1329522

Fisher, Henry Donald, III

DESIGN OF REVIEW CONSOLE FOR RADIOLOGY APPLICATIONS

The University of Arizona M.S. 1986

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages __✓__
2. Colored illustrations, paper or print __✓__
3. Photographs with dark background __✓__
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages ______
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __64__ seem to be missing in numbering only as text follows.
13. Two pages numbered __63__. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ________
16. Other ______________________________

DESIGN OF REVIEW CONSOLE FOR
RADIOLOGY APPLICATIONS

by

Henry Donald Fisher III

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 6

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: H Donald Fisher III

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Ralph Martinez — 25 Nov 86

R. Martinez
Associate Professor of Electrical and Computer Engineering

Date

PREFACE

This research project consists of the design and implementation of an image display console for use in diagnostic radiology. It evolved from a much broader program in photoelectronic imaging systems for medicine being conducted in the Radiology Department at the University of Arizona. The long range goal of that research program is the replacement of film-based imaging with photoelectronic imaging.

The development of photoelectronic imaging systems for medicine has become a very active area in applied research. The evolution of VLSI technology has made the economical development of sophisticated imaging systems possible. This evolution in technology, coupled with the increasing costs associated with the existing film-based imaging system, has created great interest in the development of photoelectronic imaging systems for medicine.

The program at the University of Arizona was initiated in 1973 by Dr. Sol Nudelman. Research performed during the first six years led to the development of Digital Subtraction Angiography. In 1979 the research emphasis was shifted towards the development of a totally photoelectronic radiology department. Progress was slow until after the

first Picture Archiving and Communication Systems (PACS) meeting held in 1981. At that time the medical industry became interested and began to sponsor research at various universities. In 1983 a five year research program was established between Toshiba Medical Systems and the University of Arizona to develop a prototype PACS system. The research described here is a component of that program.

During the time this research was being performed, the author was the group leader of a team of hardware and software engineers. The design and fabrication of the display console described here is a result of the joint efforts of the members of that team. The author claims credit for the establishment of the review console concept, the design requirements definition, the high level system design, and the detailed design of the image processor. The detailed circuit design and implementation was performed by Mr. R. Vercillo and Mr. R. Lamoreaux. The vast majority of the software was written by Mr. K. McNeill. Mr. J. Percy was responsible for the extensive fabrication effort.

Thanks go to the members of the design team for their assistance, technical feedback, and patience during the development ordeal. Special thanks go to Dr. T. Ovitt M.D. for the invaluable consultations that provided incite into what the realistic system requirements should be. I wish to thank Dr. Martinez for his guidance in the preparation of this thesis and for being my advisor. And finally, thanks to Dr. M. Capp M.D., chairman of the Radiology Department, for providing the resources and administrative latitude that allows work such as this to be performed.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

ABSTRACT

Hospitals are currently considering upgrading their manual film-based imaging systems with automated Picture Archiving and Communication Systems (PACS). The image display workstation, which is to provide the interface between the PACS and the radiologist, will be a critical component of such a system. A workstation design optimized for the PACS application was required in order to provide image processing and image handling capabilities equivalent to those in use now. The work described here encompasses the design and fabrication of a prototype display workstation for Radiology.

CHAPTER 1

INTRODUCTION

The cost of health care in the United States increased at a rate of 30% per year for the three years preceding 1983. This rate has been twice that of inflation and currently represents 12% of the GNP. Many approaches have been taken to reduce these costs without introducing a negative impact upon patient care. The Federal Government has tried to legislate a limit on the rate increase of health care costs by introducing DRG's (Diagnostic Related Groups). This program imposes restrictions upon the amount of reimbursement that Medicare will provide for various types of diseases. It is anticipated that the third party carriers (e.g. Blue Cross) will adopt similar reimbursement strategies. These limits on reimbursement have forced hospitals to revise the ways in which they provide medical care and to move toward automation.

An obvious approach to the reduction of hospital costs is to make the health care delivery system more efficient. This approach has resulted in the introduction of computers in many areas of Hospital Management. Computers have been used for many years in areas such as patient registration and in clinical laboratories. A major new application area is within radiology departments. Within the radiology department, computers are being used in two areas. They are being used as management tools to aid in such functions as scheduling and inventory control. They are also increasingly being used to replace the diagnostic film-based imaging systems currently in use with digital images [1],[2]. The latter application is referred to as a Picture Archiving and Communication System, or PACS.

The research described here encompasses the design and fabrication of a PACS image display subsystem. This display subsystem will initially be used in a series of psychophysical experiments designed to determine the radiologist's requirements for such a system. If these experiments validate the design concepts of the display subsystem, it will serve as a prototype for future PACS display system development.

In this introduction, an overview of the current film-based system will be provided, followed by the design requirements for the replacement PACS. The PACS is composed of four major subsystems. These are image acquisition, image transmission, image archiving, and image display. Each of these subsystems will also be briefly described. Finally, the scope of this research project will be defined. Problems encountered in the design of the image display, and a detailed description of the display implementation will be provided in subsequent chapters.

Overview of Film-Based System

Traditionally, most radiographic procedures have been performed using film as the image receptor. Since the mid 1970's more and more imaging modalaties have relied upon digital techniques for image handling. Film is still being used as the image receptor in procedures such as chest radiography, where images are produced that require a large format (14x17 inch images). Other modalities, which generate images in a digital format, include: Computed Tomography (CT), Nuclear Medicine, Magnetic Resonance Imaging (MRI), Ultrasound, and Digital Subtraction Angiography(DSA). When a study using one of the above modalities has been completed, the digital images are presently being transferred to film. The digital information is not retained and these film images become the only records of the procedure. The film images from all of the above imaging modalities are put into patient jackets (folders) which are filed for a prescribed length of time (short term archive). Patient jackets are then transferred to a warehouse (long term archives). The images are eventually destroyed when the film is reprocessed to reclaim its silver content. It is envisioned that the PACS, which are presently under development, will eventually replace the manual film system described above.

PACS Design Requirements

As the photoelectronic equivalent of a film-based radiological imaging system, the minimal requirement of a PACS is that it be at least as effective as the system which it is to replace. This imposes the following overall design requirements:

1. Images from all imaging modalities should be able to be incorporated into a PACS environment.
2. Images should be equally accessible in a PACS environment as they are in a film-based one. They must be able to be stored and retrieved from storage with the same facility as were their antecedents.
3. Image processing functions should be available which allow the radiologist to perform operations equivalent to those which are performed in a film-based system (ex. PACS contrast manipulation should be equivalent to bright lighting in a film-based system).

4. A radiologist should be able to work at the same speed, viewing as many images consecutively and concurrently, as he does using a film-based system.

PACS Subsystems

A PACS is composed of four subsystems [3] as shown in Figure 1.1. These subsystems are:

1. Photoelectronic Image Acquisition;
2. Image Archiving;
3. Electronic Image Transmission; and
4. Image Display.

Each of the subsystems is responsible for a major function. The following sections describe each of the subsystems listed above.

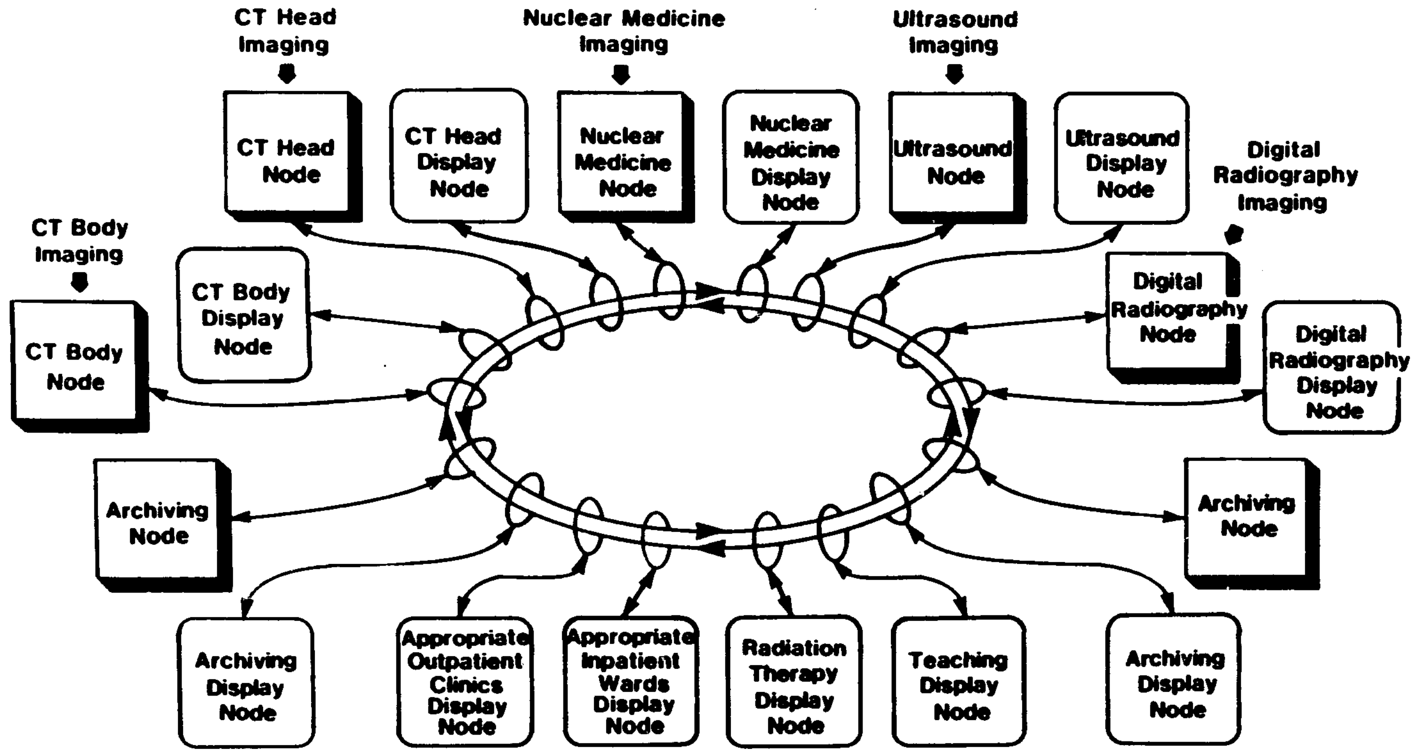


Figure 1.1 PACS System Block Diagram (Ref. 3)

Photoelectronic Image Acquisition

The first of the four PACS subsystems is photoelectronic image acquisition. A photoelectronic image acquisition device is one which generates an image in a digital format. The output image can be described in terms of its spatial and contrast resolutions. Spatial resolution refers to the number of pixels required to retain the information detail in an image (ex. 512X512 pixels). Contrast resolution, or dynamic range, describes the number of gray levels present in the image. An image with a dynamic range of 60 db (measured in volts) contains 1000 shades of gray. The higher the spatial resolution becomes, the smaller are the structures that can be visualized in an image. The greater the dynamic range, the more subtle the differences in brightness which can be visualized. As mentioned earlier, numerous digital imaging modalities exist which could be incorporated into a PACS environment without modification. Each of these imaging modalities is characterized by intrinsic spatial and contrast resolutions that determine the number of pixels and the number of bits per pixel required to represent its images. PACS subsystems should be able to accommodate the varied image formats which the different modalities present. In general, these images are characterized by spatial resolutions of up to 2048X2048 pixels with 40 db to 70 db dynamic ranges (100 to 4000 shades of gray) [4].

Image Archiving

The second PACS subsystem is image archiving, or storage. Adult medical images must be retained for about 5 years. Pediatric images must be retained until the minor reaches the age of 18. It has been estimated [5] that 4 Tera bits of data would be generated annually by all imaging modalities in a 300 bed hospital. The total storage requirement for a 5 year period would exceed twenty tera bits. Using conventional storage media, this would require 10,000 reels of tape recorded at 6250 BPI.

On-line storage of the entire patient data library is not feasible using the storage technologies available now or in the foreseeable future--nor is it required. The current approach [5],[6] is to divide the image archive into two sections. The first, a short term archive, is designed to handle the active images for those patients currently under treatment. The short term archive is characterized by on-line storage which provides rapid access time, read/write capability, and limited storage. Such a system could be manufactured using state of the art technology magnetic computer disks operating in parallel. The long term archive is characterized by the use of higher density off-line storage which could be non-erasable. Laser optical disks are being proposed for this application. Current technology provides approximately three gigabytes of storage on a ten inch diameter disk. A combination of these two technologies will provide both rapid access to active images and compact long term archival storage.

Electronic Image Transmission

The third PACS subsystem is image transmission. The transmission system must provide bidirectional communication between acquisition systems, display workstations, and image archives. It is envisioned that a PACS will eventually become a component of a national network linking together hospitals all across the country. Standardization is a primary design consideration for such an inter-hospital national network. The primary design considerations for the intra-hospital local area networks are speed, reliability and the ability to interface to the other manufacturers' PACS components. Speed is being achieved through the use of parallel fiber optic data transmission paths. The American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) formed a joint committee ACR-NEMA to develop a standard defining hardware and data interfaces within a PACS. This standard defines error detection and correction (EDAC) codes that will be used to guarantee the integrity of the data transmitted. Single point failures will be circumvented through the use of redundant data paths. PACS image transmission systems are designed in this way so that a single failure will have only local effect. The ACR-NEMA standard also defines a standard data interface that is intended to permit the interfacing of any manufacturer's equipment to the PACS.

Image Display

The fourth PACS subsystem is image display. An image display workstation, or console, recovers an image from the archive, processes it, and displays it on a viewing screen according to the dictates of the user. To be as effective as the current film-based systems, image display workstations must be capable of simultaneously displaying multiple images acquired via the varied photoelectronic imaging modalities (CT, MRI, DSA, etc.) [7]. To accomplish his task, an image display console must be capable of accomodating images with a wide latitude of spatial and contrast resolutions present. To allow comparisons of results from different radiographic procedures, these images must be displayed concurrently. The above capability is referred to as the 'multimodality display capability', and is a key requirement of a PACS image display subsystem.

Consoles of the image display subsystem will be required to address the various imaging requirements that exist throughout a hospital. It is anticipated that there will be three types of viewing consoles [1] representing different levels of sophistication in terms of:

1. The number of of viewing screens and the number of images displayed simultaneously;
2. The amount and sophistication of image processing capability;
3. The speed of image handling (speed of retrieval from storage and image processing).

These three types of consoles are referred to as the analysis console, the review console, and the utility console. Each will be described in more detail below.

The analysis console, the most sophisticated of the three consoles, will be intended for use by the radiologist who has the primary responsibility for the interpretation of a study. It will be intended to functionally replace the large panels of light boxes currently in use in a radiology department's reading room. It will be capable of displaying multiple high resolution images on approximately 8 viewing screens, and of performing sophisticated image processing operations on each image independently. It has been estimated [8] that a spatial resolution of 2048X2048 pixels will be required to accommodate large format images such as chest radiographs. The analysis console will be capable of handling images with dynamic ranges of up to 70 db. It will provide the most sophisticated image processing capabilities of any of the three consoles. A minimal set of capabilities will include image subtraction, contrast enhancement, magnification and minification. Additional capabilities will include real time movie loops for cine angiography, image registration, and color capability. The speed of image handling should be no slower than that of the current film-based systems. Radiologists consume information at an incredible rate [5],[7]. In a typical situation, 6 to 8 images are examined and a report is dictated in less than a minute. This places severe design requirements upon the image retrieval, processing and display components of an analysis console.

The review console will be targeted for use by referring physicians and other doctors who must review cases after the radiologist has submitted his report. A referring physician may use this console in consultation with the patient, or a surgeon may use it to view radiographs during a surgical procedure. The review console will have a subset of the analysis console's capability. It will have a spatial resolution of approximately 1024X1024 pixels, the same dynamic range as the analysis console, one or two display screens, and less sophisticated image processing functions than those of the analysis console.

The utility console will be designed to satisfy the requirement where a physician only needs to obtain a casual view of a radiograph. It could be located in his office. An example might be when a technologist needs to ask a radiologist if the image taken would satisfy his needs. A low resolution single screen display with very limited processing capability would suffice.

As mentioned earlier, in order to be equivalent to film-based systems, PACS will need to be capable of handling large format as well as smaller format images. Without the large format capability a PACS would be of little value since these images comprise 75% of all those taken in a radiology department [9]. Large format images are characterized by a spatial resolution of 2048X2048 pixels with up to a 70 db dynamic range. Thus 40 MBits of information will be generated by each chest radiograph. To transmit such an image from an archive to a display workstation in .5 seconds requires a transmission bandwidth of more than 80 MBits/sec. To display such an image on a video monitor (with a single gun CRT) requires that data be presented at a rate exceeding 200 MPixels/second (for a 30 Hz interlaced raster).

Though extensive research is being done concerning the handling of large format images, no current imaging system other than film has achieved broad acceptance by the medical community. The large capacity image archiving subsystem and the high bandwidth data transmission subsystem described above are adequate to handle the quantity of data present in large format images. However, current technology has proved inadequate to provide the full range of image display capabilities required for large format images.

The first commercial products capable of producing a large format radiograph in a digital format were introduced at the 1985 meeting of the RSNA [10]. Images generated by these systems are processed to increase their spatial frequency resolution and decrease their dynamic range. The processed images are then written to film for interpretation.

There now exists only one company that markets a 2048X2048 resolution display system. It was introduced in 1986 (two years after this research project was initiated) in the hopes that it would gain acceptance as a PACS analysis console. The initial units have exhibited poor contrast resolution and a fixed pattern "banding" artifact. They provide only one display screen and are very expensive.

Thesis Objective

The research project described here implements a different approach to the design of the image display subsystem. The objective of this thesis was to design a review console that provides a virtual 2048X2048 image display using a 1024X1536 display format. A virtual display is one in which the radiologist is given the impression he is viewing a 2048X2048 image even though the actual resolution is less. The intent of the virtual display was to provide more than just the illusion of 2048X2048 capability. The display console is designed such that the radiologist's performance is equivalent to what would be achieved if he were viewing the actual 2048X2048 image data. This is accomplished through the use of a unique fractional zoom capability that will be described later.

This research project involves the design and implementation of the PACS review console as described above. It consists of a system capable of processing and displaying the digitally formated diagnostic images that will be present in a PACS environment. The review console will also provide a test bed that will permit related research into the psychophysics of the user/display interaction. This research has been sponsored in the Radiology Department, AHSC, University of Arizona, by Toshiba Medical Systems.

CHAPTER 2

CONSOLE DESIGN PROBLEMS AND PERFORMANCE REQUIREMENTS

The long term goal for a PACS is to replace the film-based imaging systems currently used in radiology with ones comprised of photoelectronic imaging devices. To be accepted by the radiological community, the PACS will have to provide operations equivalent to those currently provided by the film-based system. Additional features which enhance the quality or speed of diagnosis will increase the probability of PACS acceptance. Image display workstations are the system components which will provide the radiologist's interface with the PACS. They must provide access to patient information distributed throughout the PACS as well as a comprehensive image display/analysis capability.

At the PACS II conference (June 83) many radiologists described the requirements they felt critical to the design of an image display console for Radiology [4]-[7]. A comprehensive requirements list with pertinent questions, from the radiologist's perspective, was distributed by Dr. J. Perry, M.D. from the University of North Carolina. The requirements list is shown in Figure 2.1. This list was

this if fig 2.1 page 1

this is fig 2.1 page 2

this is fig 2.1 page 3

intended to define the features desirable in an analysis console. As described in the introduction, a review console is intended to provide a less sophisticated implementation of the capabilities provided by the analysis console. Therefore, this list was used to define the initial requirements for the review console designed in this study. These requirements can be grouped into two categories. First are those requirements that relate to how efficiently a radiologist can perform his functions. These include the speed of image processing and image handling, and the user interface. Second are those requirements that define the ability of a display station to accurately portray diagnostic image data. These include spatial and contrast resolutions, image processing capabilities, and the multimodality display capability.

Efficiency Requirements

Image Processing Speed

Radiologists have been very concerned about any possible increase in the time required to perform their task [11]. Any system that forces the radiologist to be less productive than he is now will probably not be accepted due to economic considerations. Thus, it was important to determine the image recall rate in the film-based system.

To achieve this, staff radiologists at the University of Arizona were observed to determine how they handled images while performing their routine diagnoses. These radiologists were also interviewed to determine what they felt an acceptable image recall rate would be. Most of the radiologists indicated that they would be satisfied if the system could recall an image in 2 seconds or less. This response time did not turn out to be an unrealistic requirement. With a transmission bandwidth of 100 MHz a 2048X2048 pixel image could be transmitted in .5 seconds. Images with less that 2048X2048 pixels could obviously be transferred in less time. A .5 second transmission time would leave 1.5 seconds to resolve access conflicts at the image archive and for other system overhead. Since the 100 MHz transmission rate was achievable with the current technology it was adopted as a design goal.

Image Handling Speed

Another consideration was how much time is expended manipulating an image before a diagnosis can be reached. Each radiologist has an individual preference for how his images are presented. The technologists that operate the film library take this into account when they present images for analysis. A similar capability should be included in electronic display stations. A file should be included that contains information on each radiologist's preference for initial image presentation. Such a capability would eliminate the time consumed by redundant preliminary adjustments.

User Interface

The efficiency of the user interface is of paramount importance. It is important for the radiologist to have direct access to commonly used operations such as contrast enhancement or magnification. The user interface should be able to provide assistance, perhaps in the form of help commands, to an operator who is unfamiliar with some aspects of the display console's operation. However, this assistance should be implemented in such a way that it does not impair the efficient use of the display station by a knowledgeable operator.

Display Accuracy Requirements

The optimum display format is one in which each image is displayed at the physical size, spatial resolution and dynamic range in which it was acquired. The requirements for these three parameters will be described in the paragraphs which follow. The design chosen for implementation in the review console is included at the end of each description.

Physical Size

The physical size requirement can only be satisfied by a display monitor of the proper size. The largest and most common image to be displayed is the 14X17 inch chest radiograph. The 14X17 inch format could be displayed in full size using a 23 inch monitor positioned in the portrait mode (long axis vertical). Such a monitor would provide a 14X19 inch vewing area. Unfortunately, monitors of this size, which simultaneously exhibited the required dynamic range (discussed below) and which could be operated in portrait mode, were not available at the time of this study. High resolution 19 inch monitors that could be operated in the portrait mode and which provided the required dynamic range were available. These monitors permit the display of the 14X17 inch image in an 11.5X14 inch format. The image displayed on the 19 inch monitor is thus 85% of full size. This situation was discussed with several of the staff radiologists at the University of Arizona Medical Center. Their impression was that the minification of 15% was within the normal variation in the size of adult patients seen, and would not present a problem. Thus the decision was made to incorporate 19 inch monitors into the review console design.

Spatial Resolution

The spatial resolution of radiological images ranges from 64X64 pixels for Nuclear Medicine images to an estimated [12],[13] 2048X2048 pixels for chest images. The review console designed in this study can display images with up to 1024X1536 pixels at their full resolution. Smaller images can be either zoomed or enlarged using the continuous zoom capability (COZO). A zoomed image is one in which the magnification of the displayed image is increased, while the borders of the image remain constant. An enlarged image is one in which the borders of the image are allowed to expand as the image is magnified. Images with resolutions of up to 1024X1536 pixels could be displayed using commercial state of the art display monitors.

There were many problems encountered with the display of 2048X2048 images in real time. First, there were no commercial monitors available that permitted the display of 2048X2048 format images. Even if such a monitor had been available, the electronics to drive it could not have been manufactured using current technology. The pixel rate required to refresh a 2048X2048 monitor is in excess of

200 MHz. This number is derived assuming a 30 Hz interlaced display with 2048 active lines/frame, 2048 pixels/line, plus the time lost due to the horizontal and vertical retrace of the electron beam. The fastest 8 bit video DACs (Digital to Analog Converters) available at the time of this study had an update rate of 100 MPixels/second. Clearly the display technology capable of displaying a 2048X2048 image in real time was not available.

