Real-Time Reconstruction Of Log Cross-Sections Using Tomographic Data

A thesis presented in fulfilment of the requirements for the degree of Doctor of Philosophy at Massey University

Philip Colin Long
November 1993

Massey University Library
Thesis Copyright Form

Title of thesis:

(1) (a) I give permission for my thesis to be made available to readers in Massey University Library under conditions determined by the Librarian.

(b) I do not wish my thesis to be made available to readers without my written consent for . . . months.

(2) (a) I agree that my thesis, or a copy, may be sent to another institution under conditions determined by the Librarian.

(b) I do not wish my thesis, or a copy, to be sent to another institution without my written consent for . . . months.

(3) (a) I agree that my thesis may be copied for Library use.

(b) I do not wish my thesis to be copied for Library use for . . . months.

Signed 

Date 13-12-93

The copyright of this thesis belongs to the author. Readers must sign their name in the space below to show that they recognise this. They are asked to add their permanent address.

NAME AND ADDRESS DATE

ABSTRACT

This thesis examines the application of internal imaging technologies to the detection of defects within felled logs. This application requires images of moderate resolution to be generated at high-speeds. Transmission tomography is suggested as the most appropriate imaging technology, with particular reference to X-ray transmission tomography. High-speed X-ray scanners suitable to the application exist.

An international literature search was performed to find reconstruction systems capable of high-speed reconstruction of the X-ray data. Those systems claiming to be high-speed are discussed in regard to their means of achieving high-speed reconstruction. The discussions show that most effort has been directed toward sophisticated hardware implementation of the reconstruction process, rather then the mathematics of the reconstruction process itself. To increase the achievable rate of reconstruction, the mathematics of each process in the reconstruction algorithm are examined in this thesis, with the aim of reducing computational complexity.

Convolution backprojection is the most commonly used reconstruction method in transmission tomography when the X-ray data are complete, and can be neatly separated into two separate processes, convolution (filtering) and backprojection, as the name suggests. This reconstruction method was chosen as suitable for the log processing application because of its mathematical simplicity and quality of image reconstruction.

Truncation of the convolution kernel is examined, and simulated results show adequate reconstruction quality with significant truncation. An inexpensive hardware design capable of performing the convolution operation in real-time is developed.

The backprojection process normally employed is computationally expensive, and is the major encumbrance to the realisation of simple high speed image reconstruction from projections. A new backprojection algorithm for use in high-speed parallel-ray tomographic reconstruction systems is presented. The algorithm has the same functionality as the standard backprojection algorithm. However it has been arranged so that fast table look-up methods may be used, eliminating the need for time consuming mathematical calculations. The modified backprojection algorithm reduces the size of the required look-up table by an order of magnitude. Simulated results using the modified algorithm are provided and compared to those obtained using the unmodified algorithm. The resulting images are comparable with respect to feature identification, confirming that the two algorithms function similarly. High-speed implementations of the modified backprojection process used in tomographic reconstruction are presented.

By combining both the convolution and backprojection implementations presented, an inexpensive reconstruction system suitable for detection of defects in felled logs is achievable.

ACKNOWLEDGMENTS

I would like to acknowledge the following people for their support during the course of my research.

- Dr. R.I. Chaplin for undertaking and performing the key supervisory role for the research presented in this thesis. Bob provided an excellent environment for discussion of the numerous ideas, aided in maintaining clear direction throughout the duration of my research, and gave me support when I needed a lift. I enjoyed working with Bob and would like to thank him for being a good friend.
- Prof. W. Monteith for undertaking and performing a supervisory role for the research presented in this thesis. Bill helped to maintain focus on the overall project when I was unable to see past the problem at hand, and disciplined me into being thorough when I wanted to bolt for the answer. I also enjoyed working with Bill and would like to thank him for talking me into undertaking the study.
- Ian Noel and Wyatt Page for their help getting to grips with some of the more obscure aspects of mathematics.
- Heather North, Ian Noel, Harold Hendel, Ali Abedian, Mohammad Sandringham, Samantha Long and Xiao-ping Pan (fellow postgraduates) for making the experience memorable. I will remember my postgraduate days with fond memories thanks to all of you.
- Samantha Long for being my wife. Samantha supported me emotionally during my research, and also financially for the latter half of my research. Samantha was patient and understanding when things were tough, and shared in my joy when things were good. I would like to thank Samantha for being my best friend.
- Holt Forest Trust for their financial support. The Trust provided financial support for myself, and a computer for the project. I would like to thank the Trust for maintaining an interest in the project, and hope the results of this thesis are beneficial to the forestry industry in the future.

I would also like to thank God through whom all truth is found, and from whom all understanding is given.

TABLE OF CONTENTS

Chapter 1

1. INTRODUCTION

This thesis is concerned with the application of technology to the process of good-wood optimisation, in order to minimise wastage of New Zealand's natural tree resource.

1.1 LOG PROCESSING

New Zealand provides a climate which is suitable for growing a wide range of trees from the slow growing native kauri, to the rapid growing pinus radiata. The New Zealand Logging industry has chosen to invest the bulk of its resources into the planting, growing, pruning, felling, and processing of the faster growing trees, in particular pinus radiata [Larsen, A. 1992]. This makes New Zealand one of the few timber producing countries in the world where first generation trees are not felled.

1.1.1 Tree Pruning

The objectives of pruning are to promote growth of clearwood (wood without defects) in the bottom 6 meters of the tree, and to prevent development of bark-encased knots in the wood by pruning while the branches are still small and green.

When the branches are pruned sapwood grows up over the pruned stumps to cover the cut. The outer wood then grows out and over the top of this leaving a sap pocket. If the branch is not pruned carefully the moisture can cause the branch end to rot which encourages insect attack and other fungal diseases. The wood which grows out over the pruned branches is free from knots and is

referred to as *clearwood*. The core containing the pruned branches, air pockets, sap pockets, and other defects is referred to as the *defect core*.

The pinus radiata seeds are grown in nurseries until they become seedlings (very young trees about 30 cm in height). The seedlings are planted with a planting density of about 500 to 600 stems (trees) per hectare [Wylie, R. E. J. et al. 1984] (although this varies considerably depending on the geographical and climatic conditions, and the desired characteristics of the trees). To maintain a uniform diameter defect core it is suggested [Levack, H. 1986] that the trees be pruned every year. However because of the cost involved in annual pruning (especially in remote locations) this is commonly carried out in three stages during the first 4 to 10 years of growth. To reduce expense it is also common to thin the trees at the same time. Table 1-1 shows the typical age at which pruning should occur, and the pruned height and tree density that is typical at each stage of prunning.

Table 1-1. Typical details of pruning/thinning for Pinus radiata [Wylie, R. E. J. et al. 1984]

Stage	Pruned Height (meters)	No. of trees pruned/hectare	Height of tallest trees (meters)	Approx. age (yrs)
1	0 - 2	550	5	4 - 6
2	2 - 4	250	8	6 - 8
3	4 - 6	200	10	8 - 10

The branches of pinus radiata grow in groups at intervals up the tree. Each group of branches is referred to as a *whorl* and between each whorl the tree grows clearwood. The spacing between each whorl is very dependent on the growing conditions for each tree, and an average figure is relevant only for a particular site. However a rough figure based on observation would be between 1 and 2 metres, with most being closer to 1 metre.

1.1.2 Log Processing

Log processing can be divided roughly into three main categories based on the end use of the log, these being timber, wood chips (used for fibre boards), and pulp and paper [Larsen, A. 1992]. Within each category the wood is graded depending on a number of characteristics which relate to quality. The timber category yields the highest profit and will be the focus of the following sections.

The timber quality is derived largely from the overall strength of the timber. High wood density is desirable because it increases strength, stiffness and hardness while reducing the shrinkage which occurs during drying [McConchie, D. 1992].

Timber sawn from plantations where pruning is not carried out will contain knots. However one of the advantages of pinus radiata over other tree species is that short lengths of clearwood 1 - 2 meters long are left between the whorls instead of the knots being scattered along the height of the tree.

Knots and/or other defects in sawn timber reduce the strength of the timber, and significantly reduce the profit that may be obtained. To remove these defects a process known as finger jointing has been developed, where the portion of the timber containing the knot is cut out, and the remaining timber lengths are glued together using a finger joint as shown in figure 1-1. This process is currently seen to be very effective by the consumer, and is internationally accepted. However the joints can be visually displeasing and are mainly used for surface treated (painted) or hidden uses.

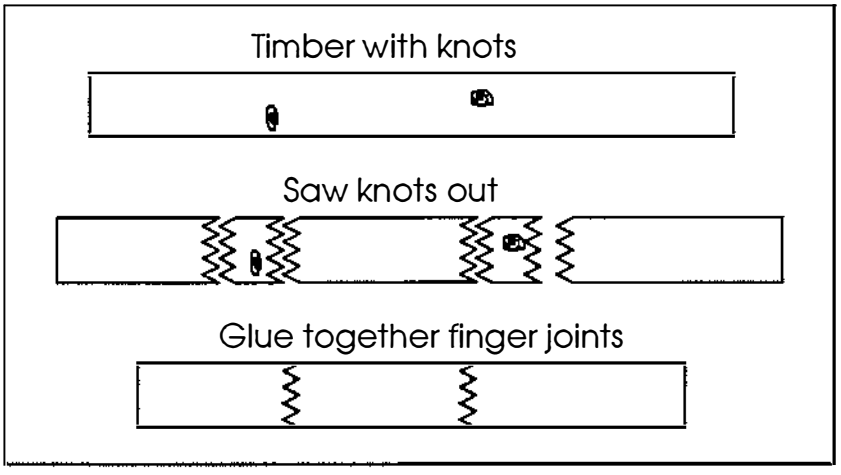


Figure 1-1. Finger Joint process.

1.1.3 Defect Detection

The defects which occur in pruned logs are mainly contained within an imaginary central core of the log slightly larger then the trunk of the tree at the time of pruning (about 16-18 centimetres in diameter [Wylie, R. E. J. et al. 1984]). This core is referred to as the *defect core* and contains the low

density heartwood as well as the knots, sap pockets, and other related defects. Knowledge of the position of this core within the log is of interest to both the log grower, and those responsible for the processing of the logs.

Log growers are currently having to accept prices which make allowance for sub-standard log quality because the seller is unable to guarantee the usefulness of the log for its end use. A fault report containing detailed information about the internal structure of each log would ensure a reasonable price for the quality of the log, and aid the buyer to select logs suitable for their end use. This both reduces wastage and increases overall profitability [Garder, K. and Mnyama, D. 1987].

When milling the logs into timber the mill operator attempts to position the saw blades close to the defect core to avoid wasting good clearwood, while at the same time avoiding inclusion of defects in the timber causing reduction in the wood quality. However mill operators currently allocate a large margin of error when sawing up each log to account for the uncertain position of the defect core.

Also the defect core size and shape may change along the length of the log due to log taper, and run out (ie. the core position changes along the length of the log, possibly due to prevailing winds during growth).

Knowledge of the internal structure of each log would ensure maximum good wood yield from each log and reduce wastage. Extending this concept it is possible to envisage a mill controlled, optimised and operated by computer.

Although it is difficult to accurately estimate the gain achievable if defect information were available, the figures from the 1991 New Zealand Forestry Statistics book [ministry of Forestry 1991] show that 11,871 million m^3 of logs are removed from New Zealand forests each year, and only 2,121 million m^3 of those logs become sawn timber. Only a small portion of this 9750 million m^3 difference is likely to be directly attributable to wastage in the defect core. However the figures give some indication that there is significant room for improvement.

1.2 THESIS OUTLINE

Chapter 2 investigates methods available for viewing the internal structure of objects with a view to selecting the most appropriate for obtaining density cross-section images from logs. Four of the methods (Transmission tomography, Emission tomography, Nuclear Magnetic Resonance, and Ultrasonic) are common in current medical practice [Webb, S. 1988] and the fifth (Electrical Impedance) has been developed more recently and is the focus of much research [1st Proceedings of the European Concerted Action on Process Tomography, 26-29 March 1992]. The chapter compares the physical principles used by each of the methods in the context of wood imaging, and concludes with the reasons for suggesting that the most appropriate method for imaging trees is Transmission Tomography (specifically X-ray tomography).

Chapter 3 gives an overview of the processes involved in X-ray tomography which are data acquisition, and image reconstruction. The first part of the chapter explains in more depth the physical principles which govern the behavior of X-rays, and how they are used to obtain measurements suitable for the reconstruction process. The main principles discussed are X-ray generation, X-ray absorption, and X-ray detection. A brief history of the development of X-ray scanners is given, with explanation of the latest X-ray scanner technology which allows generation of X-ray data in real-time. Finally the reconstruction process is introduced showing the link between the physical process, and the mathematical model of this process.

Chapter 4 details the reconstruction process, explaining the mathematical problem as first proposed by J. Radon in 1917 [Helgason, S. 1980]. The problems encountered with direct solution of the reconstruction problem are discussed, and the main approximation methodologies are introduced (Transform methods and Algebraic Reconstruction Techniques). Each of the methods available are appropriate for different situations, for example restricted data acquisition, missing data, poor contrast, etc. The current reconstruction methods are discussed to determine which method is best suited to the problem of real-time reconstruction of log defects. The main reconstruction methods for both Transform and Algebraic methodologies are

derived in this chapter, with special focus on the transform method known as *convolution backprojection*, as this forms the mathematical basis for the following chapters.

Chapter 5 examines a number of reconstruction systems claiming to be real-time, based on literature found from an international literature database search. Three high-speed data convolution systems (two analog and one digital), and three high-speed backprojection systems (one composite analog/digital and two digital) are discussed in total. Of the six systems discussed, only one convolution system and one backprojection system actually provide results adequate for any form of image reconstruction, and both of these systems require complex digital signal processing in order to perform the computations at high speeds. Also the ability to perform high-speed processing is traded off against the quality of the output in both cases.

Chapters 6 and 7 contain the author's original contribution, which describe a reconstruction system suitable for real-time implementation. The convolution backprojection reconstruction algorithm is modified to simplify the mathematical complexity, so that a hardware realisation can be implemented with simple low-cost components.

Chapter 6 starts by examining the use of Algebraic Reconstruction Techniques to achieve high-speed reconstruction by using less X-ray data as discussed by Garden and Mnyama [Garden, K. and Mnyama, D. 1987].

Following this a modified reconstruction algorithm is introduced and developed which significantly reduces the numerical complexity of backprojection. The underlying mechanism used for simplifying the mathematics is a modification of the discrete reconstruction area. This modification allows high-speed look-up methods to be used where previously these had been impractical due to their large data requirements. The performance of this algorithm is compared with the standard convolution backprojection algorithm.

A reduction in the mathematics of the convolution process is also discussed. This reduction involves truncating the convolution kernel to its most significant terms. Tests are performed using the truncated convolution kernel

with the standard convolution backprojection method, and the results are compared to those obtained using the full convolution kernel.

Chapter 7 re-structures the modified backprojection algorithm for implementation. Tests are then performed using a computer program with two forms of the modified algorithm, and these results are compared to results obtained using the standard convolution backprojection algorithm. The tests are then repeated using data contaminated with different levels of noise.

Schematic design for a hardware implementation is then suggested, and two low cost solutions are discussed. The first hardware implementation uses a Field Programmable Gate Array (FPGA) for all of the computation and logic, and two external static RAM chips for storage. A chip design for the FPGA is given, and the worst case timing analysis shown for that chip design. An implementation of the schematic design is then discussed using inexpensive discrete hardware components. The timing for this design is estimated based on the worst case and typical case timing given for each component.

The modified algorithm is then re-arranged into a systolic array of similar processes for implementation using parallel processing. The path of the data through the array and the function of each component in the array are discussed, and the achievable timing of the implementation is estimated.

Chapter 8 concludes by suggesting the most appropriate method for obtaining cross-sectional images of the density distribution of logs, based on the work in the previous chapters. Finally suggestions of future work which may help to advance this technology are offered.

Chapter 2

2 NON-DESTRUCTIVE IMAGING METHODS USED FOR VIEWING THE INTERNAL STRUCTURE OF OBJECTS

Several imaging techniques exist for observing a cross-section of the internal structure of an object. These methods rely on penetrating the object with some form of radiation, electrical current, or acoustic energy, and observing the radiation, current, or acoustic energy exiting from the object.

Although the objectives of the imaging methods discussed here are comparable in a broad sense, the data acquisition process, image reconstruction process, and functional use of each of these methods differ considerably.

This chapter describes the main facets in each process, and concludes with a comparison of their usefulness to the application of real-time imaging of log defects. Only those imaging methods which give cross-sectional information are discussed.

The five main methods used for imaging the internal structure of objects are Transmission Tomography, Emission Tomography, Nuclear Magnetic Resonance, Ultrasonic imaging, and Electrical Impedance Imaging.

2.1 TRANSMISSION TOMOGRAPHY (X-RAY)

Transmission tomography refers to all classes of tomography which transmit radiation through an object. However this section will assume that the radiation is X-ray. A more in-depth discussion of X-ray tomography is given in later chapters, so the following section is kept brief for the purposes of comparison.

Medical imaging has utilised X-ray tomography since the first commercial Computer Aided Tomography (CAT) scanner in 1973 [Roder, F.L. and Scheinman, E. and Magnuson, P. 1986]. This system was extremely slow and patients were exposed to relatively large radiation doses. However in spite of this X-ray tomography was a great success and much research has been directed towards this new technology because it was the first time the internal structure of a patient could be viewed in detail without invasive surgery.

Transmission tomography involves the transmission of multiple pencil-like beams through an object using one or several radiation sources, and an array of sensors (initial systems used only one sensor). The source and sensors are typically located on a transaxial plane at opposite sides of the object, and the source-sensor plane is rotated about the axis of the object as in figure 2-1. As the X-ray passes through the object it is attenuated. The amount of attenuation undergone by the X-ray is determined by several physical properties of the material in the X-ray path including density (see section 3.1). The combined effect of the various attenuating components is quantitatively described by an attenuation coefficient assigned to a particular material. The attenuation is linear for a uniform material, and by measuring the intensity difference of the X-ray before and after passing through the object the integral of the linear attenuation coefficient distribution through that path is obtained.

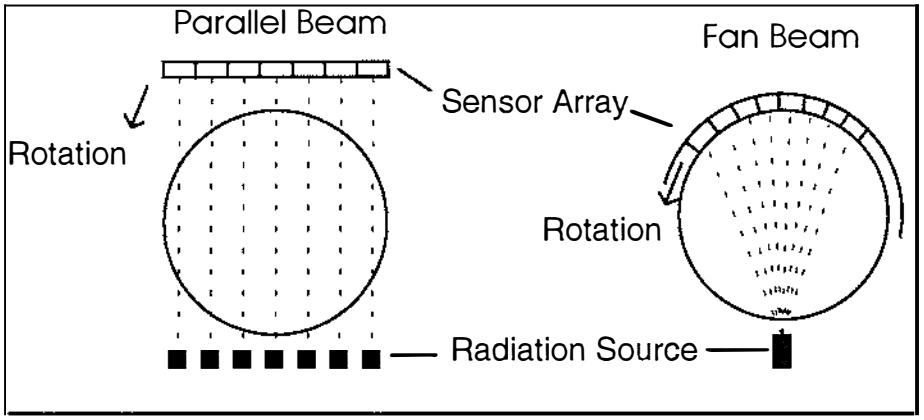


Figure 2-1. Source/sensor geometry used in transmission tomography.

