desired filter functions since all filter arrays must be padded to a size of 128x128 before convolution.

The spatial convolution program written on the IBM XT uses image arrays of size 128x128 to allow valid execution speed comparisons between spatial and Fourier convolution. The filter function size, on the other hand, is variable. By calculating the time required to perform each instruction in the convolution program, its execution speed was estimated to be:

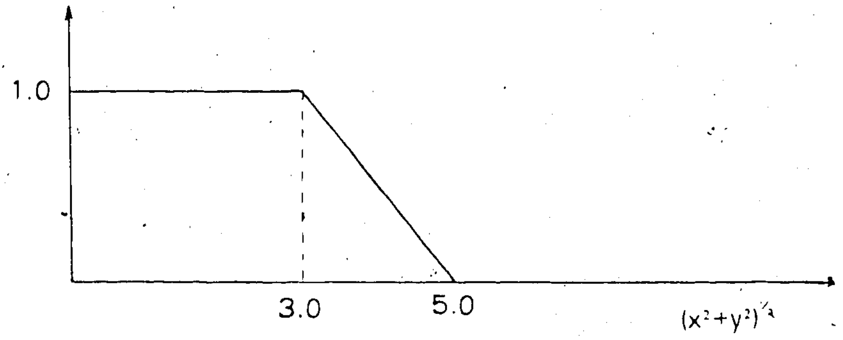
$$\text{speed}= (128)^2 n^2 (76\times10^{-6}) \text{ seconds} \qquad (4)$$

where the size of the filter array is nxn.

For filter functions of size less than 13x13, spatial convolution was faster than Fourier convolution, while for larger filter sizes, convolution in the Fourier domain was faster. For this reason, all filter functions had to be windowed to a size smallier than 13x13 before performing spatial convolution in order to realize an advantage over convolution in the Fourier domain.

Reducing the size of a filter array with a window function will usually modify its frequency response. It is therefore useful to observe the effect of windowing on the frequency response of a filter. By looking at the Fourier transform of the filter before and after windowing, the change in the response of the filter can be observed. Plate 3 shows the response of a filter before and after windowing with the function shown below.

Figure 4



As can be seen from Plate 3, windowing can appreciably change the response of a filter.

B. Fourier Domain

Filtering is easily performed in the Fourier domain by multiplying the Fourier transform of an image with a suitable two-dimensional function. This two-dimensional function must merely be of the same shape as the desired filter. For example, an ideal high pass filter in the Fourier domain is of the form:

$$f(u,v)=1 \text{ for } u^2+v^2>b$$

$$f(u,v)=0 \text{ for } u^2+v^2\leq b$$

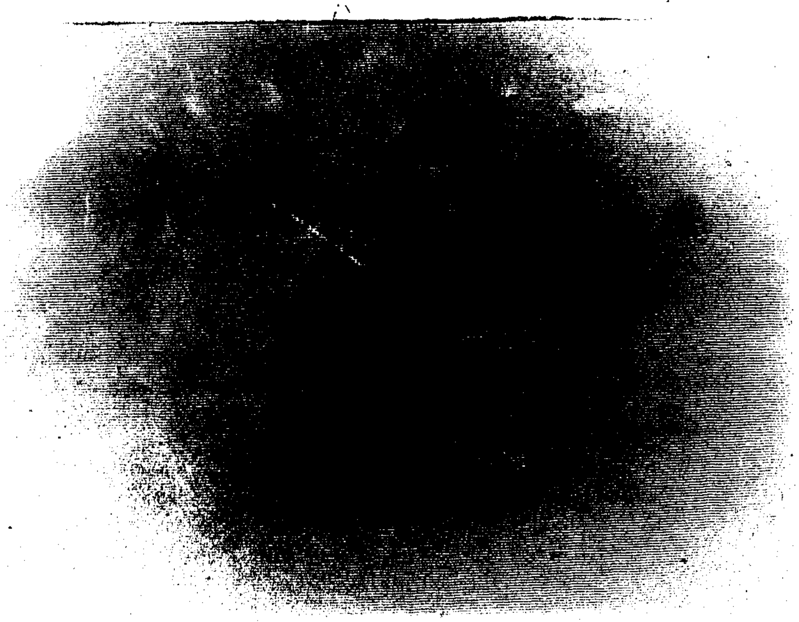
where f(u,v) is the filter function and b is the cutoff frequency. The filtering algorithm is then

$$i(u,v)=o(u,v)f(u,v) \qquad (5)$$

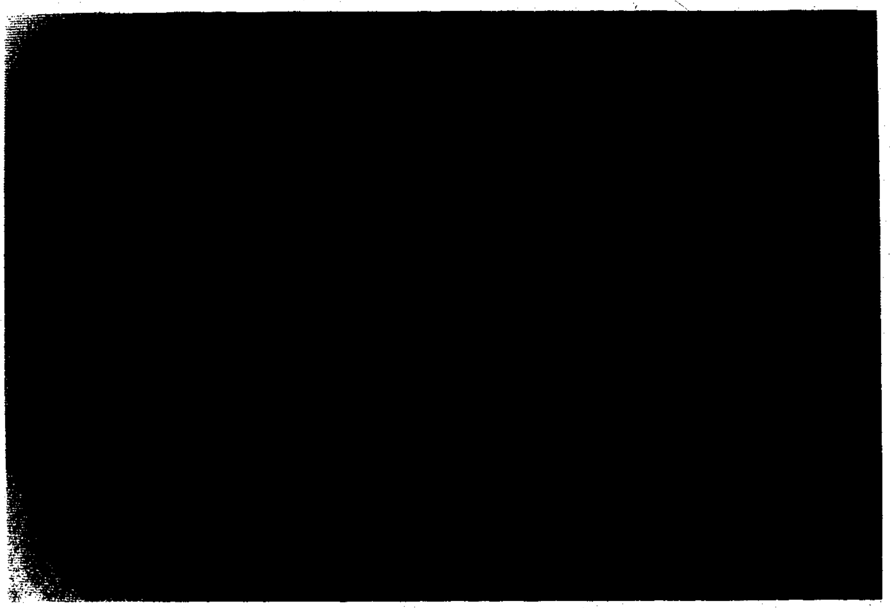
The filtered image is then obtained by inverse transforming the result.

Filtering in the Fourier domain can produce ringing in the output image if the filter function is not smooth and continuous at all points. This is demonstrated in Plate 4a by

PLATE 3



Original filter response



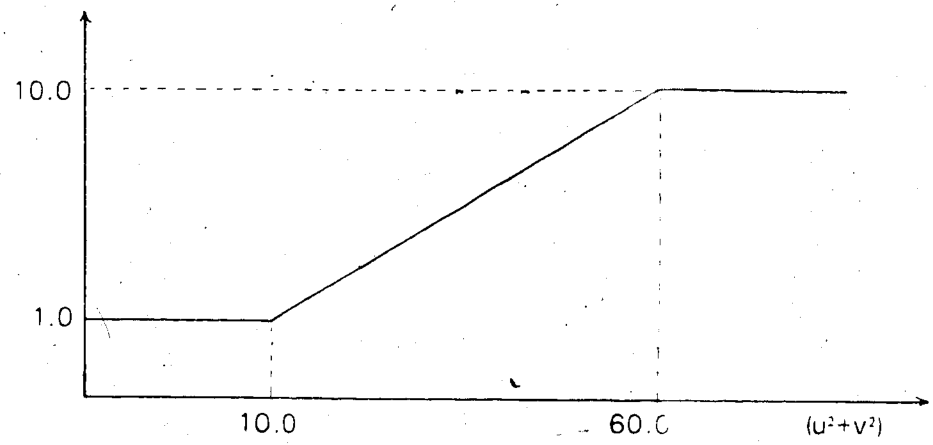
Filter response after windowing

an image that has been filtered with a filter function of the form

$$f(u,v)=1 \text{ for } u^2+v^2 \leq 10.0$$

$$f(u,v)=5 \text{ for } u^2+v^2 > 10.0$$

Ringing can be reduced by applying a filter to an impulse function and then observing the amount of ringing produced by the filtering. If any excessive ringing occurs, it can be reduced by limiting the extent of the ringing in the space domain using a suitable window function. This windowed filter function is then transformed into the Fourier domain and used as the new filter function. It must be remembered that windowing tends to smooth the response of a filter and therefore a windowed filter will usually differ from the original design. Plate 4b shows an image that has been modified with a filter of the form



Filter used to obtain Plate 4b.

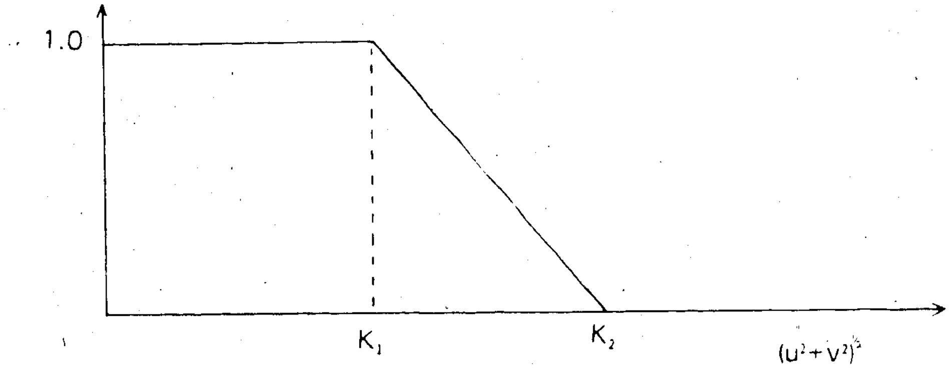
The image observed in Plate 4c has been filtered with a windowed version of the filter which modified Plate 4b. The extent of the impulse response of the filter was limited by a window function to a maximum of five pixels from the origin of the impulse. The resulting effect of the windowed filter is different than the original filter. The response of the filter after windowing can be observed in the Fourier domain.

To filter an image using Fourier techniques required approximately 150 seconds. An FFT and inverse FFT calculation must be performed as well as the actual filtering of the transformed image. Two filter programs designed to operate in the Fourier domain were developed for the IBM computer. One program was a low pass filter which had

Plate 4a

Ringing in image produced by non-smooth filter

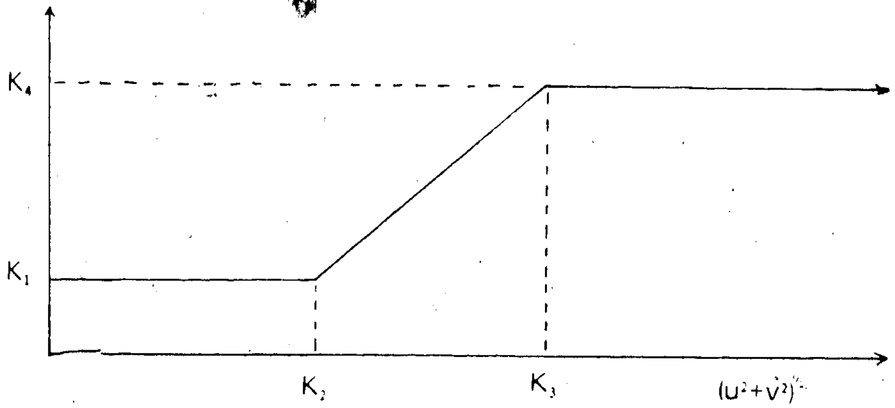
the response shown below.



Frequency response of low pass filter program.

The constants K_1 and K_2 are entered when the program is called to allow for variable cutoff frequencies and slope rates. The low pass filter was not extremely useful for this project since it is mostly effective on noisy images.

A high pass filter was also developed. The shape of the filter is shown below.



Frequency response of high pass filter program.

The constants K_1 to K_4 are entered when the program is run, enabling the filter shape to be altered. K_1 determines the amount of low frequency attenuation, K_2 and K_3 determine the cutoff frequency and also the slope of the cutoff region. The value of K_4 determines the amount of high frequency boost. After the filter has processed the Fourier transform, the inverse transform is then calculated to yield the enhanced image.