However, it is important to accommodate 2048X2048 images since they comprise such a large percentage of those examined. A psychophysical experiment was performed (see Appendix 1) to determine what spatial resolution would be required to retain the information required by a radiologist to perform his analysis. An outcome of this study was that a radiologist could perform a large portion of his work at a resolution equivalent to that provided by a 1024X1024 display format. Increased resolutions would be required only for the examination of detail in local areas. This result formed the basis for the review console's virtual display design. Images with spatial resolutions of up to 1024X1536 could be displayed directly. Images with spatial resolutions between than 1024X1536 and 2048X2048 could be displayed using the review console's continuous zoom (COZO) capability.

The human eye has a spatial frequency response that peeks at about 9 cycles/deg. [14] (Figure 2.2). To achieve the best interpretation of structures in an image, the peak frequency response of the eye and the frequency spectrum of the desired structures must be made to match. Humans achieve this match unconciously by positioning their eyes at a distance appropriate to the structures in the scene that they are viewing.

When the radiologist first views an image using the film-based system, he obtains an overview impression by sitting away from the image. When he wishes to examine detail in some area of an image he moves his eyes closer to the image. By using the continuous zoom capability the radiologist can keep his eyes at a fixed distance from the monitor and use electronic magnification to match the spatial frequency present in the image to that of his eyes.

The review console has a 4 mega pixel memory which is large enough to accommodate a 2048X2048 image. The problem was how to make the 2048X2048 pixel data available to the radiologist using a 1024X1536 display screen. This was accomplished in the review console using the COZO capability in a mode where the displayed image is first minified (displayed using fewer pixels than are available). A full 2048X2048 image is displayed using every other pixel, generating a 1024X1024 pixel display format. In this mode

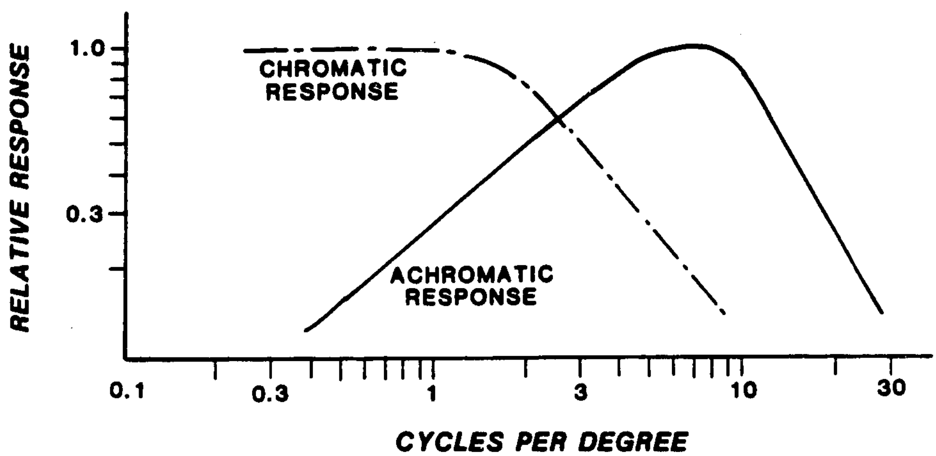


Figure 2.2 Frequency Response of Human Eye

the COZO is operating at a zoom factor of one half. As the radiologist requires more detail in some area and starts to increase the zoom factor from .5 to 1, more of the source pixels are included in the reconstruction of the displayed image. When the operator has reached a zoom factor of 1 (which appears to him to be a magnification of 2) all of the information present in the original image is being displayed. Additional increases in magnification enlarge the image but do not introduce any additional information.

The COZO, which will be described in greater detail below, has thus been able to provide the capability of examining 2048X2048 images without having to develop a true display of that resolution. The validity of this concept is currently being evaluated with a set of psychophysical measurements.

Dynamic Range

There are two aspects to the discussion of the displayed dynamic range in an image. The first is the dynamic range present in the recorded image. The second is the dynamic range that is available for interpretation on the display monitor.

A study to determine the dynamic range present in the different imaging modalities has not been performed;

therefore, it was decided to perform a worst case analysis. This analysis would at least place a bound on the required dynamic range. The density of chest radiographs ranges from a minimum of .04 (fog level) to a maximum of 3.5 (very black). 3200 gray levels are required to display an image that exhibits this range of densities. A display console with 12 bits per pixel can display 4096 shades of gray. 10 to 12 bits are required to display Computed Tomography (CT) images [6], [15]. Ultrasound, Digital Subtraction Angiography and Nuclear Medicine generate images with less than 10 bits of dynamic range. Again, the chest radiograph is seen to set the design criteria for the review console. Therefore, a system designed to accommodate images with up to 12 bits of dynamic range would satisfy all known digital image acquisition devices.

The human eye can only distinguish about 16 shades of gray in a complex image [16]. This does not imply however, that a 4 bit system would be adequate for display. Additional gray levels are required to avoid contouring in regions with subtle changes in gray scale [17]. The review console designed in this study has an 8-bit capability allowing it to display 256 shades of gray. A contrast mapping look-up table (LUT) is provided to allow the viewing of any portion of the 4096 levels possible in the input image on the 256 level display monitor.

A monitor was required that would provide the wide dynamic range described above using a long persistence phosphor. The wide dynamic range was desirable in order to simultaneously display as many gray levels as possible. The long persistence phosphor was required to eliminate inter-field flicker. These turned out to be conflicting requirements. The long persistence phosphors cannot generate the same peak luminance levels found with some of the shorter persistence phosphors. It was determined that this reduction in the number of perceived gray levels would be unacceptable. The decision was made to incorporate a monitor with a P104 medium persistence phosphor and to reduce the flicker by increasing the frame rate from 30 Hz to 40 Hz.

Multimodality Display Capability

The phrase 'multimodality display' implies the capability of analyzing images from many different imaging sources on the same display monitor simultaneously. These images are assumed to have different spatial and contrast resolutions. Each image must be capable of being manipulated independently of the others. It must also be possible to add, remove, and move images on the display monitor. It must be possible to perform image processing operations on each of the images independently. In certain types of studies a time sequence of images is acquired. Cardiac wall motion analysis is an example of such a study. A movie loop display capability is required to exhibit this type of data.

The multimodality display capability is provided to mimic the panel of light boxes in a radiologist's current film-based environment. That panel of light boxes provides a large area for the display of images. All of the films relating to a particular diagnosis can be displayed simultaneously, regardless of their size or format. A display console cannot provide a display area comparable to that available using light boxes. Because of this, features were incorporated into the review console that allow the radiologist to perform functions that are equivalent to those performed with the panel of light boxes.

In a film-based environment the radiologist begins his analysis by hanging all of the images that have been acquired on light boxes. From these he selects those images that provide the information required for his diagnosis. An equivalent operation can be performed using the review console's 'gallery mode' display capability. The gallery mode allows the display of a large number of images at a very reduced size. From this display the radiologist can select the images that he wishes to examine in more detail.

The radiologist may select more images for examination than can be accommodated on the CRTs. Magnification, minification and overlay capabilities have been included in the review console design to make optimum use of the limited image display area. Magnification and minification can be used to adjust the size of the images. The review console has a move capability that allows the operator to position the images anywhere on the screen. The overlay capability can be used to form what appears to be a stack of images. Any image in the stack can be brought to the top of the stack for analysis. The selection of which image is to appear on the top is accomplished using the touch screen capability.

Movie Loops

In many imaging studies, in particular imaging of the vascular system, the motion of the organ contains the diagnostic data [8]. To allow this type of data to be visualized, a movie loop capability was included in the review console design. The number of images acquired and the acquisition rate vary depending upon the study performed. DSA studies are acquired in a 512X512 format with as few as 10 frames acquired at one second intervals. Adult Cardiac studies consist of hundreds of frames acquired at 60 frames per second. Pediatric cardiac studies are acquired at rates exceeding 90 frames per second.

The review console was designed to include a movie loop capability. Each display memory has a replicative zoom that provides a factor of 4 magnification. This allows a 256X256 image to be displayed in full size. Twenty four of these 256X256 images can be held in each of the four display memory buffers. Using two of these buffers allows the construction of a movie consisting of up to 48 frames. The rate at which the images are presented and the direction (forward or reverse) are under the control of the radiologist. He is also provided the capability to edit the movie loop by defining the initial and ending frames. This capability allows him to select a single heart beat from a longer sequence.

Image Processing Capability

Image processing functions are required for image enhancement and image analysis operations. Image enhancement operations are used by the radiologist as an aid in the visualization of obscure pathology. Image analysis operations are available to provide a quantitative analysis of features within an image.

<u>Image Enhancement Operations</u>. Contrast modification operations, image subtraction, spatial filtering, and image magnification (COZO) are the most common image enhancement operations. They are also operations that are frequently performed with the film-based imaging system. Thus, they represent a minimum image enhancement capability for a display station designed for radiology.

As mentioned in the introduction, contrast enhancement is accomplished in the film-based system by a technique called "bright lighting". Bright lighting consists of using a high intensity lamp to back light opaque regions in a radiograph. Electronic image display systems perform an equivalent operation by altering the distribution of the displayed image gray levels. This operation is usually implemented in hardware by a look-up table (LUT) placed between the video refresh memory and the digital to analog converter.

Image subtraction is often used to eliminate high contrast structures that conceal subtile features of interest. A quantity 'B' can be subtracted from 'A' by first negating B add adding it to A. This is exactly how film-based subtraction is accomplished. A negative is first made of the image to be subtracted. Then the two images are manually aligned and contact printed to form their sum (which is the difference of originals). Digital image subtraction is performed by direct arithmetic subtraction on a pixel by pixel basis. Sometimes the patient or the imaging system will have moved and the image pairs will need to be registered. Correlation techniques can then be used to align the images with subpixel accuracy.

Low pass spatial filtering, or image smoothing, is used to reduce the presence of random noise in images. The simplest image smoothing algorithms consist of replacing a pixel by the average of the pixels in its neighborhood. More sophisticated algorithms weight the pixels before they are summed. High pass filtering, or edge enhancement, is used to enhance the visibility of small structures in the image. It also has an undesirable characteristic in that it enhances the visibility of the noise present in the image. Simple high pass filter operations can be performed using a technique analogous to unsharp masking. In unsharp masking two images of the same area are used. The second image is simply a blurred version of the first image. The second image (which contains only low frequency data) is subtracted from the first image (which contains both high and low frequency data) leaving an image containing only the high frequency data. Electronic imaging systems can perform an edge enhancement operation using the same technique.

As mentioned above, the spatial resolution of medical images ranges from 64X64 to 2048X2048 pixels. To examine the lower resolution images, as well as the detail in higher resolution images, a magnification capability is required. The simplest way to achieve magnification is by pixel replication. Using this technique, each pixel is replicated a specified number of times in both the horizontal and vertical directions. There are two disadvantages to this approach. First, the lowest zoom factor that can be achieved is a factor of 2, i.e. each pixel replicated once. Radiologists typically desire magnifications of much less than 2. They also desire the image magnification to change smoothly. If the image jumps from a magnification of 1 to 2 the radiologist loses his appreciation of where the zoomed image is in relation to the whole. The second disadvantage of replicative zoom is that the pixel structure becomes apparent for magnifications larger than 3. The human eye-brain combination has its own spatial frequency response that is very sensitive to edges. As the pixel structure becomes apparent, the radiologist's ability to discern small contrast differences is diminished. The disadvantages inherent in pixel replication can be avoided by estimating the pixels required to achieve a desired magnification with an interpolation kernel.

The ideal interpolation kernel is the sinc function. The sinc function is physically unrealizable due to its infinite extent. It does however, provide a basis for the comparison of other interpolation kernels [18] (which serve as approximations of the sinc function). Pixel replication, or a zero-order hold, uses the simplest interpolation kernel. Here the additional pixels required to achieve a desired magnification are computed by convolution with a rectangular function. The second-order interpolation kernel is bilinear interpolation. Here the additional pixels required to achieve a desired magnification are estimated from their nearest neighbors by convolving them with a triangular-shaped interpolation function. Higher order interpolation kernels, such as the cubic B-spline, offer more accurate approximations of the sinc function but are difficult to implement in hardware. Bilinear interpolation was chosen for implementation in the review console. Both of the disadvantages evident in pixel replication were substantially overcome using this technique. First, the requirement for non-integer magnifications, which permits fractional zoom factors, can be implemented using this technique. Second, by using this technique, visible pixel structure is eliminated for all but the largest magnifications. Minification is required to view an entire 2048X2048 pixel image on the 1024X1536 pixel format display screen. In this mode a magnification factor of .5 is required to display an overview of the entire image. The radiologist must then be able to magnify the image to examine detailed structures. When he has increased the magnification to what appears to him to be a factor of two, the display actually contains a full resolution image at a hardware magnification of 1. A radiologist currently performs the same function by moving his eyes closer to the film. Fractional bilinear interpolated minification, similar to that used for magnification, is required to achieve a smooth transition from minification to magnification. Minification may also be required to allow the simultaneous display of many medium resolution images on the same screen. The review console magnification hardware has been augmented to allow it to also perform this minification function.

Image Analysis Operations. The image analysis operations required of a radiographic display station include density and distance measurements. Density measurements can be performed by having the operator position a cursor over a target pixel. The system can then compute the actual density from the gray level in the image. The average density over an area can be obtained by performing a region of interest calculation. To accomplish this, the operator is instructed to trace a border around the desired area. The computer can then compute a histogram of the gray levels within the defined region. Statistics that describe the distribution of densities within a region can be computed from this histogram.

Distance measurements can be performed by having the operator define the end points of the structure to be measured. The computer can then compute the distance between the end points in pixels. Using the image formation data that accompany the radiograph, the computer can translate the distance in pixels to a decimal length.

CHAPTER 3

REVIEW CONSOLE DESIGN

A block diagram of the review console architecture designed as part of this research project is shown in Figure 3.1. This console is composed of the following six interrelated subsystems:

1. Base Memory
2. Image Processor
3. Display Memory
4. Mass Storage Subsystem
5. Network Interface Module
6. User Interface

This discussion will begin with a brief overview introducing each of the above subsystems and describing the way in which they interact. The overview will be followed by a detailed description of the design of each of the subsystems.

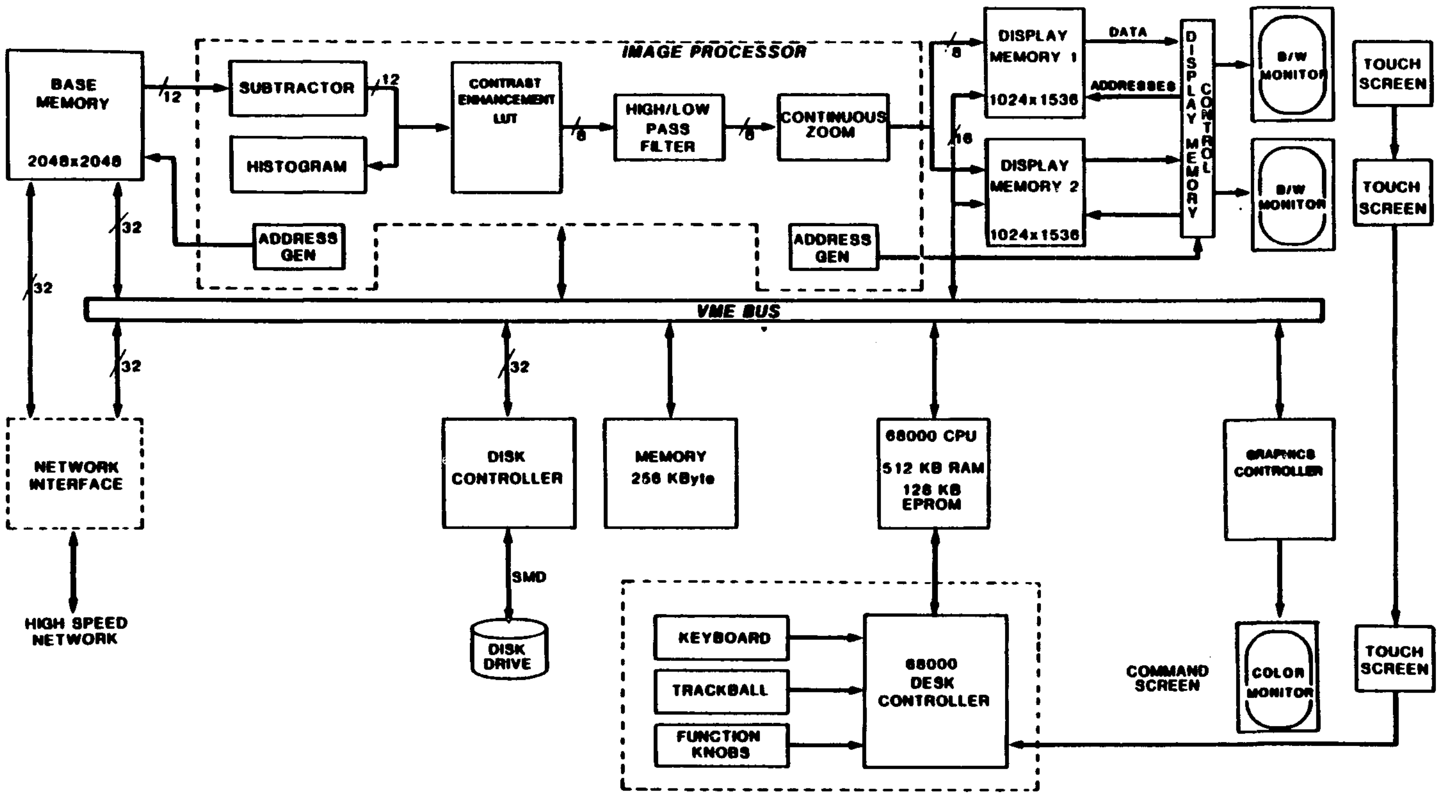


Figure 3.1 Review Console Block Diagram

Subsystem Overview and Interaction

The review console includes a unique architecture that was designed specifically to implement the multimodality display requirement defined above. This architecture can be broken down into two sections. The first section consists of those components that make up the high speed pipelined image display channel, operating at video rates. The image display channel consists of the base memory, the image processor and the display memory components, all of which appear above the VME bus in Figure 3.1. The second section includes the nonvideo rate subsystems required for image data management and console operation. The transfer of image data to and from the PACS is accomplished by the network interface module. The mass storage subsystem provides a local image storage capability for the review console. The user interface subsystem provides operator control of the review console as well as an interface to the rest of the PACS.

An image that has been selected for viewing must first be loaded into the base memory. The base memory is required to store the images at their full resolutions. Access to the base memory by the network interface module and the image processor is provided by dedicated high speed data ports. Access to the base memory by the CPU or the mass storage subsystem is provided by a low speed port to the VME bus. The source of images to be loaded into the base memory can be either the local mass storage subsystem or another node of the PACS. Images available on the local mass storage subsystem are transferred to the base memory over the VME bus. Movement of images to or from other nodes in the PACS is accomplished through the network interface module. The network interface module accepts ACR-NEMA format data packets, extracts the desired image data, and enters it into the base memory through a dedicated data channel.

The local mass storage subsystem is also used as a cache for images. This cache is required when an attempt is made to display more images than can be accommodated by the base memory. As an example, consider the simultaneous display of an AP (Anterior to Posterior i.e. front to back) image and a lateral chest image on the two display monitors. This would be accomplished by first loading the 2048X2048 AP image into the base memory and having the image processor write it to one of the display memories. The next step would be to load the lateral image into the base memory and write it to the other display memory. At this point the base memory is full and the lateral image has yet to be loaded. The solution is to preserve the contents of the base memory on disk before loading the lateral image. Using this approach much more data can be displayed than can be accommodated in the base memory alone.

The image processor is responsible for reading a desired image from the base memory, processing it, and writing it to the display memory subsystem. The image processing performed during this transfer can include image subtraction, contrast modification, spatial filtering, magnification, and minification. A histogram of the gray levels present in a user defined region of the image can be computed if desired. The image processor generates the addresses required to read an image in the correct format from the base memory. It also generates the addresses required to determine where and in which display memory the processed image is to be written.

The display memory subsystem consists of two 1024X1536 video refresh memories and their associated display monitors. Each location in a video refresh memory corresponds to a pixel on the display monitor associated with that memory. During image processing, only those display memory locations which correspond to the image being updated are modified. The addressing of the display memory is not modified to accomplish roam and zoom as in conventional image display systems. In the review console architecture these functions are accomplished by the image processor. In this manner each image on the display monitor can be processed independently, achieving the multimodality display design goal.

The user interface subsystem provides the user with control over the review console and access to the rest of the PACS. An effort has been made to make the interface feel natural to the user by simulating the functions performed using the film-based system wherever possible. Touch screens have been incorporated on the display monitors and the command screen. Image selection is accomplished by simply touching the desired image simulating the familiar act of pointing. Commands are entered by touching the desired label in a color coded set of menus. A standard computer keyboard has been included for the entry of text such as patient names. Knobs are the most natural interface for certain operations. Individual knobs have been included for each function to eliminate the confusion caused by redifining controls. Clockwise rotation causes an increase in all cases, similar to the familiar volume control on a radio. Knobs are provided to control contrast, brightness, magnification and movie framing rate. A trackball is provided to allow precision delineation of regions of interest.

The vast majority of the multimodality display requirements are imposed upon the image display channel. It must present an uninterrupted display containing multiple images of varying spatial and contrast resolutions to the radiologist. It must allow each of the images that comprise the display to be processed independently. A description of how each of the subsystems has been designed to achieve the multimodality design objective will now be presented.

Base Memory

The base memory is the first component of the image display channel. It provides the interface between the image data at its full resolution and the review console's image display channel. A block diagram of the base memory is shown in Figure 3.2. The base memory is an 8 megabyte triple ported memory with independent data and address busses dedicated to each port. The three independent ports were required to allow the image processor to operate independently of any data transfers that might be occurring on the other two ports.