The array of sensors provides a set of data specific to a particular source-sensor rotation. Each of these data sets can be thought of as a single one-dimensional function and are commonly known as the Radon transform of an object for a given angle [Helgason, S. 1980]. Combining these functions by means of a reconstruction algorithm provides a cross-sectional image of the 2-dimensional distribution of linear attenuation coefficients of the object through the plane of the transmitted beams. The resultant image is often referred to as a density image (where density is defined as electrons per cubic meter) even though the attribute being imaged is the linear attenuation coefficient distribution of the object. However in most cases the linear attenuation coefficient distribution cross section of an object is found to be closely related to the density distribution of the cross section by a series of linear relationships [Webb, S. 1988], for instance the higher density features (eg. bone) in general have higher attenuation coefficients .

X-ray radiation is very penetrating allowing the imaging of large or dense objects. Hence this form of imaging is often applied to non-destructive testing (NDT) of industrial components [Kak, A.C. 1988, pp132].

X-ray scanners have developed considerably over the last decade. The mechanical rotation has been replaced by electronic rotation (see section 3.1.4) allowing high-speed data acquisition which is useful for imaging of moving objects such as the heart [Roder, F.L. et al, 1986]. The work contained later in this thesis describes methods for high-speed image reconstruction which make full use of this high-speed data acquisition.

2.2 EMISSION TOMOGRAPHY

Emission tomography uses the decay of radioactive isotopes to image the distribution of the isotope within an object as a function of time, as compared to transmission tomography which uses the attenuation of a radiated source to infer information about the attenuation properties of the object being imaged. Consequently this method of imaging is often applied to imaging of mammals. An example of a typical application of emission tomography is tracing the path of an isotope from the lungs to the rest of the body after inhalation.

Different isotopes are used for imaging different characteristics. This is because the required decay time of the isotope needs to match the processing time of the particular body functions. These isotopes are usually introduced into the body either by injection or inhalation.

Radioactive isotopes are characterised by emission of gamma-ray photons or positrons. The concentration of an isotope in any part of the body changes with time due to radioactive decay, flow of body fluids, and biochemical kinetics within the body. This implies that the data acquisition for a given cross-section must be completed in a time interval that is short compared to the time constant associated with the changing concentration. On the other hand, this changing concentration provides a means of determining the functional state of different organs within a body by imaging the same cross-section at different times.

There are two main types of emission tomography. Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET) [Kak, A. C. 1988], [Webb, S. 1988]. These two imaging techniques differ considerably and consequently will be discussed individually.

2.2.1 Single Photon Emission Computed Tomography (SPECT)

As the name suggests SPECT is concerned with imaging objects using isotope sources emitting gamma-ray photons. Note that gamma-ray photons are indistinguishable from X-ray photons, and the different names are simply to indicate their origin.

When a radioactive isotope is introduced into a body, the gamma-ray photons caused by radioactive decay are emitted in all directions at once as

compared to transmission tomography where the photons travel in approximately straight lines from the source to the detector(s) (figure 2-2). For this reason, it is crucial to collimate the photons before detection. However the narrower the collimation the longer the time required to make a statistically meaningful observation. Consequently even with collimation the images produced by SPECT tend to be blurred due to the unfocused nature of detection.

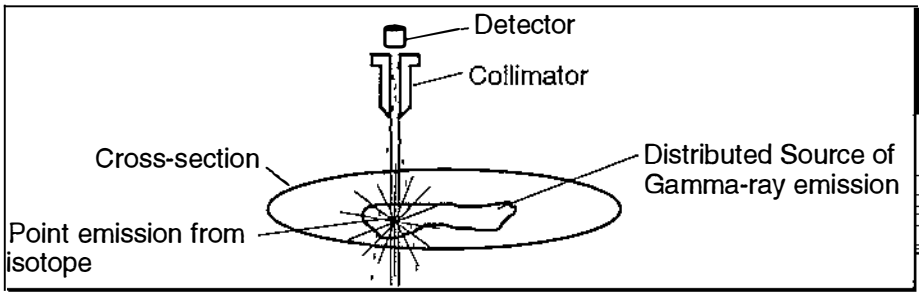


Figure 2-2. Gamma-ray emission from decaying isotope used in SPECT.

Another difficulty with emission tomography is caused by the attenuation that photons suffer during their travel from the emitting nuclei to the detector. The extent of this attenuation depends on both the photon energy and the nature of the tissue in the photon path. A set of photon intensity measurements has little meaning in the absence of some knowledge of the internal attenuation of the body section being imaged. Compensation for this lack of knowledge is dealt with using several methods aptly called attenuation compensation methods [Kak, A.C. 1988].

The simplest attenuation compensation method is to assume that the body cross-section is an ellipse of uniform attenuation. This approach may benefit by measuring the patient prior to the examination. However the assumption of uniform attenuation does not hold true and consequently leads to significant blurring.

At the other extreme, the most comprehensive attenuation compensation method involves performing a cross-sectional scan using transmission tomography prior to the examination. This method provides a good attenuation map and consequently the SPECT images are of relatively high quality. However the additional scan means that the process takes longer and the radioactive dose for the patient is significantly higher.

Data acquisition is usually performed using a linear array of detectors which are rotated transaxially about the body providing a set of projections for a given cross-section. The projections are then modified to compensate for attenuation effects, and an image of the isotopic distribution within the cross-section is reconstructed using the projections. Although the information contained in those produced using emission tomography is different to the information contained in the projections produced using transmission tomography, the reconstruction techniques used are the same for both [Webb, S. 1988].

2.2.2 Positron Emission Tomography (PET)

Radioactive isotopes which emit positrons are used in PET. However it is not the positrons which are detected since a positron can only exist in nature for a very short length of time before it interacts with an electron. When this happens the masses of the electron and positron are *annihilated*, creating two gamma-ray photons of equal energy (511 keV each) travelling in opposite directions. The distance which the positron travels before coming to rest is proportional to the initial energy of the positron, and typically ranges from 5mm (for a positron with approximate energy of 1-MeV in water) to 25mm (approximate energy 5-MeV in water). Clearly this positron movement from the isotope source can greatly effect the resulting image and must be considered when choosing the isotope.

The anti-parallel property exhibited by the two annihilated photons is used in PET to determine the path of the emitted photons. This is done by means of *coincident detection*, which involves detecting both photons generated by the positron using a detector pair at opposite sides of the body section being imaged (figure 2-3). Coincident detection assumes that where both detectors of a detector pair register a photon within a given time interval (typically in the order of 10 to 25 ns) then the positron annihilation occurred somewhere along the path between the two detectors. This detection timing function is provided electronically and is known as *electronic collimation.* Although the above assumption is not always true (the coincidence may result from two different annihilations), it provides a simple and accurate means for detecting the path of the photons without the need for collimating devices.

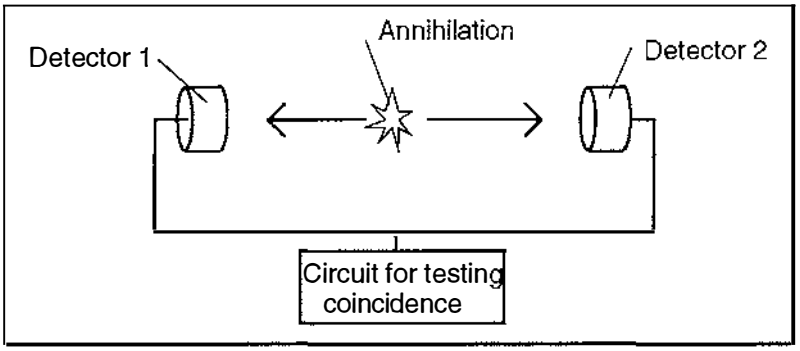


Figure 2-3. Detector pair arrangement used in PET.

Data acquisition systems for PET vary widely from two arrays of detectors to an entire ring of detectors around the body. Because the detectors have electronic collimation it is possible to use any pair of detectors for detection of a photon pair which greatly increases the usefulness of the detectors.

Attenuation compensation is not as much of a problem in PET systems because the integral attenuation along a given path can be determined from the attenuation undergone by the two photons (each with initial energy of 511 KeV and travelling in opposite directions). Therefore one can readily compensate for attenuation by first doing a transmission study to record total transmission loss for each ray in each projection.

Reconstruction techniques for PET systems are the same as those used for SPECT and transmission tomography systems [Webb, S. 1988].

2.3 NUCLEAR MAGNETIC RESONANCE IMAGING (NMR OR MRI)

Although only a brief discussion of NMR is given here, it is helpful to first consider a single isolated proton within a magnetic field. The proton has angular momentum m_p, and a charge of $+e$, where the charge on the proton can be considered as being distributed and rotating about a central axis as a result of the angular momentum m_p. This spinning charge gives rise to a magnetic dipole moment normal to the plane of charge circulation, similar to the magnetic flux density produced by an electric charge circulating in a wire coil.

When a proton is placed within an external magnetic flux density B_o, the couple exerted by the external magnetic flux on the magnetic dipole moment of the proton will cause the proton moment m_p to precess about B_o at a rate known as the *Larmor frequency* (figure 2-4). The frequency of precession is proportional to the applied magnetic field B_o. However the angle between m_p and B_o is not changed by this precession, and depends only on the orientation of m_p when B_o was established. It is useful to resolve the magnetic dipole moment m_p into two components, m_{pz} which is parallel to the external magnetic flux B_o (where B_o defines the direction of the z-axis), and m_{pxy} which is the plane perpendicular to B_o.

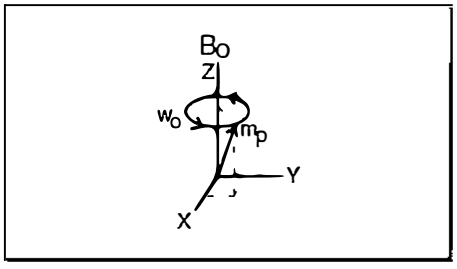


Figure 2-4. Precession of proton moment caused by external magnetic flux density B_o.

In practice the object under examination will consist of many nuclei, and the net magnetic moment M will be derived from the vector sum of all the individual nuclear magnetic moments. In the presence of an external magnetic flux B_o the net magnetic moment M at equilibrium will be aligned with B_o. M can once again be resolved into the components M_z and M_{xy}, and only the M_{xy} signal produces a measurable signal. Therefore in order

to measure M, it must first be *tilted* away from B_0 to produce a measurable component in the M_{xy} plane.

Let x' and y' be the axes of a plane rotating about the z-axis at the Larmor frequency. A magnetic field or flux density B_1 applied parallel to the x'-axis will exert a second couple on M which will rotate in the xy plane causing the net moment M to spiral from the z-axis to the rotating xy plane (figure 2-5). The angle which M will move through from the z-axis will depend upon the magnitude and duration of B_1. In NMR experiments B_1 will have a frequency in the radio frequency (RF) range and usually consist of a 90^o pulse of duration such that M tilts through 90^o into the xy plane, such that $M_z = 0$. The xy component of M is now a rotating RF field which can be detected using a suitable coil. The resulting signal (induced current) is known as a *free induction decay* (FID), and is composed of a sine wave with exponentially decaying amplitude. The decay of the signal is due to a loss of phase coherence after the second applied flux density B_1 is removed.

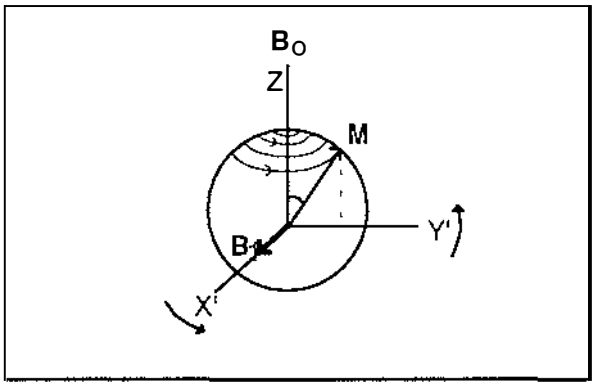


Figure 2-5. Spiral motion of net moment M approaching equilibrium in the presence of a rotating external magnetic flux.

There are several different procedures for generating FID signals using secondary pulsing to obtain significant M_{xy} components, and the reader is referred to [Webb, S. 1988] for detailed descriptions of these.

The classical description used above is only adequate as a model when considering the net magnetic moment of the nuclei. It is necessary to use quantum-mechanical descriptions when describing single isolated nuclei. The quantum-mechanical model defines the different allowable energy states of an isolated nuclei, which combine to create the net effect described by the classical model.

The underlying basis of many NMR systems is the relationship between the Larmor frequency of protons and the applied magnetic field. Consider a sample of two objects of equal size and uniform density separated by s along the z-axis in the presence of a uniform magnetic field B_0 (figure 2-6). After the sample is irradiated with a 90^0 magnetic pulse, the frequency of the FID signal (obtained by Fourier transformation) caused by both objects will be equivalent, and consequently it will be impossible to differentiate between the contributions to the resulting signal. If however, a small magnetic flux gradient G_z is added to the basic magnetic field B_0 along the z-axis, then the precession frequency will increase in proportion to displacement in the z direction. The Fourier transform of the FID signal will now contain two main components with separation proportional to s. Equally, if the object is irradiated with a narrow band of frequencies in the presence of the z gradient, only those nuclei resonating with frequencies within that frequency band will be excited. This process is known as *selective excitation* and is used to select a transaxial cross-section from a sample.

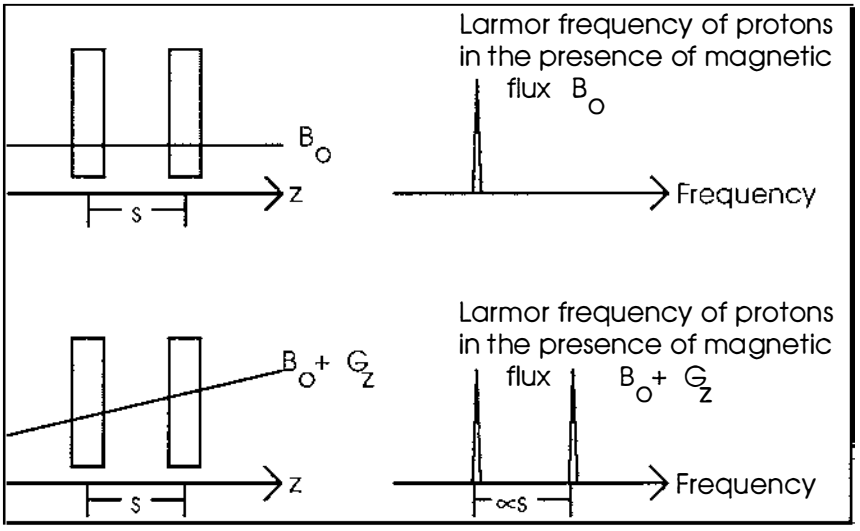


Figure 2-6. Using NMR to measure displacement using a magnetic gradient G_z.

After the plane of interest is selected and excited, the FID signal is read out in the presence of an x gradient. Each signal in the FID will now relate to a particular x displacement. The Fourier transform of the FID signal is equivalent to a projection obtained from transmission tomography. By carrying out the same procedure using a combination of x and y gradients that are changed for each measurement, a series of projections can be built up from different angles around the object. An image can then be

reconstructed from the projections using projection reconstruction methods as in transmission tomography.

The other main methods of data acquisition are similar to projection reconstruction imaging as described above, although they make use of phase encoding for additional information about the image. Two methods which use phase encoding are two-dimensional Fourier imaging and spin-warp imaging [Webb, S 1988].

NMR systems vary in size and resolution depending on the application. It is also a preferable imaging tool for many applications because it is low risk compared to other imaging techniques. NMR scanners are currently capable of producing medical quality images of entire human body cross sections.

2.4 ULTRASONIC IMAGING

The radiation energy used in ultrasonic imaging is acoustic, which renders this method very safe. The hardware required is also inexpensive (in comparison with the other imaging methods) making it easily accessible. Unlike X-ray radiation, ultrasonic radiation not only penetrates into an object but it also interacts with the object through refraction and reflection. In the same way that light is reflected and refracted at interfaces between media of different refractive index, so sound is reflected and refracted at interfaces between media of different *acoustic* refractive index.

It is the reflected acoustic signal which is detected and used in ultrasonic imaging. The attenuation of this signal is of course dependent on the depth of the reflecting interface, and the nature of the media in the path of the sonic wave. It is common for the wave to be focused by the source in order to strengthen the wave energy along the path of interest.

The most common reconstruction technique used for imaging cross-sectional segments of an object with ultrasound is called ***pulse-echo*** imaging. As the name suggests a *pulse* of ultrasound (typically 2-15 MHz [Webb, S. 1988]) is emitted for 2-4 cycles from a transmitting transducer, and the reflected waves or *echoes* are detected by a receiving transducer. The basic element of this type of ultrasonic imaging is known as the '*A-scan*'. The essential components in the signal processing chain of an ultrasonic pulse-echo imaging system are as follows:

Amplification

> This initial stage is more complex then the name suggests because as well as maintaining a significant signal to noise ratio in the presence of high-voltage pulse generation from the nearby transmitter, the amplifier must be linear with high gain, and possess a large dynamic range (as the echoed signal may have an overall dynamic range from the noise level to the largest echoes of 70 to 80 dB).

Time gain control

> The amplitude of the received echoes diminishes considerably as time from the initial pulse increases due to the attenuation undergone by the signal. Consequently some form of time-varying gain

function is required to compensate for the attenuation of the sound. The simplest function used is a logarithmic voltage ramp, usually set to compensate for some mean value of attenuation, although the resultant signal is generally a rough approximation to the ideal due to the changing attenuation throughout the object. This function is usually provided by a voltage-controlled attenuator, and the general result of this attenuation correction is a reduction in the dynamic range of the signal (often to around 40 to 50 dB).

Compression amplifier

The gain characteristics of this stage vary considerably dependent upon the application. However a general feature is that the applied gain decreases with increasing magnitude of input signal. An example is an amplifier with a logarithmic response. This allows a signal which has a dynamic range of 40 to 50 dB to be displayed on a cathode ray tube which may only have a dynamic range of 20 to 30 dB. The effect of this amplification stage is an increase in the significance of the less prominent interfaces.

Demodulation

At this stage the envelope of the signal is extracted using half or full-wave detection. This is usually followed by smoothing with a time constant about 1.5 times the initial ultrasound wavelength. The resulting envelope contains high magnitude peaks at positions proportional to media interfaces within the region being imaged, and is known as an A-scan.

There are three commonly used formats for displaying ultrasonic echo data, and one less common format. The simplest is the A-scan, which has been described above. This method uses a one-dimensional display and is used when precise measurements of axial length are required (figure 2-7a). The velocity of sound differs in media of different density [Kak, A.C. 1988] making the distance travelled by the sound not only a function of time, but also of the media travelled through. And because only the time is measured the precision is increased where the media is of relatively uniform density.

The M-mode (or TM-mode) format is the next simplest. This is used for observing changes in the A-scan as a function of time. The A-scan is used

to modulate the brightness of a line in the vertical direction of a two dimensional display, where the horizontal position of the line is incremented as a function of time (figure 2-7b). This format is usually used for imaging a constant line of interest through an object, where motion in the direction of the transducer is of concern.

The B-mode format is the most commonly adopted mode, and is used when scanning motion is applied to the transducers. Once again the A-scan is used to modulate brightness in a line onto a two-dimensional display. However the position of the line on the display is dependent upon the position and direction of the ultrasonic transducers at that time. A common simple scanning method used in ultrasonic imaging involves rotating the transducer through a particular sector of interest while maintaining a stationary axis position (this rotation is generally done using a motor driver). The resulting B-scan will therefore be shaped as a segment on the display (figure 2-7c).

The C-mode (or constant depth) format is the less common format for displaying ultrasonic data. As the name suggests, C-mode is used for displaying images of an object at a constant depth. This is done by selecting a portion of the A-scan, and using this signal to brightness modulate a point on a two-dimensional display (figure 2-7d). Generally either a one-dimensional array of transducers is moved in a direction orthogonal to that array, and the display is derived using the same geometry, or less commonly, a two-dimensional array of transducers is used, and each section of the display is derived from a single transducer. The C-mode format has the advantage that the transducers can be focused to optimise the response at a particular depth. However out-of-plane attenuating structures may significantly reduce the received signal.