Plate 4b and 4c

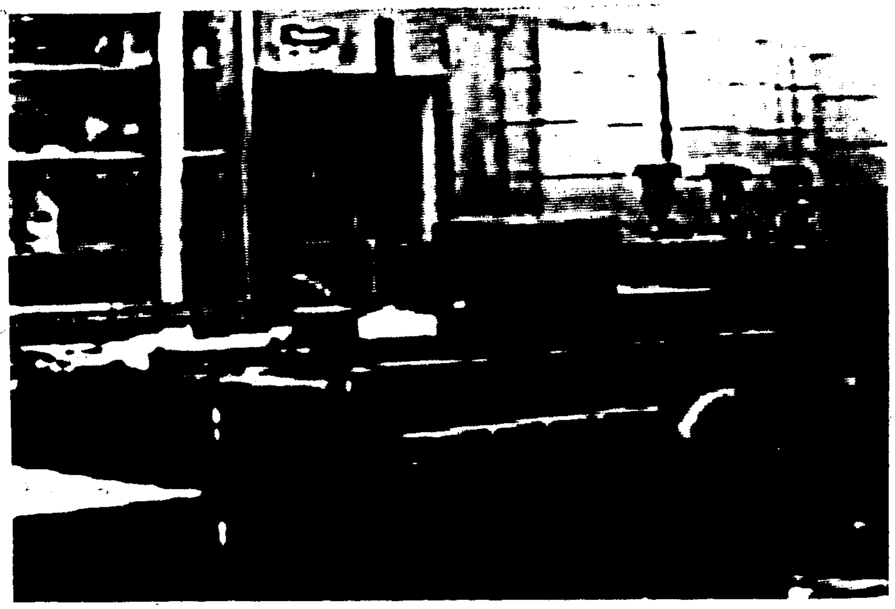


Plate 4b: Image filtered by non-windowed function

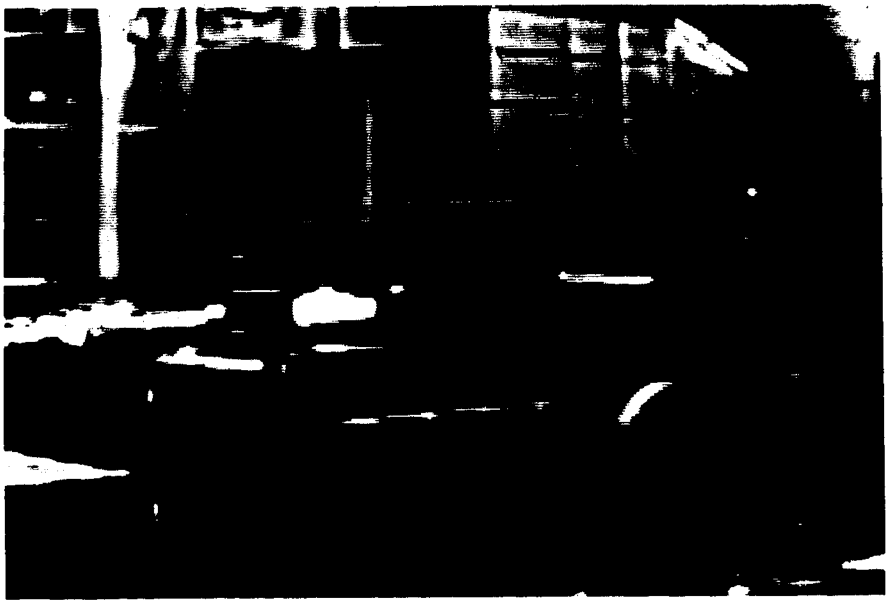


Plate 4c: Image filtered with a windowed version of function

V. Edge Sharpening

The subjective quality of a slightly blurred image, or an image without distinct edges, can be greatly improved by the use of edge sharpening algorithms. The human visual system is much more responsive to an abrupt edge than to an edge which gradually changes from one grey level to another [18]. Images with sharp edges tend to be more pleasing to the eye and therefore easier to analyze. There are many techniques used to enhance the edges in an image and some of these will be discussed and demonstrated in this chapter.

A. Derivative

Edges in an image can be detected by calculating the partial derivative of the image. The partial derivative df_{xy}/dx will detect vertical edges while df_{xy}/dy will detect horizontal edges. Unbiased detection of all edges requires a rotationally invariant algorithm. It can be shown that $df_{xy}/dx + df_{xy}/dy$ is not rotationally invariant. It can be made rotationally invariant by squaring both parts of the equation before the summation, thus

$$(df_{xy}/dx)^2 + (df_{xy}/dy)^2 \qquad (6)$$

While this algorithm will detect and display the edges of an image, the subjective quality of the derivative image is very poor. Since only edges are present, much of the information in the original image is lost. The addition of the derivative of an image to the original image will restore the lost information but the edges will not be sharpened effectively. This effect is demonstrated in Figure 5.

It can be shown that the Laplacian operator

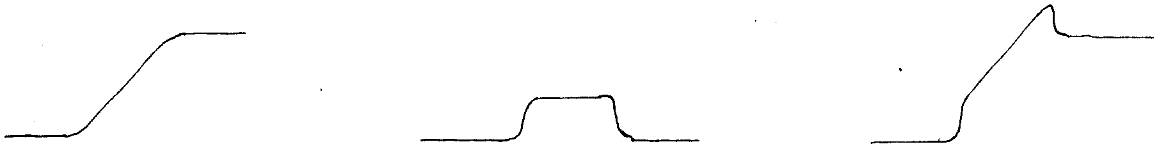
$$\nabla^2 = d^2f_{xy}/dx^2 + d^2f_{xy}/dy^2 \qquad (7)$$

is rotationally invariant.

The addition of the Laplacian of an image to the original performs edge sharpening while preserving the natural quality of the image. This algorithm produces overshoots on both sides of an edge causing the HVS to perceive the edge as being sharper. This effect is demonstrated in Figure 6.

The degree to which edges are sharpened by the Laplacian algorithm is controlled by scaling the value of the Laplacian before adding it to the original image. These edge

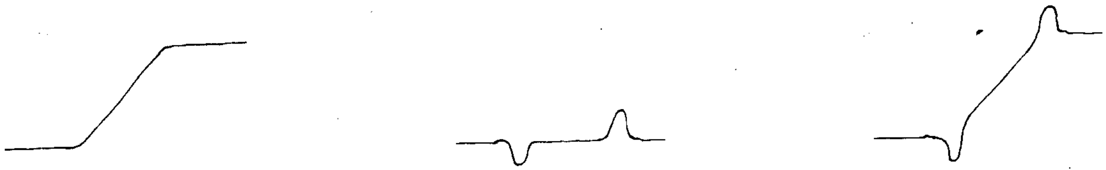
Figure 5



Original Edge   Derivative of edge   Sharpened Edge

Edge sharpening with derivative algorithm

Figure 6



Original Edge   Laplacian of edge   Sharpened Edge

Edge sharpening using Laplacian algorithm

sharpening techniques enhance the high frequency content in an image and therefore also enhance any high frequency noise present in the original image.

A program which calculates the Laplacian of an image and then adds the result to the original image was developed and tested on the experimental system. Plate 5 demonstrates the improvement that can be obtained with this algorithm. The algorithm was also applied to the image of a transilluminated thumb. The result, shown in Plate 6, is not terribly impressive. To sharpen the appearance of the thumb, a great deal of emphasis had to be applied to the edge information. This amplified the noise to such a degree that it started to overwhelm the data and became noticeable before much improvement was obtained in the sharpness of the image.

This algorithm was then applied to an enhanced version of the transilluminated thumb. The original image was processed with a filter designed to improve the contrast in all areas of the image. The enhanced image was then sharpened with the Laplacian algorithm. As can be seen from Plate 7, the algorithm did improve the subjective quality of the image.

For certain types of images, an improvement in the Laplacian algorithm can be obtained by squaring the magnitude of the Laplacian at each point, while still preserving the original sign of the number. This has the effect of increasing the weight of the more distinct edges in relation to the fainter ones. Since the magnitude of the noise is usually small, the more pronounced edges are amplified more than the noise.

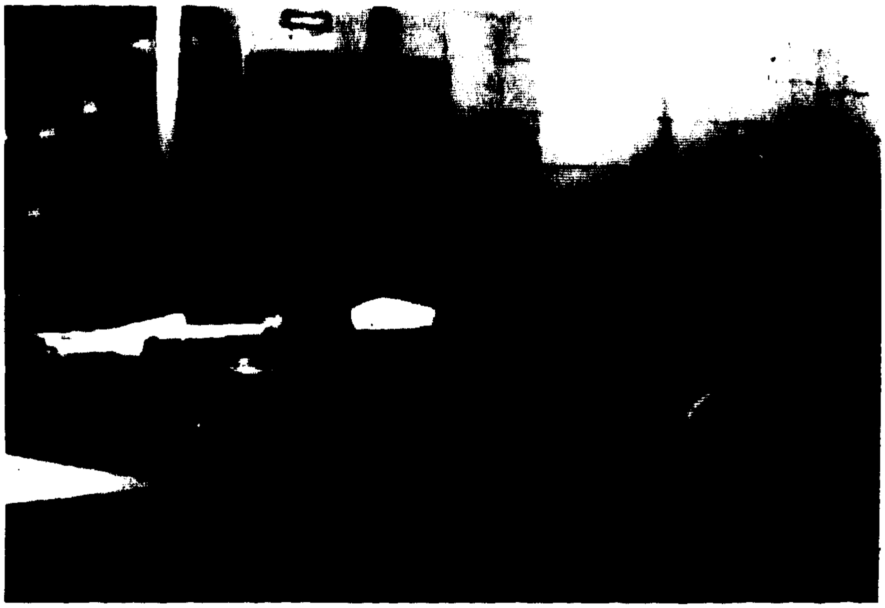
B. Edge sharpening by filtering

Since the edge information in an image is contained in the high frequencies, edges in an image can be sharpened by boosting the high frequency content of the image with the use of a filter. This was attempted by filtering in the Fourier domain.

If an ideal high pass filter is applied to an image, the resulting image contains no low frequency content. This produces an image consisting only of edges and points, giving it a washed out appearance. A more pleasing effect is obtained by leaving the low frequency content intact and boosting the high frequencies, resulting in a natural but sharper image.

Various high frequency boost functions were applied to a number of images. The improvement in edge sharpness produced by this method was not as impressive as the improvement obtained with the Laplacian algorithm that was mentioned earlier. Plate 8 shows an image which has been sharpened with a high boost filter. The amount of time required to perform the filtering was quite long since it was done in the Fourier domain. The Laplacian executed much faster and produced a better result on all the images that were processed.