The base memory can store images of arbitrary sizes and aspect ratios from multiple 8X8 pixel images up to a single 2048X2048 pixel image in increments of 8 pixels. The bits within each pixel in the base memory are assigned as a function of the dynamic range and graphics requirements of the input image. The least significant 8 bits (0:7) of each pixel are reserved for image data. The most significant 4

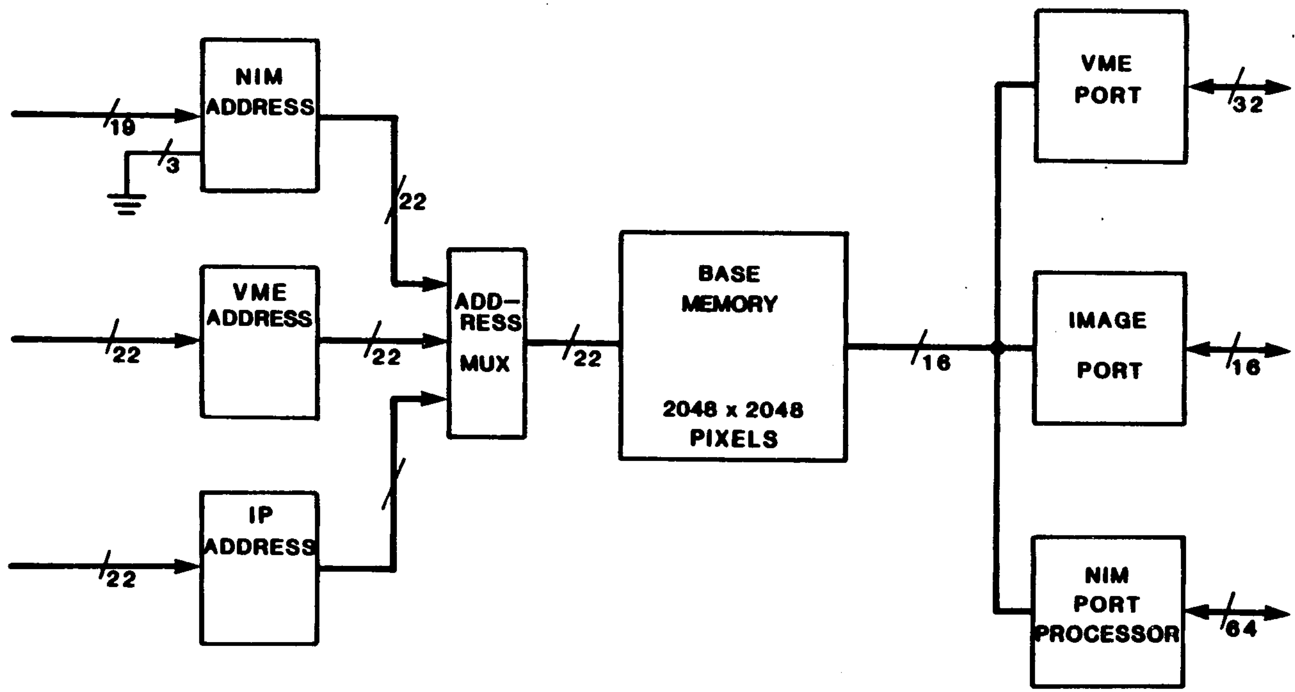


Figure 3.2 Base Memory Block Diagram

bits (12:15) of each pixel are reserved for graphics overlays. The remaining 4 bits (8:11) of each pixel can be assigned to either image data or graphics data as required by the input image. This assignment can be made for each image independent of other images present in the base memory. This allows the base memory to accommodate images with up to 4096 shades of gray. It also allows from 4 to 8 bits to be assigned as graphics overlays.

The first data port consists of a read/write port to the VME bus that allows the CPU and disk subsystem to access the image data. The second port is a high speed read-only port that allows the image processor access to the image data. The third port is another high speed read/write port used by the network interface module.

The VME port permits read and write access of either word (16 bit) or long word (32 bit) data. The word access has been provided to allow the CPU access to data in pixel format. The ACR-NEMA standard allows images to be represented by either unsigned binary or two's complement integers. The VME port provides automatic sign extension of the most significant data bit when 2's complement data are present. A bit in the control register specifies that 2's complement data are present and identifies which bit is to be sign extended. The graphics bits can also be masked to allow access to just the gray scale image data. This has been included to allow the CPU to process the image data independent of its original format.

The image processor port is a 16 bit read-only port. The base memory has been implemented using a high degree of parallelism to achieve the 10 megapixel transfer rate required by the image processor. Each request by the image processor results in 8 pixels being transferred from the base memory in 550 nsec. High speed buffer memory is provided within the base memory to allow sequential access of the image data. The image processor has the capability of performing image subtraction between two images located in the base memory. The high speed buffer memories also allow access to alternate pixels in the 8 pixel groups.

The network interface module port is a read/write port that is 8 pixels wide. A transfer can occur over this port every 550 nsec. At this rate the network interface module achieves the design objective of being able to transfer a 2048X2048 image in less than .5 seconds.

The ACR-NEMA standard allows graphics overlay data to be stored either as part of the gray sale data or in separate graphics bitplanes. To accommodate this graphics data, the network interface module must be able to update the base memory graphics bit planes without disturbing the image data already present. This is accomplished by treating each bit in the memory as one of 16 possible bitplanes. The write logic for each of these bitplanes can be enabled individually. The network interface module can transfer the image data into the base memory first. It can then disable the write logic to the image data bit planes and write the graphics overlays.

Image Processor

As shown in Figure 3.1, the image processor is the second component of the image display channel, positioned between the base memory and the display memory. The image processor is responsible for controlling the transfer of image data from the base memory to the display memory. Certain image processing operations are performed on the image data during this transfer. A block diagram of the image processor is given is Figure 3.3.

The data path from the base memory to the image processor consists of 12 bits for gray scale data and 8 bits for graphics data. The method of assigning the 16 data bits in the base memory to represent either image or graphics data has been explained in the preceding section on the base memory. The data path from the image processor to the display memory subsystem consists of 8 data bits and one graphics overlay bit. The M68000 CPU controls the operation of the image processor by loading values into the desired

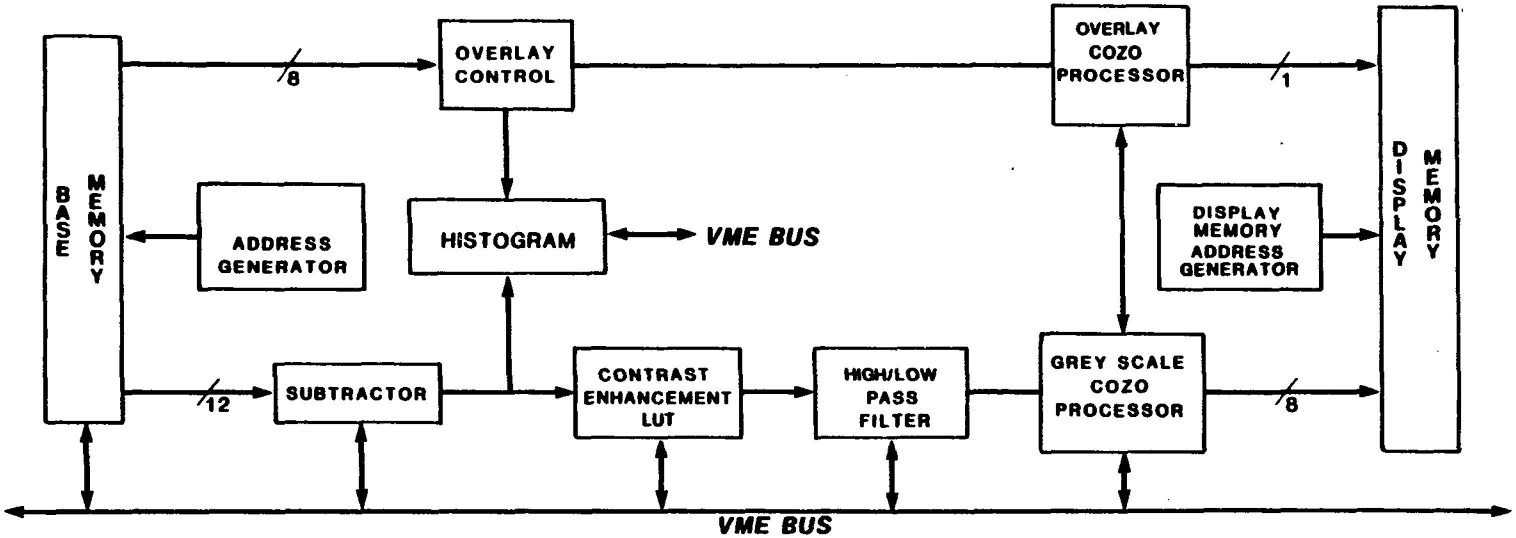


Figure 3.3 Image Processor Block Diagram

registers and tables. The image processor is a pipelined architecture that performs the following sequence of functions on the gray scale data:

1. It generates the sequential addresses required to read the source image data from the base memory.
2. It performs image subtraction between two images that reside in the base memory.
3. It performs contrast enhancement modifications on we image data.
4. It performs spatial filtering on the image data.
5. It performs a continuous zoom (COZO) to magnify or minify the displayed image.
6. It generates the sequential addresses required to write the processed image data to either display memory.
7. It computes a histogram of the gray levels contained in a selected region of the image identified by the Region of Interest (ROI) overlay bit.

Functions 1 through 6 are concerned with manipulation of the gray scale image data. The last function permits the computation of certain image statistics that can be used by the CPU when executing various image processing algorithms.

Each of these processing functions will be described in detail following a description of their interactions.

Function Interaction

The sequence of processing operations performed by the image processor on the gray scale image data is critical to its successful operation. The reasons for the sequence included in this design are:

1. The subtraction operation must be the first operation performed by the image processor. For the subtraction to be successful, it requires the full dynamic range present in the source images.
2. The contrast manipulation LUT (Look-up Table) must precede the spatial filtering operation for two reasons. First, contrast mapping needs to operate on the full dynamic range of the input data to visualize subtle contrast variations. Second, the filtering operation can reduce the number of gray levels in the output image. This reduction in the number of gray levels could cause contouring to be visible in the image if contrast enhancement were performed after a spatial filtering operation.

3. The filtering operation must be performed before the zoom operation. The zoom changes the spatial frequency (in cycles/pixel) of the displayed image. If the filter operation were performed after the zoom, the effect of the filtering operation would change as a function of the image magnification.

A computer program was written that modeled the image processor architecture (see Appendix 2). This model was used to verify the interaction between the image processing operations. It was also used to identify problems with the integer math used in the spatial filter and continuous zoom operations. The 'pixel pitching' algorithm (described in the section on Continuous Zoom) required for minification was developed using this model.

The base memory can contain images with up to 8 graphics overlay bits defined. This capability was included to accommodate as many different types of images as possible. The current implementation of the review console uses monochrome monitors for the display of image data. With a monochrome monitor only one overlay can be uniquely identified. Because of this limitation, the image processor converts the 8 graphics bitplanes into 2 bitplanes. One of these is used as the region of interest bit. It is used to selectively enable the histogram generator over user identified regions. The other bits are combined using a logical OR function and appear as the ninth bit written to the display subsystem.

Input Pixel Address Generator

A fundamental requirement for a PACS image display workstation is the multimodality display capability. As stated previously, a multimodality display workstation must be able to display images of different physical sizes and dynamic ranges. To satisfy this requirement, the image processor was designed to accommodate images of any size, from 8X8 pixels to 2048X2048 pixels. The first section of the image processor is the base memory address generator shown in Figure 3.4. It consists of the logic necessary to generate the address and control signals required to read a desired image from the base memory.

The central component of the address generator is the pixel address counter. The pixel address counter contains the address in the base memory of the next pixel to be processed. It must be 22 bits wide so that it can address the 4 million pixels present in a 2048X2048 image. As part of the image processor initialization sequence, the CPU loads the address counter with the address in the base memory of the first pixel to be transferred. Subsequent

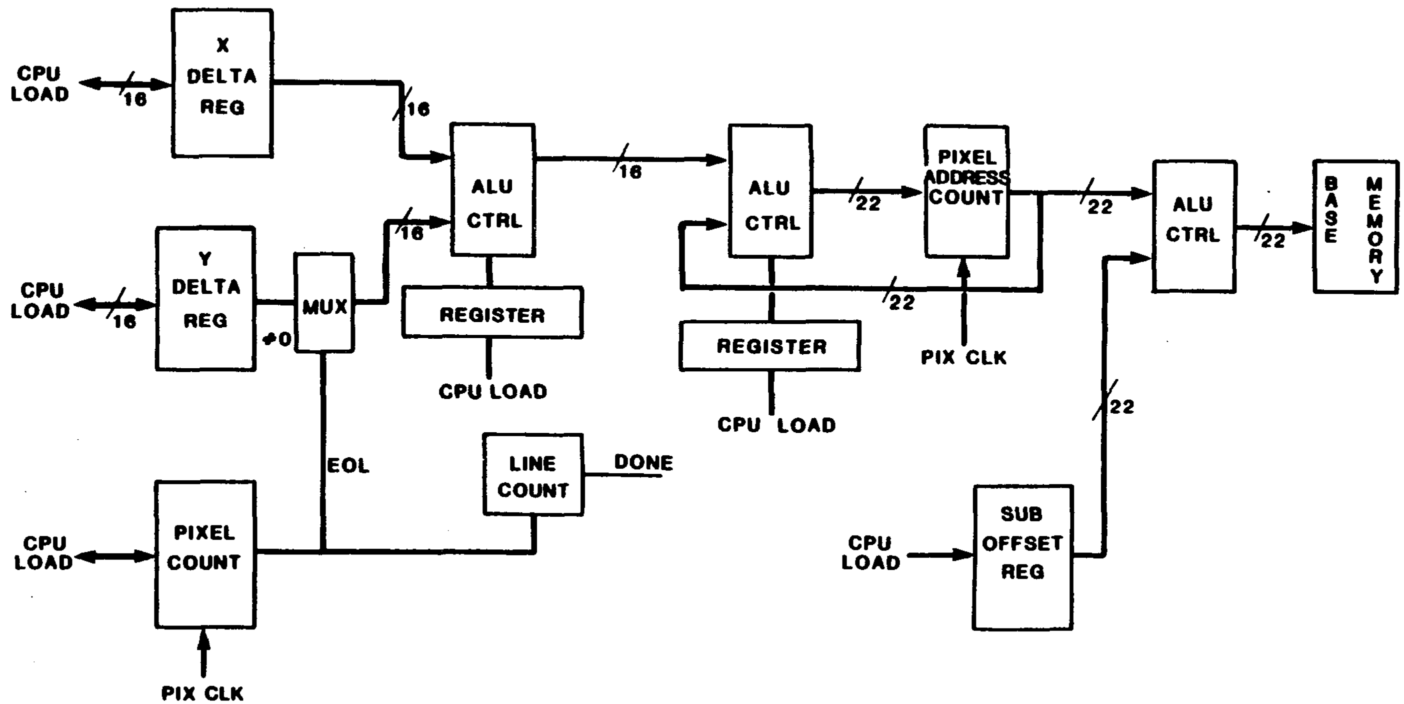


Figure 3.4 Base Memory Address Generator Block Diagram

address locations are determined from the sum of the current location and the offsets generated in the X Delta and Y Delta registers.

The X and Y Delta registers are initialized as part of the image processor initialization sequence. The offsets generated by the X Delta register are used to access pixels sequentially along a line of an image. The X Delta is always loaded to present a positive one as input to the ALU. In a normal left to right display, the control input of the ALU is set to add the X Delta to the current pixel address. If a left to right image reversal (X flip) is requested, the ALU is set to subtract the X Delta from the current pixel address. Depending upon the roam and zoom requested, the last pixel required by the image processor comes from some position along the line, in general not the last pixel of the line. The Y Delta register contains the offset needed to position the pixel address counter to the beginning of the next line to be transferred. The offset contained in the Y Delta register is only added at the end of each line. To accomplish this, a 2 input MUX is inserted in the data path between the Y Delta register and the ALU. One input to this MUX is the contents of the Y Delta register. The other input is the number 0. While sequential pixels along a line are being accessed, a zero is input to the ALU. At every end of line (EOL), the Y Delta input to the MUX is enabled allowing the offset to be added. A counter that contains the number of pixels to be read generates the EOL signal. Another counter that contains the number of lines to be read is used to stop the image processor at the end of the image.

As mentioned above, the CPU loads the pixel address counter with the first pixel to be transferred. In a normal display this is the pixel that appears in the upper left corner of the displayed image, and the X delta register increments the pixel address counter forwards to the last pixel on the line. In a display with an X flip, the first pixel addressed is the last pixel on the line and the X delta register is used to count backwards towards the first pixel. In a display with a Y flip, the first pixel addressed is the first pixel on the last line and the Y delta register is used to count backwards to the first line. In a display with both X and Y flips enabled, the first pixel addressed is the last pixel on the last line.

The last component of the base memory address generator is the subtract offset register. This register is loaded with the offset, in pixels, between the first image and the one that is to be subtracted from it. In this mode of operation, two read operations are performed on the base memory for each value computed in the pixel address counter.

This allows subtraction to be performed independent of requests for roam, zoom, or X and Y flips.

Image Subtracter

The second section of the image processor is the image subtracter as shown if Figure 3.5. Image subtraction is provided to permit visualization of low contrast details that have been obscured by overlaying structures. An example is in Digital Subtraction Angiography (DSA), where a pair of images is recorded at exactly the same imaging position. A reference image is first recorded prior to the administration of a contrast medium. A second image is recorded after the contrast medium has been introduced. The obscuring structures, such as bones and soft tissue, are present in both images while the contrast medium appears only in the second image. By subtracting the first image from the second, an image of only the vessels that contain the contrast medium is left.

The method of generating the addresses of the conjugate points in the two images to be subtracted has been presented above. The image data points are read in pairs from the base memory. The first pixel is clocked into the N-1 pixel register and appears at the A input of the ALU. The second pixel appears at the B input of the ALU. The ALU

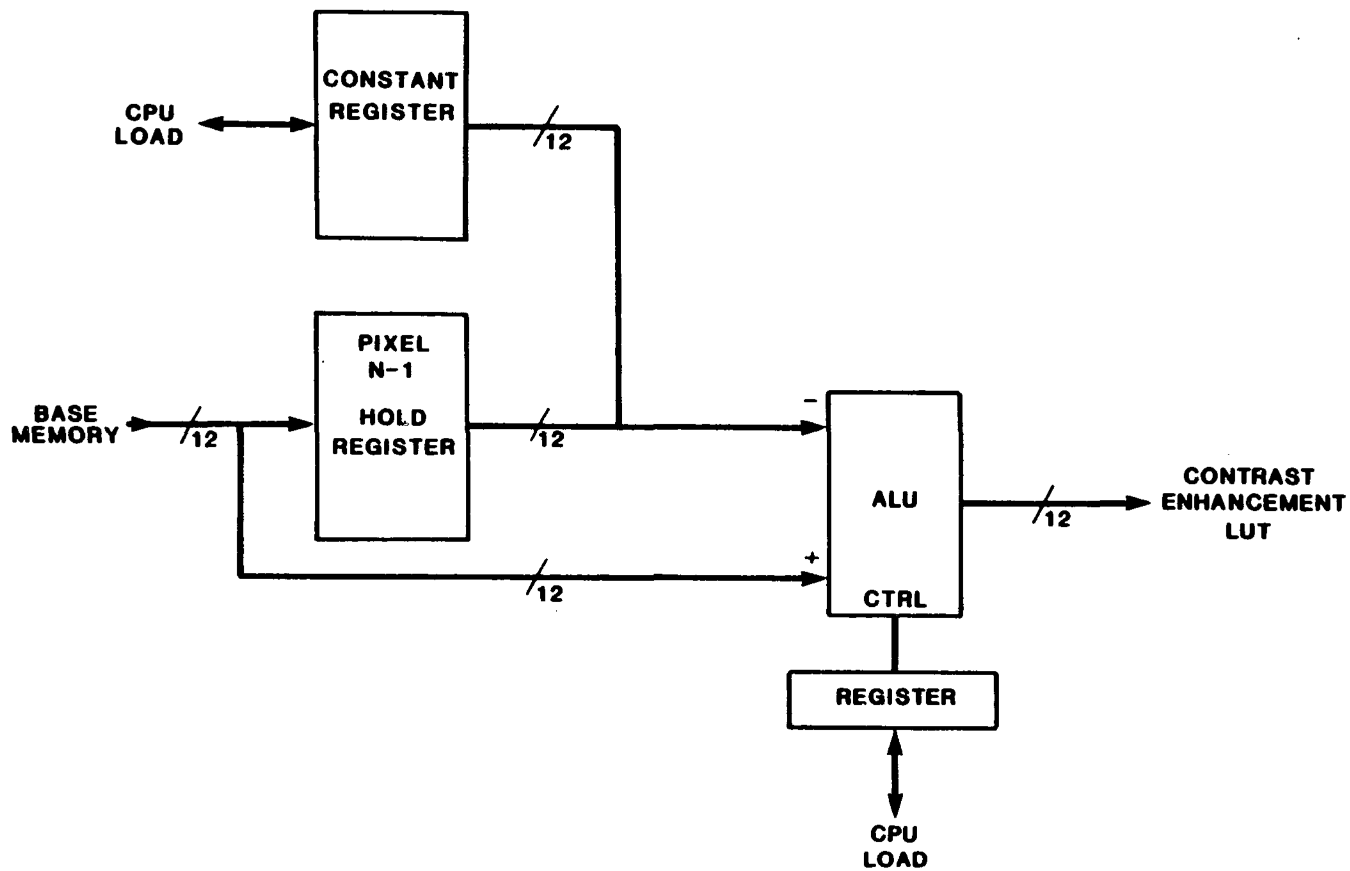


Figure 3.5 Image Subtractor Block Diagram

can be set up to perform arithmetic or logical operations between the two images.

As indicated above, the most common operation is subtraction. The B input to the ALU can come from a constant register rather than from the N-1 pixel register. This allows logical or arithmetic operations to be performed between input image data and a constant (e.g. add a constant to an image). The output of the ALU forms the input to the contrast enhancement LUT.

Contrast Enhancement

The contrast enhancement LUT forms the third section of the image processor. Contrast enhancement is a requirement in all electronic medical display systems. The terms 'window' (which corresponds to contrast) and 'level' (which corresponds to brightness) are used in the medical community to describe this capability. The display workstation includes a look-up table (LUT) that is used for contrast modification.

The LUT is 4096 words long, each word being 8 bits deep. This LUT is actually used to perform three functions. The first is contrast enhancement. The operator instructs the CPU to compute a set of values that will perform the desired contrast modification and loads it into the LUT. If the user has selected gray scale reversal, the values loaded into the LUT are computed to produce that effect. An aspect of the multimodality display requirement is the ability to simultaneously display images that have different dynamic ranges. The second function performed by the LUT is the transformation of the input image dynamic range to the 8 bit value required by the subsequent stages of the image processor. In this way input images with dynamic ranges from 1 bit to 12 bits can be accommodated. The third function that can be performed in the LUT is the conversion of unsigned binary data to 2's complement data. The subsequent stages of the image processor expect the data to be in 2's complement format.

Spatial Filter

A spatial filter is included as the fourth section of the image processor. Much of the wide dynamic range encountered in medical images is generated by large structures which are not of diagnostic interest. High pass filtering of the image data can be used to reduce the intensity of these structures. This will allow more of the diagnostically significant anatomy to be visible simultaneously on the CRT.

A block diagram of the spatial filter is provided in Figure 3.6. The filter consists of a 32 delay one-dimensional finite impulse response (FIR) filter. It was constructed using 16 of the TDC1028 VLSI Digital Filter Building Blocks manufactured by TRW. Each of these ICs can provide 8 stages of filtering using 4 bit data and 4 bit filter coefficients. In the review console design, 8 bit data with 8 bit filter coefficients were used such that 4 ICs were required for each 8 delays. The 32 stage spatial filter consists of 4 of these 8 delay stages in cascade, resulting in the use of 16 of the TDC1028 ICs.

The spatial filter can be used to perform any filter function that can be accommodated in 32 delays. The most common use for this type of filter is to provide high pass and low pass filtering of the image data. A symmetric FIR filter is desirable for this application since it produces linear phase response which will not introduce any ringing into the image. To implement a symmetric filter, an odd number of coefficients is required. An odd number of coefficients is achieved by always forcing the coefficient of the last stage of the filter to be 0. The image processor timing is designed such that the center of the filter is 16 delays from the start.