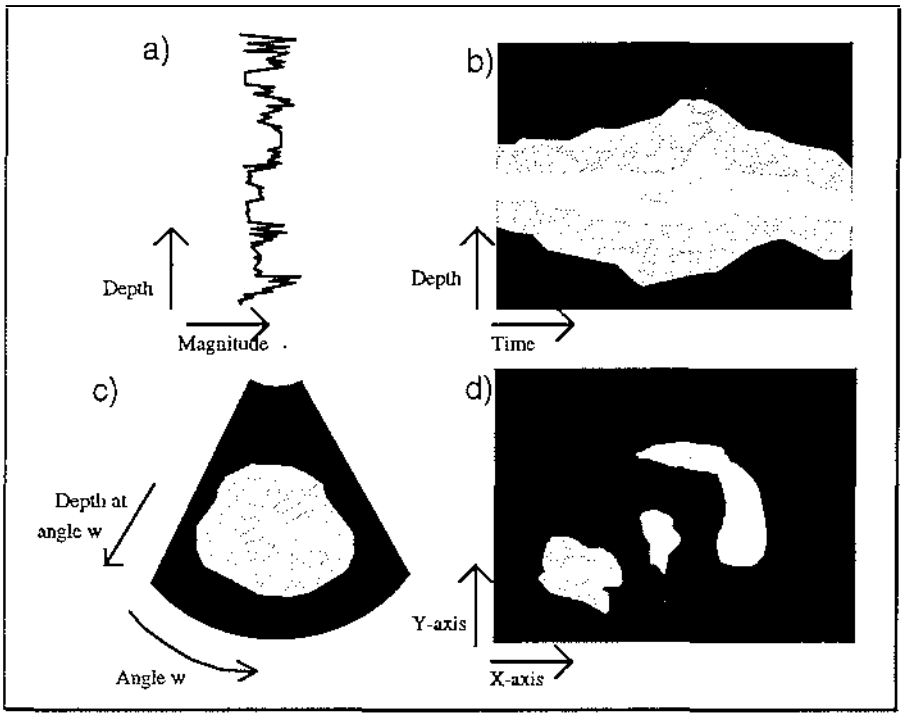


Figure 2-7 a)example of an A-scan, depicting a measurement through the abdomen. b) example of an M-mode scan, depicting the motion of a heart ventrical. c) example of a B-mode scan, depicting a cross-section of a fetus in a womb. d) example of a C-mode scan, depicting a planar scan of organs in the abdominal region.

2.5 ELECTRICAL IMPEDANCE TOMOGRAPHY

Electrical impedance tomography (EIT) also known as applied potential tomography (APT) attempts to reconstruct an impedance cross-section of an object using a number of electrodes placed around the surface of the object. The electrodes are typically spaced at equal distances around the perimeter of the object.

When current is passed between two electrodes a potential distribution is established within the object (the current does not travel in a straight line between the two electrodes). Since a finite number of electrodes is used, it is usual to approximate the potential distribution by a finite number of isopotential areas. To get the most information from the finite set of electrodes, current is passed between all possible pair combinations of electrodes. Figure 2-8 shows the isopotentials for a 16-electrode system assuming uniform impedance where current passes between an adjacent pair of electrodes.

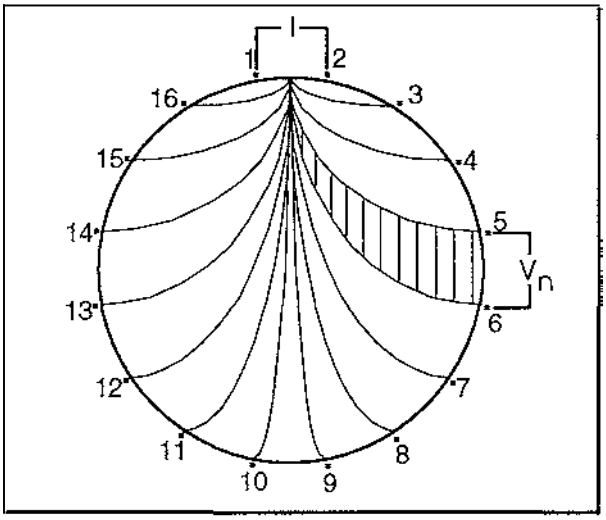


Figure 2-8. Isopotential regions of an EIT system assuming uniform inductance. The applied current I causes potential differences V_n between electrode pairs.

If the composition of the object were known, then the potential distribution within the object could be estimated accurately using finite element analysis, and the resultant potential differences between each of the electrodes could be calculated. However since the composition of the object is not known, it is necessary to make an initial estimate of the composition in order to estimate the potential values at points around the object corresponding to the electrodes. When no prior information is available it is

common to make the initial assumption that the object has uniform impedance throughout.

The reconstruction process involves calculation of estimated potential differences for all electrode pairs given the estimated object impedance and current passing between two electrodes. These calculated estimates are then compared with measurements of the actual potential differences from the electrode pairs. The ratio of the measured potential to the estimated potential is then used to modify the estimated impedance image by modifying the values contained in each isopotential area (defined by the estimated impedance image) based on the ratio. After all of the potential ratios have been calculated and used to modify the impedance image, the process is repeated using a different electrode pair for applying current. A more detailed description of the reconstruction process can be found in [Webb, S. 1988], and more recent developments in [Proceedings of the European Concerted Action on Process Tomography, 26th-29th March 1992].

The resulting cross-section is very dependent on the initial estimate of the isopotential regions, and consequently it is usual to repeat the process several times using the improved estimate as the initial estimate in the subsequent reconstruction. This complex iterative reconstruction technique makes application of EIT to real-time systems unlikely with current technology.

2.6 Comparison of Imaging Techniques for Log Defect Detection.

The stimulus for the discovery the above imaging methods has been the application to medical diagnosis. It is therefore appropriate to consider the differences between the problem of imaging the internal structure of a mammal, and a log. These differences are shown in table 2-1.

Table 2-1. Comparison of mammal and log with respect to imaging internal structure.

Mammal	**Log**
Cross-section relatively small (20-30 cm), low attenuation	Cross-section medium to large (30-60 cm), greater attenuation
Radiation dose limited	Higher radiation allowable
Electrical current limited	Higher electrical current allowable
Air and blood fluids flowing through body	No internal fluid movement (tree is dead so sap stops)
High water content, constant over time	Low water content, dries over time
Real-time reconstruction not critical (can collect data real-time and reconstruct off-line)	Real-time data collection and reconstruction critical.

Emission tomography is not practical for a number of obvious reasons. Firstly the injection of a radioactive isotope into the log at regular intervals is not practical because the log is hard and penetration to the central region would be difficult, especially while the log is moving. Secondly emission tomography is generally used for imaging the distribution of the radioactive isotope as a function of time, which is not appropriate for the log application. Finally the reconstruction methods are complex and time consuming because of the need for attenuation compensation.

Nuclear magnetic resonance imaging is in essence used for imaging the hydrogen molecules within an object usually found in water [Webb, S. 1988]. This causes problems when imaging logs because of their low water content (assuming the water content of the log reflects the different matter at different positions when the tree is felled). Also the outer section of the log may have dried over time causing severe changes in the resulting estimate image.

It may be possible to penetrate the log with ultrasound at certain frequencies. However coupling the ultrasonic transducers and receivers to the log is a large problem. Air will severely attenuate sound of high frequency [Kak, A.C. 1988] over a very short distance, so it is usual to submerge objects in a bath of water to provide acoustic coupling. It is not practical to submerge logs in a bath of water because air pockets trapped in the pith and bark (if this has not been stripped) of the tree, and bubbles caused by moving the logs in and out of the bath, will attenuate the sound more then the tree itself. These bubbles and air pockets will show up as black spots in the reconstructed image, masking any detail due to the changing refractive index of the log.

It is possible to penetrate the log using electrical current (consider lightning). However practical problems arise with this method also. Firstly the problem of connecting the electrodes to the log may be difficult because of the rugged handling of the equipment, varying log size, and varying log shape. It may be possible to overcome such physical problems. However the main problem with EIT is the time required to reconstruct the estimated image with the required resolution.

Transmission tomography can be used for objects causing large attenuation because the X-ray radiation is very penetrating. The X-ray source and detectors do not need to be directly coupled to the log so the system can be made fairly robust. The presence of air pockets or lack of water do not effect the X-ray radiation adversely. Real-time data acquisition systems and reconstruction systems for X-ray transmission tomography are currently available, making this imaging method the only practical option for application to the detection of defects within felled logs.

Chapter 3

3 Overview Of X-ray Computed Tomography

Computed Tomography involves two processes. The first is the data acquisition process where X-rays are passed through the object, and measurements of X-ray attenuation are made. The second is the image reconstruction process where the scalar X-ray measurements are transformed into a cross-sectional image using a reconstruction algorithm.

Section 3.1 summarises the behaviour of X-rays from generation, through absorption, to detection, and describes the evolution of the X-ray scanner. Section 3.2 briefly describes the reconstruction process, although details of image reconstruction are left until chapter 4.

3.1 Data Acquisition

The following account is not extensive or complete. It is provided to demonstrate the basic properties of X-rays, and some of the potential problems associated with their generation and detection.

In 1895, while investigating electrical discharges through gases and the effects of cathode rays, Roentgen came across a strange phenomena which led to his discovery of the X-ray. Roentgen noticed that a fluorescent screen outside the cathode ray tube would light up, even when the screen was shielded from the direct light of the gaseous discharge. It soon became apparent that some penetrating radiation was being emitted by the gaseous discharge tube, causing the screen to illuminate. Roentgen called these new radiations X-rays. He then proceeded to perform an extensive investigation into X-rays, and his findings are summarised below:

1. X-rays are generated whenever a beam of electrons strikes an obstacle, and elements of high atomic weight are the most efficient sources of X-rays.

2. Fluorescence is excited in many materials, such as calcium tungstate, barium-platino-cyanide, and others.

3. X-rays can be collimated with slits or pin-holes, showing that the radiations travel in straight lines, like light.

4. Photographic emulsions are affected by X-rays, thus providing an excellent method for the study of the radiations.

5. Electrified objects lose their charge when irradiated with X-rays.

6. Various materials are more or less transparent to X-rays, the differences in transparency providing the basis for the study of the internal structure of objects which are opaque to visible light.

7. Magnetic fields do not deflect an X-ray beam, showing that X-rays are not a stream of charged particles.

8. Reflection and refraction of X-rays do not occur (although it was later found that these effects could be observed to a small degree).

These findings (which summarise the important qualitative facts concerning X-rays) were published in 1896, and within three months X-rays were being used in medical diagnosis (as a photographic technique rather then a tomographic technique).

3.1.1 X-ray Generation

As the above findings indicated, to generate X-rays, a stream of rapidly moving electrons are required to strike a target. The original source of electrons was produced by a potential difference within a gas-filled tube. However when a gas-filled tube has been in use for some time, the pressure of the gas in the tube becomes lower, since some of the gas is absorbed by the walls of the tube, causing problems with tube current control. In order to maintain a given electron current through the tube, a higher potential difference must be used, which has the undesirable affect of producing 'harder' X-rays (ie. the electrons have higher energies, thus producing higher energy, more 'penetrating' X-rays). This problem was overcome by using a heated filament as a source of electrons in a tube with pressure as close as possible to a vacuum. With this arrangement the tube current could be

controlled by varying the heating power supplied to the filament, and any loss of gas by absorption was an advantage.

It is not surprising that a large amount of heat is generated at the target (anode) by the intense electron bombardment. Heavy refractory metals such as wolfram and molybdenum not only provide the most efficient generation of X-rays, but are also less susceptible to heat problems. Lighter metals such as copper can be used as targets, but cooling would be needed where X-ray generation was required for any substantial length of time. Figure 3-1 shows an X-ray tube used in modern commercial X-ray systems. Note that a rotating target is used to dissipate the heat.

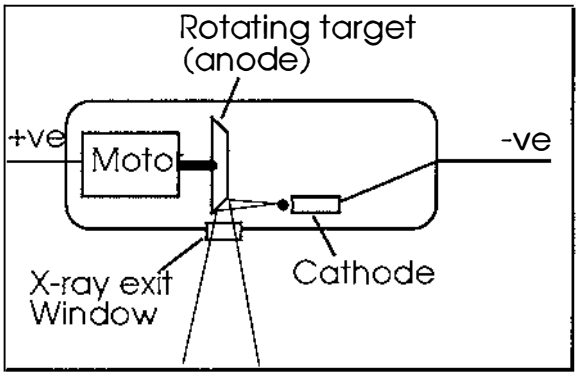


Figure 3-1. Construction of commercial X-ray tube

3.1.2 X-ray Absorption

X-rays are the result of a stream of 'photons' being emitted from an object. The X-ray is said to be 'monochromatic' if all the photons possess the same energy, and 'polychromatic' if the photons possess different energy levels. The monochromatic property is desirable in X-rays because photons of different energy levels are absorbed at different rates by the same material, although in practice a monochromatic X-ray source is difficult to achieve.

As an X-ray propagates through an object, the photon intensity of the X-ray is reduced. This reduction in photon intensity is a result of three different phenomena. The first of these is known as 'photoelectric absorption', and was observed by Hertz in 1887 [Brown, J.G. 1966]. This phenomena consists of an X-ray photon imparting all its energy to a tightly bound inner electron in an atom. The electron uses some of this acquired energy to overcome the binding energy within its atomic shell. The rest of the energy appears as kinetic energy of the now free electron. Photoelectric absorption is the major cause of photon absorption at lower energies (below 800 KeV).

The second mechanism creating a photon intensity drop was named 'Compton scatter' after its discovery in 1923 by Compton [Brown, J.G. 1966]. This phenomenon consists of the interaction of the X-ray photon with either a free electron, or one that is loosely bound in one of the outer shells of an atom. This interaction causes the X-ray photon to be deflected from its original direction of travel with some loss of energy, which is gained by the electron. Since X-rays are assumed to travel in straight paths, it is unlikely that this deflected photon will be detected by the desired X-ray detector. Compton scatter effect is the major cause of X-ray photon intensity loss for photons with energy levels of between 1200 KeV and 4300 KeV.

The third contributor to photon absorption was discovered in 1932 by C. D. Anderson, and is known as 'pair formation' [Brown, J.G. 1966]. This phenomenon only occurs at energy levels above 1022 KeV, and is therefore not usually included in discussions of X-ray absorption. However it is included here to complete the general picture of X-ray absorption. Pair formation consists of a high energy photon converting itself into two particles of equal mass. These particles being a negatively charged electron (just called an electron) and a positively charged electron (called a positron). The energy of the photon is shared equally between the electron and the positron, and appears as kinetic energy. Pair formation is the major cause of photon absorption for photons with energy levels above 6000 KeV.

Note that a typical medical X-ray scanner source produces photons with energy levels between 50 and 120 KeV. Consequently only photoelectric absorption and Compton scatter need be considered with regard to X-ray scanners used in tomography.

The amount a homogeneous material will attenuate an X-ray passing through it is dependent upon the additive combination of the above three phenomena, and is indicated by the absorption coefficient for that material.

3.1.3 X-ray Detection

It was observed by Roentgen that fluorescent screens coated with materials such as calcium tungstate are useful for observing qualitative features of X-ray distribution. However this method is limited for quantitative measurements.

There are two common methods of measuring X-ray intensity. The first is the gas ionisation chamber [Brown, J. G. 1966], where a central rod is maintained at a positive potential with respect to an outer cylinder which encompasses the rod. The chamber has a window at one end which allows the X-rays to pass into the chamber causing ion production in the gas which fills the chamber. Because of the potential difference between the rod and the cylinder, the positive ions are collected by the outer cylinder while the negative ions (or electrons) are collected by the central rod. If a large enough potential difference is applied across the rod and cylinder of the chamber then all the ions produced by the X-rays will be collected. The schematic construction of a gas ionisation chamber is shown in figure 3-2

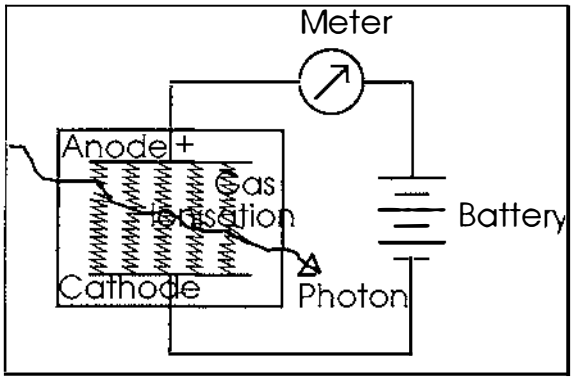


Figure 3-2. Construction of gas ionisation chamber.

The number of ions produced per second can be measured by amplifying the current produced by the ionisation chamber. This can be done rapidly and accurately, and this method for comparing X-ray intensities has been accepted as a standard method.

The second common method of X-ray detection is the scintillation detector [Birks, J.B. 1964]. The crystals which make up the scintillation detectors (commonly sodium iodide, bismuth germanate, or cesium iodide crystals) serve two purposes. Firstly the crystal traps most of the X-rays which strike the crystal. The X-ray photons then undergo photoelectric absorption (or Compton scatter followed by subsequent photoelectric absorption) resulting in the production of secondary electrons. The second purpose of the crystal is to transform the kinetic energy of the secondary electrons into flashes of light. These light flashes are usually detected by either a photomultiplier tube or a solid state photo-diode, providing a quantitative comparison between two X-ray intensities. Figure 3-3 shows the schematic construction of a scintillation detector.

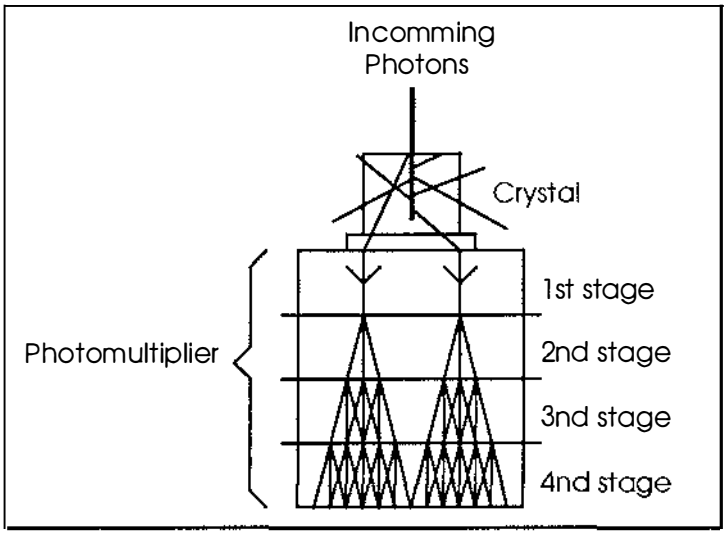


Figure 3-3. Schematic construction of scintillation detector

3.1.4 X-ray Scanners

The first commercial CT scanner was introduced by EMI in 1973 for use in the medical industry. The scanner utilised a single X-ray source on one side of the object being scanned, and a single scintillation detector on the other side. The object size, shape and composition determined how much a narrow X-ray beam was attenuated as it was passed from the source to the detector through the object. This action constituted a single ray value. The source-detector pair then moved linearly (sideways) to acquire a set of parallel ray values known as a 'projection' or 'view'. After a projection was acquired for one aspect angle, the gantry (to which the source and detector were affixed) was rotated and a further projection acquired. This process was repeated until projections had been acquired over 180 degrees (taking around 4 minutes to complete a scan). X-ray scanners which used this method of data acquisition were later known as 'first generation scanners'.