Plate 5



Original image

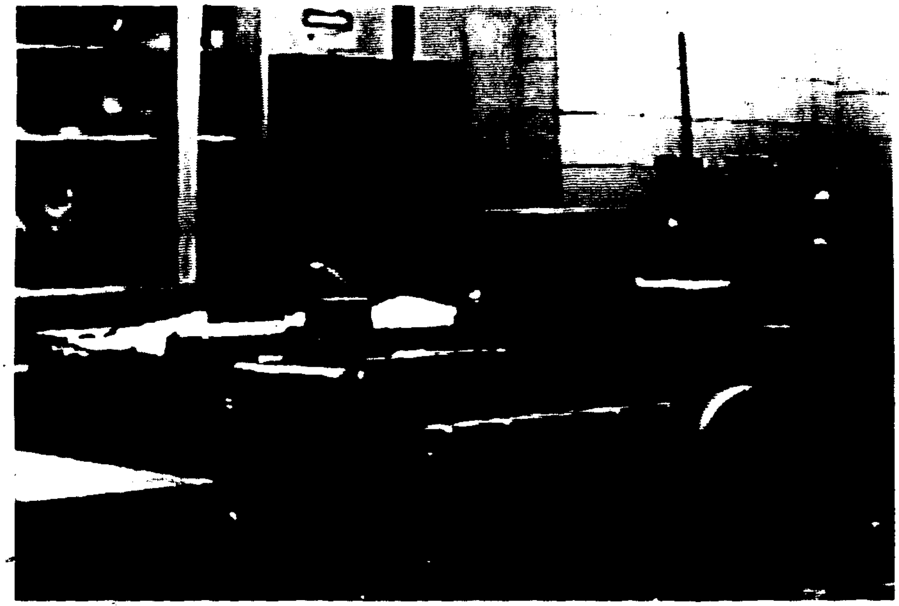
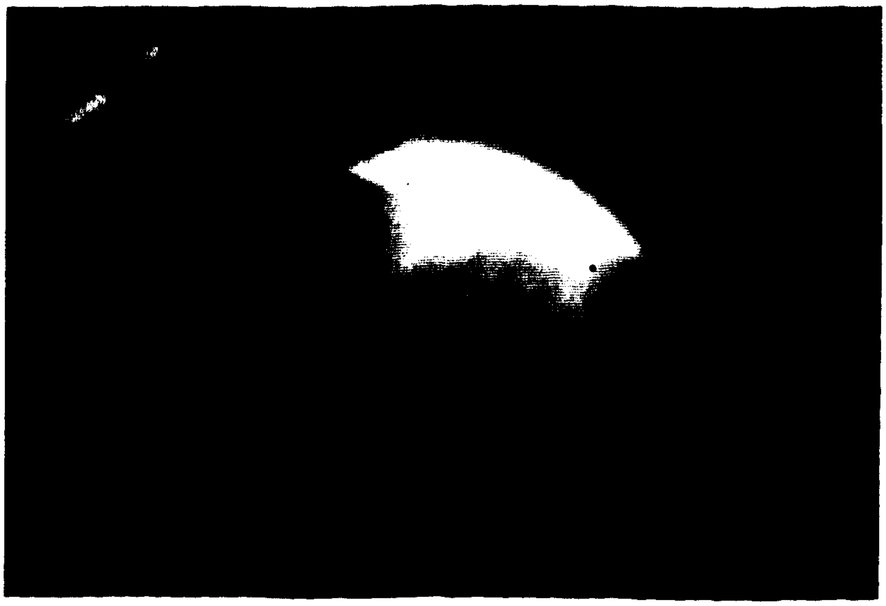


Image after laplacian enhancement

Plate 6



Transilluminated thumb

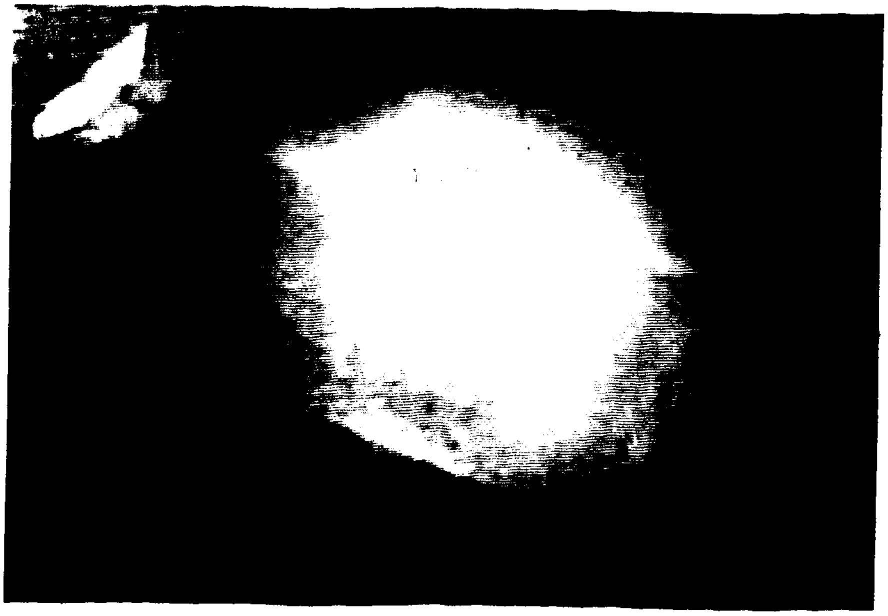
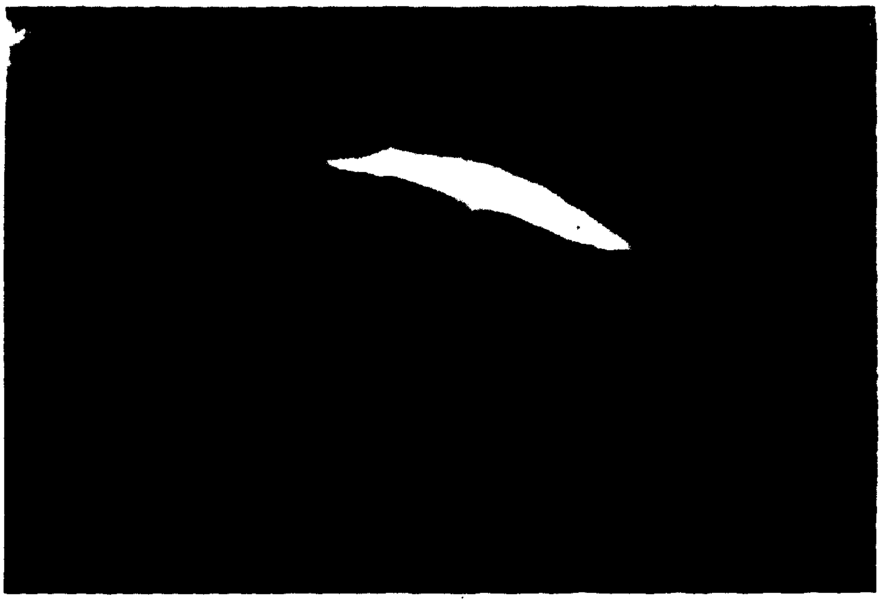
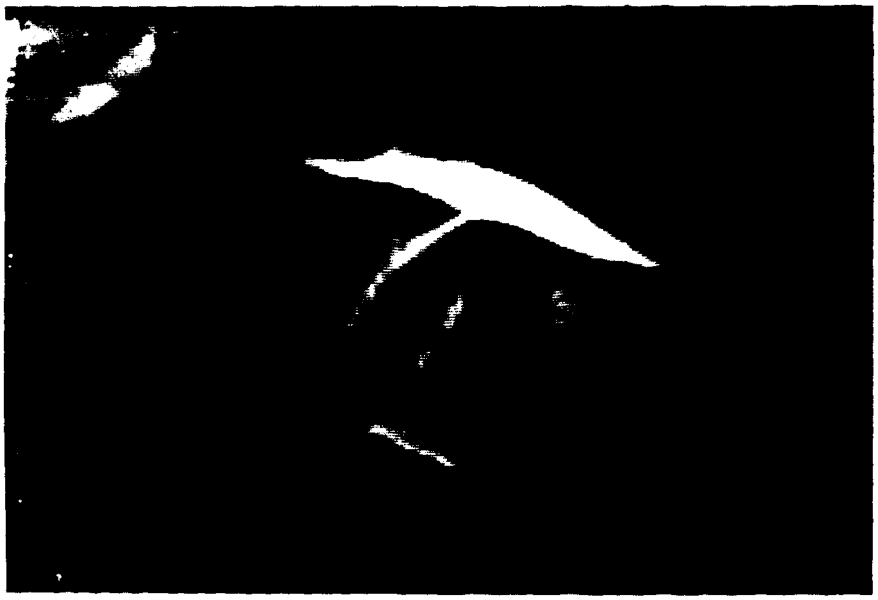


Image of thumb after laplacian enhancement

Plate 7



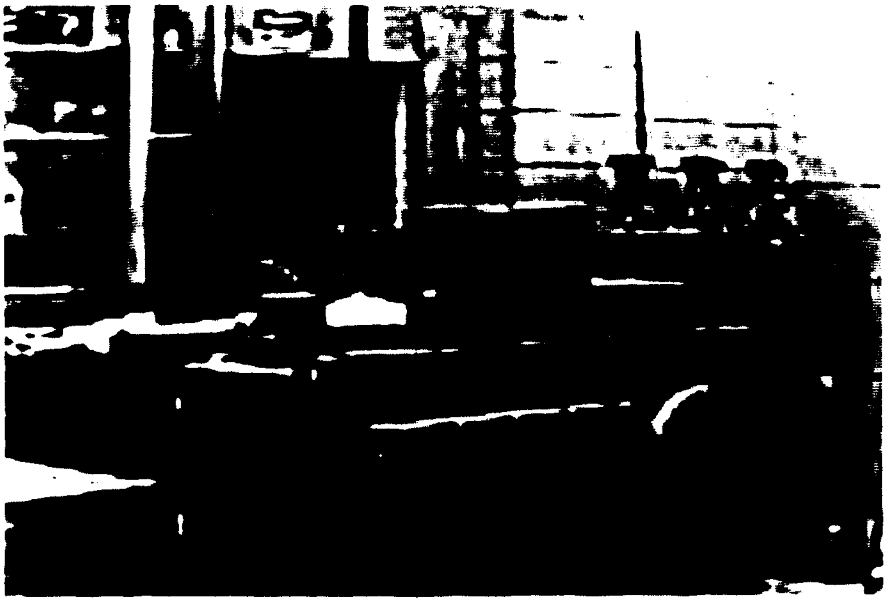
Enhanced image of transilluminated thumb



Edge sharpened version of the top image.

Plate 8

Top: original image



Bottom: Image enhanced with high boost filter

VI. Contrast Enhancement

A. Grey Scale Modification

A digital imaging system maps an object or scene into an array of digital numbers. The ideal system is linear and maps each point in the scene into a point in the array with the transfer function:

$$i(x,y)=k*o(x,y)$$

where $o(x,y)$ is the intensity of the scene at the point (x,y) and k is the mapping constant.

If a scene is of low contrast, then the image mapped into *i* will not use the full dynamic range available and improvements in the image can be obtained by using contrast enhancement techniques.

The histogram of an image denotes the number of times each possible pixel level occurs in the image. Images that are of low contrast will exhibit a histogram with most of the pixels concentrated over a small part of the available range. Dramatic improvements in image quality can be gained by modifying the pixels in the image so that its histogram is *flattened*. This is accomplished by changing the value of each pixel in the image using a suitable mapping function.

$$e(x,y)=h(x,y)g(x,y)$$

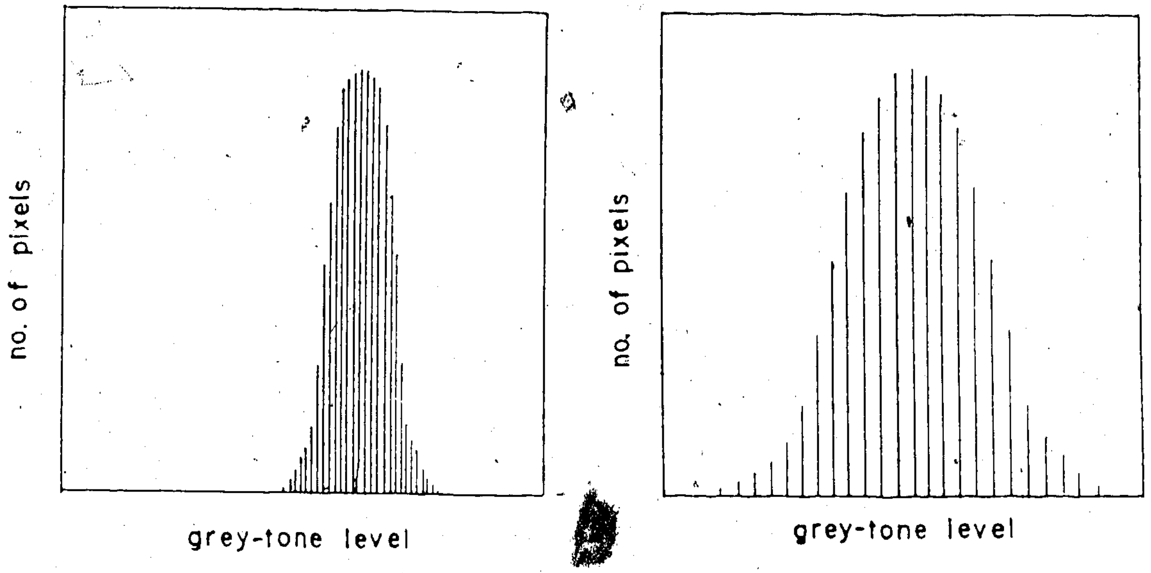
where $h(x,y)$ is the mapping function, $g(x,y)$ is the original image and $e(x,y)$ is the enhanced image. Figure 7 demonstrates the improvement in the histogram of an image after a suitable mapping function is applied to it.