The system default when an image is first displayed is to perform no spatial filtering. This is accomplished by

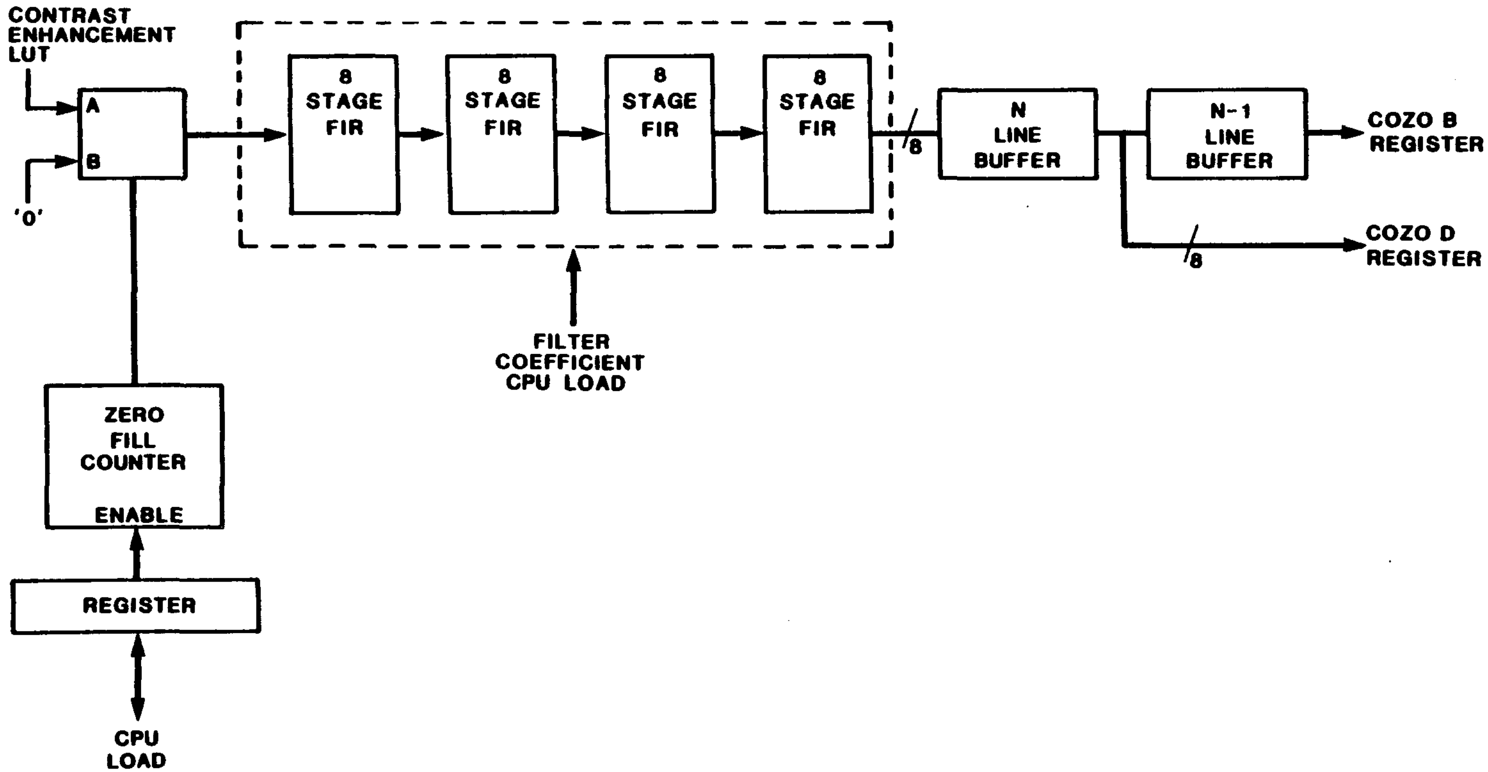


Figure 3.6 Spatial Filter Block Diagram

loading a Dirac function into the filter coefficient registers. The coefficients for a Dirac response consist of zeros in all the coefficients except the one at the center (at delay 16) which is assigned a value of one. The operator can specify that a high pass or low pass filter operation is to be performed by rotating the filter control knob on the user desk. Clockwise rotation indicates that high pass filtering is desired while counterclockwise rotation indicates that low pass filtering is desired. As the knob is rotated further in either direction the intensity of the selected filter function is increased.

The spatial filter must be initialized at the beginning of each line. Two initialization schemes were incorporated into the spatial filter. The first is used when the first displayed pixel corresponds to the first pixel of the line in the base memory. In this case there are not enough input pixels to fill the filter before the first output pixel is required. Therefore, the first 15 data values are initialized to zero. The 'B' inputs to the MUX, located at the input to the filter, are tied to ground and permit the insertion of these 15 zeros. The second initialization scheme is used when the roam and/or zoom place the first displayed pixel at least 16 pixels from the start of the line in the base memory. In this mode the image processor generates addressing to read the 15 pixels that proceed the first one that is desired for display. In this way the entire 31 stages of the FIR filter will contain valid data when the first pixel value is computed.

The output of the FIR filter is directed to two one line buffer memories that form the input to the continuous zoom (COZO) processor. When a magnification greater than one is desired, the COZO processor must compute more than one output line for each input line. Thus, the first four sections of the image processor must be halted while the COZO processor is computing the extra output lines. The two buffer memories provide the COZO processor with the data it requires to interpolate the additional lines of output.

Continuous Zoom

A problem with previous attempts to electronically display medical images has been the inability to match the spatial frequency of the image structures to that of the observer's eye. In a film-based system this match is accomplished by moving the radiologist's eyes in relation to the film. In early electronic systems hardware zooms were attempted. These systems could only provide integer zooms, i.e. factors of 2 or 3 etc. An increase in size by a factor of 2, the minimum step for these systems, was more than a radiologist desired. With a magnification factor of 2 about

75% of the image was not visible. The review console attempts to solve this problem with a continuous zoom capability. Continuous zoom provides zoom factors from a minification of 1/32 (zoom factor of .031) to a magnification of 2048 (one pixel mapped to the entire output screen) in steps of .031.

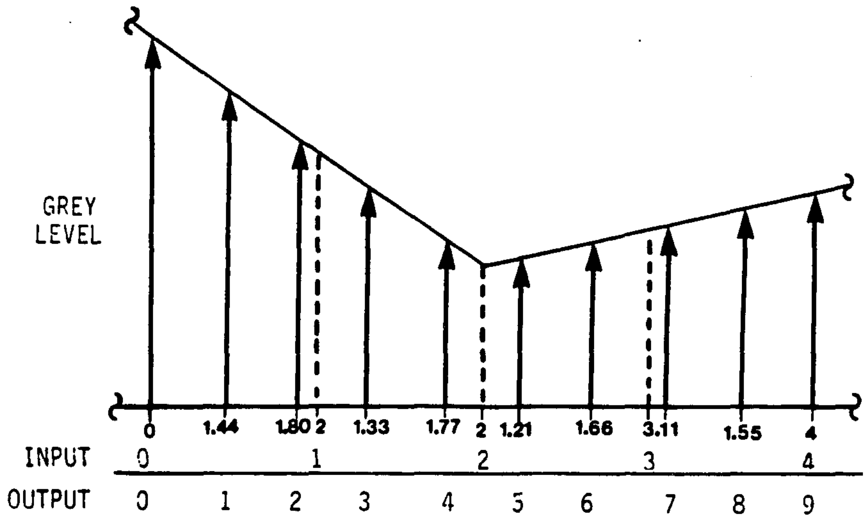
The continuous zoom processor was designed to provide an accurate reproduction of the input image at all magnifications. As described in the preceding chapter, bilinear interpolation is used to compute the output pixel values. Bilinear interpolation was chosen to reduce the "blocking" introduced by pixel replication.

Figure 3.7 is an example of how an output pixel G, at coordinate (X,Y), is estimated from its neighbors using bilinear interpolation. In this figure, X is the fractional distance from A to G measured in the horizontal direction. Y is the fractional distance from A to G measured in the vertical direction. The value of the pixel G is estimated from its four neighbors by the successive application of linear interpolation. Linear interpolation is first applied in the horizontal direction and then in the vertical direction. The pseudo pixel E, at location (X,0), is first computed from its neighbors A and B (Equation 1). Similarly, the pseudo pixel F, at location (X,1), is computed from its neighbors C and D (Equation 2). The this is fig 3.7

estimated value of G can now be computed by applying linear interpolation in the vertical direction between pseudo pixels E and F (Equation 3). The combination of these three equations yields an expression that relates the output pixel G to its input neighbors and the offsets X and Y (Equation 4). The equation, in this form, was suitable for hardware implementation.

The operation of the COZO processor will be described with the aid of an example and Figures 3.8, 3.9 and 3.10. Figure 3.10 is a block diagram of the COZO processor hardware used to compute Equation 4 described above. The inputs to the COZO processor consist of the image data and the multiplier coefficients that correspond to the offsets X and Y. A block diagram of the hardware designed to compute the X and Y coefficients is presented in Figure 3.9. An example calculation that illustrates the states present in the coefficient processor is given in Figure 3.8.

An example calculation will be used to describe the operation of the X and Y coefficient processors. The example will be given for the X processor, the operation of the Y processor being identical except for the method of initialization. This example has been formulated for an image magnification of 2.25. To achieve a magnification of 2.25 nine output pixels must be computed for every four input pixels (9/4 = 2.25). The dashed lines in the plot of



CYCLE	X COEF DELTA REG (1/magnification)	PIPE REG	ADDER OUTPUT	X COEF	LINE BUFFER MEMORY SHIFT	VALUE OF COZO INPUT REG'S A	VALUE OF COZO INPUT REG'S B
0	.4443	.9995	1.4438	.996	1		0
1	.4443	.4438	.8881	.441		0	1
2	.4443	.8881	1.3324	.879	1	0	1
3	.4443	.3324	.7767	.332		1	2
4	.4443	.7767	1.221	.773	1	1	2
5	.4443	.221	.665	.219		2	3
6	.4443	.665	1.109	.664	1	2	3
7	.4443	.109	.554	.105		3	4
8	.4443	.554	.998	.055		3	4
9	.4443	.998	1.442	.996	1	3	4

X COEF PROCESSOR STATES

Figure 3.8 X Coefficient Processor Example

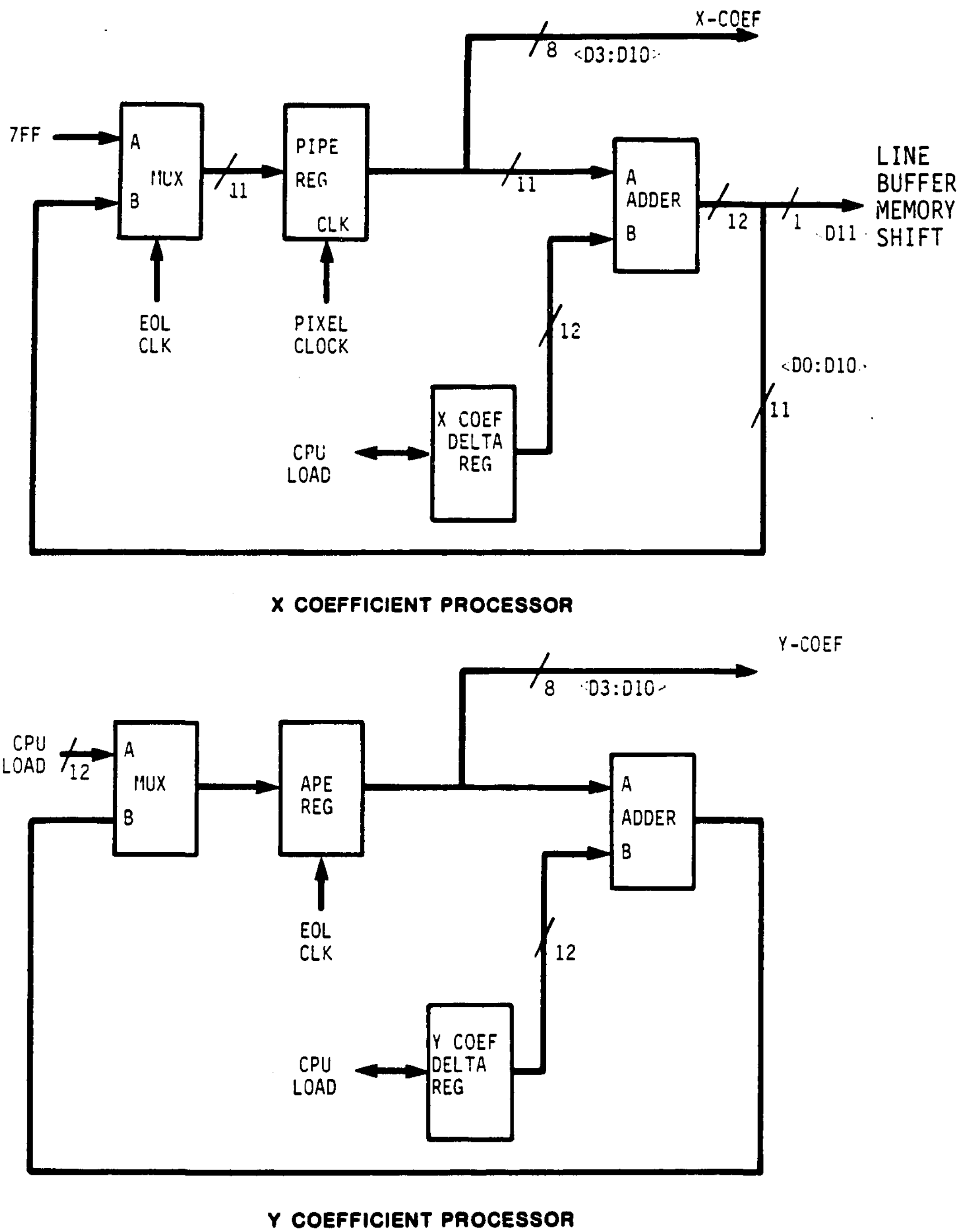


Figure 3.9 Cozo Coefficient Processor Block Diagram

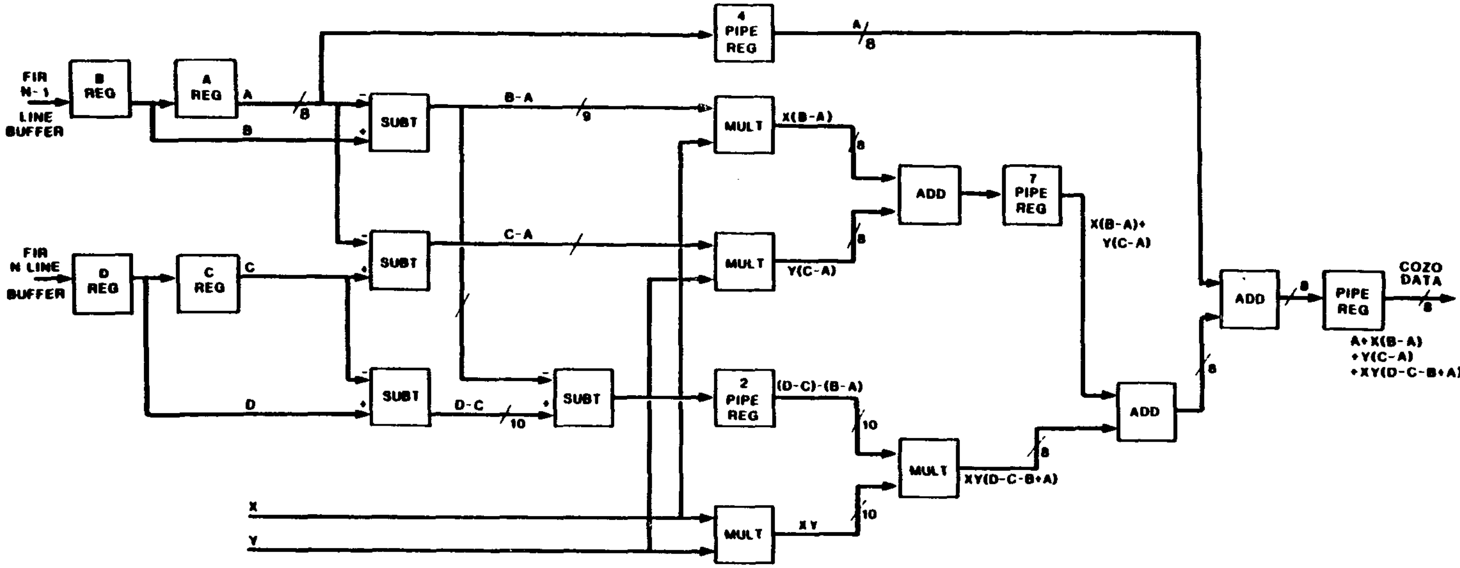


Figure 3.10 Cozo Processor Block Diagram

Figure 3.8 indicate the position of four typical samples of input image data. The solid lines indicate the position of the nine output pixels. One additional input and output data point has been included to illustrate how the process repeats.

The table at the bottom of Figure 3.8 shows the values present in the X coefficient processor over a nine pixel cycle (the value of 2.25 was chosen for this example because it repeats after nine pixels). To achieve a magnification of 2.25, the CPU must load the reciprocal of the magnification (1/2.25 = .4443) into the X COEF Delta Register (Figure 3.9). The X coefficient processor is initialized at the beginning of each line of video data by loading the Hex value 7FF (.9995 decimal) into the Pipe Register. The most significant 8 bits of the Pipe Register output become the X COEF input of Figure 3.10. The Pipe Register output is also added to the contents of the X Delta register. The result of this addition (.9996 + .4443 = 1.4438) yields a number with a one in the most significant bit (MSB) of the adder output. Generation of a number greater than one implies that the interpolation should be performed between the next two pixels along the line. Thus, the MSB of the output of the adder can be used as the Line Buffer Memory shift clock.

Cycle 0 of Figure 3.8 illustrates the beginning of the video scan line. The initialization loads the first pixel of the video line into the B REG of Figure 3.10. The X Coef value will be .996, equivalent to a value of 1.0 in this design. The value of the first pixel output is thus equal to the value in the B REG which is the first input pixel as desired. After the circuit has had time to settle, the MSB of the adder output will be high. The next occurrence of Pixel clock will cause a Line Buffer Memory shift advancing the first pixel from the B REG to the A REG and second pixel from the Line Buffer Memory to the B REG.

Cycle 1 illustrates the calculation of the second output pixel. At this point the End of Line (EOL) clock will be low, allowing the least significant 11 bits (the fraction) of the adder output (remaining from Cycle 0) to appear at the Pipe Register input. When the next Pixel Clock occurs this value will become the new Pipe Register output. The 8 MSBs of pipe register output have a value of .441 as required for the calculation of pixel 1. The output of the adder (.4443 + .4438 = .881) will yield a value less than one. Thus there is no Line Buffer Memory shift and the next calculation will be performed between the same two pixels as desired.

Cycles 2 thru 9 proceed in a similar fashion. As stated above, the example chosen was one that repeats. This can be verified by recognizing that the values in Cycle 0 and Cycle 9 have the same values.

Figure 3.10 shows a block diagram of the Continuous Zoom Processor. The processor was designed using a pipelined architecture. Because of this, the output data is not available until five clock cycles after the input data has been presented. All of the calculations are performed using integer arithmetic. The processor is designed to accept 8 bit input data, 8 bit X and Y coefficients and to output an 8 bit pixel value. Additional bits were added to the data paths between the processing elements to preserve an overall numerical precision of 8 bits. The equations included along the data paths of Figure 3.10 show what portion of Equation 4 is being computed by the preceding processing elements.

Minification is also performed using the continuous zoom processor. Minification is performed by first computing an appropriate magnification factor that falls between one and two. Then an integer divide (performed by discarding pixels, or "pixel pitching") is used to achieve the desired minification. As an example, consider an image where the desired minification is 0.8. This minification is accomplished by first computing coefficients that will yield a magnification of 1.6. Then every other pixel is thrown away, performing a divide by 2, and yielding the desired minification of 0.8. A similar operation is performed where every other line is discarded to achieve minification in the vertical direction.

Output Pixel Address Generator

The output pixel address generator consists of a pixel counter and a line offset register. The address that represents the position of the upper left pixel of the output display is first loaded into the pixel counter register. A value equal to the width of the screen (1024) in pixels minus the number of pixels on the displayed line is loaded into the offset register. Pixels are then stored sequentially (four at a time) until the current line has been completed. The contents of the offset register are then added to the pixel counter to address the first pixel on the next line. This cycle is repeated until the entire output image has been written.

Histogram Processor

The histogram processor is designed to compute a histogram of gray levels of an image as it passes through the image processor. The histogram processor is positioned after the subtracter and before the contrast mapping LUT.

This position was chosen to yield a histogram of the input image data unaffected by the image enhancement processing which follows in the image processor. The histogram of any region (or regions) in the image can be computed. The region over which the histogram is to be computed is identified by one of the Region of Interest (ROI) bits. The first ROI bit encountered enables the computation of the histogram. A subsequent ROI bit disables it. The only restriction that is imposed is that there must be a start bit and a stop bit on each line where the histogram is to be computed.

The histogram processor consists of a high speed memory and an add by one counter. The first requirement in using the histogram processor is to set all values of the memory to zero. The CPU performs this operation by writing zeros into all of the memory locations. Next, the image data is cycled through the image processor. The gray level value of each pixel forms the address of the corresponding bin in the high speed memory. The value at that address is read, incremented by one, and written back to the same location. This process continues for each pixel in the image that has been enabled by the graphics overlay.

Display Memory

The display memory subsystem is the final component of the image display channel. Its purpose is to provide an interface between the image processor and the display monitors. The display memory subsystem consists of two identical display channels as shown in Figure 3.11. Each channel is capable of displaying 8 bit gray scale data on a 1024X1536 portrait mode display. A single graphics overlay bit is included that permits image annotation of the display or region of interest outlines. The display memory incorporates a triple ported architecture. The first port is a write only port from the image processor. The second port is a high speed (98 Mpixels/second) port that drives the video digital to analog converters (DAC). The third port is a read/write port to the VME bus to allow software access to the processed image data.

Each of the display memory channels is double buffered. The double buffers were included so that the image processor could update an image without the operator's being distracted by the changes. The double buffering operates as follows. Assume that the current image display is being refreshed from Buffer A in Figure 3.11. Buffer B will contain a copy of the data present in Buffer A. The next time the image processor is requested to perform an operation, it will write the result into Buffer B. The

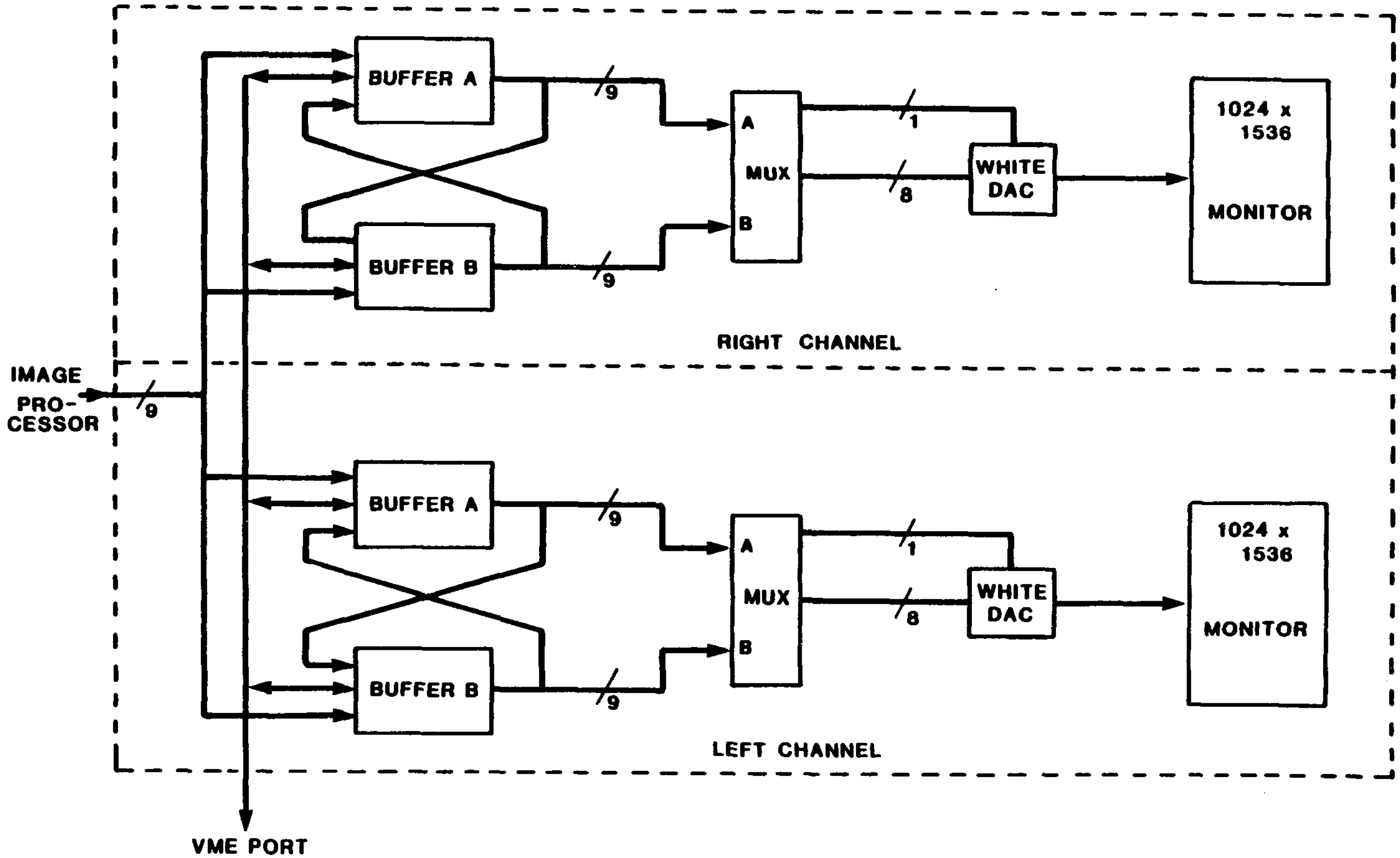


Figure 3.11 Display Memory Block Diagram

buffers are then flipped such that the monitor is now being refreshed from Buffer B. The contents of Buffer B are then copied to Buffer A during the first video frame subsequent to the flipping of the buffers. Buffer A now contains the same data that appears on the display monitor. The next output from the image processor will be directed to Buffer A. It will then become the refresh memory, etc. This technique allows the image processor to update any area of the display memory without consideration of the processing previously performed on that channel.