The cross sections produced by these scanners were of poor quality, and the combined scan and reconstruction times ranged from 8 minutes to over 20 minutes (largely dependent on the system used for reconstructing the data). However these systems gained immediate acceptance because for the first time it was possible to detect and identify internal features in scanned objects.

The next 'generation' of scanners to enter the marketplace during the late 70's [Roder, F.L. and Scheinman, E. and Magnuson, P. ,1986] reduced the scan time from 4 minutes to 20 seconds by increasing the fan angle of the X-

ray source and introducing a small array of detectors in place of the single detector of the first generation scanners. Although the scan process required the same basic motion as the previous generation (ie. linear and angular increments), the scanners were able to obtain several measurements at several different aspect angles from a single position, thus allowing mechanical increments to be larger.

The third generation scanners (known as 'rotate-rotate' scanners) developed in the early 80's employed a system where both the single source and the array of detectors are rotated around the object over 360 degrees (without the need for linear motion). These scanners required a large-angle X-ray fan beam (ranging 30 to 60 degrees), and a broad array of X-ray detectors (ranging 500 to over 700) to replace the linear motion of the source and detector(s) of the previous two generations. These detectors were arranged in either a straight line to produce equispaced data (measurements taken at equal spacing along the detector array line) or as an arc to produce equiangular data (measurements taken at equal angular displacements around the source). Scan times for these scanners usually ranged from 1 to 10 seconds, and during this time more then 1000 projections could be taken. Note that scanners which achieve the faster scan times generally take the projection measurements 'on the fly' (ie. the source-detector pair motion is continuous), which results in rotational smearing of the data. However the smearing is usually so small that its effects are not noticeable in the final image.

The fourth generation scanners (known as 'rotate-fixed' scanners) followed soon after the third generation scanners and replaced the rotating array of detectors with a fixed ring of detectors covering the entire 360 degrees. Inside this fixed ring of detectors the source continually rotates around the scan area, while the detectors within the 'view' of the X-ray source are sampled every few milliseconds. The collection of samples from a single detector are known as a 'detector-vertex' fan, which approximates a single view from a third generation scanner. Since the detectors are placed at fixed equi-angular intervals around the ring, the data collected by each detector are approximately equi-angular rather than exactly equi-angular, as the source radius is in most cases smaller than the detector radius. In general interpolation is used to convert the data to equiangular fan data before reconstruction. The resulting scan times for rotate-fixed scanners are in the order of 1 to 5 seconds.

The important difference between third and fourth generation scanners is that a third generation scanner has the number of measurements for a single projection fixed by the number of detectors, while there is an infinite number of possible projections (although the maximum 'useful information' would be limited by the spread of the X-ray, and the width of the detector aperture). However a fourth generation scanner has its maximum possible number of projections fixed by the number of detectors, while the number of rays per projection is determined by the detector sampling rate, and has no theoretical limit (limited by the width of the focal spot on the X-ray tube and by the size of the detector aperture).

During the mid 1980's a breakthrough in X-ray technology was developed. The main focus of this development was the way the X-ray source position was varied. Up until this time all X-ray scanners had mechanically moved the position of the X-ray source in order to measure projections from different aspect angles over the 360 degrees. These designs reached their limit at scan times in the order of one second due to the centrifugal force on the X-ray tube. The new technology overcomes this barrier by eliminating the mechanical motion of the X-ray tube. In its place a stream of electrons are produced and accelerated within a high-vacuum electron gun, focused and deflected by electromagnetic coils onto tungsten target rings which produce X-rays. This electron beam sweeps around the target rings producing an entire set of scan data within 50 milliseconds, and can be repeated up to 17 times a second. Scanners of this type (often referred to as 'fifth generation' scanners) eliminate the problem of overheating as the X-ray source points (target rings) are fixed, and hence cooling can be achieved by conventional methods. A diagram of a fifth generation scanner is shown in figure 3-4.

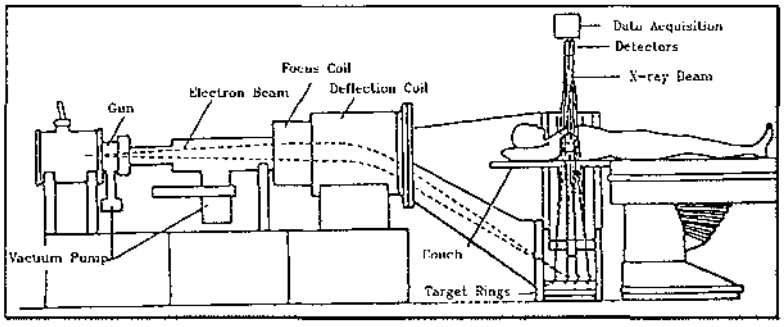


Figure 3-4. Fifth generation X-ray scanner eliminates physical movement to achieve high speed data acquisition.

3.2 IMAGE RECONSTRUCTION

Image reconstruction involves transforming the scalar line integral X-ray data into a two-dimensional image estimating distribution of the linear attenuation coefficient over the transaxial cross-section from which the data were acquired. Since X-rays travel in straight lines each X-ray measurement is an integral of the X-ray linear attenuation coefficient distribution through the object over the X-ray path.

The equation relating the intensity of an X-ray passing through an object to the linear attenuation coefficient distribution is given in equation 3-1. This relationship assumes that (i) the X-ray beams are a very narrow pencil shape, (ii) the X-ray source is monochromatic and (iii) no scattered radiation reaches the detector.

$$I_\theta(s) = I_\theta^0(s) \times e^{-\int_{AB} u(x,y)dt} \tag{3-1}$$

where u(x,y) is the two-dimensional distribution of the linear attenuation coefficient, θ and s define the position of the ray path AB and $I_\theta^0(s)$ is the unattenuated intensity. The (s,t) axis rotates with the X-ray source position such that the source is on the t-axis.

A projection at an angle θ is defined as

$$\begin{aligned} p_\theta(s) &= -\ln[I_\theta(s) / I_\theta^0(s)] \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} u(x,y)\delta(x.\cos\theta + y.\sin\theta - s)dxdy \end{aligned} \tag{3-2}$$

where the Dirac delta function δ picks out the path of the line integral, since the equation of AB is $s=x.\cos\theta + y.\sin\theta$.

Equation 3-2 expresses the linear relationship between the object function *u* and the measured projection data p. The problem of reconstruction is precisely that of inverting equation 3-2 to recover u from the set of projections p.

Methods used for obtaining a solution to equation 3-2 are discussed in the next chapter.

Chapter 4

4 DERIVATION OF RECONSTRUCTION ALGORITHMS

In 1917 J. Radon proved that a differentiable function in three dimensional space could be determined explicitly by means of its integrals over the planes in that space [Helgason, S. 1980]. However it was not until the invention of the X-ray scanner in 1972 by Hounsfield that the importance of this was fully realised. Radon's equations proved that an estimate of an object's X-ray attenuation coefficient function could be reconstructed from a number of X-ray line integrals through the object.

Section 4.2 describes algorithms known as Transform methods. These methods model the reconstruction process in the continuous domain and this model is applied to the discrete projection data producing a discrete image required for storage and display.

Section 4.3 describes iterative reconstruction methods known as series expansion methods. These algorithms begin by defining a discrete reconstruction area where each value represents an area of uniform density (ie. uniform linear attenuation coefficient) similar to pixels of a computer display. The projection data are mapped onto the reconstruction area using interpolation.

4.1 USEFUL OPERATORS

In order to describe the derivation of the transform methods it is useful to first define a few operators which are used to perform the transformations.

4.1.1 Radon Transform Operator

The Radon transform operator, or the projection operator (as it is often called) transforms a function from the spatial (x,y) domain to the Radon domain (t,θ) where s is displacement from the origin and θ is angle from the x-axis as in figure 4-1.

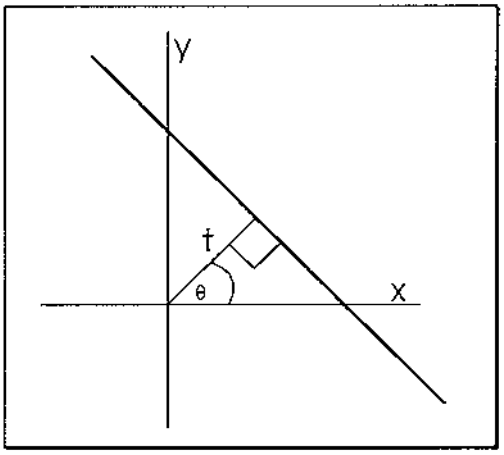


Figure 4-1. Coordinates of the Radon domain (t,θ).

A single point in the (t,θ) domain represents a line in the (x,y) domain. However a single point in the (x,y) domain is represented by a sinusoid in the (t,θ) domain, which indicates that a point in the (x,y) domain is formed by the intersection of infinity lines from different angles through a common point.

If the function p(t,θ) denotes the Radon transform of the function f(x,y) then the mathematical definition of the Radon transform is defined as

$$p(t,\theta) \equiv \Re f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x.\cos\theta + y.\sin\theta - t)dxdy \qquad (4\text{-}1)$$

$$-\infty<t<\infty\ ,\ 0\leq\theta<\pi \qquad (4\text{-}2)$$

The symbol $\Re$ is used to denote the Radon transform or *projection* operator. The reason for the name projection operator is because the function p(t,θ) represents the one-dimensional parallel projection of the function f(x,y) onto

a line for a given angle θ. Each individual value of p(t,θ) is known as a line integral since its value is obtained by integrating the values of f(x,y) along the ray path.

4.1.2 Backprojection Operator

The backprojection operator is closely associated with the Radon transform operator, and is defined as

$$b(x,y) \equiv Bp(t,\theta) = \int_0^{\pi} \int_{-\infty}^{\infty} p(t,\theta)\delta(x.\cos\theta + y.\sin\theta - t)dt\,d\theta \qquad (4\text{-}3)$$

where the function b(x,y) is the backprojection of p(t,θ).

This equation shows that each point in the backprojected image b(x,y) is generated by the addition of each of the projection values p(t,θ) corresponding to the X-rays which passed through that point (x,y).

It is possible to disassemble the above equation into a number of equations where each equation describes the backprojection of a single projection from angle θ_n onto an image $b'_n(x,y)$.

$$b'_n(x,y) = \int_{-\infty}^{\infty} p(t,\theta_n)\delta(x.\cos\theta_n + y.\sin\theta_n - t)dt \qquad (4\text{-}4)$$

The function b(x,y) can be calculated by superposition of all of the *sub-images* $b_n'(x,y)$.

$$b(x,y) = \sum_n b_n'(x,y)dn \qquad (4\text{-}5)$$

This is an important result showing that backprojecting the image b(x,y) can begin as soon as the first projection is available for processing. It also shows how the image can be reconstructed iteratively, where the estimate is the superposition of the available sub-images $b_n'(x,y)$, and the iterative estimate approaches the actual backprojected image b(x,y) as the projections are processed.

4.1.3 Fourier Slice Theorem

The Fourier Slice Theorem states that the one-dimensional Fourier transform of a parallel projection is equal to a central slice of the objects two-dimensional Fourier transform, where the angle of the projection determines the angle of the slice about the centre.

This can be proved by considering the two-dimensional Fourier transform of the object, and restricting the transform to the u-axis in the frequency domain. This results in an equation equivalent to the one-dimensional Fourier transform of the projection taken perpendicular to the u-axis. This proof follows.

The two-dimensional Fourier transform of the object f(x,y) is defined as

$$F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{4-6}$$

Likewise the one-dimensional Fourier transform of the projection $p(t,\theta)$

$$S(w,\theta) = \int_{-\infty}^{\infty} p(t,\theta)e^{-j2\pi wt}dt \tag{4-7}$$

Consider the Fourier transform of the object along the u-axis in the frequency domain (ie. let v=0).

$$F(u,0) = \int_{-\infty}^{\infty}\int f(x,y)e^{-j2\pi ux}dxdy \tag{4-8}$$

The phase factor no longer depends upon y, so the integral may be split into two parts

$$F(u,0) = \int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f(x,y)dy\right]e^{-j2\pi ux}dx \tag{4-9}$$

The term in the brackets is clearly recognisable as the projection equation for a projection taken at angle $\theta=0$ (x = constant).

$$p(t,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x.\cos 0 + y.\sin 0 - t)dxdy = \int_{-\infty}^{\infty} f(x,y)dy \qquad (4\text{-}10)$$

Substituting this into the above equation and changing the variable of integration gives

$$F(u,0) = \int_{-\infty}^{\infty} p(t,0)e^{-j2\pi ut}dt \qquad (4\text{-}11)$$

The right hand side of this equation represents the one-dimensional transform of the projection at θ=0. Clearly this result is independent of the orientation between the object and the coordinate system, and proves the theorem stated above.

The importance of the Fourier transform and Fourier Slice theorem to the problem of image reconstruction is demonstrated in the following sections where they are used as the basis for transform reconstruction methods.

4.2 TRANSFORM METHODS

Although the projection data are discrete, transform methods model the reconstruction problem as continuous. These methods are in general faster then iterative methods, and produce better images of low contrast objects [Webb, S. 1987].

Derivation of the transform methods is specific to either parallel or divergent-ray projection data due to the different representations in the Fourier domain.

4.2.1 Direct Fourier Method (DFM)

It follows from the Fourier slice theorem that an estimate of the object density can be determined from its projections. If the one-dimensional Fourier transform of a projection taken from angle θ can produce a central slice of the object's two-dimensional Fourier transform at that same angle, then the set of projections for all angles $0<\theta\leq\pi$ grouped onto a single plane will produce an estimate of the object's two-dimensional Fourier transform. Performing a two-dimensional inverse Fourier transform produces an estimate image of the cross-section.

The frequency domain samples fall in radial lines about the centre (see figure 4-2). If digital Fourier inversion is to be applied, then samples are required at regular intervals over a square grid as shown in figure 4-2. Samples on the square grid are obtained by interpolating from the radial points in the Fourier domain. The density of the radial points becomes sparser as one gets farther away from the centre and the interpolation error becomes larger. This implies that the reconstruction will suffer greater degradation in its calculation of the high frequency components than of the low frequency components.

Interpolation from the radial to the square grid points is a critical component in the Fourier inversion method. Much work has been carried out in developing efficient interpolation methods for high-speed reconstruction. The reconstruction quality is more sensitive to interpolation accuracy in the radial direction then in the angular [Matej, S. and Bajla, I. 1990]. Hence hybrid spline-linear interpolation has been suggested for improved results [Matej, S. and Bajla, I. 1990] where spline interpolation is used in the radial direction and linear interpolation in the angular. Another method of

optimising the frequency domain reconstruction methods is reported in [Tabei, M. Ueda, M. 1990] where the Fourier transformed projections are projected onto a rectangular grid in the frequency domain using alias-free interpolation of FFT bins.

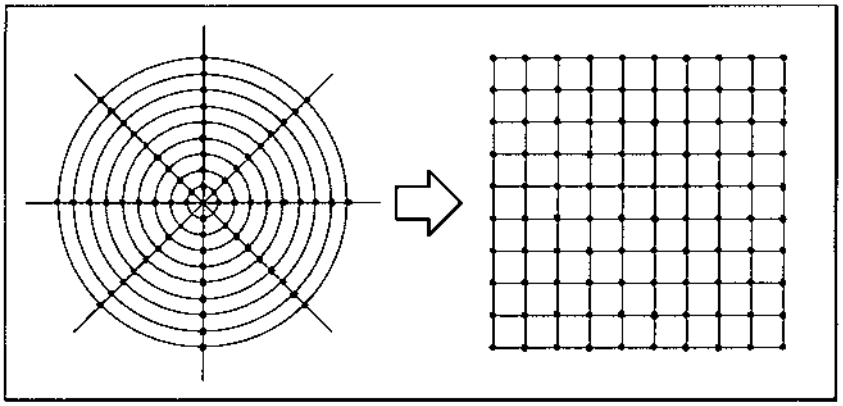


Figure 2 The sampled projections form radial points in the frequency domain. Square grid points must be interpolated from the radial points to use IFFT methods.

4.2.2 Convolution Backprojection

This reconstruction method is the most commonly used in almost all areas of transmission tomography, particularly in medical applications where the required images are often of low contrast.

The backprojection operator is a simplistic attempt to redistribute the projection data created by the Radon operator. As there is no prior knowledge of the object, the backprojection operation evenly distributes each ray value over the ray path. This approach leads to a reconstruction which is blurred with a point spread function of $1/|r|$ where r is the distance from the point. Consequently it is necessary to preprocess the data before backprojecting to minimise blurring. (Note that as the data is available in sampled form only, and is inherently band limited by the sample spacing, the blurring can not be completely eliminated.).

To begin the derivation of the preprocessing operation, consider the formula for the inverse Fourier transform of an image.

$$f(x,y) = \int_{-\infty}^{\infty}\int F(u,v)e^{j2\pi(ux+vy)}dudv \qquad (4\text{-}12)$$

Manipulation of equation 4-12 is easier when polar coordinates are used in the frequency domain (w,θ) instead of cartesian coordinates (u,v). Substituting

$$\begin{aligned} u &= w.\cos\theta \\ v &= w.\sin\theta \end{aligned} \tag{4-13}$$

and changing the differentials

$$dudv = w.\,dwd\theta \tag{4-14}$$

gives

$$f(x,y) = \int_0^{2\pi}\int_0^{\infty} F(w,\theta)e^{j2\pi w(x.\cos\theta+y.\sin\theta)}w.\,dwd\theta \tag{4-15}$$

Considering the integration in two parts ($0\le\theta<\pi$, $\pi\le\theta<2\pi$) the formula can be rewritten as

$$\begin{aligned} f(x,y) = &\int_0^{\pi}\int_0^{\infty} F(w,\theta)e^{j2\pi w(x.\cos\theta+y.\sin\theta)}w.\,dwd\theta \\ &+\int_0^{\pi}\int_0^{\infty} F(w,\theta+\pi)e^{j2\pi w[x.\cos(\theta+\pi)+y.\sin(\theta+\pi)]}w.\,dwd\theta \end{aligned} \tag{4-16}$$

Using the property $F(w,\theta+\pi) = F(-w,\theta)$ the above expression may be written as

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} F(w,\theta)|w|e^{j2\pi w(x.\cos\theta+y.\sin\theta)}dw\right]d\theta \tag{4-17}$$

Using the Fourier slice theorem $F(w,\theta)$ in equation 4-17 can be replaced by the Fourier transform of the projection p at angle θ, $S_\theta(w)$, since θ is constant within the square brackets. This gives

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} S_\theta(w)|w|e^{j2\pi w(x.\cos\theta+y.\sin\theta)}dw\right]d\theta \tag{4-18}$$

This equation may be split into

$$f(x,y) = \int_0^{\pi} Q_\theta (x.\cos\theta + y.\sin\theta) d\theta \tag{4-19}$$

$$Q_\theta (t) = \int_{-\infty}^{\infty} S_\theta (w)|w| e^{j2\pi wt} dw \tag{4-20}$$

Q_θ represents a filtered projection, where the filter function is given by |w| in the frequency domain. Each filtered projection is then backprojected over the image space f(x,y) by equation 4-19. Note, the argument of the filtered projection Q replaces the Dirac delta function used in the backprojection equation 4-3. Although both forms of the equation are functionally equivalent the Dirac delta function has been introduced in earlier sections to simplify derivation of the modified backprojection in chapter 7.

As the name indicates (convolution backprojection) the filtering operation is performed in the spatial domain by means of convolution instead of multiplication in the frequency domain. The latter is known as *Filtered Backprojection.* By convolving the functions in the spatial domain, the need for Fourier inversion is eliminated. However, since |w| is not a square integrable function, its inverse transform does not exist. A common method used to overcome this is to truncate the bandwidth of the filtering kernel to that of the sampled projections before it is transformed into the spatial domain.