A simple program that performs a linear mapping function was written. The minimum and maximum pixel values in the image that is to be contrast enhanced, is first calculated. Each pixel in the image is then transformed using the algorithm

$$e(x,y) = \frac{(g(x,y)\text{-min})\text{range}}{(\text{max-min})} \qquad (8)$$

where *min* is the minimum pixel value in the image $g(x,y)$, *max* is the maximum pixel value in the image and *range* is the available range in the output display system. This algorithm spreads the pixel values of the image over the entire range that is available, without loss of information. That is to say, the mapping function always produces new pixel values which are within the allowable range, which in this case is between the values of 0 and

Figure 7



Original histogram *Histogram after modification*

256. An example of the improvement possible in images of low contrast is shown in Plate 9. This algorithm does not work well on low contrast images that have a few pixels with values near the minimum and maximum. For example, almost no improvement will be realized on an image that has almost all of its pixel values concentrated in the low end of the available range if it has at least one pixel with a value near the maximum.

An interactive algorithm of the form

$$e(x,y)=\frac{(g(x,y)-K_1)range}{(K_2-K_1)} \qquad (9)$$

eliminates the problem of the previous algorithm. Pixel values in the original image near the maximum and minimum will not prevent contrast expansion from taking place. The constants K_1 and K_2 are entered when the program is initiated. The resulting image has the pixel values which fell between K_1 and K_2 stretched over the entire available dynamic range. Pixels values originally outside the range of K_1-K_2 will yield a new value which will be outside the available range of the system. These illegal values must be set to either the maximum or minimum allowed by the display system, with a resulting loss in information. This method has the advantage of allowing the operator to expand the pixel values in any range desired.

Plate 9

original image

image with contrast expanded

Another method is to determine the range in which a certain percentage of the total number of pixels is contained and then expand this region over the entire available output range. For example, if the range containing 80% of the pixels is to be expanded, the following is done. The constant K_1 of (9) is calculated so that 10% of all the pixels are in the range between 0 and K_1. K_2 is calculated so that 10% of all the pixels are in the range between K2 and the maximum value. The region K_1-K_2 is then expanded over the total available range. This method has a longer execution time since the histogram of an image must first be calculated before K_1 and K_2 can be determined. There is also loss of information in pixels with values that are outside the K_1-K_2 region. This method has the advantage of not requiring K_1 and K_2 to be specified when the program is entered.

Simple contrast enhancement can be done in real time and is therefore a useful aid when doing real time preliminary observations on a patient. This can be done digitally by programming the look-up tables of the video processor boards to perform the desired mapping on the input image. Images are then enhanced and displayed at video rates.

Simple contrast enhancement can also be performed using analog circuitry. An analog video amplifier placed between the video camera and the image processing boards will perform this function in real time. If the noise level is low, analog enhancement will usually produce better results than digital contrast enhancement since more information is actually obtained.

B. Local Area Contrast Enhancement

Often, a region of an image will be very dark, or very bright, in relation to the rest of the image. In these areas, information is compressed over a small range of values, tending to mask any detail. Improvements can be obtained by enhancing the contrast in these areas of the image, leaving the rest of the image alone. This is called local area contrast enhancement.

Local area contrast enhancement is more difficult and time consuming than simple contrast enhancement. An initial attempt at local area enhancement was made by dividing an image into a number of local areas. The maximum, minimum and average pixel values contained in each local area were then obtained. The difference between the maximum pixel value and the minimum pixel value of each local area was calculated and used as a basis to determine the amount of contrast expansion to be applied in each local area. The algorithm used to expand the contrast in each local area was

$$g(x,y)= \frac{\{e(x,y)-avg\}*range}{(max-min)} \qquad (10)$$

where *max* is the maximum pixel value contained in the local area, *min* is the minimum pixel value contained in the local area and *range* is the range over which the pixel values are to be expanded.

The results of this algorithm were unsatisfactory because contours were produced along the edges of each local area. This algorithm also expanded the contrast in each local area to the value of *range*, resulting in false contours being produced in areas of relatively constant grey-tone.

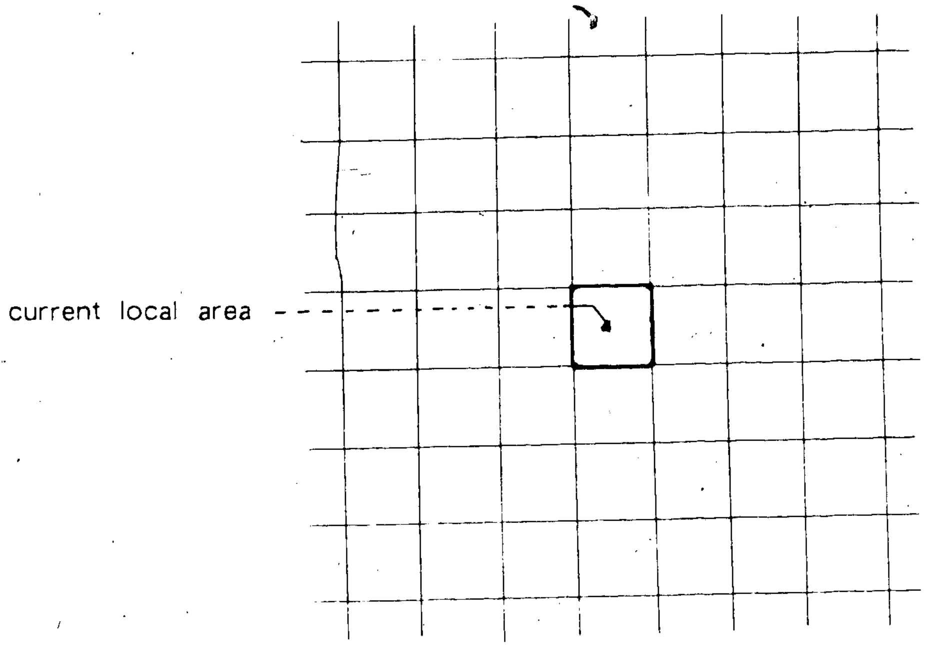
In an attempt to eliminate these problems, the following changes were made to the enhancement algorithm. The size of each local area was reduced so that the total number of local areas per image was greater. The average pixel value in each local area, required by the expansion algorithm, was determined using an average of the pixel values in the current local area as well as the average pixel value in 48 neighboring local areas. The maximum and minimum pixel values in each local area were also calculated. An *average maximum pixel value* of a local area was then calculated for each area by taking the average of the maximum pixel values of a local area and 48 of its neighbors. An *average minimum pixel value* was calculated in the same way. The pixel values in each local area were then modified using the algorithm

$$g(x,y)=\{e(x,y)-Pavg\}*K+128 \qquad (11)$$

where P*avg* is the average pixel value in the current local area and 48 of its neighbors. The constant *K* performs the actual contrast expansion. In the previous algorithm of (10) this multiplier was different for each local area and was one of the causes of the contouring between local areas.

The average pixel value of each local area, P*avg*, was calculated over 49 local areas so that the difference in P*avg* between two adjacent local areas would be small. This was an attempt to eliminate contouring problems between local areas. From (11) it can be seen that each pixel value is expanded around the average pixel value of the local area in which it is contained. This is accomplished by subtracting P*avg* from the pixel

Figure 8



This diagram shows a local area and 48 of its neighbors. The local area contrast enhancement program divides the image into a large number of local areas. The minimum, maximum and average pixel values are calculated for each area. *Average minimum, average maximum* and *average average* values are then calculated for each local area. This is done by calculating the average of the minimum, maximum and average values of a local area and 48 ~~of its~~ closest neighbors.

and then multiplying by the factor K. The contrast expansion factor, K, determines the amount of contrast expansion that will be applied to each local area. The constant of *128*, in (11), effectively sets the new average pixel value in each local area to 128. The values of *average minimum* and *average maximum* are used to calculate a maximum allowable expansion factor K*max* for each local area using the equation

$$K\mathit{max} = \text{range}/(\text{max}-\text{min}) \qquad (12)$$

where 'range' is the available range in the output system, *max* is the *average maximum* of the current local area and *min* is the *average minimum* of the current local area. The standard expansion factor *K* is applied to all local areas except in areas where K*max* is smaller then the value of *K*. In these areas the contrast is expanded by a factor of K*max*.

This is done in an attempt to keep most of the new pixel values within the allowable range of the output system. Any values that do fall outside of this range are simply set to 0 or 256. Plate 10 shows an image which has been processed using this method. Contouring can be seen between local areas but it is fairly light.

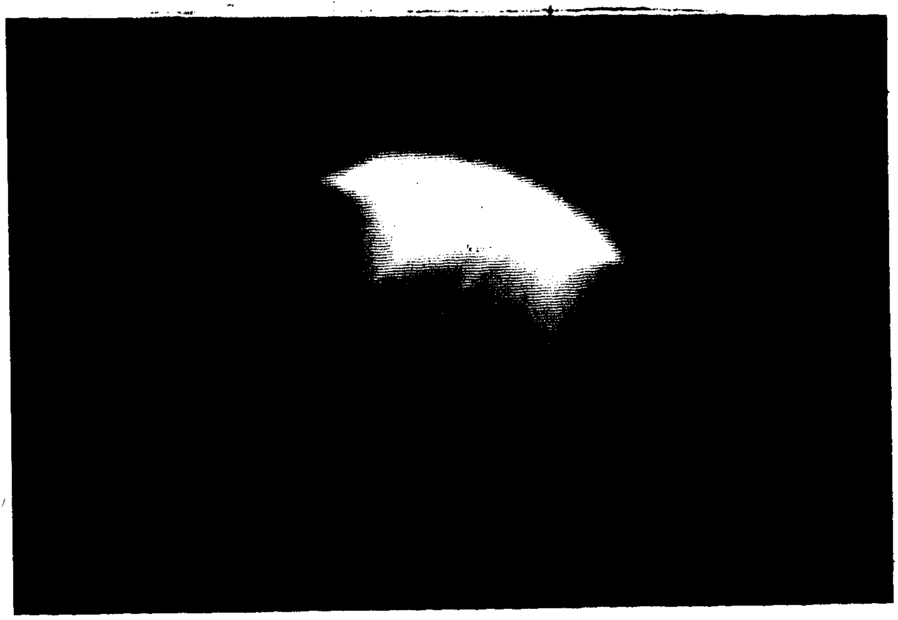
The difference in the average intensities of various regions of an image can also be reduced by filtering an image. This is accomplished with a filter that attenuates the low frequencies in an image, since it is the low frequency information that allows the average intensity of various parts of the image to differ. This low frequency attenuation prevents one region of an image from being much brighter or much darker than some other region of the image. If the filter also amplifies the high frequencies, then the edges in the image will be sharpened as well.

Trial and error was required to determine the filter shape that produced the best results. When the cutoff frequency was set too high or if the amount of low frequency attenuation was too great, the resulting image appeared washed out due to lack of low frequency information. On the other hand, if too much low frequency information was left in the image, there was little improvement in the local area contrast. If there was too much high frequency boost, noise started to become a problem.

The results obtained with this method are fairly good as can be seen in Plate 11. The execution time of this procedure is quite long since the Fourier and inverse Fourier transforms of the image must be calculated. It has the advantage over the spatial enhancement technique in that their are no local areas and therefore no resulting false contours.