Certain types of clinical studies result in a sequence of images taken in time. A movie type display is required to visualize the motion present in these sequences of images. The image processor is not fast enough to allow the processing of movies in real time. Special hardware had to be included in the display subsystem to allow it to display these movie loops. First, a hardware zoom was included such that a 256X256 image would be displayed in 1024X1024 format. Each of the display memory buffers can hold up to 24 256X256 images. Using both memory buffers, a total of up to 48 256X256 images can be stored. Special addressing was included that allows the selection of any one of these images for display. An interrupt was also included to notify the CPU when a video frame has been completed. The movie speed is determined by the time each image is displayed, measured in frame times. The fastest rate is thus one frame time or 40 frames per second. To display a movie, the CPU enables the hardware zoom and loads the display subsystem with the address of the first image it wishes to display. It then waits for the number of frame times required to achieve the desired rate. Once the interframe delay has been achieved, the CPU loads the address of the next image of the sequence. This process continues until terminated by the operator.

A gray scale wedge was included along the right edge of the monitor to aid in the adjustment of the window and level. Three markers are superimposed on the gray scale wedge. The center one is provided to indicate the level, or brightness adjustment. The other two are provided to show the level, or range of input gray levels, that has been selected for display.

Mass Storage Subsystem

The Mass Storage Subsystem was designed to provide a local storage capability for medical images. It also provides the cache storage required for images when space in the base memory has been exhausted. The most important requirements imposed upon the mass storage subsystem are speed and size. The speed requirement is imposed by the system response time desired by the radiologist. The size requirement is driven by the quantity of information present in medical images.

The speed of the mass storage subsystem is ultimately limited by the transfer rate of the disk drive employed. High speed disk drives are available in two formats. The first type has a single data channel active at any time. For this type of disk drive additional speed is obtained by increasing the density of storage on each track. The second type of disk drive has multiple data channels enabled in parallel. This type of disk drive is call a parallel transfer drive (PTD). It was determined that the PTDs were too large and expensive to be considered in the review console design. The fastest single channel disk drive available was the Fujitsu M2333 with a 2.458 Mbyte/sec transfer rate. It has an unformatted storage capacity of 337.7 Mbytes. The M2333 met the size, speed and cost constraints imposed upon the review console and was incorporated into the design.

The requirement was to design a disk controller that would achieve an effective transfer rate that was as close as possible to the drive's 2.458 Mbyte/sec peak rate. To achieve this goal:

1. Each track on the disk was formatted to contain a single sector (40880 bytes). This was done to reduce the overhead implicit when multiple sectoring is implemented.
2. Automatic track switching was implemented to avoid losing a rotation of the disk at the end of each sector. This was critical since loss of every other rotation would effectively reduce the transfer rate by a factor of two. The sector length was chosen (shortened by 80 bytes) to allow time between the end of the sector and the occurrence of the index pulse. This time is required to allow the phase lock loop in the read/write electronics time to sync up on the new head.
3. Automatic seek to the next cylinder was implemented. This assures that the seek will complete with the loss of only one revolution.
4. 32 bit transfers are performed on the VME bus between the disk controller and the base memory.
5. No error correction was implemented. A cyclic redundancy check is computed on each track to validate the integrity of the data.

The two design goals for the disk controller were size and speed. The storage space utilization is 99.8

percent efficient. The only loss is the 80 bytes reserved for timing at the end of each sector. The transfer speed is 88 percent of the peak rate for the transfer of a 1024X1024 image. The loss here is due to 1) the insignificant 80 byte loss at the end of each track and 2) the loss of one revolution for each seek to a new cylinder. There are 10 tracks per cylinder which implies that 10 out of 11 rotations are actually used to transfer data.

The mass storage subsystem can transfer a 1024X1024 pixel image, each pixel occupying two bytes of storage, in .9 seconds. This is four times the transfer rate achieved by conventional disk systems that appear as peripherals on computer systems.

Network Interface Module

The Network Interface Module (NIM) was designed in order to provide an ACR-NEMA standard interface port to the review console. The ACR-NEMA standard was developed to assure that PACS components provided by different manufacturers would be able to communicate with each other. A large effort was expended in trying to define where the standard interface would best be placed. The architecture finally postulated by the ACR-NEMA standard divides the PACS into two sections. The first section consists of the network and the electronics required to interface with it. This interface electronics was called the Network Interface Equipment (NIE). The second section consists of the remaining PACS components (the acquisition, archiving, and display subsystems). All of the components included under the second section were to interface with the NIE, provided by the manufacturer of the first section, over the interface defined by the ACR-NEMA standard.

In addition to the hardware definition, the ACR-NEMA standard also defined message data formats and a basic set of global commands. The network interface module in the review console was designed to decode the ACR-NEMA data packets and to enter the data into the review console. Image and graphics data were to be entered through a dedicated high speed bidirectional port into the base memory. Using this port, data transfer rates of up to 10 MPixels/second (or 120 MBits/second) could be achieved. Patient demographic data, commands, and other ASCII text would be transferred by the NIM through the VME port.

The ACR-NEMA document had not been accepted as a standard when the design of the review console was initiated. The research contract that funded the development of the review console was with Toshiba Medical Systems Corp (TMS). TMS decided to adopt an interim network approach until the remaining standardization and technological questions had been answered. The interim network approach employs a star topology and is based on the Digital Equipment Corp. (DEC) DR11-W parallel interface. TMS requested that this type of interface be included in the review console making it compatible with the rest of their development. Thus, development of the ACR-NEMA NIM was placed on indefinite hold. The remaining components of the review console were designed to allow the inclusion of the NIM at a later time. Backplane space was reserved for the NIM in the hardware prototype. The dedicated data paths to the base memory were designed and implemented.

User Interface

The user interface was designed to provide the radiologist with a natural interface to the review console. Wherever possible, parallels were drawn between the operations a radiologist performs using the familiar film-based system and the equivalent operations he would perform using the review console.

The user interface consists of a control desk and three video monitors. Two of the monitors provide the monochrome image display capability described previously. The third monitor is the control monitor. It is a color monitor designed to provide a menu driven interface to the review console. The control desk is a computer driven console that incorporates a trackball, six control knobs, and a standard alphanumeric keyboard.

The display screen of the control monitor is divided into two sections. The monitor surface is covered with a touch sensitive screen to allow the computer to determine what point on the screen has been touched. The top half of the control monitor is used to present the color coded menu keys. This section also includes a display that indicates what image processing functions have been activated and to what degree. The bottom half of the screen is used to present an alphanumeric display similar to that of a standard computer terminal. The combination of the touch sensitive screen and the color coded menus allows the implementation of a "soft key pad". The majority of the commands entered into the review console use this approach. A main menu is presented that permits the direct selection of the frequently occurring functions. After a main function has been selected, and if additional choices are required, subfunction menus are displayed. The menus were designed to minimize the number of times submenus would be required. Most operations can be selected directly from the main menu.

The alphanumeric portion of the control monitor is used for the display of textual information, such as patient demographics or the status of procedures. When the directory key is touched (on the soft key pad) the studies present on the local disk are listed in this region. The alphanumeric display is also used, in conjunction with the keyboard, to communicate with other stations in the PACS.

Each of the monochrome display monitors is also equipped with a touch sensitive screen. Here the touch sensitive screen is used to select which image the radiologist wishes to process. It can also be used to move an image to another position on the monitor. This interface allows the normal act of pointing to control operations on the review console.

The most commonly used image processing functions are contrast enhancement, filtering and zoom. Individual controls were provided for each of these operations. Separate control knobs were provided for 1) the window, 2) the level 3) the magnitude and type of spatial filter, and 4) the magnitude of the continuous zoom. Two additional controls were provided that are available for assignment to other operations. The function of these last two controls is dependent upon what function is being performed by the review console. For example, one of these controls is used to adjust the speed and direction of the movie loops when that operation has been selected. All of these control knobs were implemented using shaft encoders so that they would not have any absolute position. This was required to allow the radiologist to examine the images in any order. Whenever he returns to an image, the last settings for that image must be reinstated irrespective of the current knob positions. This requirement could not have been met if conventional potentiometers had been employed.

A trackball was included in the desk to allow the precise positioning of a cursor on an image. This feature can be used to define a region of interest to identify anatomical structures when annotating an image.

CHAPTER 4

RESULTS AND CONCLUSIONS

The objective of this research was to design an imaging display station suitable for use in a filmless radiology department. The minimum requirement imposed upon this design was that it permit the radiologist to perform image analysis operations equivalent to those currently performed using the film-based system. This set of requirements has been identified in the PACS literature as the multimodality display capability. The display station which resulted from this research will be used in a series of psychophysical experiments designed to validate the multimodality design criterion.

The display console, shown in Figure 4.1, was designed and built that achieves the multimodality design goals as outlined in Figure 2.1. Many original architectural concepts were required to meet this multimodality design goal.

Innovative System Design Concepts

The first of these was the pipelined display channel consisting of a base memory, image processor and display

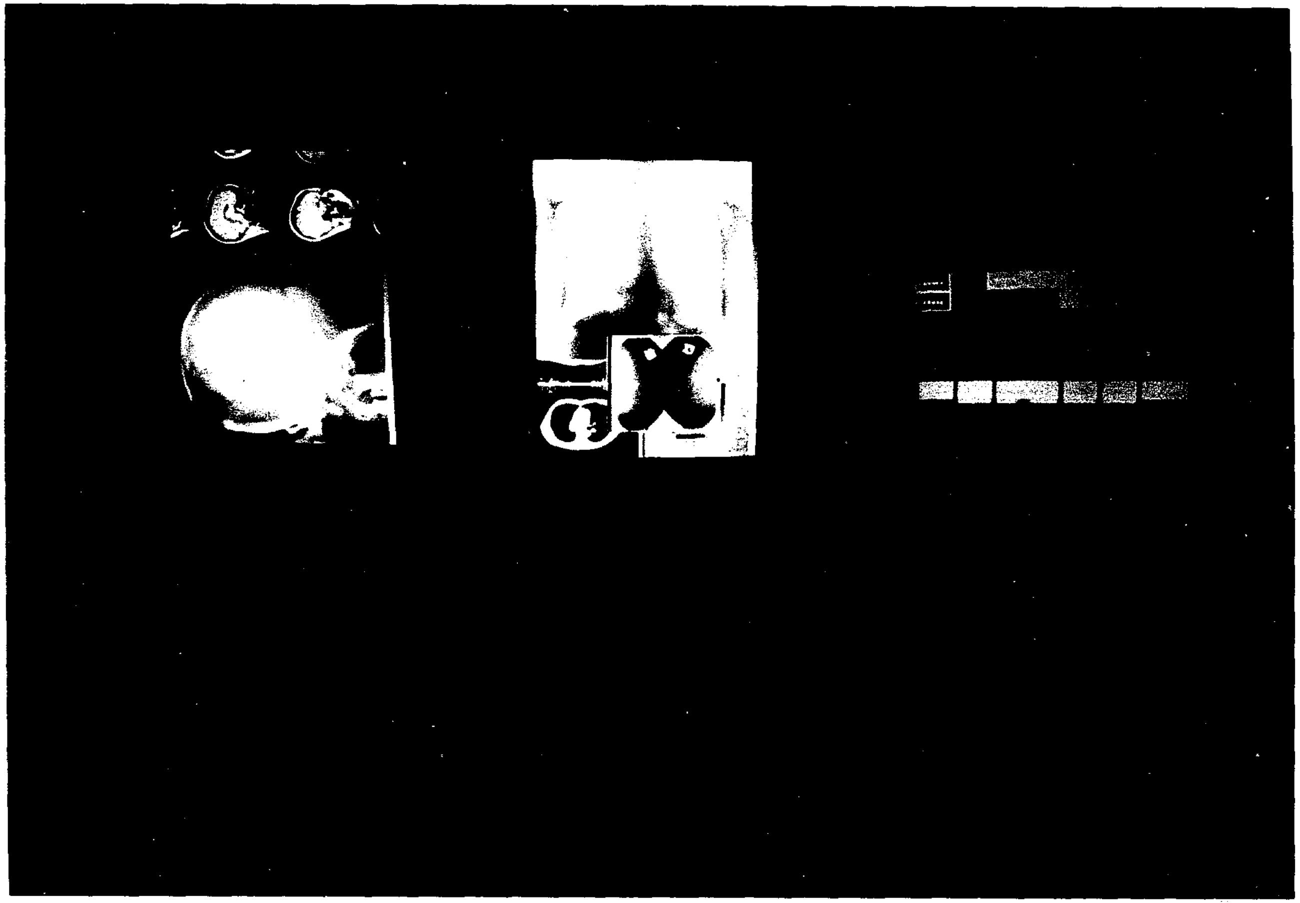


Figure 4.1 Photograph of Completed Review Console

memory. This architecture was developed specifically to permit the processing of a single image on the display screen independent of other images present. This is a unique approach that has not as yet appeared in any commercial products (Patent applied for in 1986).

The second innovative design concept was the use of continuous zoom processing for both the magnification and minification of images. A substantial cost benefit could be achieved if it could be demonstrated that a radiologist can process 2048X2048 resolution images using this approach. The author is aware of only one manufacturer (Comtal) that offers a hardware continuous zoom processor. Its design is limited in that 1) it can only perform magnification and 2) it can only support resolutions of up to 512X512 pixels. The 1024X1536 capability and the algorithm that allows minification are unique to the review console's design (Patent applied for in 1986).

System Constraints and Future Research

A final evaluation of the design will not be possible until psychophysical experiments have been completed. Even without this data it is possible to make some preliminary statements and suggestions for further research. The overall response time of the system needs to be improved.

For images of size 1024X1024 or less, the 10 MPixels/second processing rate (resulting in a 0.1 second image update rate) is adequate. As larger images are processed, the increased delay interferes with the visual feedback the operator is receiving from the display monitor. Parallel processing techniques could be enlisted to reduce this delay. The one dimensional spatial filter design implemented in the image processor should eventually be replaced by a two dimensional design. The asymmetrical effect the one dimensional filter has on the image data is much more apparent to the radiologists than the simulations had predicted. The addition of a pseudo color capability to the image display subsystem is desirable for some of the imaging modalaties (e.g. MRI and Nuclear Medicine). It may soon be possible to replace the Base Memory in the image channel with a high speed parallel transfer disk drive. This extension would make all of the images on the disk available instantly, thus eliminating the swapping delays implicit in the current design.

The ultimate test of whether the design objectives have been met will be if the display console is accepted by the radiological community. While no formal psychophysical experiments have been completed, the display station has been demonstrated for many radiologists visiting the University Medical Center. All of their responses have been positive. Toshiba Medical Corp. has seen enough merit in the design to pay all of the legal fees involved with the patent application. Toshiba Medical Corp. has also requested that the display console be demonstrated at the November 1986 meeting of the RSNA.

APPENDIX A

PSYCHOPHYSICAL DETERMINATION OF THE SPATIAL RESOLUTION REQUIREMENTS OF RADIOGRAPHS

This appendix is included to document a research effort that was conducted in parallel to and in support of the review console design. The author requested that the study be performed in an attempt to add rigor to the specification for maximum spatial resolution. The author participated in the design of the experiment, wrote the required software, and performed the required image processing.

The spatial resolution of a displayed image was perhaps the most critical parameter in the design of hardware for a display console. The size of the memory and the data rate required of the display components increase as the square of the spatial resolution. In spite of its importance, the minimum spatial resolution requirement for radiographic images has never been determined. This is in part due to the difficulty in making such a measurement. Radiologists have difficulty relating the actions they perform to a numerical value for spatial frequency.

An upper bound on the required resolution can be obtained by an analysis of the image acquisition system. The film-screen combination routinely used for chest radiography has a limiting resolution of 5 lp/mm. This results in a spatial resolution requirement of 4096X4096 pixels (assuming a 14x17 inch format). A spatial resolution of 5 lp/mm is the maximum that can be placed on the film and thus the maximum resolution a radiologist can use. The actual resolution requirements may be lower than this.

A psychophysical experiment was designed to measure the spatial frequency required by the radiologist in making his diagnosis. For this experiment the isolation of spatial frequency as the only variable was desired. To accomplish this, it was decided to use film as the imaging medium.

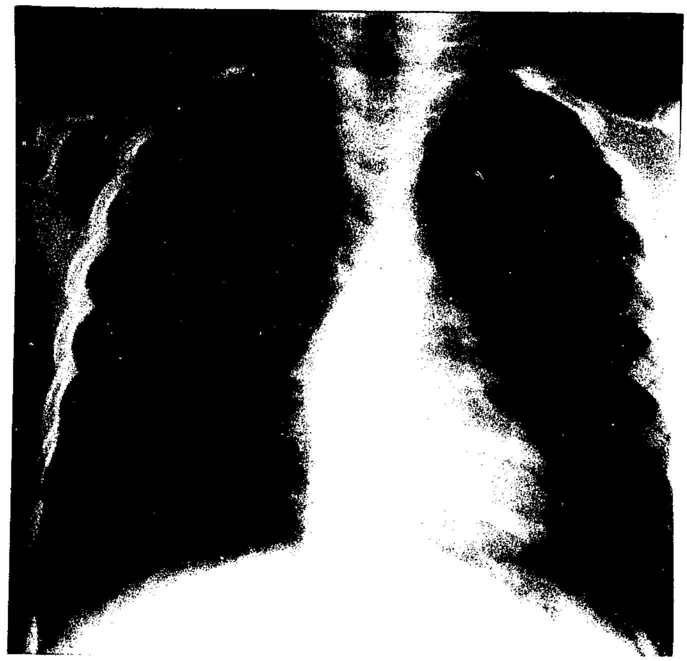
Eight clinical studies were selected by one of the staff radiologists. These images were chosen because high spatial frequencies (sharp images) were required to diagnose the known pathology. The spatial resolution in these images was degraded in known increments. Radiologists were given sets of these images and asked to perform a routine diagnosis. Their performance was then correlated to the spatial frequency content of the images they had used.

The results of the study [19],[20] demonstrate that the required spatial resolution may be around 2.5 lp/mm. It was not possible to make a more definite statement at this stage of the study due to the limited number of images and radiologists available. The accuracy of this type of study is limited by the number and type of images used as well as by the number of radiologists available to see an image from each set. Another study is planned in which the range of resolutions will be clustered around 2.0 lp/mm. The object of this upcoming study will be to determine if there is an optimum spatial resolution that falls between 1024X1024 and 2048X2048 pixels.

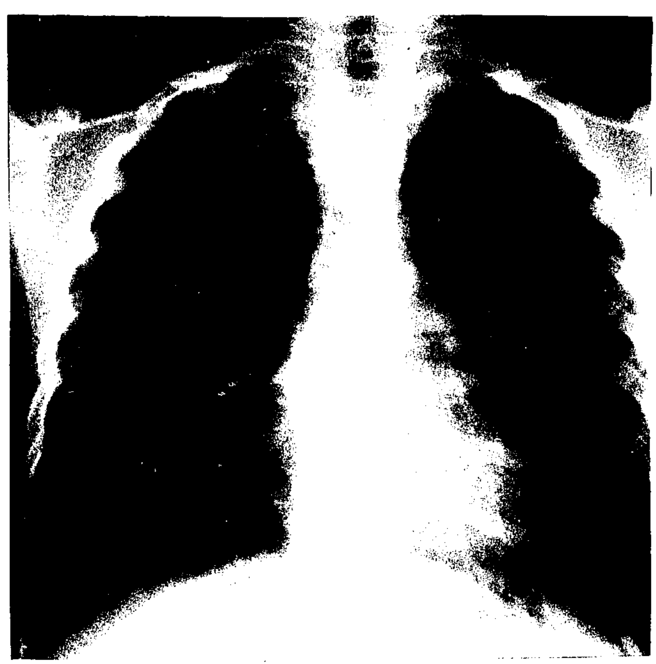
Digital image processing techniques were used to degrade the images used in this study. This approach was chosen to assure a uniform and controlled approach to the blurring process. The selected images were first digitized at 12 bits/pixel on a microdensitometer. The highest resolution image was sampled at 10 samples per mm corresponding to the 5 lp/mm maximum that can be achieved by the conventional film/screen imaging system. These digitized images were then convolved with a Gaussian blur function to reduce their spatial resolution. Images were made having spatial resolutions of 5 lp/mm, 2.5 lp/mm, 1.25 lp/mm, and 0.625 lp/mm for each of the eight originals. The original and the three blurred images were then written back to film using the same microdensitometer.

The images shown in Figure 5.1 show the four resolutions for one of the sets of eight images. The images in Figure 5.2 are a magnification taken from the upper left corner of each of the images in Figure 5.1. These images are presented to show the degradation in image quality caused by the blurring process. The images shown in Figure 5.3 were formed by subtracting each of the three blurred images from the 5 lp/mm image. The structures present in these images are the information that is lost due to the reduction in the resolution. During the psychophysical experiments, the 5 lp/mm image was used as the control. Radiologists were not able to distinguish the 5 lp/mm image created by our blurring system from the original.