If the maximum frequency component of $S_\theta(w)$ is w_{max} then the band limited filtering kernel in the frequency domain is

$$\begin{aligned} H(w) &= |w| \qquad |w| < w_{max} \\ H(w) &= 0 \qquad |w| \geq w_{max} \end{aligned} \tag{4-21}$$

Transforming into the spatial domain gives

$$h(t) = \int_0^{w_{max}} we^{j2\pi wt} dw - \int_{-w_{max}}^{0} we^{j2\pi wt} dw \tag{4-22}$$

This can be evaluated as follows

$$
\begin{aligned}
h(t) &= \left[\frac{we^{j2\pi wt}}{j2\pi t}\right]_0^{w_{max}} - \int_0^{w_{max}} \frac{e^{j2\pi wt}dw}{j2\pi t} - \left[\frac{we^{j2\pi wt}}{j2\pi t}\right]_{-w_{max}}^{0} + \int_{-w_{max}}^{0} \frac{e^{j2\pi wt}dw}{j2\pi t} \\
&= \frac{w_{max}\sin(2\pi w_{max}t)}{\pi t} - \left[\frac{e^{j2\pi wt}}{(j2\pi t)^2}\right]_0^{w_{max}} + \left[\frac{e^{j2\pi wt}}{(j2\pi t)^2}\right]_{-w_{max}}^{0} \\
&= 2w_{max}^2 \mathrm{sinc}(2w_{max}t) + \frac{\cos(2\pi w_{max}t) - 1}{2\pi^2 t^2} \\
&= 2w_{max}^2 \mathrm{sinc}(2w_{max}t) - w_{max}^2 \mathrm{sinc}^2(w_{max}t)
\end{aligned}
\tag{4-23}
$$

ie.

$$
h(t) = w_{max}^2\left[2\mathrm{sinc}(2w_{max}t) - \mathrm{sinc}^2(w_{max}t)\right] \tag{4-24}
$$

This is the continuous form of the filter function |w| in the spatial domain. In practical systems this function will be evaluated at discrete points. The filtering operation may now be expressed as a convolution

$$
Q_\theta(t) = \int_{-\infty}^{\infty} P_\theta(t').h(t-t')dt' \tag{4-25}
$$

where $P_\theta(t)$ is the parallel projection taken at angle θ.

Using the shorthand symbol $\otimes$ to denote the convolution operation equation 4-20 may be written as

$$
Q_\theta(t) = P_\theta(t) \otimes h(t) \tag{4-26}
$$

4.2.3 Convolution Backprojection for Divergent-Ray Data

The Convolution backprojection method for divergent-ray (fan-beam) data does not have a simple interpretation in terms of Fourier transforms [Lewitt, R.M. 1983] as does the parallel data case, because each projection non-uniformly samples the reconstruction area with a range of cut-off

frequencies. As a consequence the resolution of the estimated image may not be uniform.

It is usual to derive the fan-beam equations using a coordinate transform on the parallel equations. The parameters used to describe a ray from a fan-beam projection are (γ,β) where β is the angle of the projections main axis, and γ is the angle between this axis and the ray (figure 4-3). The fan-beam coordinates (γ,β) are related to the parallel coordinates (t,θ) by

$$\theta = \beta + \gamma, \qquad t = D.\sin\gamma \tag{4-27}$$

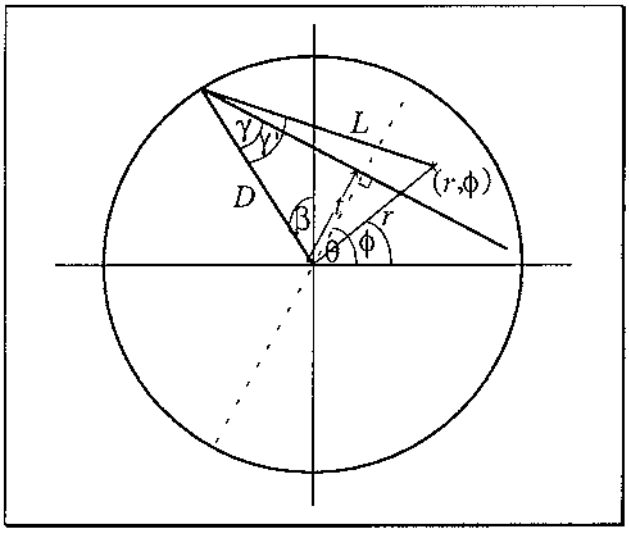


Figure 4-3. Coordinates used to describe divergent-ray (fan-beam) projection data.

Combining equations 4-19 and 4-20 gives

$$f(x,y) = \int_0^{\pi}\int_{-\infty}^{\infty} S_\theta(w)|w|e^{j2\pi w(x.\cos\theta + y.\sin\theta)}dwd\theta \tag{4-28}$$

where $S_\theta(w)$ is the Fourier transform of projection $P_\theta(t)$

$$P_\theta(t) = \int_{-\infty}^{\infty} S_\theta(w)e^{j2\pi wt}dw \tag{4-29}$$

From equation 4-22 the filter function h(t) can be written

$$h(t) = \int_{-\infty}^{\infty} |w|e^{j2\pi wt}dw \tag{4-30}$$

Instead of multiplying P and h in the frequency domain as S and |w|, convolution of P and h in the spatial domain is performed. (Note that the two operations are equivalent).

$$f(x,y) = \int_{0}^{\pi} \int_{-\infty}^{\infty} P_{\theta}(t).h(t'-t)dtd\theta \tag{4-31}$$

where $t' = x.\cos\theta + y.\sin\theta$

The projections from a fan beam source do not repeat after π degrees so the integration of θ is extended to cover 2π. Converting the coordinate system to polar coordinates using the relationships

$$\begin{aligned} x &= r.\cos(\phi) \\ y &= r.\sin(\phi) \end{aligned} \tag{4-32}$$

gives

$$f(r,\phi) = \frac{1}{2}\int_{0}^{2\pi} \int_{-\infty}^{\infty} P_{\theta}(t).h(r.\cos(\theta-\phi)-t)dtd\theta \tag{4-33}$$

Substituting the fan-beam coordinates from equations 4-27 gives

$$f(r,\phi) = \frac{1}{2}\int_{0}^{2\pi} \int_{-\Gamma}^{\Gamma} P_{\beta+\gamma}(D.\sin\gamma).h(r.\cos(\beta+\gamma-\phi)-D.\sin\gamma)D.\cos\gamma d\gamma d\beta \tag{4-34}$$

where the integral limit Γ is half the fan angle and is equal to $sin^{-1}(t/D)$. The term $D.\cos\gamma$ is introduced by the change of coordinates, where the determinant of the Jacobian J is calculated by

$$\frac{\partial\theta}{\partial\beta}=1 \quad \frac{\partial\theta}{\partial\gamma}=1 \quad \frac{\partial t}{\partial\beta}=0 \quad \frac{\partial t}{\partial\gamma}=D.\cos\gamma \tag{4-35}$$

$$J=\begin{bmatrix} 1 & 1 \\ 0 & D.\cos\gamma \end{bmatrix} \quad , \quad |J| = D.\cos\gamma \tag{4-36}$$

The parameters of the projection function P are not appropriate for parallel ray data, so the function $R_\beta(\gamma)$ will be used to indicate fan beam projection data

$$f(r,\phi)=\frac{1}{2}\int_0^{2\pi}\int_{-\Gamma}^{\Gamma} R_\beta(\gamma).h(r.\cos(\beta+\gamma-\phi)-D.\sin\gamma)D.\cos\gamma\,d\gamma\,d\beta \qquad (4\text{-}37)$$

It is desirable for computational purposes to include only projection parameters in the convolution. Using the trigonometric identity [cos(A+B)=cos(A).cos(B)-sin(A).sin(B)] the argument of h can be expressed

$$\begin{aligned} r.\cos(\beta+\gamma-\phi)-D.\sin\gamma &= r.\cos(\beta-\phi)\cos\gamma-r.\sin(\beta-\phi)\sin\gamma-D.\sin\gamma \\ &= r.\cos(\beta-\phi)\cos\gamma-[r.\sin(\beta-\phi)+D].\sin\gamma \end{aligned} \qquad (4\text{-}38)$$

From figure 4-3

$$\begin{aligned} L.\cos\gamma' &= D+r.\sin(\beta-\phi) \\ L.\sin\gamma' &= r.\cos(\beta-\phi) \end{aligned} \qquad (4\text{-}39)$$

where L and γ' relate to a point (r,ϕ) in the reconstruction area by

$$L(r,\phi,\beta)=\sqrt{(D+r.\sin(\beta-\phi))^2+(r.\cos(\beta-\phi))^2} \qquad (4\text{-}40)$$

$$\gamma'=\tan^{-1}\left(\frac{r.\cos(\beta-\phi)}{D+r.\sin(\beta-\phi)}\right) \qquad (4\text{-}41)$$

Using equations 4-38 to 4-41 the argument of the function h (equation 4-37) may be expressed as

$$L.\sin\gamma'.\cos\gamma-L.\cos\gamma'.\sin\gamma \qquad (4\text{-}42)$$

Using the trigonometric identity [sin(A-B)=sin(A)cos(B)-cos(A)sin(B)] this simplifies to

$$L.\sin(\gamma'-\gamma) \qquad (4\text{-}43)$$

Equation 4-37 becomes

$$f(r,\phi) = \frac{1}{2}\int_{0}^{2\pi}\int_{-\Gamma}^{\Gamma} R_{\beta}(\gamma).h\left(L.\sin(\gamma'-\gamma)\right)D.\cos\gamma d\gamma d\beta \tag{4-44}$$

From equation 4-30

$$h\left(L.\sin(\gamma'-\gamma)\right) = \int_{-\infty}^{\infty} |w| e^{i2\pi w L\sin(\gamma'-\gamma)} dw \tag{4-45}$$

To show the change in the filter function h(t) caused by the introduction of fan beam geometry the function h must be expressed in terms of a single argument $h(\gamma)$.

Defining

$$w' = w.\frac{L.\sin\gamma}{\gamma} \tag{4-46}$$

gives

$$\frac{dw}{dw'} = \frac{L.\sin\gamma}{\gamma} \tag{4-47}$$

$$dw' = \frac{\gamma}{L.\sin\gamma}.\frac{1}{dw} \tag{4-48}$$

and

$$|w| = |w'|\left(\frac{\gamma}{L.\sin\gamma}\right) \tag{4-49}$$

Note that because sin is an odd function

$$\left|\frac{\gamma}{L.\sin\gamma}\right| = \left(\frac{\gamma}{L.\sin\gamma}\right) \tag{4-50}$$

Equation 4-48 has been separated to show that dw in equation 4-45 is cancelled and a constant is introduced. Equation 4-49 shows that the same

constant is introduced by the |w| to |w'| exchange. Therefore substituting w' for w in equation 4-45 gives

$$h(L.\sin\gamma)=\left(\frac{\gamma}{L.\sin\gamma}\right)^2\int_{-\infty}^{\infty}|w'|e^{i2\pi w'\gamma}dw' \tag{4-51}$$

Recognising from equation 4-30 that

$$\int_{-\infty}^{\infty}|w'|e^{i2\pi w'\gamma}dw'=h(\gamma) \tag{4-52}$$

equation 4-51 may be expressed as

$$h(L.\sin\gamma)=\left(\frac{\gamma}{L.\sin\gamma}\right)^2 h(\gamma) \tag{4-53}$$

In this way the argument of the convolution function contains only parameters relating to the projection data. The final form of the convolution backprojection equations for fan-beam data are

$$f(r,\phi)=\int_0^{2\pi}\frac{1}{L^2}\int_{-\Gamma}^{\Gamma}R_\beta(\gamma).g(\gamma'-\gamma)D.\cos\gamma d\gamma d\beta \tag{4-54}$$

$$g(\gamma)=\frac{1}{2}\left(\frac{\gamma}{\sin\gamma}\right)^2 h(\gamma) \tag{4-55}$$

Rearranging to separate the backprojection operation gives

$$f(r,\phi)=\frac{D}{2}\int_0^{2\pi}\frac{1}{L^2}.g_\beta(\gamma)d\beta \tag{4-56}$$

$$g_\beta(\gamma)=\int_{-\Gamma}^{\Gamma}R_\beta(\gamma).\cos\gamma.\left(\frac{(\gamma-\gamma')}{\sin(\gamma-\gamma')}\right)^2 h(\gamma-\gamma')d\gamma' \tag{4-57}$$

The factor $1/L^2$ is now grouped with the backprojection operation, forming a *weighted backprojection.* The squared factor before the function *h* merely serves to modify the convolution function *h*.

4.2.3.1 Rebinning divergent-ray projection data

An alternate approach to reconstruction using divergent-ray projection data is to re-group or re-bin the projection data so that projections of parallel rays are grouped together. After the data has been re-binned the standard parallel reconstruction algorithms may be used.

Using parallel reconstruction algorithms not only simplifies the reconstruction process, but also yields a more correct estimate due to a more accurate interpretation in the Fourier domain (see sections 5.1.3 and 5.2.1).

The coordinate transform from parallel ray to fan beam is

$$t = D.\sin\gamma \qquad \theta = \beta + \gamma \tag{4-58}$$

Comparing the parallel projection $P_\theta(t)$ and the fan beam projection $R_\beta(\gamma)$

$$R_\beta(\gamma) = P_{(\beta+\gamma)}(D.\sin\gamma) \tag{4-59}$$

Assuming that the angular increment used for sampling the parallel projections is the same as the angular increment used for sampling the fan-beam projections, and is equal to the angular increment between successive fan beam rays

$$\Delta\theta = \Delta\beta = \Delta\gamma = \alpha \tag{4-60}$$

then β and γ can be expressed as $m\alpha$ and $n\alpha$ respectively where m and n are integer. Equation 4-59 becomes

$$R_{m\alpha}(n\alpha) = P_{(m+n)\alpha}(D.\sin n\alpha) \tag{4-61}$$

This states that the *nth* ray in the *mth* fan beam projection is the *nth* ray in the *(m+n)th* parallel projection. Note that the parallel projections will not be equi-spaced because of the sin $n\alpha$ term. However this can either be rectified by interpolation, or incorporated into the algorithm (which is simple where precalculated look-up tables are used to define the pixel-ray intersections as in the implementations described in chapter 8).

4.3. SERIES EXPANSION METHODS

Unlike the transform methods, iterative methods may be applied to either parallel or divergent-ray projection data by simply altering the values in the weighting matrix. Iterative methods are often used in industrial situations where the available data are incomplete [Censor, Y. 1983].

The series expansion methods begin by discretising the reconstruction space with a number of uniform density picture elements (pixels) $f_1, f_2, \ldots f_N$ where N is the total number of pixels. These pixels are usually square, and form an array type structure (figure 4-4).

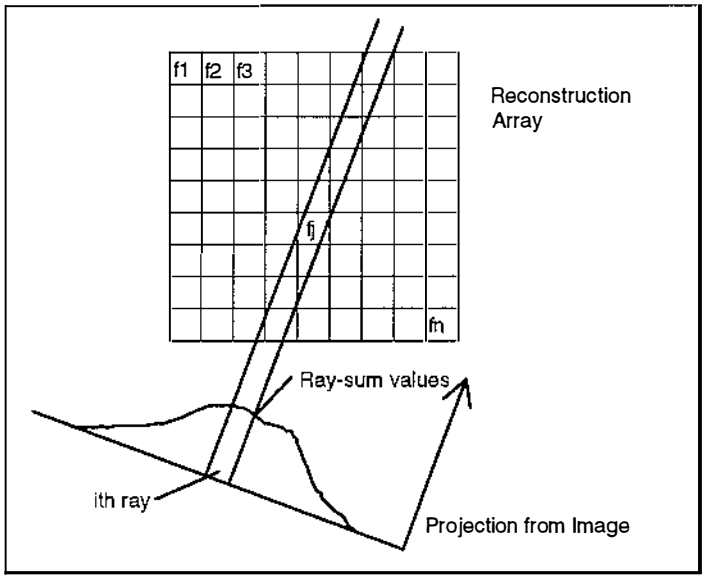


Figure 4-4. Typical geometry of discrete reconstruction area and ray-sum calculated from the ray path and the pixel-ray intersection areas.

The line integral discussed in conjunction with the transform methods has now been replaced by a *ray-sum.* As the name suggests a ray-sum p_i for the *ith* ray is generated by summing the pixel values of the array by a series of weights. The weights w_{ij} are determined by the intersection area between the *ith* ray path and the *jth* pixel. If the number of values (over all projections) is M then the ray-sums are

$$p_i = \sum_{j=1}^{N} w_{ij} f_j \qquad i = 1,2,\ldots,M \tag{4-62}$$

Each ray-sum is derived from a unique equation relating the measured samples to the estimate image. If the number of unknown pixel values is

less then or equal to the number of equations, then theoretically a solution is attainable.

4.3.1 Direct Matrix Inversion

Consider the ray-sum equations in matrix format

$$\underline{P} = \underline{W}.\underline{F} \tag{4-63}$$

where $\underline{P}$ is the vector of all ray-sums p_i.

$\underline{W}$ is a weighting matrix in which each row contains the N weights relating to the *ith* ray.

$\underline{F}$ is the vector of N unknowns (pixels).

Making the pixel values $\underline{F}$ the subject gives

$$\underline{F} = \underline{W}^{-1}\underline{P} \tag{4-64}$$

This is a trivial method of solution in theory. However a number of problems arise when it is implemented.

Consider the size of the weighting matrix $\underline{W}$. For a typical system the number of measurements will be 256 projections each containing 256 ray values, and the number of pixels will be 256^2. $\underline{W}$ will therefore contain 65535 rows each containing 65535 values giving a total of over 4296 million values. This alone poses a significant problem. However a further problem occurs when calculating the inverse of $\underline{W}$. In its basic form (ie. as it appears in the above equation) $\underline{W}$ is extremely sparse having less then one percent of its entries non-zero [Censor, Y. 1983 pp.411]. In this case this leads to severe ill-conditioning or singularity [Dusaussoy, N.J. and Abdou, I.E. 1991, pp1166], and consequently the system cannot be solved with precision this way.

The difficulty encountered when solving the set of equations directly has lead to the development of many iterative solution methods.

4.3.2 Algebraic Reconstruction Techniques (ART)

These methods start with an initial estimate of the image. In an iterative step, the current estimated image is refined to a new image by taking rays one at a time, and changing the values of the pixels which intersect this ray.

The amount of refinement is determined by the discrepancy between the measured ray value (from the physical system) and the value obtained when a pseudo-ray is passed (calculated) along the same path in the estimated image. Generating pseudo projections from an estimate image is known as *reprojection.* This is shown in figure 4-5. The difference value is called the residual, and the group of residuals associated with an entire projection is known as a residual projection.

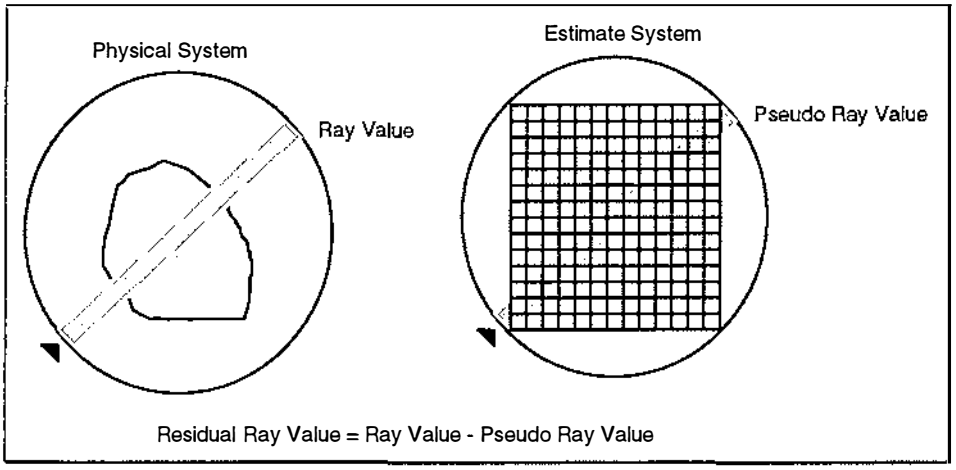


Figure 4-5. Residual ray values are calculated from the actual ray values and pseudo ray values generated by reprojection.