Plate 11 shows an image of a transilluminated thumb which has been enhanced in the Fourier domain using a suitable filter to perform local area contrast enhancement. The high pass filter program described in the *Filtering* section of this thesis was used with the constants K_1-K_4 set to 0.5, 5, 10 and 4.0 respectively. The results obtained using Fourier filtering were very similair to the results obtained with the use of the spatial local area contrast program except for the minor contouring produced by the spatial program. The spatial local area contrast program executes in approximately 25 seconds while Fourier filtering takes about 150 seconds. The spatial method also has the advantage of being able to enhance 256x256 images while the Fourier method was limited

Plate 10



original image

image enhanced using local area contrast to images of 128x128, when using the IBM XT computer.

PLATE 11

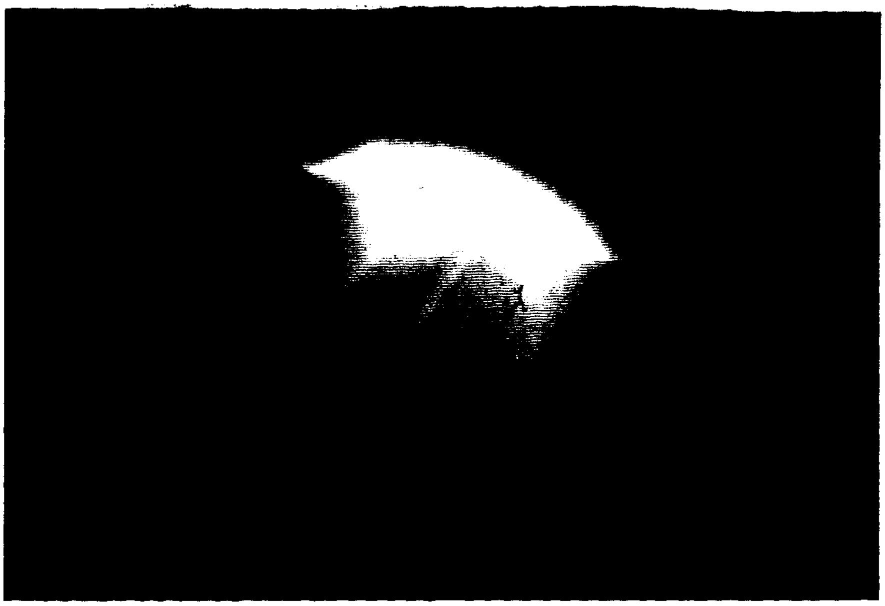


Image of transilluminated thumb

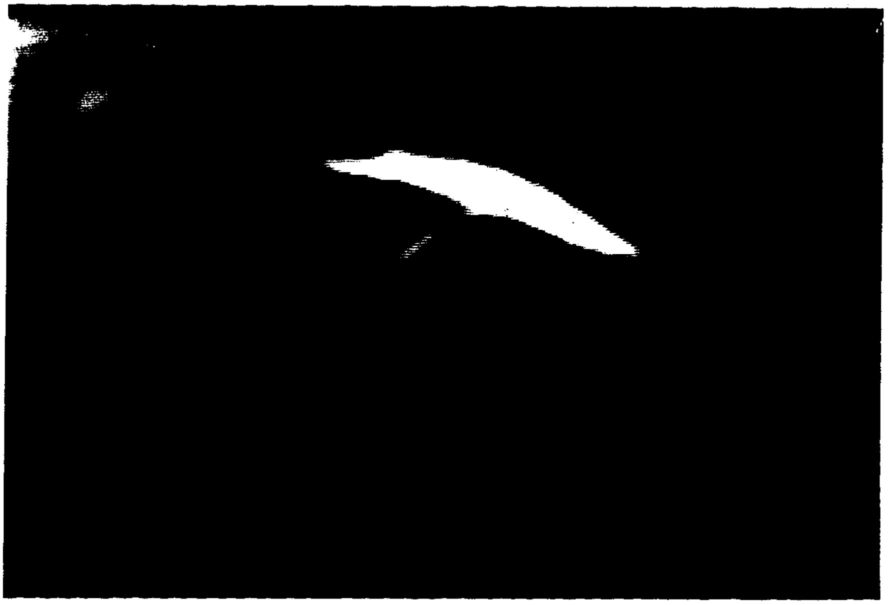


Image filtered in Fourier domain to enhance contrast.

VII. Deconvolution

All images acquired by a system are imperfect. They will always be degraded to some extent. Hopefully, the acquired images will have enough useful information to be of use. If an acquired image is considerably degraded restoration of the image can be attempted if some *a priori* knowledge of the degradation is known. With this *a priori* knowledge the original undegraded image can be estimated from the degraded image. Research to develop methods to restore an image quickly and accurately from a noisy degraded image is being done at many institutions.

If an imaging system i's known to have poor high spatial frequency response then restoration could simply be the processing of the degraded image with a filter that attempts to restore the high frequencies. If there is motion blur in the degraded image then restoration methods would try to reduce the effects caused by the motion.

The degradation of an ideal image f(x,y) to its degraded image g(x,y) can be mathematically modelled [18] as

$$g(x,y)= \iint h(x,y,x ,y)f(x ,y')dx\, dy + n(x,y) \qquad (13)$$

where h(x,y,x ,y) is the degradation function and n(x,y) is the random noise. If we assume the degradation to be shift invariant then the degradation function reduces to

$$g(x,y)= \iint h(x-x ,y-y)f(x ,y)dx\, dy + n(x,y) \qquad (14)$$

If there is no noise present, the degraded image is the convolution of the ideal image with a degradation function. The degradation function is called the point spread function (PSF). The ideal image can then be estimated from the degraded image by deconvolution. It can be shown that the convolution of two functions is the inverse Fourier transform of the product of their Fourier transforms. Therefore the Fourier transform of (14), for the ideal case of zero noise, can be written as

$$G(u,v)=H(u,v)F(u,v) \qquad (15)$$

where G(u,v),H(u,v) and F(u,v) are, respectively, the Fourier transforms of g(x,y),f(x,y) and h(x,y). Theoretically, the transform of the undegraded image F(u,v) could be estimated using

$$F(u,v) = G(u,v) / H(u,v) \qquad (16)$$

To deconvolute a degraded picture, it would seem, requires only to divide the Fourier transform of a degraded image by the Fourier transform of the estimated PSF.

This technique has several problems. Division is undefined at points where the Fourier transform of the PSF is zero, and noise in the degraded image can overwhelm the actual data in the restored image.

When deconvoluting continuous functions, undefined points can be set to zero with no loss in information since the area in one infinitely narrow slice of an integral is zero. In discrete deconvolution, loss of information occurs since there is a finite area associated with each point in the summation.

Noise in the image results in another problem when attempting to inverse filter an image. Often, the magnitude of the Fourier transform of an image G(u,v), will drop as the distance from the origin in the uv-plane is increased. The transform of the noise N(u,v), on the other hand, usually drops less rapidly [18]. Therefore the quotient N(u,v) / G(u,v) has a tendency to increase as the distance from the origin is increased. Any attempts to enhance information at spatial frequencies where N(u,v) / G(u,v) is large results in false contours and noise being produced in the enhanced image. Many degradations attenuate the high frequencies more than the low frequencies (focus blur, image blur) and consequently compensating for this attenuation with inverse filtering will boost the high frequency information and in many instances will cause excessive noise to be seen in the output image.

One method of overcoming these problems is by the use of an algorithm called *least squares filtering* or *Weiner filtering*. It can be shown that to minimize the mean squared error between the original undegraded image and the estimated restored image

$$e^2=E\{f(x,y)-\hat{f}(x,y)\}^2 \qquad (17)$$

requires a filter of the form

$$M(u,v)=\frac{1.0}{H(u,v)}\ \frac{H(u,v)}{H(u,v)\quad+[S_{\eta\eta}(u,v)/S_{ff}(u,v)]} \qquad (18)$$

where H(u is the Fourier transform of the estimated degradation function and $S_{\eta\eta}$ and S_{ff} are the spectral density of the noise and the image respectively.

The Weiner filter will produce a defined number at all points in the Fourier transform of the restored image provided that a non-zero value for the spectral noise density is used. It also prevents the noise in the image from overwhelming the data when the estimated spectral noise density is as large or larger than the actual noise density in the image.

An attempt at restoring noisy degraded images was made using inverse filtering. Due to the complexity of the image restoration problem the amount of time spent in this area was kept at a minimum.

The generation of noise data using a computer program is a complex problem To develop programs which produce random numbers with the correct probability statistics requires a large amount of effort. To circumvent this problem and obtain noise data quickly the video camera and processor boards were used to produce an array filled with noise data. Two images of an identical scene were obtained via the video processor boards and stored in the computer. The difference between the two images was then calculated and used as the noise array. Defocused images of a constant grey-tone surface were used so that any misalignment between the two images would not be the main cause of any difference between the two images. The resulting array was then used as the noise data and added to images to degrade them.

To attempt image restoration on the IBM computer, programs were written to calculate two dimensional Fourier transforms, inverse Fourier transforms and to multiply and invert arrays. Images were degraded by convolving them with a degradation array and then adding a noise array to obtain a degraded noisy image.

Least Squares Filtering was first performed on an image by using an arbitrary constant *K* for S_{nn} / S_{ff} in the Weiner filter algorithm to yield

$$M(u,v) = \frac{1.0}{H(u,v)} \quad \frac{H(u,v)}{H(u,v) \quad +K} \qquad (19)$$

This eliminates the need to have any statistical knowledge of the noise or of the image. The effect of the factor

$$\frac{H(u,v)}{H(u,v) \quad +K}$$

is to put a maximum limit on the value of 1 / H(u,v), thereby reducing the effect of any zeroes in H(u,v). The value of *K* which produces the best results for a given image is determined experimentally. The degraded image of Plate 12 was obtained by convolving an image obtained from the video processor boards with the degradation function shown below.

$$\begin{matrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 1.0 & 1.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{matrix}$$

Noise was then added to the degraded image. As can be seen from Plate 12, the restored image is an improvement over the degraded image, although it contains more noise than the original degraded image.

To perform true Weiner filtering the spectral density array of both the image and the noise must be obtained. The noise can be assumed to be spectrally white ($S_{nn}(u,v)$=constant) if $S_{ff}(u,v)$ falls off much faster than $S_{nn}(u,v)$ [18]. For the work done in this thesis, this was assumed to be the case. The spectral density of an image is the Fourier transform of its autocorrelation array. Therefore some measure of the autocorrelation of the images to be enhanced, must be obtained. Obviously some types of images will have different statistics than others. For example a transilluminated image will have a different spectral density than a motion degraded image of an outdoor scene.

A set of images is defined as being homogeneous if the expected value of each pixel in an image, over the set of all possible images, is the same and if the autocorrelation function of the images is translation invariant. For a real homogeneous random image field, the autocorrelation function is

$$R_{ff}(a,b)=E\{f(x+a,y+b)f(x,y)\} \qquad (20)$$

where E{ } is the expected value of a random variable and f(x,y) is the value of the pixel at the point (x,y) in an image. If the random image field exhibits an even higher symmetry such that the autocorrelation is a function of only one variable 'l', such that $l=(a^2+b^2)^{\frac{1}{2}}$, then it is homogeneous and isotropic.

To simplify the problem of estimating the statistical properties of the input image, the random image field was assumed to be homogeneous and isotropic. An approximate measure of the autocorrelation of the input images was obtained in the following manner.

Plate 12

Top. degraded image

Bottom: restored image

The expected value of (20) for a degraded image $i(x,y)$ and a given value of a and b was calculated by taking the average of $i(x,y)*i(x+a,y+b)$ for all values of x and y for which $i(x+a,y+b)$ is defined. This was done for values of a and b such that $0<a\leq64$ and $0<b\leq64$ to produce an autocorrelation array of size 128x128, since the image is assumed to be homogeneous and isotropic.

A computer program was written for the IBM to calculate the autocorrelation array of an image using the method described above. Each point in the autocorrelation requires between 4000 and 16,000 multiplications and additions, resulting in a very long execution time for the program (approximately one hour). This is not unreasonable since the same estimated autocorrelation array can be used for all images which are subject to the same degradation and exhibit similair statistical properties.