The computer programs written to perform the convolution processing are included in this appendix. The images were blurred by convolving a 128X128 pixel Gaussian blur function with the 2048X2048 input image data. The convolution processing was performed in the Fourier domain to reduce the magnitude of the computation. A computer generated image of a Gaussian function was generated by the first program. The magnitude of the blurring is determined by the FWHM (Full Width at Half Maximum) which is entered by the user. The program then computes a two dimensional Gaussian with the area under the curve equal to one. This was done for FWHM's equal to 2 (for 2.5 lp/mm), 4 (for

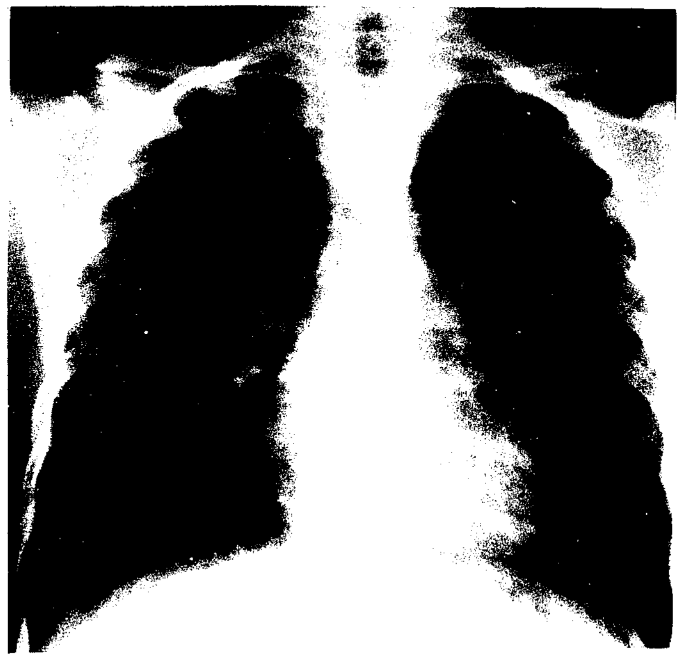


(a)

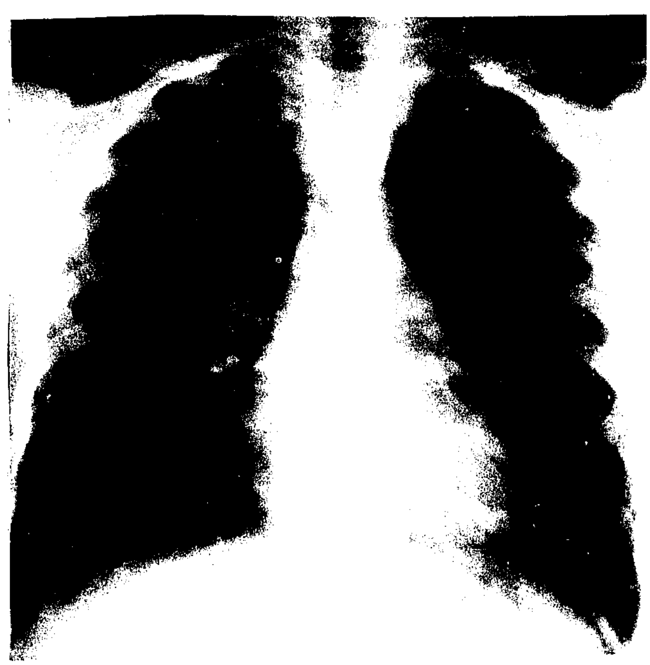


(b)

Figure 5.1 Chest Radiograph at Four Spatial Resolutions
(a) 5 lp/mm. (b) 2.5 lp/mm.
(c) 1.25 lp/mm. (d) 0.625 lp/mm.

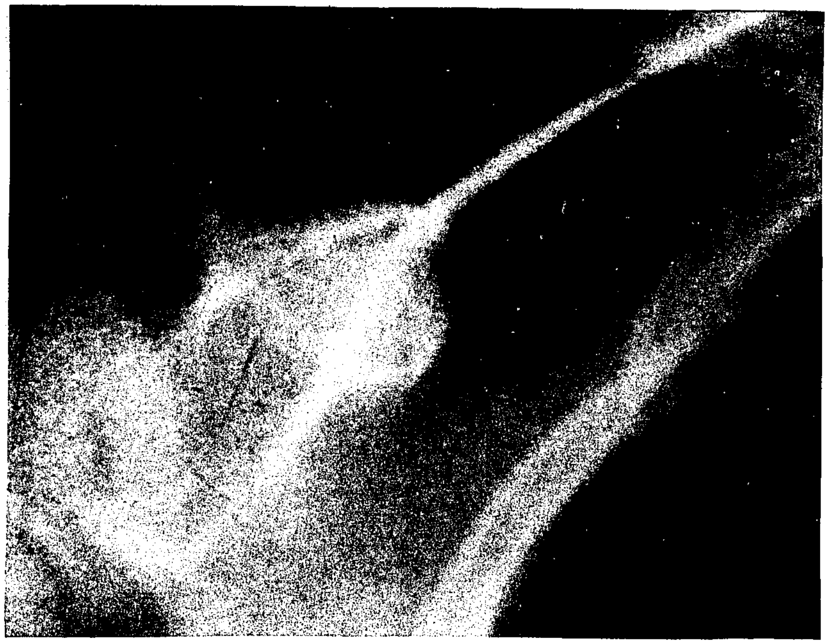


(c)

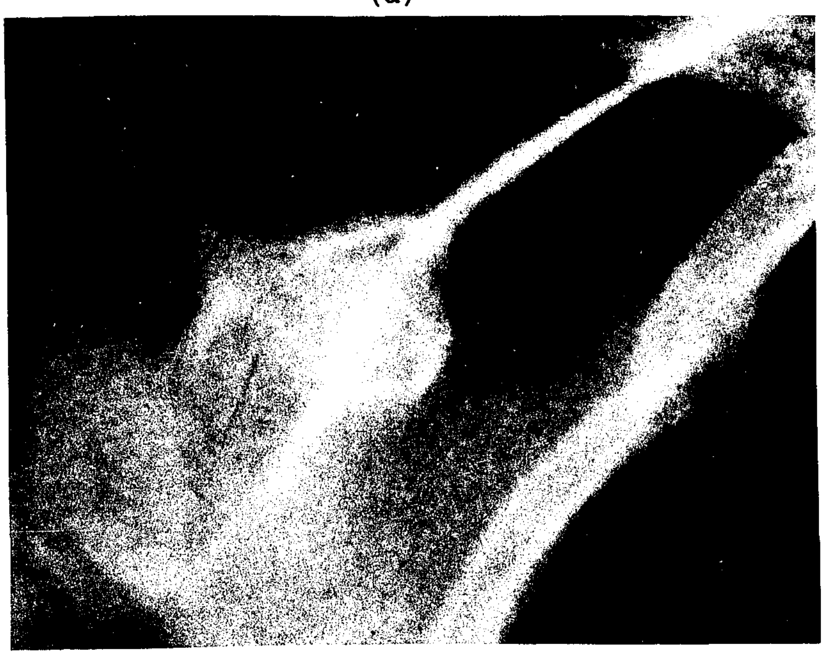


(d)

Figure 5.1 (Continued)

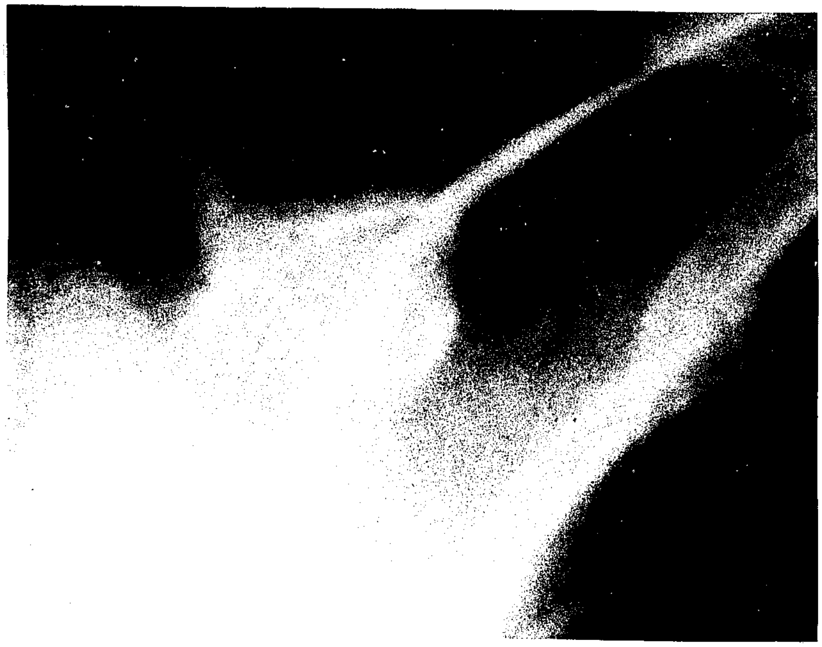


(a)



(b)

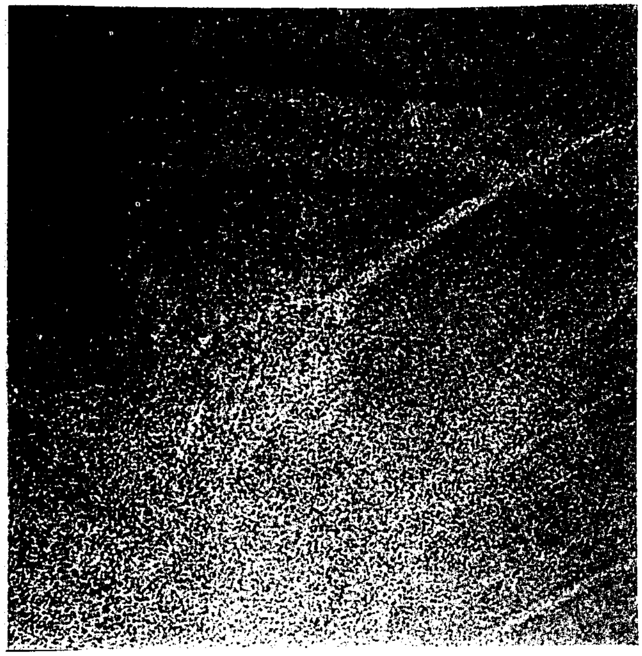
Figure 5.2 Upper Left Corner of Chest Radiograph at Four Spatial Resolutions
(a) 5 lp/mm. (b) 2.5 lp/mm.
(c) 1.25 lp/mm. (d) 0.625 lp/mm.



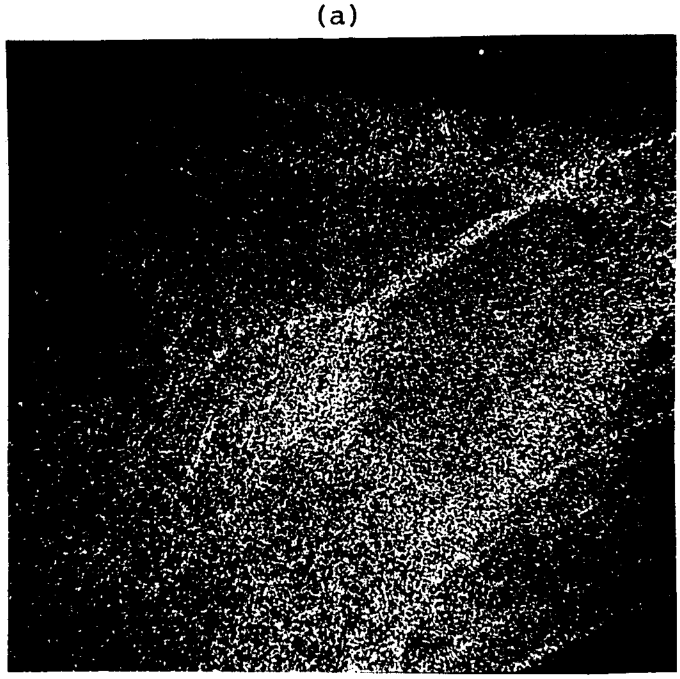
(c)

(d)

Figure 5.2 (Continued)

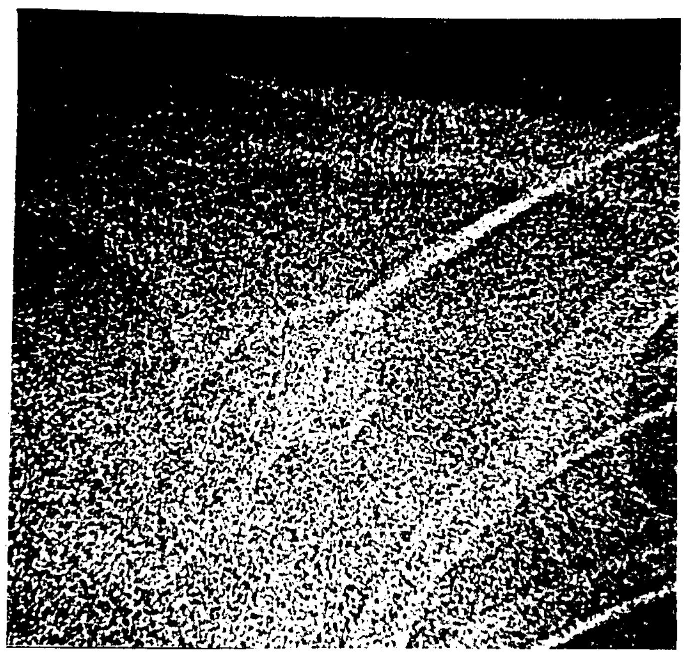


(a)



(b)

Figure 5.3 Information Lost Due to Reduction in Spatial Resolution
(a) 2.5 lp/mm. (b) 1.25 lp/mm.
(c) 0.625 lp/mm.



(c)

Figure 5.3 (Continued)