The residual value is evenly redistributed among the pixels which intersect with the ray path. This operation is similar to the backprojection operation encountered with transform methods, although on a discrete level. The mathematical description of this process is

$$\underline{F}^{i} = \underline{F}^{i-1} + \frac{P_i - (\underline{F}^{i-1} \cdot \underline{W}_i)}{\underline{W}_i \cdot \underline{W}_i} \underline{W}_i \qquad (4\text{-}65)$$

where $\underline{F}^{i}$ is the estimated image vector after the *ith* iteration

P_i is the *ith* ray value measured from the physical system

$\underline{W}_i$ is the vector containing the weights for the *ith* ray.

$\underline{W}_i \cdot \underline{W}_i$ is the dot product of $\underline{W}_i$ with itself

The derivation of this equation begins by expanding the expression $\underline{P}=\underline{W}.\underline{F}$ as

$$\begin{aligned} p_1 &= w_{11}f_1 + w_{12}f_2 + w_{13}f_3 + \ldots + w_{1N}f_N \\ p_2 &= w_{21}f_1 + w_{22}f_2 + w_{23}f_3 + \ldots + w_{2N}f_N \\ &\vdots \\ p_M &= w_{M1}f_1 + w_{M2}f_2 + w_{M3}f_3 + \ldots + w_{MN}f_N \end{aligned} \qquad (4\text{-}66)$$

The image $(f_1,f_2,f_3,\ldots f_N)$ denoted by the vector $\underline{F}$ can be considered as a single point in an N-dimensional space. In this space each ray-sum equation represents a hyperplane, and the solution of the set of equations is the intersection of all of the hyperplanes (assuming a single solution exists).

For a two variable system the equations would be

$$\begin{aligned} p_1 &= w_{11}f_1 + w_{12}f_2 \\ p_2 &= w_{21}f_1 + w_{22}f_2 \end{aligned} \qquad (4\text{-}67)$$

Figure 4-6 shows how the initial guess is projected onto the first hyperplane, then onto the second, back to the first, and so forth until reaching the intersecting point, which is the solution. Note that in this section the word projected refers to the mathematical operation of projecting one vector onto another [Kreyszig, E. 1983, pp 262].

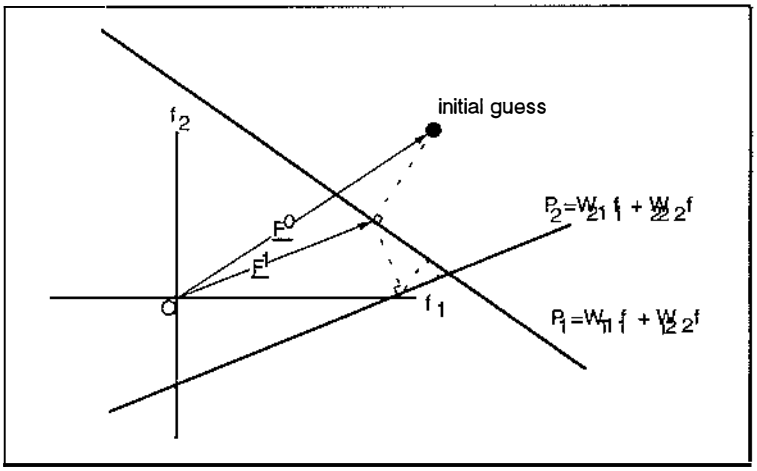


Figure 4-6. The initial estimate is projected onto other hyperplanes, causing the estimate to converge to a solution.

The first equation from equations 4-67 may be expressed as

$$P_1 = \underline{W}_1.\underline{F} \qquad (4\text{-}68)$$

The vector $\underline{W}_1$ is perpendicular to the hyperplane represented by the first equation. Figure 4-7 shows that the projection of any vector $\underline{OB}$ (whose initial point lies on the origin and terminating point lies on the hyperplane represented by the first equation) onto the vector $\underline{w}_1$ is of constant length and is equal to |OA| .

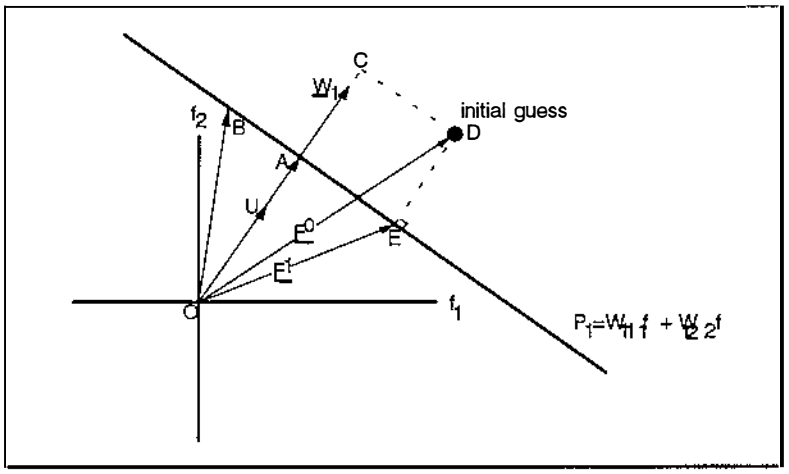


Figure 4-7. The projection of any vector $\underline{OB}$ onto the vector $\underline{w}_1$ is of constant length and is equal to |OA|

Backprojecting the initial guess $\underline{F}^0$ onto the hyperplane given by $p_1 = w_{11}f_1+w_{12}f_2$ gives the vector $\underline{F}^1$ which can be calculated by

$$\underline{F}^1 = \underline{F}^0 - \underline{ED} \tag{4-69}$$

To calculate $\underline{ED}$, the unit vector in the direction of $\underline{W}_1$ must first be calculated as

$$\underline{OU} = \frac{\underline{W}_1}{\sqrt{\underline{W}_1 \cdot \underline{W}_1}} \tag{4-70}$$

When a vector $\underline{b}$ is multiplied by a unit vector $\underline{a}$ the result is equivalent to the projection of $\underline{b}$ onto $\underline{a}$ [Kreyszig, E. 1983, pp263]. Therefore the perpendicular distance of the hyperplane from the origin $\underline{OA}$ is

$$|\underline{OA}| = \underline{OU}.\underline{OB} = \frac{1}{\sqrt{\underline{W}_1 \cdot \underline{W}_1}}(\underline{W}_1 . \underline{OB}) \tag{4-71}$$

The vector $\underline{F}$ is included in the set of vectors $\underline{OB}$, therefore

$$|\underline{OA}| = \frac{1}{\sqrt{\underline{W}_1 \cdot \underline{W}_1}}(\underline{W}_1 . \underline{F}) = \frac{P_1}{\sqrt{\underline{W}_1 \cdot \underline{W}_1}} \tag{4-72}$$

The length of the vector $\underline{ED}$ is

$$\begin{aligned} |\underline{ED}| &= |\underline{OC}| - |\underline{OA}| \\ &= \underline{F}^0 . \underline{OU} - |\underline{OA}| \end{aligned} \tag{4-73}$$

Substituting equations (4-70) and (4-72) into this equation (4-73) gives

$$|\underline{ED}| = \frac{\underline{F}^0 . \underline{W}_1 - P_1}{\sqrt{\underline{W}_1 \cdot \underline{W}_1}} \tag{4-74}$$

and since the direction of the vector $\underline{ED}$ is the same as the unit vector $\underline{OU}$

$$\underline{ED} = |\underline{ED}| . \underline{OU} = \frac{\underline{F}^0 . \underline{W}_1 - P_1}{\underline{W}_1 \cdot \underline{W}_1} . \underline{W}_1 \tag{4-75}$$

Substituting (4-75) into (4-69) gives (4-65).

Consider the effect equation (4-65) has on a single pixel of the image f_j.

$$f_j^{(i)} = f_j^{(i-1)} + \frac{p_i - \underline{f}^{(i-1)} . \underline{w}_i}{\sum_{k=1}^{N} w_{ik}^2} . w_{ij} \tag{4-76}$$

The new pixel value is determined from the old pixel value and the difference between the ray values and the reprojected ray values passing through the position corresponding to that pixel.

4.3.3 ART With Relaxation

[Eggermont, P.P.B. and Herman, G.T. and Lent, A. (1981)] expanded the standard ART method to include a relaxation parameter, where the relaxation refers to a sequence of real numbers λ_i , usually confined to the interval

$$\varepsilon_1 \le \lambda_i \le \varepsilon_2, \qquad \varepsilon_1, \varepsilon_2 \ge 0 \tag{4-77}$$

used to vary the magnitude of the change being applied to the image.

ART with Relaxation is very similar to standard ART (equation 4-76) and usually takes the form

$$f_j^{(i)} = f_j^{(i-1)} + \lambda_i \cdot \frac{p_i - \underline{f}^{(i-1)} \cdot \underline{w}_i}{\sum_{k=1}^{N} w_{ik}^2} \cdot w_{ij} \qquad (4\text{-}78)$$

This method is useful when the hyperplanes defined by the system do not have a common intersection point. When this happens the image f^i continually changes and the system will not converge. By reducing the value of λ_i for large values of i the system will remain between all of the intersection points, stabilising the estimated solution.

Equation 4-78 is the usual form used when implementing ART on a digital computer.

4.3.4. Maximum Entropy Methods

Maximum entropy methods include all iterative methods which converge to a solution which maximise the function

$$-\sum_{j=1}^{n} F_j \ln F_j \qquad (4\text{-}79)$$

subject to the constraints

$$\begin{aligned} &\underline{W} \cdot F_i = P_i \quad , \quad i = 1,2,\ldots,m \\ &\underline{F} \geq 0 \end{aligned} \qquad (4\text{-}80)$$

An example of an entropy maximisation reconstruction technique is MART (Multiplicative ART) [Censor, Y. 1983]. Starting with the initialisation $\underline{F} = e^{-1}\boldsymbol{1}$ (where $\boldsymbol{1}$ is a vector of ones), a typical step is

$$f_j^{(i)} = f_j^{(i-1)} \cdot \left(\frac{p_i}{\underline{f}^{(i-1)} \cdot \underline{w}_i} \right)^{\lambda_i} \qquad (4\text{-}81)$$

The relaxation parameter λ_k where $0<\lambda_k\leq 1$ is introduced to stabilise the solution when no unique solution exists.

The maximum entropy methods in general require many iterations to converge to a solution and each iteration requires more computation then the standard ART method [Dusaussoy, N.J. Abdou, I.E. 1991].

4.4 COMPARISON OF RECONSTRUCTION METHODS

The transform methods produce an estimate image in one iteration (ie. using the projection data only once). The estimate is of high quality assuming all projection data is available.

The convolution backprojection method gives better results then the direct Fourier inversion method because of the importance of radial interpolation in the frequency domain. Increasing the complexity of the interpolation gives improved results but increases the computation.

Iterative methods start with an initial estimate and improve the estimate converging to a solution. The projection data are often used more then once resulting in reconstruction times longer then those of the transform methods. Also the added burden of reprojecting every ray increases the computational requirements resulting in long reconstruction times. The iterative methods perform better when the projection data are incomplete because they converge to a solution based upon the data available.

A major factor in the selection of an appropriate reconstruction method for the defect detection application is the time required to reconstruct an image of suitable quality. The standard ART method is the fastest of the iterative methods because of the simple mathematics involved, and hence it is appropriate to focus on the standard ART method for comparative purposes, rather then the mode elaborate algebraic methods like entropy optimisation, and expectation maximisation which require significantly more processing, and are better suited to situations where projection data is limited or misssing [Dusaussoy, N.J. and Abdou, I.E. 1991, Browne, J.A. and Holmes, T.J. 1992].

Reconstruction times using the convolution backprojection method are relatively fast, requiring only one iteration of the projection data, (unlike the Algebraic methods which often require several iterations before converging to a satisfactory solution), and the mathematics are simpler and less sensitive to interpolation methods.

Chapter 6 applies both the Algebraic and Transform methods to high-speed image reconstruction, and discusses the suitability of each to the defect detection application.

Chapter 5

5 CURRENT REAL-TIME RECONSTRUCTION SYSTEMS

The technological breakthrough [Roder, F.L. et al. 1986] which eliminated the need for moving parts in X-ray scanners has enabled the introduction of high-speed X-ray scanners, which in turn has motivated the development of high-speed reconstruction systems. The term *real-time* has been introduced to differentiate between those systems which operate at close to video rates (25-30 frames/second), being termed real-time, and those which are simply faster then the old systems. Also included in the chapter title is the word *system* which has been used to indicate a hardware system using a reconstruction algorithm to process X-ray data.

Many algorithm improvements claiming to be high-speed have been developed. A large number of these [Matej, S. 1990], [Matej, S. Bajla, I. 1990], [Tabei, M. Ueda, M. 1990], perform modifications in the frequency domain and claim computational requirements of up to 1/6 that of the standard convolution backprojection algorithm [Tabei, M. Ueda, M. 1990]. Others [Cho, Z.H. Chen, C.M. Lee, S. 1990], [Goutis, C.E. Drossos, S.N. 1984] deal with backprojection in the spatial domain and claim speed improvements ranging from 1.8 to 4.4 times when compared to the standard backprojection algorithm.

Three very different hardware systems are discussed in the following sections, all claiming to reconstruct images in real-time. Two of the three systems [Gmitro, A.F. Tresp, V. Gindi, G.R. 1990], [Baringer, W.B. Brodersen, R.W. Petkovic, D. 1991] have been built and tested, and the other system [Kingswood, N. et al 1986] is unrealistic at this time due to excessive hardware requirements.

5.1 COMPOSITE DIGITAL/ANALOGUE RECONSTRUCTION SYSTEM - VIDEOGRAPHY

[Gmitro, A.F. and Tresp, V. and Gindi, G.R. (1990). Videographic Tomography-Part I: Reconstruction with Parallel-Beam Projection Data. IEEE Transactions on Medical Imaging, Vol.9 (No.4), pp365-375.]

The term Videography has been adopted by the authors of the paper [Gmitro, A.F. Tresp, V. Gindi, G.R. 1990] to describe a system of data acquisition and image reconstruction performed at video rates (30 images per second). The paper describes the implementation and performance of a reconstruction system which operates at these rates.

A number of methods are discussed for filtering the projection data. The least successful of these is the Acousto-Optic (AO) convolver, followed by the Surface Acoustic Wave (SAW) convolver. However the best results are obtained using a digital convolver.

The backprojection process is performed using a combination of mechanics, optics, and electronics, and is referred to as the Optical backprojector.

5.1.1 Acousto-Optic Convolver

This device (AO) performs the convolution operation by transforming the projection data into the frequency domain, and multiplying it with the frequency domain convolution (filtering) kernel.

An acousto-optic cell is used to convert the analogue electrical signal representing the projection data into a travelling acoustic wave in a transparent crystal. The analogue signal is used to amplitude modulate a carrier frequency bringing the projection data into the pass band of the AO device. A coherent optical wavefront is produced by a laser, a spatial filter and a collimating lens, and used to illuminate the acousto-optic cell (see figure 5-1). This acoustic wave produces a corresponding refractive index variation in the AO cell that is capable of modulating the phase of the coherent optical wavefront passing through it.

The wavefront exiting from the AO cell is transformed into the Fourier domain by a Fourier transforming lens, and directed with the aid of an optical lens through a binary transmission mask. This mask contains a

number of slits of different length, where the various lengths encode the convolution kernel. Passing the Fourier domain wavefront through this mask produces the product of the frequency domain projection data and the filtering kernel which is encoded in the mask.

After passing through the mask a Fourier transforming lens and stop (aperture) are used to convert the frequency domain wavefront into the spatial (time) domain once again. The stop is used to select only those diffraction components that are meaningful. A photomultiplier is used to spatially integrate the light intensity incident on it. The result is a signal representing the convolution of the projection function and the convolution kernel amplitude modulated on the carrier frequency. This carrier frequency is removed, and any temporal bias is removed by filtering. The layout of the acousto-optic convolver is shown in figure 5-1.

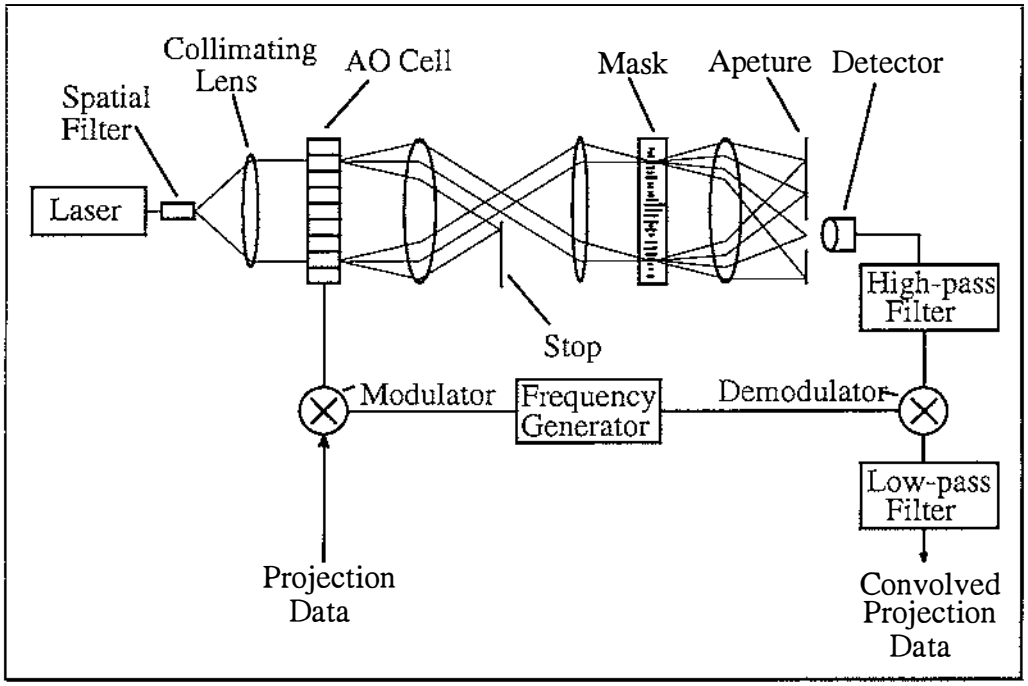


Figure 5-1. Layout of acousto-optic (AO) convolver.

The performance of the acousto-optic convolver was far inferior to the other convolution devices giving an image with little contrast between features. Several factors are attributed to this, one being the rounding which occurs in the low-frequency response of the acousto-optic filter because of the limited extent of the of the kernel encoded in the mask. Another arises from optical phase errors in the coherent optical system.

The most critical optical component in the acousto-optic convolver is seen to be the lens between the acousto-optic cell and the mask, and although a

high quality lens was used, significant error was attributed to lens aberrations.

5.1.2 Surface Acoustic Wave (SAW) Convolver

The surface acoustic wave convolver is seen to have a number of practical advantages over the acousto-optic system. It is constructed on a monolithic substrate yielding a device that is stable, compact, rugged, and relatively inexpensive to produce.

The operation of the surface acoustic wave convolver may be thought of as a conventional resistor-capacitor filter circuit which filters an input signal according to the frequency domain characteristics determined by the component values specified. The surface acoustic wave device is simply a filtering device whose frequency domain characteristics can be accurately controlled.