The resulting autocorrelation array was Fourier transformed to produce the Spectral density of the input image. This spectral density was then used to Weiner filter a degraded image. The spectral noise density was an arbitrary constant experimentally choosen to produce the best results. A program was written and developed to perform the Weiner filtering and the result is shown in Plate 13. It can be seen that no improvement was obtained with this method over the method used to obtain Plate 12. The reasons for this were not clear since it is reported in the literature that the use of the Weiner filter using the spectral density of the image should produce better results than the use of an experimentally derived constant[18]. The calculation of the autocorrelation function of the input array may not have produced reliable results since only one image was used for the calculation. More reliable results may be obtained by taking the average autocorrelation of a number of images and using the result to produce a spectral image density array. Due to time limitations, this problem was not explored more fully.

While the Weiner filtering method did not give as good a restoration as was expected, the restored images in Plates 12 and 13 show a fair improvement over the original degraded image. To use these restoration methods on actual images requires an estimation of the degradation that has been applied to them. This can be a difficult task in many cases. In some instances, the PSF can be inferred from the underlying physical process causing the degradation, a defocused image for example. If the degradation is of an unknown nature or if the underlying cause of the degradation is very complex, it may

Plate 13

Degraded image

Image restored using Weiner filter

be required to estimate the PSF from the degraded image itself. While the determination of the PSF of a degradation is an important problem in regard to image restoration, it will not be explored here.

VIII. Infrared Color

A. Acquiring Color Images

A transilluminated breast will exhibit different absorption characteristics at certain wavelengths depending on the condition of the breast [15], therefore it is useful to obtain infrared color images. Three successive images of the same breast illuminated with different spectrums of light in the near infrared can be digitally combined to produce a color image. This is accomplished by assigning one of the three primary colors to each image and displaying the result on an RGB monitor. The three images must be taken quickly so that there is little or no movement of the subject while the images are being captured. To accomplish this, a filter wheel was placed between the light source and the fibre optic bundle. Three bandpass filters are mounted in the wheel with center passbands of 650nm,800nm and 900nm. By rotating the filter wheel, the fibre optic bundle successively delivers three different colors of light to the breast.

To capture three images of the breast with each image corresponding to a different spectrum of light the filter wheel must be synchronized to the video camera. Some background information on video equipment will first be presented to assist in the explanation of this synchronization.

Most video equipment uses an interlaced format for acquiring and displaying an image. A standard television image consists of 525 horizontal lines. These lines are divided into two fields, the odd field consisting of the odd numbered lines and the even field consisting of the even numbered lines. When an image is acquired or displayed, the odd field is first produced by scanning all the odd lines sequentially. After the odd field has been completed, the even field is scanned. This interlaced scan produces less flicker in the output display system than a straight sequential scan of all 525 lines. One complete image is called a frame and consists of an odd and even field.

To acquire an image from a video camera requires the digitization of both the even and odd fields. Each field is displayed in 1/60th of a second for a total frame time of 1/30th of a second. The Hitachi KP-120 video camera used in this research has a resolution of 190 horizontal lines by 240 vertical lines and consequently does not output an even and odd field. A complete image is scanned in one field time, enabling a complete image to be digitized in only 1/60th of a second. Theoretically each filter needs to illuminate the fibre optic bundle for a minimum of 1/60th of a second to allow the video processor boards to capture a complete image. Unfortunately, the video processor boards could not be programmed to acquire only one field of data. A minimum of one frame could be captured at one time by the video boards, resulting in a minimum image acquisition time of 1/30th of a second. The net effect is that when a *freeze image* command is sent to the video processor boards, two successive images from the camera are digitized. The data from the first image being written over by the data from the second image. The scene to be digitized must be illuminated during the 17 milliseconds that are required to acquire this second image from the camera.

A finite length of time is required to allow the filter wheel to rotate from one filter to the next when attempting tc acquire successive images with the use of the filter wheel. Since one field of data is ignored by the video processor boards, it would be theoretically possible to use this interval to allow the filter wheel to rotate to another filter. This procedure was attempted but the timing requirements were very critical and image acquisition could not be properly synchronized to the position of the filter wheel. It was decided that additional time was required between the acquisition of successive images to allow the filter wheel additional time to rotate from one filter to the next. Since the video camera continually transmits frames of data, it is not possible to control the start of a new frame. If there is to be any delay between the acquisition of successive images, it must be an integer multiple of one frame time. A delay time of any other length is not possible since the video processor boards must wait for the start of a new frame before they can begin to capture an image. The final design for the acquistion of color images used a delay of one frame time, or 1/30th of a second between successive images. As a result, the filter wheel must complete one revolution in

$$3\text{x(one frame time)}+ 3\text{x(one frame time)}= 1/5 \text{ of a second}$$

to yield a rotational speed of 300 rpm.

The filter wheel is rotated with a small d.c. electric motor and synchronized to the vertical sync pulses of the video camera using a phase locked loop circuit. This synchronization allows the video camera to capture a complete image while a filter is in position to allow light to pass through it, on to the fibre optic bundle. The synchronization also allows the computer to know which filter is currently in position.

A phase lock loop circuit is used to control the speed of the motor which drives the filter wheel. A light emitting diode and a phototransistor mounted on opposite sides of the filter wheel assembly, detect each revolution of the filter wheel. The pulses from the phototransistor are converted into a square wave using a flip/flop. The phase locked loop circuit synchronizes the square wave derived from the filter wheel to a 2.5 Hz square wave derived from the vertical sync pulses of the camera. The complete operation of this circuit is described in section 'C' of this chapter.

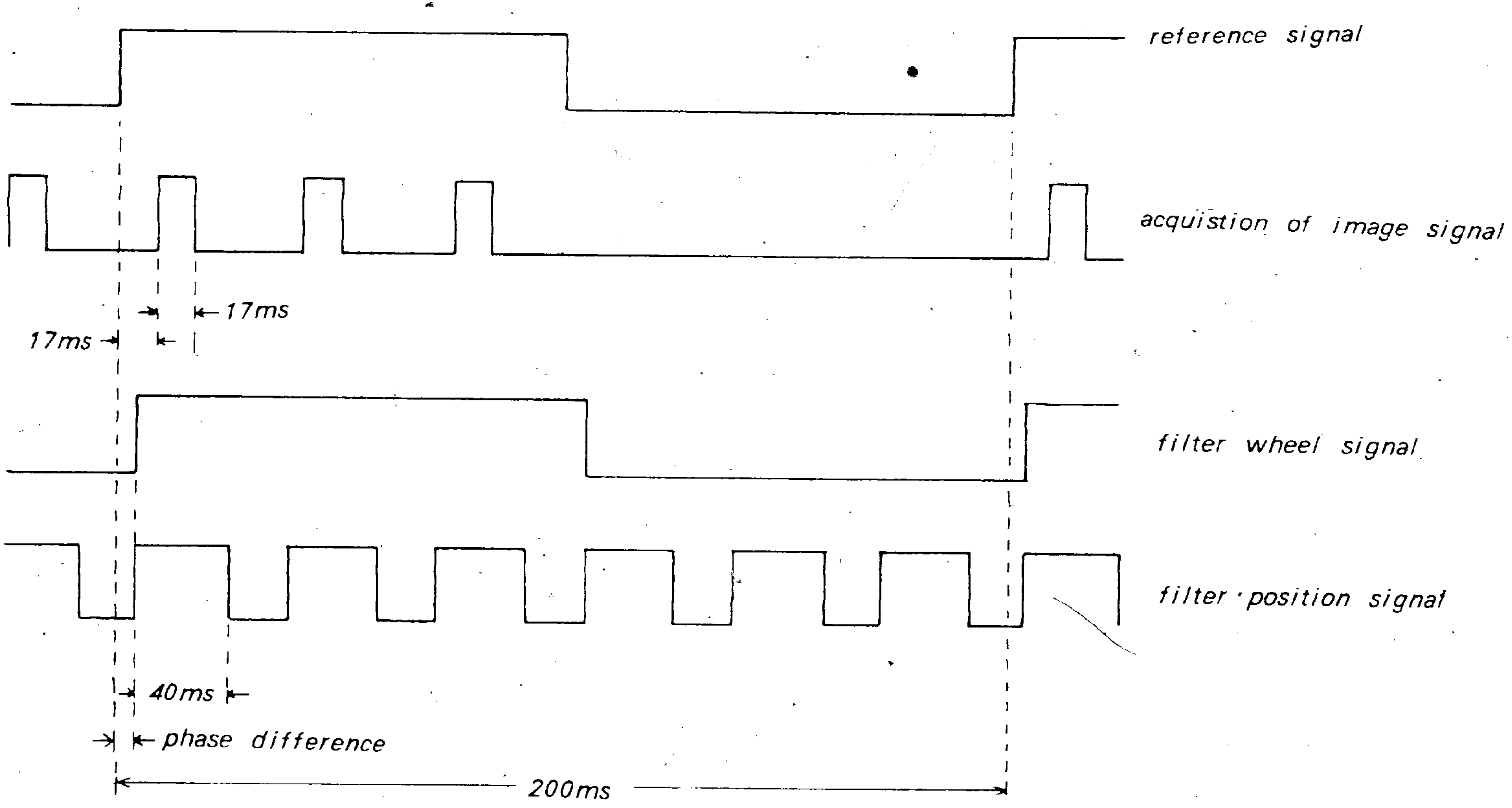
The length of time each filter illuminates the 1/8 inch fibre bundle, when the filter wheel is rotating at 300 rpm, was determined by observing the light emitted from the fibre with the use of a phototransistor. The signal from the phototransistor was displayed on an oscilloscope and the length of time each filter illuminated the fibre optic was recorded. A 1/4 inch fibre bundle was also ordered for use in this project but did not arrive in time to allow experimental work to be done with it. Therefore the data for the 1/4 inch fibre is theoretical whereas the data for the 1/8inch fibre is experimental.

1/8 in. fibre	650nm	-- 40 ms
	750nm	-- 40 ms
	850nm	-- 40 ms
1/4 in. fibre	650nm	-- 28 ms
	750nm	-- 28 ms
	850nm	-- 28 ms

The data shows that the length of time each filter illuminates the fibre bundle is greater than the required time of 17 ms. Using the data for the 1/8 in. fibre, the maximum phase difference between the filter wheel signal and the vertical sync signal that still allows the camera to capture a complete image, was calculated. Figure 9 demonstrates that if there is more than a 6 degree phase difference between the reference signal and the filter wheel signal, the proper filter will not illuminate the fibre bundle for the total time required for the video camera and the video processor boards to capture an image.

Originally, the acquisition of the three images required to produce a color picture, was to take place in the following manner: The filter wheel would be synchronized to the reference signal to allow images to be acquired at the appropriate time. The computer program that controlled the acquisition process would wait until the filter wheel

Figure 9



From the diagram, it can be seen that a filter must be in position during the first *acquisition of data* signal after the *reference* signal arrives. To allow proper acquisition of data, the *filter wheel* signal cannot lag the *reference* signal by more than 1·7 milliseconds. It also cannot lead the *reference* signal by more than 7 milliseconds. This corresponds to a maximum phase difference of -6 degrees *and* +10 *degrees* between the *reference* signal and the *filter wheel* signal.

position signal changed its state before it would start acquiring images. This change of state signifies that the filter wheel is in position to allow an image to be acquired. The computer would then command the video processor boards to acquire an image. When one image was complete, the program would delay for 1/30th of a second to allow the filter wheel to rotate to the next filter. The video processor would then be commanded to acquire another picture. This process would be repeated until all three pictures were acquired. This method of image acquisition depends on very tight synchronization between the filter wheel position and the reference signal.

The phase lock loop controller (described in section C) was not able to achieve the degree of synchronization between the filter wheel and the video camera that was required for proper operation. A modification was made to the image acquisition process and to the computer program that controls it to loosen the timing requirements. A phototransistor was installed on the filter wheel assembly, beside the fibre bundle, to detect the presence of light on the fibre bundle. The signal from this transistor is then fed into the computer. When an image is to be captured the computer first waits for a signal from the phototransistor mounted by the fibre bundle, signifying that a filter is coming into position (light is starting to fall on the phototransistor). It then waits for the camera to start a new frame. If the time interval between when the filter first moves into position and the start of a new image frame by the vidicon camera is greater than a preset value (corresponding to about 6 degrees), the computer simply waits for the next filter to come into position and then repeats the process. In this manner, the computer waits for the phase between the sync signal and the filter wheel position signal to be small enough to allow the acquistion of the images.