routines) computes the two dimensional FFT (Fast Fourier Transform) of images up to 2048X2048 pixels. This program is used to transform the three Gaussian blur functions and the input image into the Fourier domain. The third program was used to perform a complex multiplication between the transform of the input image and the transforms of each of the blur functions. The final program (two routines) computes the IFFT (Inverse Fast Fourier Transform) of the products of the transforms computed above. The outputs of the IFFT program are the desired blurred images.

```
C    PROGRAM TO GENERATE GAUSSIAN FILTER
C    FUNCTION IN NXN DATA SPACE.
     CHARACTER NAME*32
     INTEGER*4 FAB(60),STAT(2)
     REAL DATA(2048,65)
10   WRITE(5,9000)                                               !PROMPT FOR OUTPUT FILE NAME
     READ(5,9001,ERR=10)NAME
     CALL OPEN_(FAB,NAME,'.DAT','N',STAT,,2048*4,32768,*10)      !OPEN OUTPUT FILE
     IBLK=1
     WRITE(5,9002)
     READ(5,9003)FWHM                                            !INPUT FULLWIDTH AT HALF MAX
     SIGMA=ABS(FWHM/2.354)                                       !COMPUTE STANDARD DEVIATION FROM FWHM
     G=.15915/SIGMA**2                                           !COMPUTE GAIN SO ENERGY = 1.0
     DO 40 I = 0,1023,64                                         !COMPUTE OVER 1024X1024 SUPPORT
     DO 30 Y = 1,64
     DO 20 X = 1,1024
     DATA(X,Y)=G*EXP(-((X-1)**2+(Y+I-1)**2)/(2*SIGMA**2))        !EQUATION OF GAUSIAN
20   DATA(2050-X,Y)=DATA(X,Y)                                    !REPLICATE LEFT TO RIGHT
30   DATA(1025,Y)=DATA(1026,Y)
     CALL WRITE_W(FAB,DATA,2048*4*64,IBLK)                       !WRITE OUTPUT FILE
     WRITE(5,9004)IBLK
     IBLK=IBLK+1024
40   CONTINUE
     DO 70 I = 1024,1,-64                                        !COMPUTE OTHER HALF OF PULSE
     DO 60 Y = 1,64
     DO 50 X = 1,1024
     DATA(X,Y)=G*EXP(-((X-1)**2+(I-Y+1)**2)/(2*SIGMA**2))        !EQUATION OF GAUSIAN
50   DATA(2050-X,Y)=DATA(X,Y)                                    !REPLICATE LEFT TO RIGHT
60   DATA(1025,Y)=DATA(1026,Y)
     CALL WRITE_W(FAB,DATA,2048*4*64,IBLK)                       !WRITE OUTPUT FILE
     WRITE(5,9004)IBLK
     IBLK=IBLK+1024
70   CONTINUE
     CALL CLOSE_(FAB)                                            !CLOSE OUTPUT FILE
     STOP ' '
9000 FORMAT(X,'ENTER OUTPUT FILE NAME ',$)
9001 FORMAT(A)
9002 FORMAT(X,'ENTER FULL WIDTH HALF MAX IN PIXELS ',$)
9003 FORMAT(G13.0)
9004 FORMAT('+',I6)
     END
```

```
C   PROGRAM TO COMPUTE THE FORWARD 2D XFORM OF UP TO
C   A 2048 X 2048 REAL ARRAY STORED IN INFILE.  OUTPUT
C   IS N/2+1 LINES OF COMPLEX #'S
C   COMMAND:
C     FFT2D INFILE OUTFILE DIM

      CHARACTER SNAME*32,DNAME*32,CMDSTR*64
      INTEGER*4 SFILE(60),DFILE(60),STAT(2),DIM,TBL(8)
      REAL DATA(512,514)
      DATA TBL / 16,32,64,128,256,512,1024,2048 /

      I=LIB$GET_FOREIGN(CMDSTR,'$_FILE: ',J)                    !DECODE COMMAND LINE
      CMDSTR(64:64)=','
      J=INDEX(CMDSTR,',')
      SNAME=CMDSTR(:J-1)
      I=INDEX(CMDSTR(J+1:),',')
      DNAME=CMDSTR(J+1:I+J-1)
      IF(DNAME.EQ.' ')GO TO 8000
      READ(CMDSTR(I+J+1:),9002,ERR=8002)DIM

      DO 10 I = 1,8                                              !VALIDATE IMAGE SIZE = PWR 2
10    IF(DIM.EQ.TBL(I))GO TO 20
      GO TO 8002
20    CALL OPEN_(SFILE,SNAME,'.DAT','O',STAT,*8000)              !OPEN INPUT AND OUTPUT DATA FILES
      CALL OPEN_(DFILE,DNAME,'.DAT','N',STAT,,4*DIM,4*DIM+4,*8000)
      IF(DIM.LE.512)THEN                                         !LOOP FOR IMAGES <= 512 PIXELS
        CALL READ_W(SFILE,DATA,4*DIM**2,1,*8001)                 !CALLING PROGRAM DOES IO
      ENDIF
      CALL AFFT(DATA,DIM,SFILE,DFILE)                            !PERFORM FORWARD FFT
      IF(DIM.LE.512)THEN                                         !FOR IMAGES <= 512
        CALL WRITE_W(DFILE,DATA,(DIM**2+DIM*2)*4,1,*8003)        !CALLING PROGRAM DOES IO
      ENDIF
100   CALL CLOSE_(SFILE)                                         !CLOSE FILES
      CALL CLOSE_(DFILE)
      STOP ' '
8000  WRITE(6,9000)STAT                                          !ERROR REPORTING
      GO TO 100
8001  WRITE(6,9001)STAT
      GO TO 100
8002  WRITE(6,9003)
      GO TO 100
8003  WRITE(6,9004)STAT
      GO TO 100
9000  FORMAT(X,'OPEN ERROR ',Z8,I10)
9001  FORMAT(X,'IO READ ERROR ',Z8,I10)
9002  FORMAT(BN,I6)
9003  FORMAT(X,'DIMENSION ERROR')
```

```
C     SUBROUTINE TO COMPUTE FFT OF A SQUARE ARRAY THAT IS A
C     POWER OF 2
C     OUTPUT IS AVGERAGE VALUE VALUE AND N/2 FREQUENCIES REPRESENTED
C     AS N/2+1 COMPLEX LINES EACH OF N POINTS

C     IF N <= 512 DATA IS PASSED IN ARRAY
C     IF N >  512 DATA IS ASSUMED IN FILE

      SUBROUTINE AFFT(DATA,N,SFILE,DFILE,*)
      INTEGER*4 N,IBLK,OBLK,SFILE(60),DFILE(60)
      REAL DATA(N,N)

      IF(N.LE.512)THEN                                  !N > 512 => DATA IN FILE
        DO 10 I=N,1,-1                                  !DO N REAL FFT'S ON ROWS
10      CALL FFTRI(DATA(1,I),N,DATA(1,I+1))
        CALL XPOSER(DATA(1,2),N)                        !TRANSPOSE DATA
        DO 20 I=N,1,-1                                  !DO N REAL FFT'S OF COLS(2=COMPLX)
20        CALL FFTRI(DATA(1,I+1),N,DATA(1,I+2))
        CALL RTOR(DATA(1,3),DATA(1,1),N*2)              !MOVE TO AVG & N/2 OUT OF WAY
        DO 30 I=1,N-2,2                                 !REAL+IMAG -> COMPLEX
30        CALL RTOZ(DATA(1,I+4),DATA(1,I+5),DATA(1,I+2),N)
        CALL RTOZN(DATA(1,2),,DATA(1,N+1),N)            !N/2 COMPLEX
        CALL RTOZN(DATA(1,1),,DATA(1,1),N)              !AVG COMPLEX
        RETURN

      ELSE IF(N.EQ.1024)THEN
        IBLK = 1                                        !INIT FILE POINTERS
        OBLK = 1
        DO 130 I = 1,1024,128                           !1024 LINES 128 AT A TIME
          CALL READ_W(SFILE,DATA(1,2),1024*128*4,IBLK,*8000)
          IBLK = IBLK+1024
          DO 110 J = 1,128                              !DO REAL FFT ON 128 LINES
110         CALL FFTRI(DATA(1,J+1),1024,DATA(1,J))
          K = OBLK
          DO 120 J = 1,1024,128                         !TRANSPOSE AND WRITE
            CALL XPOSE1(DATA(J,1),DATA(1,129),1024,128)
            CALL WRITE_W(DFILE,DATA(1,129),128*128*4,K,*8001)
120         K = K+1024
130     OBLK = OBLK + 128                               !BUMP OUTPUT BLOCK BASE
        OBLK=1
        DO 180 I = 1,1024,128                           !COLUMNS 128 AT A TIME
          K = OBLK
          DO 140 J = 1,1024,128                         !FINISH READ AND XPOSE
            CALL READ_W(DFILE,DATA(1,129),128*128*4,K,*8001)
            K = K+128
140         CALL XPOSE2(DATA(1,129),DATA(J,1),1024,128)
          DO 150 J = 128,1,-1                           !FFT ON COLS
150         CALL FFTRI(DATA(1,J),1024,DATA(1,J+2))
          IF(I.EQ.1)THEN
            CALL RTOR(DATA(1,3),DATA(1,1),1024*2)
```

```
            DO 160 J = 3,127,2
160            CALL RTOZ(DATA(1,J+2),DATA(1,J+3),DATA(1,J),1024)
            CALL RTOZN(DATA(1,2),,DATA(1,129),1024)           !N/2 COMPLEX
            CALL RTOZN(DATA(1,1),,DATA(1,1),1024)             !AVG COMPLEX
            CALL WRITE_W(DFILE,DATA(1,129),1024*4,8193,*8001) !WRITE N/2
          ELSE
            DO 170 J = 1,127,2
170            CALL RTOZ(DATA(1,J+2),DATA(1,J+3),DATA(1,J),1024)
          ENDIF
          CALL WRITE_W(DFILE,DATA,1024*128*4,OBLK,*8001)
180       OBLK=OBLK+1024
        RETURN

      ELSE IF(N.EQ.2048)THEN
        IBLK = 1                                              !INIT FILE POINTERS
        OBLK = 1
        DO 230 I = 1,2048,64                                  !2048 LINES 64 AT A TIME
          CALL READ_W(SFILE,DATA(1,2),2048*64*4,IBLK,*8000)
          IBLK = IBLK+1024
          DO 210 J = 1,64                                     !DO REAL FFT ON 64 LINES
210         CALL FFTRI(DATA(1,J+1),2048,DATA(1,J))
          K = OBLK
          DO 220 J = 1,2048,64                                !TRANSPOSE AND WRITE
            CALL XPOSE1(DATA(J,1),DATA(1,65),2048,64)
            CALL WRITE_W(DFILE,DATA(1,65),64*64*4,K,*8001)
220         K = K+1024
230     OBLK = OBLK + 32                                      !BUMP OUTPUT BLOCK BASE
        OBLK=1
        DO 280 I = 1,2048,64                                  !COLUMNS 64 AT A TIME
          K = OBLK
          DO 240 J = 1,2048,64                                !FINISH READ AND XPOSE
            CALL READ_W(DFILE,DATA(1,65),64*64*4,K,*8001)
            K = K+32
240         CALL XPOSE2(DATA(1,65),DATA(J,1),2048,64)
          DO 250 J = 64,1,-1                                  !FFT ON COLS
250         CALL FFTRI(DATA(1,J),2048,DATA(1,J+2))
          IF(I.EQ.1)THEN
            CALL RTOR(DATA(1,3),DATA(1,1),2048*2)
            DO 260 J = 3,63,2
260            CALL RTOZ(DATA(1,J+2),DATA(1,J+3),DATA(1,J),2048)
            CALL RTOZN(DATA(1,2),,DATA(1,65),2048)            !N/2 COMPLEX
            CALL RTOZN(DATA(1,1),,DATA(1,1),2048)             !AVG COMPLEX
            CALL WRITE_W(DFILE,DATA(1,65),2048*4,32769,*8001) !WRITE N/2
          ELSE
            DO 270 J = 1,63,2
270            CALL RTOZ(DATA(1,J+2),DATA(1,J+3),DATA(1,J),2048)
          ENDIF
          CALL WRITE_W(DFILE,DATA,2048*64*4,OBLK,*8001)
280       OBLK=OBLK+1024
```

```
        RETURN
      ENDIF
8000  RETURN(1)
8001  RETURN(2)
      END
```

```
C       PROGRAM TO COMPUTE THE COMPLEX PRODUCT OF TWO
C       FILES, SRC AND DEST, AND RETURN PRODUCT TO
C       DEST ARRAY.  DOUBLE BUFFERD ACCESSES ARE USED
C       OVERLAP THE FILE IO AND MULTIPLY OPERATIONS

        CHARACTER NAME(3)*32,CMDSTR*64
        INTEGER FILE(60,3),STAT(2,3)
        REAL SRC(512,16,2),DEST(512,16,2)
        I=LIB$GET_FOREIGN(CMDSTR,'$_FILE: ',J)
        J=INDEX(CMDSTR,' ')
        NAME(1)=CMDSTR(:J)
        I=INDEX(CMDSTR(J+1:),' ')
        NAME(2)=CMDSTR(J+1:I+J)
        NAME(3)=CMDSTR(I+J+1:)
        CALL OPEN_(FILE(1,1),NAME(1),'.DAT','R',STAT(1,1),*8000)          !OPEN DATA FILES
        CALL OPEN_(FILE(1,2),NAME(2),'.DAT','O',STAT(1,2),*8002)
        CALL OPEN_(FILE(1,3),NAME(3),'.DAT','N',STAT(1,3),,512,512,*8003)
        N=1
        M=2
        CALL READ_(FILE(1,1),SRC(1,1,1),32768,1,*8001)                    !READ THE DATA
        CALL READ_W(FILE(1,2),DEST(1,1,1),32768,1,*8001)
        CALL WAIT_(FILE(1,1))
        IF((STAT(2,1).NE.32768).OR.(STAT(2,2).NE.32768))GO TO 50
10      CALL READ_(FILE(1,1),SRC(1,1,M),32768,*8001)
        CALL READ_(FILE(1,2),DEST(1,1,M),32768,*8001)
C       CALL MUL('AZTOAZ',SRC(1,1,N),DEST(1,1,N),4096)
        CALL CPXMUL(SRC(1,1,N),DEST(1,1,N),4096)                          !PERFORM COMPLEX MULTIPLY
        CALL WRITE_W(FILE(1,3),DEST(1,1,N),32768,*8001)                   !WRITE THE RESULT
        CALL WAIT_(FILE(1,1))
        CALL WAIT_(FILE(1,2))
        L=N
        N=M
        M=L
        IF((STAT(2,1).NE.32768).OR.(STAT(2,2).NE.32768))GO TO 50
        GO TO 10
50      CALL CPXMUL(SRC(1,1,N),DEST(1,1,N),STAT(2,1)/4)                   !FINISH UP DOUBLE BUFFER LOOP HERE
        CALL WRITE_W(FILE(1,3),DEST(1,1,N),STAT(2,1),*8001)
100     CALL CLOSE_(FILE(1,1))                                            !CLOSE DATA FILES
        CALL CLOSE_(FILE(1,2))
        CALL CLOSE_(FILE(1,3))
        STOP ' '

8000    WRITE(6,9000)STAT(1,1),STAT(2,1)                                  !ERROR REPORTING
        GO TO 100
8001    WRITE(6,9001)STAT
        GO TO 100
8002    WRITE(6,9002)STAT(1,2),STAT(2,2)
        GO TO 100
8003    WRITE(5,9003)STAT(1,3),STAT(2,3)
        GO TO 100
```

```
9000   FORMAT(X,'OPEN ERROR SRC1 FILE ',Z8,I10)
9001   FORMAT(X,'IO READ/WRITE ERROR ',Z8,I10)
9002   FORMAT(X,'OPEN ERROR SRC2 FILE ',Z8,I10)
9003   FORMAT(X,'OPEN ERROR ON OUTPUT FILE ',Z8,I10)
       END
```

```
C       PROGRAM TO COMPUTE THE INVERSE 2D XFORM OF A COMPLEX
C       ARRAY OF DATA STORRED AS N/2+1 LINES.  OUTPUT IS A
C       SQUARE ARRAY OF REAL NUMBERS NXN.  16<=N<=2048.
C       COMMAND:
C         IFFT2D INFILE OUTFILE DIM

        CHARACTER SNAME*32,DNAME*32,CMDSTR*64
        INTEGER*4 SFILE(60),DFILE(60),STAT(2),DIM,TBL(8)
        REAL DATA(512,514)
        DATA TBL / 16,32,64,128,256,512,1024,2048 /

        I=LIB$GET_FOREIGN(CMDSTR,'$_FILE: ',,J)                !DECODE COMMAND LINE
        CMDSTR(64:64)=','
        J=INDEX(CMDSTR,',')
        SNAME=CMDSTR(:J-1)
        I=INDEX(CMDSTR(J+1:),',')
        DNAME=CMDSTR(J+1:I+J-1)
        IF(DNAME.EQ.' ')GO TO 8000
        READ(CMDSTR(I+J+1:),9002,ERR=8002)DIM

        DO 10 I = 1,8                                          !VALIDATE IMAGE SIZE = PWR 2
10      IF(DIM.EQ.TBL(I))GO TO 20
        GO TO 8002
20      CALL OPEN_(SFILE,SNAME,'.DAT','O',STAT,*8000)          !OPEN INPUT AND OUTPUT DATA FILES
        CALL OPEN_(DFILE,DNAME,'.DAT','N',STAT,,4*DIM,4*DIM,*8000)
        IF(DIM.LE.512)THEN                                     !LOOP FOR IMAGES <= 512 PIXELS
          CALL READ_W(SFILE,DATA,(DIM**2+2*DIM)*4,1,*8001)     !CALLING PROGRAM DOES IO
        ENDIF
        CALL IAFFT(DATA,DIM,SFILE,DFILE)                       !PERFORM INVERSE FFT
        IF(DIM.LE.512)THEN                                     !FOR IMAGES <= 512
          CALL WRITE_W(DFILE,DATA,4*DIM**2,1,*8003)            !CALLING PROGRAM DOES IO
        ENDIF
100     CALL CLOSE_(SFILE)                                     !CLOSE FILES
        CALL CLOSE_(DFILE)
        STOP ' '

8000    WRITE(6,9000)STAT                                      !ERROR REPORTING
        GO TO 100
8001    WRITE(6,9001)STAT
        GO TO 100
8002    WRITE(6,9003)
        GO TO 100
8003    WRITE(6,9004)STAT
        GO TO 100

9000    FORMAT(X,'OPEN ERROR ',Z8,I10)
9001    FORMAT(X,'IO READ ERROR ',Z8,I10)
9002    FORMAT(BN,I6)
9003    FORMAT(X,'DIMENSION ERROR')
        END
```

```
C       SUBROUTINE TO COMPUTE IFFT OF SQUARE ARRAY THAT IS A
C       POWER OF 2
C       INPUT IS COMPLEX ARRAY CONSISTING OF THE AVERAGE VALUE
C       AND N/2 FREQ'S.  OUTPUT IS SQUARE ARRAY NXN PIXELS.

C       IF N <= 512 DATA IS PASSED IN ARRAY
C       IF N >  512 DATA IS ASSUMEN IN FILE

        SUBROUTINE IAFFT(DATA,N,SFILE,DFILE,*)
        INTEGER*4 N,IBLK,OBLK,SFILE(60),DFILE(60)
        REAL DATA(N,N+2)

        IF(N.LE.512)THEN                                        !N > 512 => DATA IN FILE
          CALL ZTORN(DATA(1,1),DATA(1,1),,N)                    !AVG TO REAL
          CALL ZTORN(DATA(1,N+1),DATA(1,2),,N)                  !N/2 TO REAL
          DO 10 I = N,4,-2
10          CALL ZTOR(DATA(1,I-1),DATA(1,I+1),DATA(1,I+2),N)    !CONVERT TO REAL XFORM
          CALL RTOR(DATA(1,1),DATA(1,3),N*2)                    !MOV AVG AND N/2 OUT OF WAY
          DO 20 I =1,N
20          CALL IFFTRI(DATA(1,I+2),N,DATA(1,I+1))              !N REAL FFT'S
          CALL XPOSER(DATA(1,2),N)                              !TRANSPOSE DATA
          DO 30 I =1,N
30          CALL IFFTRI(DATA(1,I+1),N,DATA(1,I))                !N REAL FFT'S
          RETURN

        ELSE IF(N.EQ.1024)THEN
          IBLK = 1                                              !INIT FILE POINTERS
          OBLK = 1
          DO 150 I = 1,1024,128                                 !1024 LINES 128 AT A TIME
            CALL READ_W(SFILE,DATA(1,1),1024*128*4,IBLK,*8000)
            IBLK = IBLK+1024
            IF(I.EQ.1)THEN                                      !FIRST TIME REAL/NYQUIST
              CALL READ_W(SFILE,DATA(1,129),1024*4,8193,*8000)  !GET N/2
              CALL ZTORN(DATA(1,1),DATA(1,1),,1024)             !AVG TO REAL FMT
              CALL ZTORN(DATA(1,129),DATA(1,2),,1024)           !N/2 TO REAL FMT
              DO 110 J = 127,3,-2
110             CALL ZTOR(DATA(1,J),DATA(1,J+2),DATA(1,J+3),1024) !CVT TO REAL XFORM
              CALL RTOR(DATA(1,1),DATA(1,3),1024*2)             !GET AVG N/2 IN POSITION
            ELSE
              DO 120 J = 127,1,-2
120             CALL ZTOR(DATA(1,J),DATA(1,J+2),DATA(1,J+3),1024) !CVT TO REAL XFORM
            ENDIF
            DO 130 J = 1,128                                    !DO REAL IFFT ON 128 LINES
130           CALL IFFTRI(DATA(1,J+2),1024,DATA(1,J))
            K = OBLK
            DO 140 J = 1,1024,128                               !TRANSPOSE AND WRITE
              CALL XPOSE1(DATA(J,1),DATA(1,129),1024,128)
              CALL WRITE_W(DFILE,DATA(1,129),128*128*4,K,*8001)
140           K = K+1024
150         OBLK = OBLK + 128                                   !BUMP OUTPUT BLOCK BASE
```

```
             OBLK=1
             DO 180 I = 1,1024,128                                   !ROWS 128 AT A TIME
               K = OBLK
               DO 160 J = 1,1024,128                                 !FINISH READ AND XPOSE
                 CALL READ_W(DFILE,DATA(1,129),128*128*4,K,*8001)
                 K = K+128
160              CALL XPOSE2(DATA(1,129),DATA(J,1),1024,128)
               DO 170 J = 128,1,-1                                   !IFFT ON ROWS
170              CALL IFFTRI(DATA(1,J),1024,DATA(1,J+1))
               CALL WRITE_W(DFILE,DATA(1,2),1024*128*4,OBLK,*8001)
180            OBLK=OBLK+1024

           ELSE IF(N.EQ.2048)THEN
             IBLK = 1                                                !INIT FILE POINTERS
             OBLK = 1
             DO 250 I = 1,2048,64                                    !2048 LINES 64 AT A TIME
               CALL READ_W(SFILE,DATA(1,1),2048*64*4,IBLK,*8000)
               IBLK = IBLK+1024
               IF(I.EQ.1)THEN                                        !FIRST TIME REAL/NYQUIST
                 CALL READ_W(SFILE,DATA(1,65),2048*4,32769,*8000)    !GET N/2
                 CALL ZTORN(DATA(1,1),DATA(1,1),,2048)               !AVG TO REAL FMT
                 CALL ZTORN(DATA(1,65),DATA(1,2),,2048)              !N/2 TO REAL FMT
                 DO 210 J = 63,3,-2
210                CALL ZTOR(DATA(1,J),DATA(1,J+2),DATA(1,J+3),2048) !CVT TO REAL XFORM
                 CALL RTOR(DATA(1,1),DATA(1,3),2048*2)               !GET AVG N/2 IN POSITION
               ELSE
                 DO 220 J = 63,1,-2
220                CALL ZTOR(DATA(1,J),DATA(1,J+2),DATA(1,J+3),2048) !CVT TO REAL XFORM
               ENDIF
               DO 230 J = 1,64                                       !DO REAL IFFT ON 64 LINES
230              CALL IFFTRI(DATA(1,J+2),2048,DATA(1,J))
               K = OBLK
               DO 240 J = 1,2048,64                                  !TRANSPOSE AND WRITE
                 CALL XPOSE1(DATA(J,1),DATA(1,65),2048,64)
                 CALL WRITE_W(DFILE,DATA(1,65),64*64*4,K,*8001)
240              K = K+1024
250            OBLK = OBLK + 32                                      !BUMP OUTPUT BLOCK BASE
             OBLK=1
             DO 280 I = 1,2048,64                                    !ROWS 64 AT A TIME
               K = OBLK
               DO 260 J = 1,2048,64                                  !FINISH READ AND XPOSE
                 CALL READ_W(DFILE,DATA(1,65),64*64*4,K,*8001)
                 K = K+32
260              CALL XPOSE2(DATA(1,65),DATA(J,1),2048,64)
               DO 270 J = 64,1,-1                                    !IFFT ON ROWS
270              CALL IFFTRI(DATA(1,J),2048,DATA(1,J+1))
               CALL WRITE_W(DFILE,DATA(1,2),2048*64*4,OBLK,*8001)
280            OBLK=OBLK+1024
           ENDIF
```

```
      RETURN
8000  RETURN(1)
8001  RETURN(2)
      END
```

APPENDIX B

COMPUTER MODEL OF IMAGE PROCESSOR

The image channel design (i.e. base memory, image processor, display memory) represents a unique architecture. In the initial design phases there was no previous design that could be used as a reference. It was also a very complicated design. The image processor was designed as a pipelined architecture consisting of 30 stages. Image data, graphics overlay data, and addresses were all required to be synchronized as they emerged from the output of the image processor. The design of the image processor pipeline became more sophisticated due to the inclusion of the COZO processor. The portion of the pipeline designed for the COZO processor was required to handle a different number of output pixels than input pixels.

A computer model was written to simulate the operation of the Image Processor. This model is presented in the computer programs included in this section. The model was used to verify the Image Processor algorithms and to identify the initialization requirements. It was also used to refine the hardware requirements of some of the algorithms. An example of this is the number of stages that were required in the spatial filter.

The MAIN program contains the initialization logic, address generation logic, subtraction logic, and lookup table processing. The second routine, called MACH 2, contains the logic to perform the spatial filtering with its associated initialization and buffering for COZO. The last routine, called COZO, implements the continuous zoom processor for both magnification and minification. The output of COZO is returned to the program MAIN where the display memory address logic is simulated.