A pair of inter digital electrode transducers, one for transmitting the signal and one for receiving it, are mounted on a piezo-electric substrate (see figure 5-2). An amplitude modulated electrical signal (as in the acousto-optic device) applied to the input transducer induces a strain in the substrate that propagates on the surface of the substrate. The acoustic waves produced are approximately plane, and travel in both the forward and backward direction. The reverse wave is damped by an acoustic absorber positioned at the ends of the substrate, and the forward wave propagates to a multistrip coupler where it is absorbed and reradiated into the region of the output transducer. The effect of the multi-strip coupler is to integrate the wave energy over the extent of the coupler, and radiate it evenly over the extent of the output transducer. The reradiated strain field induces an electrical potential in the output transducer which then has the carrier frequency removed and is filtered.

The impulse response of the filter is the convolution of the responses of the input and output transducers. The input and output responses are determined by the geometry of the electrodes and are designed to match the desired filtering kernel. The layout of the surface-acoustic wave convolver is shown in figure 5-2.

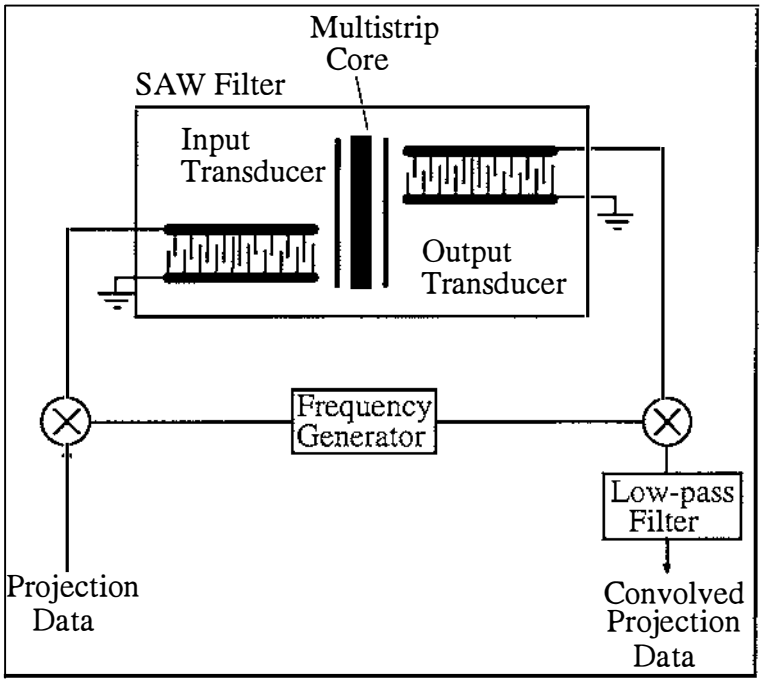


Figure 5-2. Layout of surface acoustic wave (SAW) convolver.

The performance of the surface acoustic wave convolver was superior to acousto-optic convolver giving less shading and consequently clearer definition of features. This is attributed to the improved low-frequency response. However some higher-frequency artefacts occur in the results as a result of acoustic reflections within the device, which lead to spurious delayed signals in the time-domain, and ripple in the frequency response.

5.1.3 Digital Convolver

Although both of the analogue convolvers described above work, it is difficult to achieve sufficient accuracy in the frequency domain. For this reason a digital convolver was designed and constructed to achieve a more accurate filter. A digital finite-impulse-response filter was designed implementing 56 kernel coefficients with 12-bit precision.

The digital convolver uses a digital signal processing (DSP) chip (TRW TDC1028) as a building block. The chip implements a FIR filter with 8 taps and four-bit precision in both the data and coefficients of the kernel. In order to achieve 12-bit precision the data is separated into 3 segments where each segment processes 4 bits. Each chip has 8 multipliers (4×4 bit) and 8 accumulators (13-bit), with delay elements between the accumulators. The TRW TDC1028 chips are cascaded to implement a 56 tap kernel (see figure 5-3).

The data are separated into segments based on bit significance, and passed down through the chain of chips. Each segment contains 3 channels to accommodate for the 12-bit kernel coefficients. Because the convolution kernel has high values near the middle it is not necessary to have chips in the high bit positions for all tap locations. 8-tap delays are used to keep all outputs of the channels in proper synchronisation.

A microcomputer provides the coefficients to the chips via a coefficient interface. At each stage the data are multiplied by the appropriate coefficients and summed together. A pipeline adder network and output multiplexer is used to combine the results from each of the 9 channels, zero-padding where appropriate, and output the convolved data. After combining the 9 channels a 24-bit value is obtained. The 12 most significant non-zero bits are taken from this and used as the output value.

Figure 5-3 shows the layout of the digital convolver.

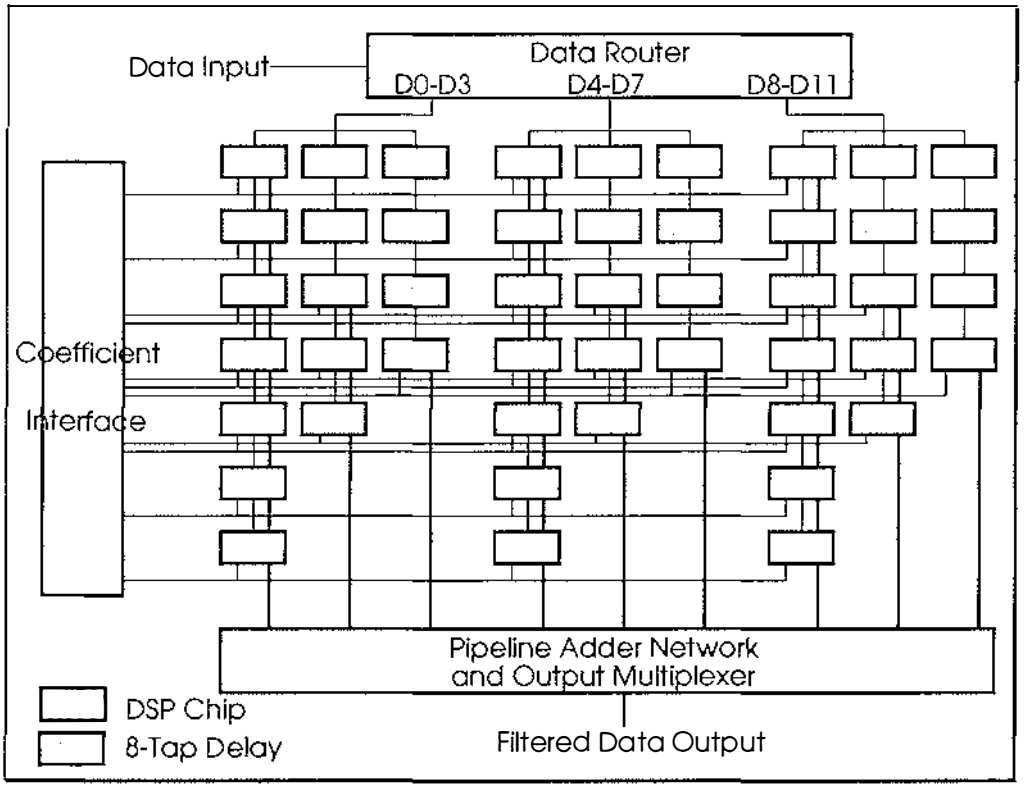


Figure 5-3. Layout for Finite Impulse Response (FIR) Filter Performing Convolution.

The performance of the digital convolver was superior to both of the analogue convolvers. Some shading error occurred because of the limited extent of the kernel. The ideal FIR filter should be 512 taps wide for projection data each containing 256 ray values, whereas the FIR kernel constructed contained only 56 taps.

5.1.4 Optical Backprojector

The optical backprojection system consists of a line (one-dimensional) display, an anamorphic optical imaging system, an optical image rotator, and a video camera as shown in figure 5-4.

The filtered projection data is used to intensity modulate the line display (x-direction only). An offset must be added to the bipolar data making it all positive in order to use the line display.

The projection (light intensity) is then smeared out in the y direction (backprojected) to cover the reconstruction area using an optical lens. A prism is then used to rotate the smeared projection to the appropriate angle where it is then integrated with the other projections using a video camera.

The prism is rotated using a stepper motor and some complex optics. Assuming one step per projection, about 15,000 steps per second are required by the stepper motor to achieve real-time reconstruction.

The video camera acts to integrate all the backprojections together to form the resultant image. A complete set of projections are processed during one video frame time so the output from the camera is a reconstructed image.

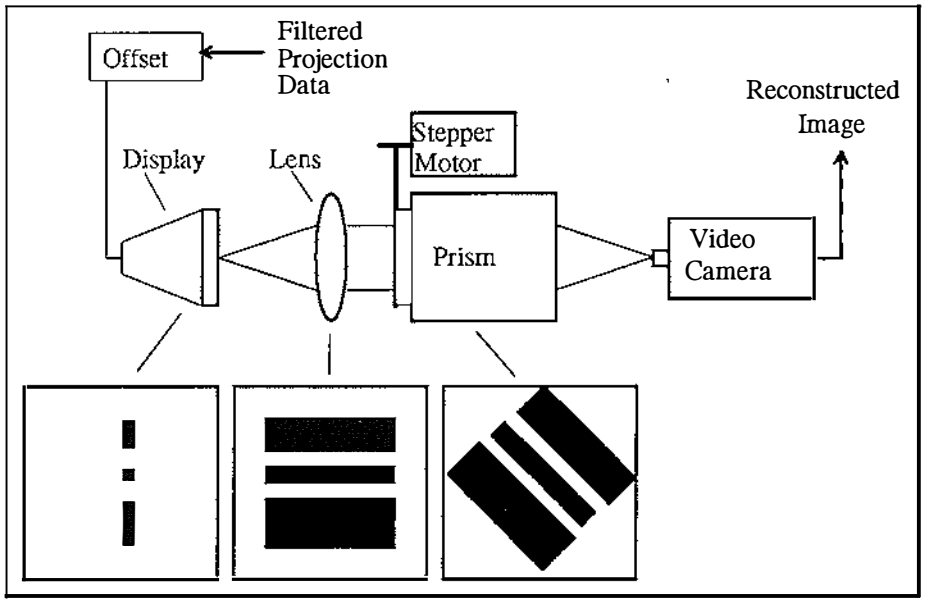


Figure 5-4. Layout of optical backprojection system.

The results obtained using the optical backprojection system described were poor, with little contrast due to saturation of the display. This resulted from the large bias added to compensate for the large negative tails created by filtering the projection data.

In an effort to overcome this saturation problem the filtered projection data were amplified and offset in such a way that both the large positive and negative excursions were clipped by the display. This resulted in better contrast in the low-contrast regions, although introduced blurring in the high-contrast regions.

To avoid both saturation and blurring the filtered projection data were separated into positive and negative components. The positive portion was reconstructed, then the negative portion was inverted making it a positive quantity, and reconstructed. A subtraction of the two images yielded a correct reconstruction without saturation or blurring. This method was performed off-line, and the resultant image was of good quality. However resolution is limited by the alignment of the optical axis of the prism with the mechanical axis of rotation. Movement in the rotor results in loss of spatial resolution. Also two reconstructions are required instead of one, and an additional real-time image subtraction device is required.

5.2 PARALLEL PIPELINE PROJECTION ENGINE (PPPE)

[Baringer, W.B., Brodersen, R.W., and Petkovic, D. (1991) Computer Vision Hardware using the Radon Transform. Proceedings of the IEEE, Vol.5, pp508-513.]

The parallel pipeline projection engine has been implemented and tested, and is capable of performing both a forward Radon transform (projection) and inverse Radon transform (backprojection) using the same hardware system. In the former mode the Image data is passed through the system, and the projections are read out, and in the latter the projection data and a null image are fed into the system, and a backprojected image results.

The system employs a processing module for every projection angle. Each module is connected to a Host computer via a VME Host bus providing the high level control. A high-speed pipelined video bus (10 MHz) runs through each of the processing modules and is used for carrying the image data. The components of each module are shown in figure 5-5. The tasks shown in the dotted box are implemented on a VLSI ASIC (application specific integrated circuit) because of the high computation and intensive I/O requirements.

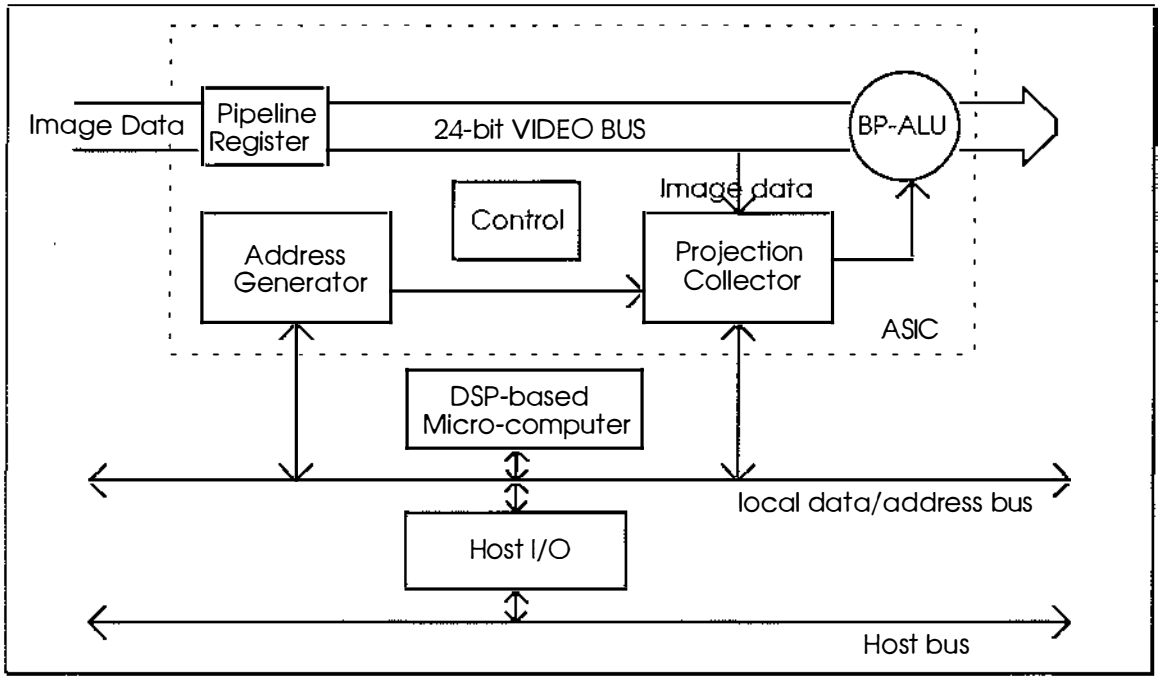


Figure 5-5. Layout of a processing module in the Parallel Pipeline Projection Engine.

In the forward mode of operation (projection) the image is pipelined through the modules via the high-speed video bus. Each processor stage reads the incoming pixel value and adds it to the appropriate accumulator in the projection collector. The correct accumulator cell is pointed to by the address computation module which has been pre-programmed by the local DSP chip for the specific angle of projection (ie. nearest neighbour interpolation is used). When the entire image has passed through the system, each module contains the projection of the image through a specific angle.

The DSP then reads the projection data from the ASIC for processing, while the next projection is processed. In the inverse mode of operation (backprojection) the projection data are passed to the projection collector either from the micro-computer within the module (usually after processing the forward mode projection data) or from the host computer or workstation. A null-image (all zeros) is passed into the pipelined video-bus. The projection data is then added to the image data using the backprojection arithmetic logic unit (BP-ALU). The cell-pixel combinations are once again provided by the address generator.

The overall system contains multiple *Radon-transform* boards, each containing four DSP micro-computers and 32 VLSI ASICs contained within a 21-slot VME card cage, giving a total processing capability of one-quarter trillion operations per second. The card-cage has a 68020-based single-

board computer and an ethernet communications card to provide support as a local host to all of the slave processors, and local-area network support to nearby workstations. Multiple video-cameras and video recording devices are connected to the real-time image pipeline in the system.

5.3 TRANSPUTER RECONSTRUCTION

[Kingswood, N. and Dagless, E.L. and Belchamber, R.M. and Betteridge, D. and Lilley, T. and Roberts, J.D.M. (1986). Image Reconstruction using the Transputer. IEE Proceedings, Vol.133 (Pt.E, No.3), pp139-144.]

The paper describes a theoretical investigation into the use of transputers for reconstruction of tomographic images. An iterative reconstruction method (ART) is employed so there is no mention of pre-filtering. However the backprojection process is the same as that of the transform methods.

Each pixel in the discrete reconstruction area is implemented using a transputer, and the four communications links are connected to those transputers in orthogonal positions. The ray paths and pixel-ray intersection areas are pre-computed using a host computer and distributed to the appropriate transputers before reconstruction.

During reconstruction the residual projections are fed into the transputer array from the sides, and propagate through the array in a pipelined manner. The host computer is used to send the appropriate ray values to the appropriate boundary transputers. Each transputer (pixel) receives a ray value, and adds the appropriate portion of the ray value (given by the pre-calculated downloaded data) to its current pixel estimate, then passes the ray value to the next transputer in the ray path. In this way a transputer is only processing rays which cross the corresponding pixel.

The paper offers theoretical predictions of speed improvements obtainable using transputers compared with sequential processing, with reconstruction areas ranging from 7 by 7 pixels to 50 by 50 pixels. These predictions assume that the estimate image will have converged sufficiently after one iteration (ie. all projections used once). The maximum utilisation achieved by any of the transputer systems is 34.7 percent (for the 7 by 7 array). The maximum speed improvement over sequential procesing achieved by any of

the transputer systems is 218 times (for the 50 by 50 array) producing about 60 backprojection operations per second.

However this theoretical solution method is not practical with current technology for a number of reasons. Firstly, most X-ray data acquisition systems generate enough data to reconstruct an image of at least 256^2 pixels [Webb, S. 1988]. Therefore a reconstruction system offering resolutions of 50^2 pixels or less would not usually be desirable. Secondly the high cost of the system would prevent implementation. For example, the paper quotes the price of a single transputer at $500 US (1986), which results in $1.25 million US for a system containing 50^2 pixels. Although this cost is reducing due to improvements in DSP fabrication, it indicates the magnitude of the cost relative to the available technology. Finally the large physical size of the larger systems would introduce problems such as signal skew, and heat dissipation. For example, standard manufactured transputer boards seldom contain more then ten transputers (eg. the FAST984 transputer board contains 9 inmos 32-bit T800 transputers) so a system containing 50^2 transputers would require in the order of 250 transputer boards unless application specific boards were manufactured, which would increase the cost of the system.

5.4 Review of Real-Time Reconstruction Systems

In an effort to increase the reconstruction speed, the quality of the reconstructed image is being reduced. Both the Acousto-Optic and Surface Acoustic Wave convolving devices (sections 5.3.1, and 5.3.2 respectively) displayed results unacceptable for most reconstruction applications. The third and most successful convolving device is the digital convolver (section 5.1.3). However this device achieves high-speeds by reducing the size of the convolution kernel from an ideal 512 taps, to 56 taps, resulting in poorer reconstruction quality.

The optical backprojection device suffered problems due to the large dynamic range of the convolved data (both in the positive and negative regions), and the resulting image quality was not acceptable. In an attempt to correct for the poor image quality, the processing time was increased.

The Parallel Pipeline Projection Engine (section 5.2) simplifies the reconstruction process by performing nearest-neighbour interpolation between the projection data and the reconstruction area, and the Transputer system (section 5.3) relies on a reduced number of pixels to reduce the inter-transputer communication times, resulting in reduced image quality. Both of these actions result in reduction of the resulting image.