The frame buffer of the video processing boards is capable of storing a 512x512 image. Thus 4 complete images of size 256x256 can be stored in the frame buffer, allowing four images to be acquired from the camera without the need of any data transfer between the video boards and the IBM computer. Without this feature, it would not have been possible to acquire the three images fast enough to prevent motion of the subject to become noticeable.

The sensitivity of the total system changes as a function of the light spectrum used to illuminate the fibre bundle. That is, the intensity of the image obtained by the system is dependant on the particular filter that is used to acquire the image. This is due to a number of reasons.

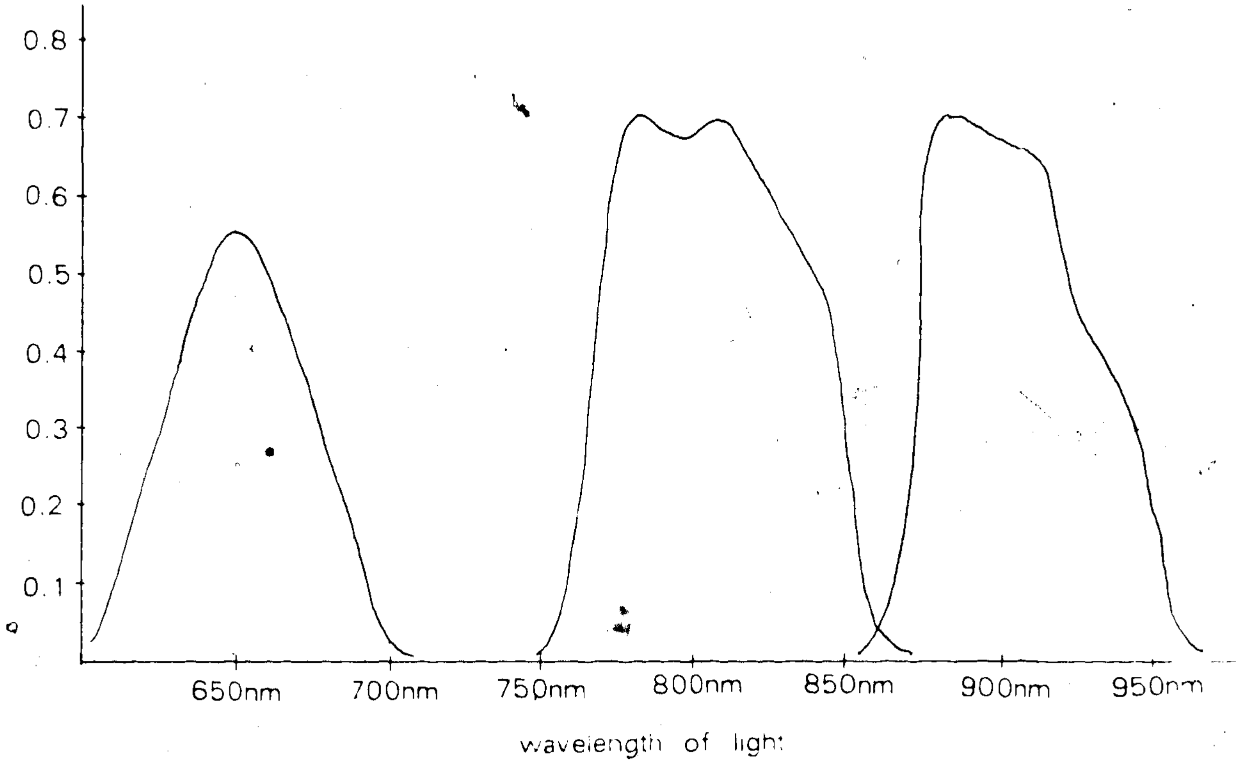
1. The sensitivity of the video camera varies as a function of wavelength.
2. Each filter in the filter wheel attenuates the light emitted by the projector lamp to a different degree.
3. The attenuation of light by the fibre bundle varies as a function of the wavelength of the light.
4. The light source has an intensity which varies as a function of the wavelength.

This non-linear response of the system must be linearized by digitally compensating each of the three images obtained with the use of the filter wheel. This is accomplished by determining the response of the system at each filter setting and modifying each image accordingly. An image of the light emitted from the fibre bundle is obtained in a dark room for each filter. The intensity levels in the three images are then compared to each other. From these levels, a compensation factor is determined for each filter. These compenstation factors are multiplied to images that are to be combined to produce one color image. In this way, the three images obtained with the filter wheel are digitally compensated to produce a linear system response for all three filters.

B. Displaying Color

The use of the filter wheel allows three images, corresponding to various wavelengths of light, to be acquired quickly. By assigning each of the three pictures a color (red, green or blue), an infrared color picture can be created. The filters mounted in the filter wheel absorb much of the light that falls on them because they allow only a small portion of the light spectrum emitted by the source, to pass through (see Figure 10). Filters with broader pass bands than the ones shown in Figure 10 would allow more light through and therefore would increase the sensitivity of the system. Though the frequency responses of the wider pass band filters would overlap, a color image could still be obtained since the response of the three filters would peak at different frequencies.

Figure 10



Frequency response of the filters used in filter wheel.

Capturing the three images required to produce a color image was a relatively straight forward problem displaying the resulting infrared color image, on the other hand was fairly difficult with the equipment available for this project. The AP-512 analog processor video board is capable of driving a color monitor, but only 8 bits of data per pixel is available to drive all three guns of the color monitor. This means that two of the color guns will have only 8 different intensity levels (3 bits of data) and one gun will have only 4 different levels (2 bits of data). The limited number of intensity levels for each color results in an unsatisfactory image.

A different technique to display a color image was tried. It displayed the three pictures acquired with the use of the filter wheel in three sequential video frames. Each picture was displayed independantly by turning on only one of the color guns in the monitor and then outputting the picture associated with that color to the monitor. This resulted in 8 bits of data being used to display each color. After one picture was displayed with the color it was assigned, the next picture was displayed. This was done for all three pictures and then repeated continuously. As a result, instead of the display monitor being updated at 30 frames per second, it was in effect being updated at 10 frames per second (three pictures each requiring 1 / 30 of a second). This resulted in a display of reduced intensity level and more importantly, intolerable flicker.

Another technique which displayed two colors in two sequential frames was then tried. This allowed each color to have 8 bits of data (256 shades) with the display being updated at a rate of 15 times a second. Although the flicker was still noticeable with this technique, it was much less annoying.

A technique that displays only two colors with 4 bits of data (16 shades) per color was then tried. Since there are 8 bits of data available per pixel, the image could be displayed at standard video rates and therefore no flicker. This technique produced contouring due to the limited number of shades for each color.

The obvious solution would be to obtain video boards capable of driving a color monitor with at least 5 bits of data per color and with 6, 7 or 8 bits being preferable. At the present this solution is relatively expensive but will become much cheaper as the price of digital memory and related hardware continues to fall. Since this solution was not available, the two color method which displayed each color in sequential frames was choosen as the [illegible]lternative.

Actual [illegible]ed color images of the female breast could not be obtained since access to a clinical setting was not available. Transillumination images of other parts of the body, such as a finger or thumb, did not exhibit any 'color' when captured and displayed using the technique described above. An image of a transilluminated breast is required to demonstrate the infrared color technique.

C. Filter Wheel Controller

The rotational speed of the filter wheel is controlled with a phase lock loop circuit. The speed of the filter wheel is locked to a reference signal derived from the vertical sync pulses of the video camera. To assist in the explanation of the control circuit, the operation of a phase lock loop will be first described.

The typical phase lock loop circuit synchronizes a signal from a voltage controlled oscillator, to an input reference signal. This is accomplished by detecting the phase difference between the reference frequency and the oscillator frequency. If the oscillator signal is leading the reference signal then its frequency is assumed to be too great and is subsequently reduced by the phase locked loop circuit. If the oscillator signal is lagging the reference signal, then its frequency is assumed to be too low. To enable the controller to *lock* on to the reference signal, the difference between the free running frequency of the oscillator (oscillator frequency without feedback) and the reference frequency must be smaller than the capture range of the phase lock loop. The phase detector circuit of a phase lock loop usually consists of an analog multiplier that multiplies the reference and oscillator signals together. If the reference signal is described by the equation $A_0\sin(w_0t)$ and the input signal is described by the equation $A_i\sin(w_it)$ then the multiplier will yield

$$A_0A_iK*\sin\{(w_0-w_i)t\}+A_0A_iK*\sin\{(w_0+w_i)t\}$$

where w_0 is the oscillator frequency, w_i is the input reference frequency and K is the gain of the multiplier. This signal is then low pass filtered to obtain

$$A_0A_iK*\sin\{(w_0-w_i)t\}$$

It is this signal that is then used as the input to the voltage controlled oscillator.

The filter wheel controller designed for this project does not employ a standard phase lock loop integrated circuit. The voltage controlled oscillator contained in a typical integrated phase lock loop circuit is not needed for this project because it is required to synchronize the filter wheel position, not a variable oscillator, to the reference signal. The phase detection circuitry in a phase lock loop chip is also of minimal value in this application because the reference signal and the signal from the filter wheel are both square waves. Multiplication of two square waves is easily accomplished using an exclusive OR circuit.

The schematic diagram of the filter wheel controller circuitry is shown in Figure 11 and will be referred to in the following explanation of the circuit. A phototransistor and a light emitting diode (LED) are mounted on the filter wheel assembly so that they are on opposite sides of the filter wheel. A small hole in the filter wheel, that lines up with the phototransisor and LED, causes the photo-transistor to produce a voltage spike for each revolution of the filter wheel. The voltage spike produced by the phototransistor is transformed into a clean pulse by the Schmitt trigger circuit of IC 8c and then converted into a square wave. Q1 is required as an interface between the Schmitt trigger and the D type f/f (5b) because the output from the single supply op amp (8c) is not directly compatible with the requirements of TTL logic. It is the square wave produced by f/f (5b) that must be locked to the reference signal.

The filter wheel is required to spin at 300 rpm or 5 revolutions per second. At this speed, the square wave derived from the filter wheel has a frequency of 2.5 Hz and consequently, the reference signal must have a frequency of 2.5 Hz. This is obtained by dividing the vertical sync signal, which has a frequency of 60 Hz, down to 2.5 Hz using the divide by 24 circuit consisting of IC's 1, 2, 4a and 4b.