Many problems with the design as conceived in the original block diagram were uncovered . Some enhancements to the basic design also became obvious as the model was developed. An example of this is the inclusion of the X and Y image flip capability. The operation of the processor was used, with examples, as input to the design team. A new block diagram of the design was developed. It was reviewed against the model before the actual chip level design was initiated.

```
      PROGRAM MAIN

C     T4 SIMULATION   FISHER  21 JAN 85

      BYTE BUFF(512*512),OUT(512*512)
      INTEGER I,J,K,N,M,LINE,ADDRESS,A,B,C,D,EOL,FLAG2,FLAG3,FLAG4,DATA2
      INTEGER PIXCNT,DATAO,DELTAB(0:2),ADPIPE1,ADPIPE2,ADPIPE3
      INTEGER BEOL,ALUREG1,ALUREG2,BPIXCNT,BXREG,BYREG,DATA1,FLAG1
      INTEGER BEOL1,BEOL2,BEOL3,BEOL4,VALID0,VALID1,VALID2,INICNT
      INTEGER PIXPCH,LINPCH,PIXNO,BXDELTA,BYDELTA
      INCLUDE 'SIMCOM.FOR'


      CALL VSINIT(4,IOSTAT)                     !INIT DISPLAY CHANNEL
      CALL VSMAP_D                              !DISABLE DISPLAY MAP
      CALL VSOVR_D                              !DISABLE DISPLAY OVERLAY

      DO 10 I = 1,512,64                        !READ IMAGE FROM DISPLAY
10    CALL VSGREY_RW(BUFF(1+(I-1)*512),64,I)
      CALL MOV('ABTOAB',BUFF,OUT,512*512)       !DUPLICATE IN OUTPUT

C     TEMPROARY PROCESSOR INIT

5     WRITE(5,9001)
      READ(5,9002,ERR=5)ZOOM,NN
9001        FORMAT(X,'ENTER ZOOM FACTOR ',$)
9002        FORMAT(BN,F6.0,I3)
      PIXLIN = 100                              !PIXELS PER LINE
      LINIMG = 100                              !LINE PER IMAGE
      ITOPPIX = 40                              !UPPER PIXEL INPUT IMAGE
      ILEFTPIX = 40                             !LEFT MOST PIXEL INPUT OMAGE
      OTOPPIX = 40                              !UPPER PIXEL OUTPUT IMAGE
      OLEFTPIX = 40                             !LEFT MOST PIXEL OUTPUT IMAGE
      SUBIMG = 0                                !1 => SUBIMAGE PROCESSING
      SUBENA = 0                                !NO SUBTRACTER
      FLIP = 0                                  !1 => XFLIP,2 => YFLIP,3 => BOTH

      IF(ZOOM.LT.1)THEN
        IF(NN.EQ.0)THEN                         !NO NEAREST NEIGHBOR
          PITCH = 1./ZOOM+.999                  !GET SUBSAMPLE FACTOR
          ZOOM = ZOOM*PITCH                     !COMPUTE ZOOM 1 < ZOOM < 2
          PIXLIN = PIXLIN*PITCH                 !ADJUST IMAGE SIZE
          LINIMG = LINIMG*PITCH
        ELSE
          PITCH = 0
        ENDIF
      ELSE                                      !IF ZOOM GE 1
        PITCH = 0                               !DISABLE SUBSAMP
      ENDIF
      IF(FLIP.EQ.0)THEN                         !IF NO HORIZONTAL FLIP
        ALUREG1 = 0                             !(X+Y)
```

```
  ALUREG2 = 0                              !ADDR + (X+Y)
ELSE IF(FLIP.EQ.1)THEN                     !HORIZONTAL FLIP
  ALUREG1 = 1                              !(-X+Y)
  ALUREG2 = 0                              !ADDR + (-X+Y)
ELSE IF(FLIP.EQ.2)THEN                     !VERTICAL FLIP
  ALUREG1 = 2                              !(X-Y)
  ALUREG2 = 0                              !ADDR + (X-Y)
ELSE IF(FLIP.EQ.3)THEN                     !FLIP BOTH
  ALUREG1 = 0                              !(X+Y)
  ALUREG2 = 1                              !ADDR - (X+Y)
ENDIF
INICNT = 0                                 !CLEAR INIT COUNTER
XDELTA = 2047./ZOOM+.5                     !COEF GENERATOR DELTAS
YDELTA = 2047./ZOOM+.5
ZOOM = 2047./XDELTA                        !CORRECT FOR INTEGER MATH
IF(ZOOM.LT.1)THEN                          !>1 => NEAREST NEIGHBOR
  XDELTA = 2048
  YDELTA = 2048
  YREG = 2047
ELSE
  YREG = 0
ENDIF
IF(SUBIMG.EQ.0)THEN                        !NO SUBIMAGE PROCESSING
  IF(ZOOM.EQ.1)THEN
    PIXNO = PIXLIN
  ELSE IF(ZOOM.GT.1)THEN
    IF(PITCH.EQ.0)THEN                     !NO FRACTIONAL ZOOM
      PIXNO = 1.+(PIXLIN-1)/ZOOM+.5        !PIXELS/LINE BASE MEMORY
    ELSE
      PIXNO = 1.+(PIXLIN-1)/ZOOM+.999
    ENDIF
  ELSE
    PIXNO = PIXLIN                         !JUMP N PIXELS
  ENDIF
ELSE
  PIXNO = PIXLIN/ZOOM+31                   !ADD 15 ON EACH SIDE
ENDIF
ADRLOD = PIXNO                             !PIXELS IN N-1 LINE BUFFER
OPIXNO = PIXLIN                            !OUTPUT PIXEL COUNT
LINNO = LINIMG                             !PROCESS 100 LINES
LINCNT = LINNO                             !N-1 BUFF REPEAT (LINE) COUNTER
IF(ZOOM.GE.1)THEN                          !NEED ALL PIXELS AND LINES
  BXDELTA = 2047                           !STEP BY ONE PIXEL
  BYDELTA = 2047                           !STEP BY ONE LINE
  BYREG = 2047                             !START COEF
ELSE
  BXDELTA = 2047./ZOOM                     !ZOOM FACTOR (2048/ZOOM)
  BYDELTA = 2047./ZOOM                     !ZOOM FACTOR (2048/ZOOM)
  BYREG = 2047                             !START COEF
ENDIF
IF(SUBIMG.EQ.0)THEN
```

```
        ADDRESS = ILEFTPIX+512*ITOPPIX        !ADDR OF FIRST PIXEL IN BASE MEM
      ELSE
        ADDRESS = ILEFTPIX-15+512*ITOPPIX
      ENDIF
      ADDRESS = ADDRESS - BXDELTA/2048        !ANTICIPATE FIRST ADD
      IF(FLIP.EQ.1)THEN                       !HOR FLIP ON
        ADDRESS = ADDRESS+PIXNO+1             !ADD DELTA TO END OF FIRST LINE
      ELSE IF(FLIP.EQ.2)THEN                  !VERTICAL FLIP ON
        ADDRESS = ADDRESS+512*LINCNT          !ADD DELTA TO BEG OF LAST LINE
      ELSE IF(FLIP.EQ.3)THEN                  !FLIP BOTH X AND Y
        ADDRESS = ADDRESS+512*LINCNT+PIXNO+1  !ADD DELTA TO END OF LAST LINE
      ENDIF
      OADDR = OLEFTPIX+512*OTOPPIX+1
      IF(PITCH.EQ.0)THEN                      !IF NO INTERPOLATIVE SHRINK
        OWRAP = 512 - OPIXNO+1                !OUTPUT PIXEL WRAP TO NEXT LINE
      ELSE
        OWRAP = 512 - OPIXNO/PITCH +1         !WRAP AFTER PITCHING PIXELS
      ENDIF
      DO 15 I = -128,127                      !INIT WINDOW LEVEL LUT
15    LUT(I) = I
      DO 16 J = 1,4                           !FIR COEF'S TO 0
      DO 16 I = 1,8
16    COEF(I,J) = 0
      COEF(8,4) = 0                           !USE ODD # OF COEFS
      COEF(8,2) = 127                         !DELTA FUNCTION IN FIR
      J = 2048                                !INIT DUMMY YREG VAriable to 0
      DO 17 I = LINCNT-1,0,-1                 !COMPUTE REPEAT COEFFS 1=>NEW
      IF(J/2048.GE.1)THEN                     !CARRY => NEW LINE
        REPBUF(I) = 1
      ELSE
        REPBUF(I) = 0
      ENDIF
      J = IAND(J,2047)                        !MASK OFF OVERFLOW
17    J = J + YDELTA
      REPBUF(LINCNT-2) = 1                    !FORCE NEW READ OF SECOND LINE
      TOCOMP = 1                              !HAVE TWO'S COMP DATA
      DELTAB(0) = 0                           !ADDRESS WRAPS FROM Y ADDR
      IF(ZOOM.GE.1)THEN                       !WHEN NOT MINIFY MODE
        IF(FLIP.EQ.0)THEN                     !FLIP OFF
          DELTAB(1) = 512 - PIXNO + 1         !FROM END OF N TO BEG OF N+1
        ELSE IF(FLIP.EQ.1)THEN                        !X FLIP ON
          DELTAB(1) = 512 + PIXNO - 1         !FROM BEG OF N TO END OF N+1
        ELSE IF(FLIP.EQ.2)THEN                        !Y FLIP ON
          DELTAB(1) = 512 + PIXNO - 1         !FROM END OF N TO BEG ON N-1
        ELSE IF(FLIP.EQ.3)THEN                        !FLIP BOTH X AND Y
          DELTAB(1) = 512 - PIXNO + 1         !FROM BEG OF N TO END OF N-1
        ENDIF
      ELSE IF(ZOOM.LT.1)THEN                  !IN MINIFY MODE
        J = BYDELTA/2048                      !ISOLATE # OF LINES TO STEP
        BYDELTA = BYDELTA-(J-1)*2048          !MAKE IN RANGE 0 TO 1
        IF(FLIP.EQ.0)THEN                     !FLIP OFF
```

```
          DELTAB(1) = 512*J-(PIXNO-1)/ZOOM    !FROM END OF N TO BEG OF N+1
          DELTAB(2) = DELTAB(1) + 512
        ELSE IF(FLIP.EQ.1)THEN                !X FLIP ON
          DELTAB(1) = 512*J + PIXNO - 1       !FROM BEG OF N TO END OF N+1
          DELTAB(2) = DELTAB(1) + 512
        ELSE IF(FLIP.EQ.2)THEN                !Y FLIP ON
          DELTAB(1) = 512*J + PIXNO - 1       !FROM END OF N TO BEG ON N-1
          DELTAB(2) = DELTAB(1) + 512
        ELSE IF(FLIP.EQ.3)THEN                !FLIP BOTH X AND Y
          DELTAB(1) = 512*J - PIXNO + 1       !FROM BEG OF N TO END OF N-1
          DELTAB(2) = DELTAB(1) + 512
        ENDIF
      ENDIF

C   MACH 1 --> ADDR COMPUTER AND BASE MEMORY IO

100 IF(INICNT.LT.100)THEN
        INICNT = INICNT+1
        INIT = 1
        ADPIPE1 = 0                           !INIT ADDRESS PIPE STAGES
        ADPIPE2 = 0
        ADPIPE3 = 0
        BEOL1 = 0
        BEOL2 = 0
        VALID0 = 0
        VALID1 = 0
        VALID2 = 0
        FLAG1 = 0
        FLAG2 = 0
        FLAG3 = 0
        FLAG4 = 0
      ELSE
        INIT = 0
      ENDIF

      IF(INIT.EQ.1)THEN
        BXREG = 2047                          !X REG TO .9995
        BPIXCNT = PIXNO                       !LOAD PIXEL COUNTER
      ELSE IF(FLAG1.EQ.0)THEN

C     STAGE 3

        IF(ALUREG2.EQ.0)THEN                  !FORWARD => ADD
          ADDRESS = ADDRESS + ADPIPE3         !ADDR + ()
        ELSE IF(ALUREG2.EQ.1)THEN             !FLIP BOTH X AND Y
          ADDRESS = ADDRESS - ADPIPE3         !ADDR - ()
        ENDIF
        IF((ADPIPE3.NE.0).OR.(VALID2-VALID1.LT.0))THEN!IF NEW ADDRESS
          DATA1 = BUFF(ADDRESS)               !READ DATA
          FLAG1 = 1                           !SET DATA AVAIL FLAG
          VALID2 = VALID1
```

```
        ENDIF
C     WRITE(5,9000)ADDRESS,MOD(ADDRESS,512),FLAG1
        BEOL2 = BEOL1

C     STAGE 2

        IF(ALUREG1.EQ.0)THEN                      !FORWARD => DIRECT ADD
          ADPIPE3 = ADPIPE1+DELTAB(ADPIPE2)       !(X+Y)
        ELSE IF(ALUREG1.EQ.1)THEN                 !X FLIP
          ADPIPE3 = -ADPIPE1+DELTAB(ADPIPE2)      !(-X+Y)
        ELSE IF(ALUREG1.EQ.2)THEN                 !Y FLIP
          ADPIPE3 = ADPIPE1-DELTAB(ADPIPE2)       !(X-Y)
        ENDIF
        VALID1 = VALID0

C     STAGE 1

        IF(BEOL1.EQ.1)THEN                        !IF AT EOL
          BXREG = 2047                            !RELOAD XREG TO .9995
          ADPIPE1 = BXREG/2048                    !ADDRESS DELATA TO PIPE
        ELSE
          BXREG = BXREG + BXDELTA                 !ADD FRACTION TO COEFF
          ADPIPE1 = BXREG/2048                    !ADDRESS DELATA TO PIPE
          BXREG = IAND(BXREG,2047)                !SAVE LS 11 BITS
        ENDIF
        VALID0 = 1                                !SET VALID DATA

        IF(BEOL1.EQ.1)THEN                        !IF AT EOL
          BYREG = BYREG + BYDELTA
          ADPIPE2 = BYREG/2048                    !ISOLATE ADDRESS BITS
          BYREG = IAND(BYREG,2047)                !ISOLATE LS 11 BITS
          LINNO = LINNO - 1                       !DEC LINE COUNT
        ELSE
          ADPIPE2 = 0
        ENDIF

        BPIXCNT = BPIXCNT - 1                     !DEC PIXEL COUNTER
        IF(BPIXCNT.LE.0)THEN                      !IF COUNTER OVERFLOW
          BPIXCNT = PIXNO                         !REINIT PIXEL COUNTER
          BEOL1 = 1                               !SET EOL FLAG
        ELSE
          BEOL1 = 0                               !CLEAR EOL FLAG
        ENDIF

C     WRITE(5,9000)ADDRESS,DATA1,FLAG1,BEOL2
C     FLAG1 = 0
C     GO TO 200
C
      ENDIF

C     MACH 2
```

```
      CALL MACH2(DATA1,FLAG1,BEOL2,A,B,C,D,FLAG2,BEOL3,*5) !CALL MACHINE 2

C     IF(FLAG2.NE.0)THEN
C       WRITE(5,9000)DATA1,A,B,C,D,BEOL2,OADDR,BEOL3
C     ENDIF
C     FLAG2 = 0
C     GO TO 200

C     MACH 3

      CALL COZO(A,B,C,D,FLAG2,DATA2,FLAG3,BEOL4)      !CALL COZO

C     IF(FLAG3.NE.0)THEN
C       WRITE(5,9000)C,D,DATA2,BEOL4
C     ENDIF
C     FLAG3 = 0
C     GO TO 200

      IF(((FLAG3.EQ.1).AND.(FLAG4.EQ.0)).OR.(INIT.EQ.1))THEN!TO SHIFT PIPE
        IF(INIT.EQ.1)THEN
          PIXPCH = 0
          LINPCH = 0
        ELSE
          FLAG3 = 0
          IF(LINPCH.LE.0)THEN
            IF((PIXPCH.LE.0).OR.(BEOL4.EQ.1))THEN
              PIXPCH = PITCH
              OUT(OADDR) = DATA2
              IF(BEOL4.EQ.1)THEN
                LINPCH = PITCH
                I = OADDR/512*512+1
                J = OADDR/512+1
                CALL VSGREY_WW(OUT(I),1,J)
                OADDR = OADDR + OWRAP
              ELSE
                OADDR = OADDR + 1
              ENDIF
C             WRITE(5,9000)DATA2,OADDR,I,J
            ENDIF
            PIXPCH = PIXPCH-1
          ENDIF
          IF(BEOL4.EQ.1)THEN
            LINPCH = LINPCH-1
          ENDIF
        ENDIF
      ENDIF

200   GO TO 100
9000        FORMAT(X,10I8)
      END
```

```
      SUBROUTINE MACH2(DATAI,FLAGI,EOLI,A,B,C,D,FLAGO,EOLO,*)

      INTEGER SUM(9,0:4),DATA(4),DELAY(38),NM1BUFF(512,0:1)
      INTEGER DATAI,FLAGI,A,B,C,D,FLAGO,EOLI,EOLO,EOLD,COUNT,PIPE1,PIPE2
      INTEGER SUBTMP,ADDR,ENDLIN,STATE,STATE0,EOL(22),EOI,LINE1

C     INCLUDE '[IP]SIMCOM.FOR'

C     INIT SECTION

      IF(INIT.EQ.1)THEN
        ADDR = ADRLOD                           !N-1 LINE ADDR POINTER
        IF((I.LT.1).OR.(I.GT.22))I = 1          !INIT DATA VALID PIPE
        DELAY(I)=0
        EOL(I) = 0
        I = I+1
        FLAGO = 0                               !INIT OUTPUT AVAIL FLAG
        STATE0 = 1                              !START FILLING FIRST LINE
        STATE = STATE0
        COUNT = 30                              !PRELOAD 15 PIXELS
      RETURN
      ENDIF

C     IF PIPE FULL MUST BYPASS THIS CYCLE

      IF((FLAGO.EQ.1).OR.(FLAGI.EQ.0))THEN
        RETURN
      ENDIF
C     WRITE(5,9000)DATAI,FLAGI,D,EOLI,ADDR
c     WRITE(5,9000)PIPE1,DELAY(1)
c     DO 40 J = 1,4
c     DO 40 I = 1,9
c40   WRITE(5,9000)SUM(I,J),DELAY(I+(J-1)*9+1)
c     WRITE(5,9000)PIPE2,DELAY(22)
c     WRITE(5,9000)DATA(1),DATA(2),DATA(3),DATA(4)
9000        FORMAT(X,6I4)
C     OUTPUT SECTION - WRITES TO NEXT STAGE IF VALID DATA AND PIPE CLEAR

      IF(DELAY(22).EQ.1)THEN
        C = D                                   !STEP PIPE
        A = B
        FLAGO = 1                               !SET OUTPUT AVAIL
        IF(REPBUF(LINCNT-1).EQ.1)THEN           !USE NEW DATA IN
          B = NM1BUFF(ADDR,0)                   !B FORM N BUFFER
          NM1BUFF(ADDR,1)=NM1BUFF(ADDR,0)       !N TO N-1
          D = PIPE2                             !NEW DATA TO D
          NM1BUFF(ADDR,0) = PIPE2               !NEW DATA TO N
          I = 0                                 !NEW DATA LOOP FLAG
        ELSE                                    !REPEAT DATA IN N BUFFS
          B = NM1BUFF(ADDR,1)                   !N-1 LINE
```

```
          D = NM1BUFF(ADDR,0)                    !N LINE
          I = 1                                  !OLD DATA LOOP FLAG
        ENDIF
        ADDR = ADDR - 1                          !DEC ADDRESS (PIXEL) COUNT
        IF(ADDR.EQ.0)THEN                        !IF AT EOL
          EOLO = 1                               !SET EOL OUT FLAG
          ADDR = ADRLOD                          !RELOAD PIXEL COUNTER
          LINCNT = LINCNT - 1                    !DEC LINE COUNTER
          IF(LINCNT.EQ.0)THEN                    !IF DONE ALL LINES
            EOI = 1                              !SET END OF IMAGE FLAG
            RETURN 1
          ENDIF
          ELSE
            EOLO = 0                             !CLEAR OUTPUT EOL
        ENDIF
        IF(I.EQ.1)RETURN                         !IF OLD DATA SKIP FIR CYCLE
      ENDIF
      PIPE2 = SUM(9,4)/128                       !OUTPUT SUM TO PIPE STAGE

      DO 220 I = 22,3,-1                         !SHIFT PIPE
  220 DELAY(I) = DELAY(I-1)
      DO 230 I = 22,2,-1
  230 EOL(I) = EOL(I-1)
      EOL(1) = EOLD
C     STATE MACHINE FOR EOL CONTROL

      DATA(4) = DATA(3)                          !SHIFT DATA IN FIR'S
      DATA(3) = DATA(2)
      DATA(2) = DATA(1)

C     STATE = 0 => NORMAL OPERATION

      IF(STATE.EQ.0)THEN
        DELAY(2) = DELAY(1)                      !OUTPUT DATA IS TO BE WRITTEN
        DATA(1) = LUT(PIPE1)                     !INPUT DATA
        IF(EOLD.EQ.1)THEN                        !IF EOL BIT SET SHIFT STATES
          STATE0 = 1                             !TO STATE 1
          COUNT = 30                             !STATE 1 OPERATES FOR 30 PIXELS
        ENDIF

C     STATE = 1 => "0" PAD OR FILL WITH BOUNDARY DATA

      ELSE IF(STATE.EQ.1)THEN
        IF(SUBIMG.EQ.0)THEN                      !IF NO BOUNDARY DATA
          DATA(1) = 0                            !THEN PAD WITH 0'S
        ELSE                                     !IF USE BOUNDARY DATA, EOL
          DATA(1) = LUT(PIPE1)                   !GET DATA
        ENDIF
        DELAY(2) = 0                             !NO VALID PIXELS
        COUNT = COUNT - 1                        !ONLY 30 PIXELS IN STATE 1
```

```
          IF(COUNT.EQ.0)THEN
            STATE0 = 0                                 !STEP TO STATE 2 AFTER 15
          ENDIF
        ENDIF

C       PROCESSING SECTION

        SUM(9,0) = 0                                   !INPUT TO FIRST FIR IS GND
        DO 210 J = 4,1,-1                              !FOUR CHIPS IN SERIES
        DO 200 I = 9,2,-1                              !EIGHT COEF'S PER CHIP
        SUM(I,J)=COEF(I-1,J)*DATA(J)+SUM(I-1,J)        !FORM SUM OF PRODUCTS
200     CONTINUE
        SUM(1,J) = SUM(9,J-1)                          !PIPE SUM FROM PREVIOUS CHIP
210     CONTINUE


C       DISABLE CLOCK FOR INITIAL PIPE STAGES WHEN PADDING WITH 0'S

        IF((STATE.NE.1).OR.(SUBING.NE.0))THEN
          IF(TOCOMP.EQ.0)THEN                          !IF TWO'S COM DATA NO CHANGE
            DATAI=DATAI-128                            !MAKE TWO'S COMP
          ENDIF
          IF(SUBENA.EQ.1)THEN                          !IF SUBTRACT LOGIC ENABLED
            IF(CLK.EQ.0)THEN                           !IF 200 NSEC CLOCK LOW
              SUBTMP = DATAI                           !SUB HOLDING REG GETS DATA
              RETURN                                   !NEED TWO CALLS FOR SUBTRACT
            ELSE
              PIPE1 = DATAI - SUBTMP                   !DIFFERENCE BETWEEN PIXELS
              DELAY(1) = FLAGI
            ENDIF
          ELSE
            PIPE1 = DATAI                              !USE INPUT DATA DIRECT
            DELAY(1) = FLAGI
          ENDIF
          FLAGI = 0                                    !SET DATA ACCEPTED
          EOLD = EOLI                                  !PIPE EOL TO FIR'S
        ENDIF
        STATE = STATE0                                 !UPDATE STATE VECTOR
        RETURN
        END
```

```
      SUBROUTINE COZO(A,B,C,D,FLAGI,DATAO,FLAGO,EOLO)

      INTEGER A,B,C,D,FLAGI,DATAO,FLAGO
      INTEGER X11,X12,X1O,X21,X22,X2O,X31,X32,X3O,X41,X42,X4O
      INTEGER PA,PB,PC,PD,P1,P2,P3,P4,XCOEF,YCOEF,PIXCNT
      INTEGER XREG,EOL0,EOL1,EOL2,EOL3,EOL4,EOLO,DEL1,DEL2,DEL3,DEL4

      INCLUDE 'SIMCOM.FOR'

      IF(INIT.EQ.1)THEN
        PIXCNT = OPIXNO
        XREG = 2047
        YCOEF = 255
        DEL4 = 0
        DEL3 = 0
        DEL2 = 0
        DEL1 = 0
        FLAGI = 0
        FLAGO = 0
        RETURN
      ENDIF

      IF((FLAGI.NE.1).OR.(FLAGO.NE.0))THEN
        RETURN
      ENDIF

C     COZO COEFFICENT GENERATOR

      FLAGO = DEL4                      !STEP VALID BIT PIPE
      DEL4 = DEL3
      DEL3 = DEL2
      DEL2 = DEL1
      DEL1 = FLAGI
      EOLO = EOL4                       !STEP EOL FLAG PIPE
      EOL4 = EOL3
      EOL3 = EOL2
      EOL2 = EOL0

      IF(EOL0.EQ.1)THEN                 !WAS AT END OF LINE
        XREG = 2047                     !RESET XREG TO .9995
      ELSE
        XREG = IAND(XREG,2047)          !MASK OFF OVERFLOW
      ENDIF
      XCOEF = XREG/8                    !SEL BITS 03:10
      IF(EOL0.EQ.1)THEN                 !INC Y COEF A EOL
        YREG = YREG + YDELTA            !preload with 2047-YDELTA
        YREG = IAND(YREG,2047)          !MASK OFF OVERFLOW
        YCOEF = IAND(YREG/8,255)        !SET BITS 03:10
C     WRITE(5,9001)YCOEF/256.
9001        FORMAT(X,F6.4)
```

```
      ENDIF

      IF(INIT.EQ.1)RETURN            !EXIT HERE AFTER CLEARING PIPE

      DATAO = P4+PA+X40/256          !STAGE 5

      PA = PB                        !STAGE 4
      P4 = P3
      X40 = X41*X42

      P3 = P2                        !STAGE 3
      PB = X10/256+X20/256
      X41 = PC
      X42 = X30/256

      P2 = P1                        !STAGE 2
      X10 = X11*X12
      X20 = X21*X22
      PC = PD
      X30 = X31*X32

      P1 = A                         !STAGE 1
      X11 = XCOEF
      X12 = B-A
      X21 = YCOEF
      X22 = C-A
      PD = D-C-B+A
      X31 = XCOEF
      X32 = YCOEF

      XREG = XREG + XDELTA           !ADD FRACTION TO COEF
      IF(XREG/2048.GT.0)THEN         !AT ADDRESS OVERFLOW
        FLAGI = 0                    !NEW PIXEL PAIR ON OVERFLOW
      ENDIF

      PIXCNT = PIXCNT -1             !PIXELS PER LINE COUNTER
      IF(PIXCNT.LE.0)THEN
        PIXCNT = OPIXNO              !RESET PIXCNT TO LINE COUNT
        EOLO = 1                     !SET EOL FLAG
        FLAGI = 0                    !CLOCK DATA PIPE ON OVERFLOW
      ELSE
        EOLO = 0                     !CLEAR EOL FLAG
      ENDIF

C     WRITE(5,9000)A,B,C,D,XCOEF,YCOEF,DATAO,FLAGO,EOLO
      RETURN

9000          FORMAT(X,10I6)

      END
```

REFERENCES

1. S. Nudelman, H. D. Fisher, M. M. Frost, M. P. Capp, T. W. Ovitt,"A Study of Photoelectronic-Digital Radiology - Part I: The Photoelectronic-Digital Radiology Department," Proc. IEEE,vol. 70, pp. 700-707, July 1982.

2. S. J. Dwyer, A. W. Templeton, N. L. Martin, et al, "Cost of managing digital diagnostic imasges for a 614 bed hospital,"1st International Conference and Worshop on Picture Archiving and Communications Systems (PACS) for Medical Applications, SPIE, vol 318, pp. 3-8, 1982.

3. S. J. Dwyer, A. W. Templeton, M. L. Tarlton, et al, "Salient Characteristics of a Distributed Diagnostic Imaging Management system for a Radiology Department," International Conference and Worshop on Picture Archiving and Communications Systems (PACS) for Medical Applications, SPIE, vol 318, pp. 256-260, 1982.

4. R. A. Kruger, 'Combined Acquisition/Processing for Data Reduction,"1st International Conference and Worshop on Picture Archiving and Communications Systems (PACS) for Medical Applications, SPIE, vol 318, pp. 256-260, 1982.

5. D. F. Preston, S. J. Dwyer, A. W. Templeton, et al, "A Digital Management System,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, 1983.

6. H. K. Huang, N. J. Mankovich, Z. Barbaric, H. Kangarloo, C Moler,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, pp. 189-198, 1983.

7. W. H. Anderson, M. A. Tarlton, K. S. Hensley, A. W. Templeton, S. J. Dwyer,"Implementation of a Diagnostic Display and Image Manipulation Node,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, pp. 225-232, 1983.

8. J. P. Neeley, M. W. Vannier, F. R. Gutierrez, P. l. Von Behren,"Practical Considerations in Digital Cardiac Angiography,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, pp. 266-279, 1983.

9. V. M. Spitzer,"The costs and Benefits of Digital Electronic Image Handling,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, 1983.

10. Radiological Society of North America, 71st Scientific Assembly and Annual Meeting, Chicago November 17-22, vol 157, pp. 323, 1985.

11. J. L. Leher,"Impact of Manual and Computer-assisted PACS for Automated PACS,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, pp. 6-13, 1983.

12. J. R. Perry, R. E. Johnston, E. V. Staab, et al,"Digital Image Display Console Design Issues,"International Symposium of Medical Images and Icons, IEEE Comp. Soc., pp. 18-22, 1984.

13. J. R. Perry, B. G. Thompson, E. V. Staab, S. M. Pizer, R. E. Johnston,"Performance Features for a PACS Display Console",Computer (IEEE), vol 16, No. 8, pp. 51-56, 1983.

14. W. K. Pratt, Digital Image Processing, John Wiley & Sons, New York, pp. 39-41, 1978.

15. J. F. Quinn, M. L. Rhodes, B. Rosner,"Data Compression Techniques for CT Image Archival,"2nd International Conference on Picture Archiving and Communications Systems (PACS II) for Medical Applications, SPIE, vol 418, pp. 209-212, 1983.

16. R. C. Gonzalez, P Wintz, Digital Image Processing, Addison-Wesley,Reading, Mass., pp. 19-27, 1977.

17. J. F. Dunn, "Cathode Ray Tube Film Recording of Video Based Medical Images,"1st International Conference and Worshop on Picture Archiving and Communications Systems (PACS) for Medical Applications, SPIE, vol 318, pp. 216-222, 1982.

18. W. K. Pratt, Digital Image Processing, John Wiley & Sons, New York, pp. 113-119, 1978.

19. H. D. Fisher, G. W. Seeley, G. W. Bjelland, T. W. Ovitt, M. P. Capp, "Psychophysical Evaluation of the Necessary Spatial and Contrast Resolution for a Picture Archiving and Communication System: Works in Progress," International Symposium of Medical Images and Icons, IEEE Comp. Soc., pp. 13-16, 1984.

20. G. W. Seeley, H. D. Fisher, M. O. Stemski, M. Borgstrom, J. Bjelland, "Total Digital Radiology Department: Spatial Resolution Requirements,"Accepted for Publication in American Journal of Roentgenology