Latch 5a determines if the filter wheel signal is leading or lagging the reference signal. The reference signal is fed into the clock input of the latch while the filter signal is fed into the data input. When the level of the reference signal changes from low to high, the state of the filter signal is latched by 5a. If IC 5a latches a high value from the filter signal, then the filter signal is assumed to be leading the reference signal. If it latches a low value, then the filter signal is assumed to be lagging the reference signal

The exclusive OR gate (IC 6a) is used as a phase detector. It's output is high whenever the filter signal and the reference signal are at different levels. The width of the pulses from the phase detector are dependant on the phase difference between the two input signals. The output of the phase detector is used to pulse width modulate the outputs from the lead lag detector to produce the signals labelled P1 and P2

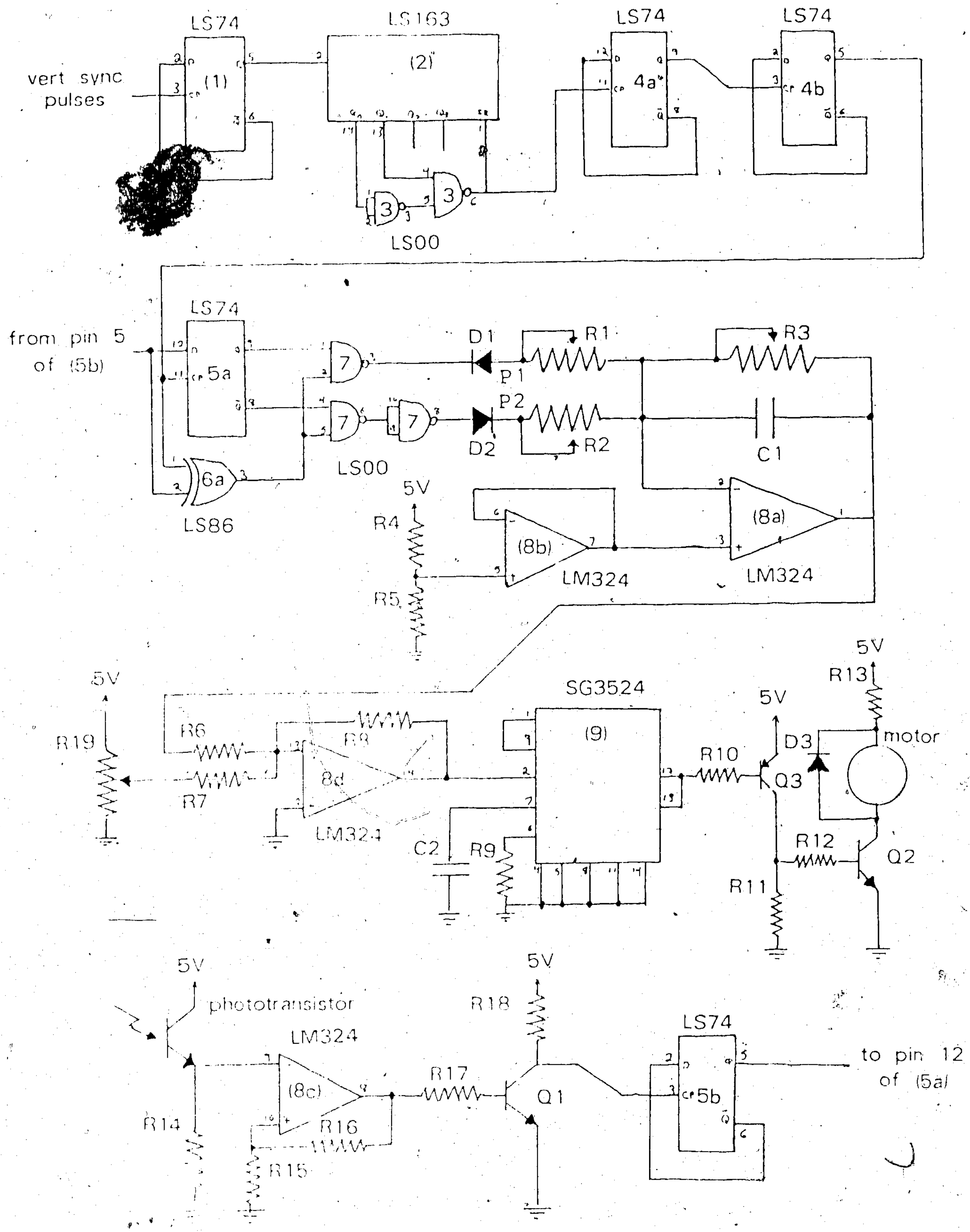
Op amp 8a integrates the P1 and P2 signals derived from the phase detector circuitry. When P1 is low, signifying that the filter signal is leading the reference, the output voltage of the op amp is increased. This causes the pulse width modulator circuit (PWM) to lower the duty cycle of the voltage waveform sent to the motor, consequently slowing down the motor. P1 has no effect on the op amp when it is high due to diode D1. When P2 is high, signifying that the filter signal is lagging the reference signal, the output of the integrator is slowly decreased, increasing the speed of the motor. P2 has no effect on the op amp when it is low, due to diode D2.

Feedback resistor R3 on the integration op amp 8a is required for stability. If R3 is removed from the circuit, the feedback capacitor C1 causes the op amp to act as an integrator. It's output would be the integral of the phase difference between the two input signals as a function of time. This was found to result in an unstable circuit with the controller not being able to lock on to the reference signal. The addition of R3 results in a transfer function that is exponential. If both inputs to the op amp are off, the output exponentially decays to its virtual ground value of approximately 2V. When either P1 or P2 are on, the output exponentially rises or falls to its maximum or minimum value.

The speed of the motor is controlled by pulse width modulating the voltage across its windings using a power transistor. An SG3524 pulse width modulator chip is used to produce the pulse width modulated signal required to drive the power transistor. The frequency of this signal is set using resistor R9 and capacitor C2 and is approximately 1 Khz for the values shown in the schematic diagram. The duty cycle of the signal produced at pin 12 of the 3524 depends on the voltage that is applied to pin 2 of the 3524 This voltage is obtained from the integration op amp described earlier.

The free running frequency of the filter wheel is first set by breaking the feedback loop and adjusting R19 to obtain a speed of 300 rpm. The feedback loop is then reconnected and variable resistors R1, R2 and R3 are adjusted to obtain optimum performance. After adjusting the controller to obtain maximum performance it was found to require approximately 3 seconds to synchronize the filter wheel to the reference frequency. This was adequate since the capture time is not critical in this application. The phase difference between the filter wheel and the reference signal could not be kept at a small enough value to allow the acquisition of images with the filter wheel. As a result, the controller is used to keep the speed of the filter wheel at 300 rpm and to roughly synchronize the position of the filter wheel to the reference signal. The tight synchronization required to obtain three images with the use of the filter wheel is accomplished using software. This is explained in the section titled 'Acquiring Color Images

Figure 11



Schematic diagram of Filter Wheel speed controller.

Table of Values for Figure 11

R1	200KΩ
R1	200KΩ
R2	200KΩ
R3	200KΩ
R4	3.3KΩ
R5	2.0KΩ
R6	3.3MΩ
R7	100KΩ
R8	100KΩ
R9	18KΩ
R10	1.0KΩ
R11	100Ω
R12	100Ω
R13	4.7Ω
R14	30KΩ
R15	10KΩ
R16	10KΩ
R17	1.0KΩ
R18	100Ω
R19	5.0KΩ
C1	1uf
C2	0.1uf

Conclusion

The goal of this research project was to develop equipment capable of capturing, processing and displaying transilluminated images of a female breast for the purpose of cancer detection. The equipment that has been developed is capable of digitizing, processing and displaying a transilluminated image. The image processing routines that have been described are intended to provide a good base for the continued development and refinement of image processing routines for transilluminated images.

The system, on the whole, performs quite well. The time required to transfer an image from the video processor boards to the computer is about 1.5 seconds. Images can be stored on the hard disk of the computer in about 6 seconds. The digitization of an image requires 1/30th of a second. The solid state camera is small and lightweight and provides adequate resolution. Most of the image processing routines execute in a reasonable length of time.

The 8087 math processor chip allowed some rather complex algorithms to be performed on the IBM microcomputer. Without this chip these programs would have run very slow and would not have been practical. In particular, the FFT programs would not have been attempted.

The next step in this project is to move the equipment into a clinical setting to evaluate its performance under actual conditions. This will determine the value of the hardware and software and will suggest methods of improving the current design of the system. One of the main concerns is to determine if the light level supplied by the fibre optic bundle is intense enough to obtain an adequate image of a transilluminated breast. If the light level is too low there are a number of modifications that can be made to the system to increase its sensitivity.

1) A more intense light source could be used to illuminate the fibre optic. Larger sizes of projector lamps are readily available but the size of their source region also becomes larger effectively limiting the amount of light that can be coupled into the fibre bundle. A very bright source with a small active region is required for maximum light coupling into the fibre bundle. The use of an arc lamp is one possibility.

2) A large fibre optic bundle would allow more light to be coupled into it.

Fibre bundles larger than 6mm would start to become unwieldy and therefore would not seem to be an adequate solution.

3) The filters currently used in the filter wheel attenuate the amount of light considerably because of their very narrow frequency response. Bandpass filters with a wider passband would increase the amount of light reaching the patient.

4) The sensitivity of the camera could be increased by placing a video amplifier between the camera and the image processing boards. This would only be useful if the noise level is much lower than the data level. The amount of amplification that could be obtained before noise becomes a problem would have to be obtained experimentally.

5) The sensitivity of the camera could also be improved with the use of a larger lense. A f2.5 lens was used in this project. Larger aperture lens are readily available.

6) More light could also be delivered to the patient with the use of multiple fibre bundles. They would not be as bulky as one large diameter fibre and would also allow the breast to be more evenly illuminated. The limit to the number of fibres that can be used is set by the size of the filters in the filter wheel. All the fibres must be illuminated while the video camera is obtaining an image.

Input from doctors familiar with the use of transillumination imaging is required to determine which of the image processing programs that have been developed are of value. Suggestions on how they can be improved in terms of being *user friendly* in a clinical environment would also be useful. This requires that the system be moved to a location that allows doctors interested in transillumination imaging easy access to the equipment

Improved packaging of the complete system is also necessary to allow the system to be used with a minimum of effort. At the present, the transportation of all the equipment is a rather cumbersome process. The filter wheel and its controller must be designed as one unit so as to occupy as small an area as possible and to be physically sturdy. The video processor boards, their power supply and their card cage must also be put into one package.

Improvements in the acquisition of near infrared color images could be obtained with the use of a color video camera containing filters in the near infrared. This would allow the acquisition and display of infrared color images at standard video rates. Video processor boards capable of digitizing and displaying color images would also be required. At the present time, this type of equipment is very expensive.

This thesis was intended to provide a good basis for the continued improvement of the diaphanographic equipment that has been developed. Hopefully it has met this goal and will be of some use in the future in regards to using transillumination imaging for the detection of breast cancer.

References

1. Advances in Digital Image Processing. Edited by Peter Stucki. Plenum Press 1979.

2. Andrews,H.,C.: Computer Techniques in Image Processing. Academic Press 1970.

3. Beauchamp,K.,G.. Signal Processing using Analog and Digital Techniques. John Wiley & Sons 1973.

4. Castleman,K.,R. Digital Image Processing. Prentice Hall 1979.

5. Cooley,J.W., Tukey,J.W.: An Algorithm for the Machine Calculation of Complex Fourier Series. Math. of Comput.. vol. 19. pp. 297-301 1965,

6. Cutler,M. Transillumination as an aid in the diagnosis of lesions. Surg. Gynecol. Obstet. vol 48, pp. 721. 1929.

7 Gros,C.M. Quenneville,Y Hummel,Y. Diapnanologic mammaire. J. Radiol. Electrol Med. Nucl. vol 53, pp. 297, 1972

8. Hall,E.L., Computer Image Processing and Recognition. Academic Press 1979.

9 Harris,J.,L. Image Evaluation and Restoration, Journal of the Optical Society of America. Vol 56. No 5. 1966.

10. Hord,R.,M. Digital Image Processing of Remotely Sensed Data. Academic Press 1982.

11. Hunt,B.,R. The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer. IEEE Transactions on Computers. Vol. C-22,No.9. 1973.

12 Lahart,M.,J. Local Image Restoration by a least-squares method. Journal of the Optical Society of America. vol 69. No 10 1979.

13 MacAdam,D. P.. Digital Image Restoration by Constrained Deconvolution. J. Opt Soc. Am. Vol 60. No. 12, 1970

14. Nussbaumer,H.,J. Fast Fourier Transform and Convolution Algorithms. Springer-Verlag 1981.

15. Ohlsson,B., Gunderson,J., Nilsson,D. Diaphanography: A method for the Evaluation of the Female Breast. World Journal of Surgery 4, 701 707. 1980.

16. Pratt.W.K.: Digital Image Processing. John Wiley & Sons 1978.

17. Rabiner,R.,L, Gold,B.: Theory and Application of Digital Signal Processing. Prentice Hall 1975.

18. Rosenfeld,A., Kak,A.C. Digital Picture Processing Academic Press 1976.

19. Sondhi,M.,M.: Image Restoreation. The Removal of Spatially Invariant Degradations. Proceedings of the IEEE. Vol. 60, No. 7 1972

20. Sternberg S. R. Biomedical Image Processing. Computer 1, 22-34, 1983.

21. Topics in Applied Physics. Edited by T.S. Huang. Springer-Verlag 1975.