Another feature of all the real-time systems discussed is the complexity and cost of the hardware required to achieve high-speeds. The following chapters describe modifications made by the author to a current reconstruction algorithm, which significantly reduce the mathematical requirements such that the hardware required to perform the modified reconstruction algorithm is both simple and inexpensive.

Chapter 6

6 HIGH-SPEED RECONSTRUCTION METHODOLOGIES

All of the high-speed reconstruction systems discussed in chapter 5 used some form of parallel processing to achieve the high rate of calculation required by the standard algorithms. The Videographic system used optics which is highly parallel, the Parallel Pipelined Projection Engine used multiple ASICs and DSPs to achieve a high level of parallelism, and the transputer solution is parallel by nature of the transputer architecture (where multiple transputers are used).

This chapter looks at other methodologies of achieving real-time reconstruction. Section 6.1 examines the feasibility of using fewer projections in the realisation that many applications, including the detection of defects within felled logs, do not require an image of high quality as is required by medical applications. Reducing the image quality allows the number of pixels comprising the image to be reduced, and the number of rays and projections to be reduced. Section 6.2 examines ways of reducing the required computation by reducing the convolution kernel, and modifying the backprojection algorithm.

The backprojection operation is used in many of the reconstruction algorithms, and constitutes the major computational cost [Cho, Z.H. et al 1990]. Section 6.2.1 examines a way of reducing the backprojection computation by modifying the backprojection algorithm to suit the geometry of the system. Section 6.2.2 shows the effect of reducing the size of the convolution kernel, using Chebychev minimisation to optimise the kernel coefficients.

6.1 REDUCTION OF PROJECTION DATA

It has been suggested [Garden, K. Mnyama, D. 1987] that as few as three projections may be sufficient to identify knots from clearwood in felled logs, resulting in a reconstructed image with an intensity ratio of 1.92

between the clearwood and knots. However the density of the wood changes after the log has been felled because of drying. This adds to the variation in the distributed attenuation coefficient within the log, reducing the intensity ratio stated above. Assuming ideal conditions, three projections would only indicate the presence of a knot, and its approximate whereabouts. The resulting image would be unlikely to contain enough information to determine the extent of the knot, the shape of the knot, or even the number of knots. For another perspective consider the available data in the Fourier domain. Recall from chapter 4 that the Fourier transform of a single projection gives information about a single line crossing the origin of the Fourier domain at the angle of the projection, and an exact reconstruction can only be obtained when the data cover the Fourier domain. Clearly three lines of data does not constitute adequate coverage of the Fourier domain.

The algebraic reconstruction techniques are not derived from the Fourier slice theorem, and cope better in situations where the projection data is sparse or missing (see chapter 4). Therefore the following discussion uses algebraic reconstruction methods for image reconstruction.

Tests were performed to determine the minimum number of projections required to give an adequate visual representation of the object being reconstructed. The test objects are shown in figures 6-1 and 6-2.

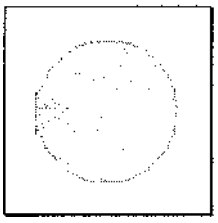


Figure 6-1. Test phantom containing two features.

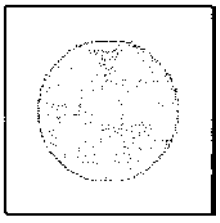


Figure 6-2. Test phantom containing four features.

The intensity value of the large circle in figure 6-1 is 100, the smaller triangle has an intensity value of 150, and the background has 0 intensity. In figure 6-2 the large circle has an intensity value of 100, the small inner circle has an intensity value of 50, the large and small triangles have intensity values of 140 and 120 respectively, and the background intensity is 0. These constitute high contrast images and are meant to represent typical log defects.

In situations where the projection data is sparse it is important to obtain the most information from the projection set. The angular increment is set close to orthogonal resulting in a large difference between subsequent projection profiles, and maximum improvement of the estimated image.

The actual angular increment between projections should also be determined with consideration for the number of available projections, such that the distribution of projection angles is evenly spread over the 360 degrees.

6.1.1 Results and Discussions

To help determine the performance of image reconstruction from few projections figures 6-1 and 6-2 were reconstructed using the standard ART reconstruction algorithm with 15 projections. The projection increment angle was calculated to be 96 degrees, making each projection almost orthogonal to the previous projection, and the resulting distribution of projections even over the 360 degrees (having 12 degree separation). The computer program used to generate the projection data is printed in appendix A, and the computer program used to reconstruct the images using the ART method is printed in appendix B.

The reconstructions of figures 6-1 and 6-2 are shown in figures 6-3 and 6-4 respectively. Interpreting the images visually is very subjective and the following discussion serves to make a point rather than to be definitive about the image qualities.

The object in figure 6-1 is extremely simple with a single feature superimposed onto an object body. It can be seen from figure 6-3 that the existence of the feature is indefinite until about the 7th iteration, and not clear until about the 11th iteration.

The object in figure 6-2 is less simple containing three features. The highest contrast feature is the small circle in the centre. The existence of this is not definite until about the 7th iteration, and the shape of the centre feature is not clear until about the 12th iteration. After about 10 iterations the existence of the triangular features is evident, but they are not clear until about the 15th iteration.

Figures 6-3 and 6-4 demonstrate that it may be possible to perform adequate reconstruction using fewer projections, therefore reducing the computation

and time required by the reconstruction process. However this reduction in computation is traded off against image quality. The next section shows how it is possible to reduce the reconstruction computation without affecting the image quality.

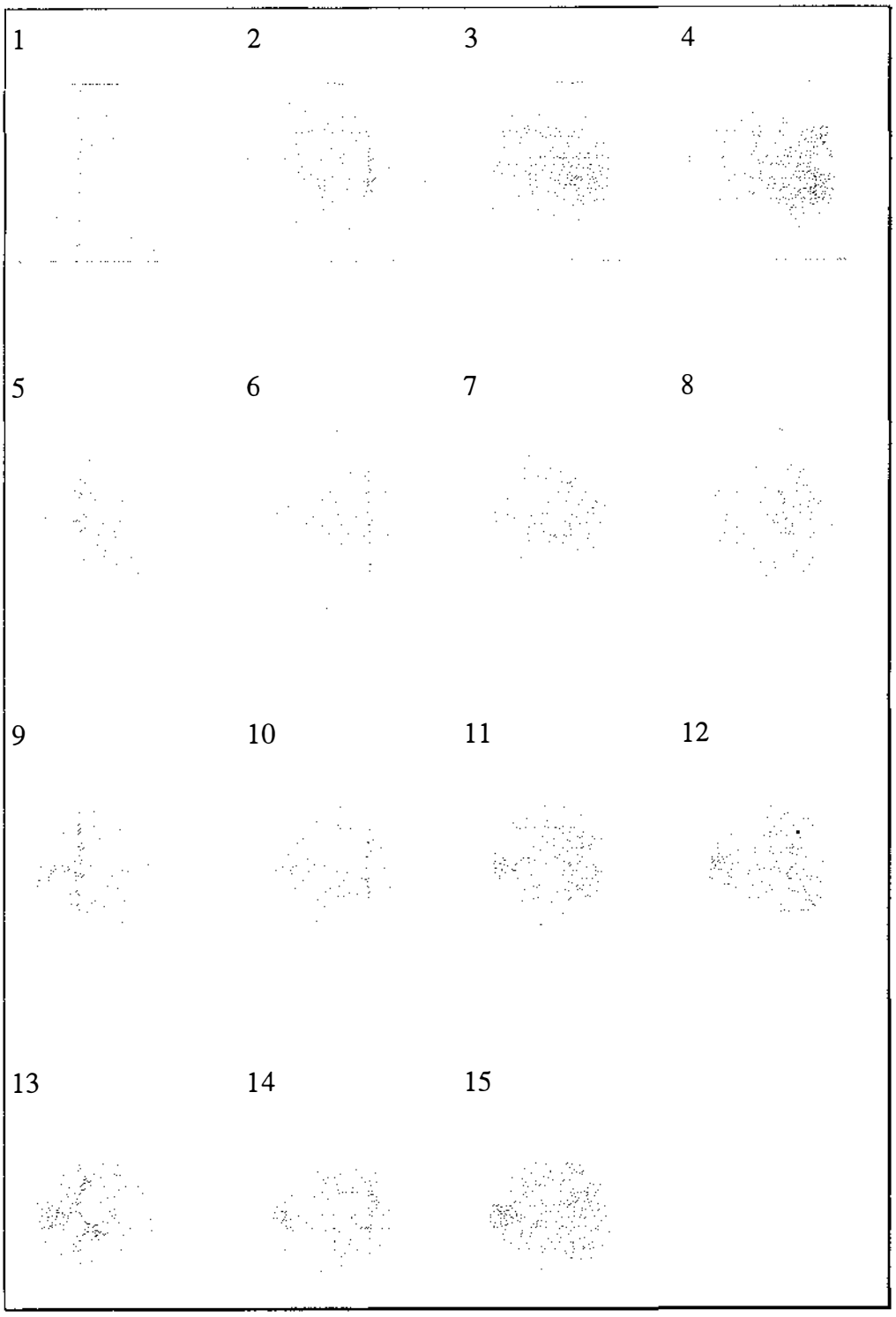


Figure 6-3. Reconstruction of figure 6-1 (15 iterations) using ART.

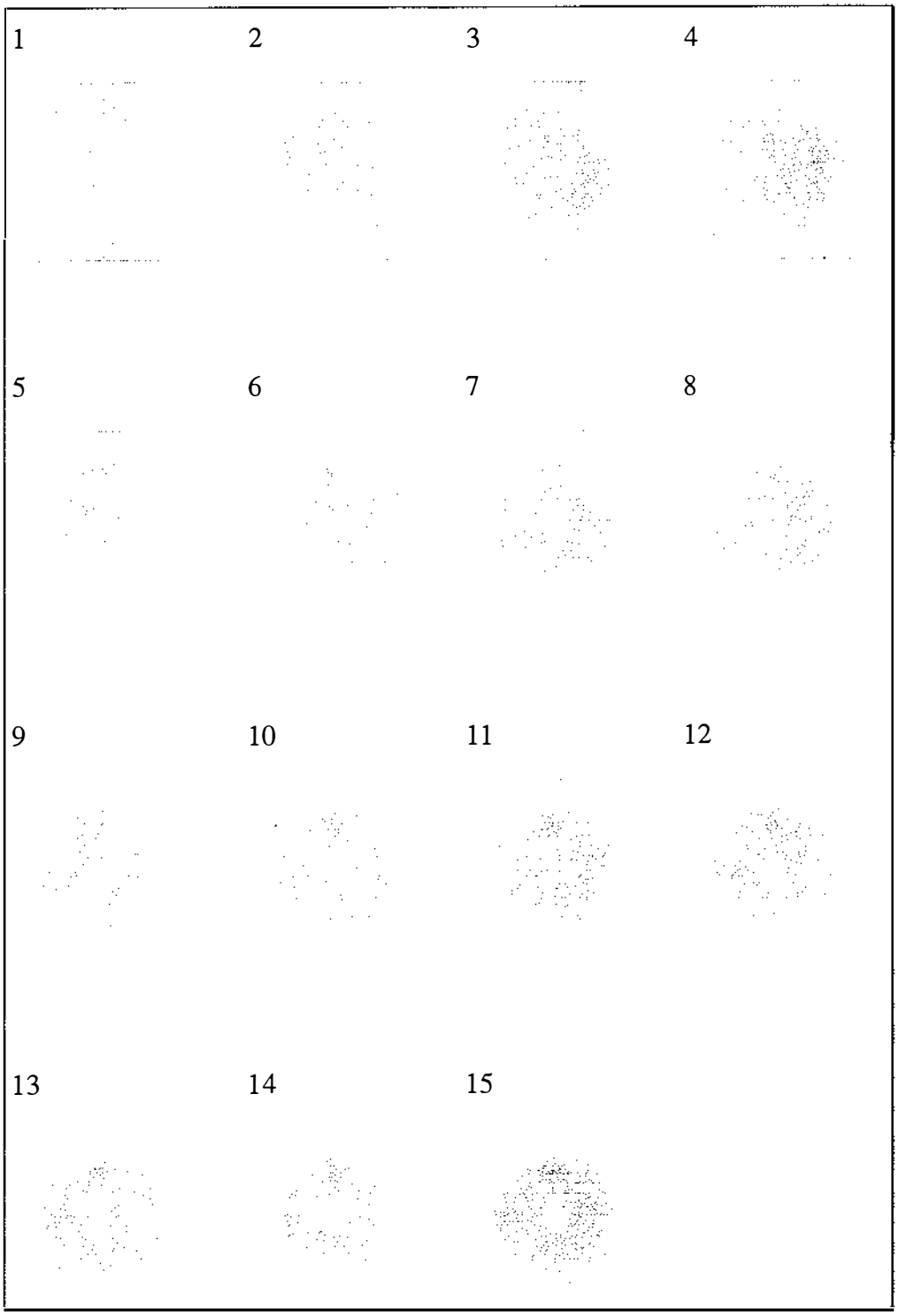


Figure 6-4. Reconstruction of figure 6-2 (15 iterations) using ART.

6.2 MODIFICATION OF THE RECONSTRUCTION ALGORITHM

The following sections describe and develop a novel reconstruction algorithm based on the standard convolution backprojection algorithms. Most of the mathematical computation in the reconstruction process arises in the backprojection operation because of the rotation of the projections. Standard reconstruction methods discretise the reconstruction area into a rectangular geometry, and hence use cartesian coordinates to perform calculations. An obvious way to overcome the run-time calculation overhead is to use precalculation and high-speed table look-up methods wherever possible. Unfortunately the standard reconstruction algorithms are not well suited to precalculation, and require look-up tables (high-speed RAM) of the order of 32 Megabytes which can be slow and difficult to manage.

The modifications described in this section overcome the problems associated with the standard reconstruction by discretising the reconstruction area in a manner suited to the rotation inherent in the data acquisition hardware. The result is to reduce the size of the required look-up table to the order of 65 Kilobytes.

Another significant computational component is the convolution operation. Reducing the size of the convolution kernel allows faster processing of the projection data, however this is achieved at the expense of image quality. Discussion of this is continued in section 6.2.2.

6.2.1 Derivation of Modified Backprojection

The standard reconstruction equations 4-19 and 4-20 can be written as equations 6-1 and 6-2. The frequency domain multiplication of equation 3-20 has been replaced by spatial domain convolution in equation 6-2, and a Dirac delta function is used in equation 6-1 instead of the argument of Q in equation 3-19.

$$f(x,y) = \frac{1}{2}\int_{0}^{2\pi}\int_{-\infty}^{\infty} Q(\theta,t).\delta(x.\cos\theta + y.\sin\theta - t)\,dt\,d\theta \qquad (6\text{-}1)$$

$$Q(\theta,t) = \int_{-\infty}^{\infty} P(\theta,\alpha).g(t-\alpha)\,d\alpha \qquad (6\text{-}2)$$

Equation (6-2) convolves the projection P with an *anti-blurring* kernel g and equation (6-1) performs the backprojection of the filtered projection Q over the reconstruction area *f* using the mapping function δ. The name *mapping function* is adopted as it maps the projection functions onto the reconstruction area.

The measurement hardware provides a finite number of projections N at equal angular displacements around the object, each projection containing 2M rays, and the reconstruction area is divided into an array (l×k) of rectangular pixels (cartesian pixel geometry). Incorporating these into equation (6-1) gives the standard discrete backprojection equation

$$f(l\Delta x, k\Delta y) = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t). \\ \delta_I(l\Delta x.\cos[n\Delta\theta] + k\Delta y.\sin[n\Delta\theta] - m\Delta t) \qquad (6\text{-}3)$$

$l=1,2,3...L,$

$k=1,2,3..K.$

An interpolating mapping function δ_I is required because the backprojection of any ray value $Q(n\Delta\theta,m\Delta t)$ will in general not coincide exactly with any of the discrete pixels $(l\Delta x,k\Delta y)$. The interpolation of the function δ_I may be nearest-neighbour, linear, spline, or other more elaborate interpolation scheme. However linear and nearest neighbour are the most common due to their low computation requirement.

To implement equation 6-3 at high-speeds an obvious solution would be to pre-calculate the values of the mapping function and store them in a look-up table for subsequent retrieval at run-time. For the standard algorithm using nearest-neighbour interpolation this requires N×W elements in the table (where N is the number of projections, and W is the number of pixels in the reconstruction area).

It is possible to reduce the size of the mapping function by a factor of N by discretising the reconstruction area using a non-cartesian coordinate system. In order to develop an algorithm which will achieve this reduction the coordinates of the standard backprojection equation (6-1) are transformed from cartesian (x,y) to polar (r,ϕ)

$$f(r,\phi) = \frac{1}{2}\int_{0}^{2\pi}\int_{-\infty}^{\infty} Q(\theta,t).\delta(r.\cos[\phi-\theta]-t)\,dt\,d\theta \qquad (6\text{-}4)$$

Equation (6-4) describes the system as continuous in both the reconstruction area and the projection space. However the projection space is discrete by nature of the physical hardware in both the ray position t and the projection angle θ. Thus equation (6-4) becomes

$$f(r,\phi) = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t). \\ \delta_I\left(r.\cos[\phi - n\Delta\theta] - m\Delta t\right) \qquad (6\text{-}5)$$

Once again an interpolating mapping function δ_I is required because the filtered projection data is only available at discrete points.

Let the radial and angular components (r,ϕ) be discretised in some arbitrary manner (r_d,ϕ_d) where the number of discrete elements (pixels) is L. The number of different argument values $(r_d.\cos[\phi_d - n\Delta\theta])$ for the mapping function will be N×L because each new projection n gives rise to L new and different argument values.

The reconstruction area can be divided into N equal angular segments of size $\Delta\theta$ where $\Delta\theta$ is the angular displacement between projections. The angle ϕ in equation (6-5) can now be replaced by $j\Delta\theta + \phi'$ where j indicates the segment and ϕ' indicates the angular offset within the segment such that $0 \le \phi' < \Delta\theta$.

$$j = \text{truncation}\left(\frac{\phi}{\Delta\theta}\right) \quad , \quad \phi' = \text{modulo}(\phi, \Delta\theta) \qquad (6\text{-}6)$$

Equation (6-5) becomes

$$f(r, j\Delta\theta + \phi') = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t). \\ \delta_I\left(r.\cos[(j-n)\Delta\theta + \phi'] - m\Delta t\right) \qquad (6\text{-}7)$$

Let the radial and angular components (r,ϕ') within a each segment be discretised in some arbitrary manner (r_d,ϕ_d'), and let the segment discretisation be repeated for all segments where the total number of pixels is once again L. The number of argument values $(r_d.\cos[(j-n)\Delta\theta + \phi_d'])$ for the mapping function is reduced from N×L to L because the values are repeated for each projection angle $n\Delta\theta$, and each projection is presented with an identical pattern of pixels.

The general modified backprojection equation is

$$f(r_d, j\Delta\theta + \phi_d') = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t). \\ \delta_I\left(r_d.\cos\left[(j-n)\Delta\theta + \phi_d'\right] - m\Delta t\right) \tag{6-8}$$

This result is independent of both the discretisation of the reconstruction area and the interpolation method chosen. Both of these issues are discussed in the following sections.

6.2.1.1 Image Area Discretisation

Discretisation using cartesian coordinates is well established. The standard video capture devices (CCD), computer displays, and memory storage units (RAM) all utilise both physically and conceptually rectangular discretisation. It follows naturally from this that most image processing algorithms will maintain a cartesian coordinate system to avoid the need for posterior interpolation before display.

However, it is not always necessary to display an image before further processing is performed although this is often desirable. For instance the process of feature detection can be performed by a computer using any coordinate system, and only the relevant information need be displayed.

Equation (6-7) describes the reconstruction area as continuous within each sector but repetitive by sectors (ie. the pattern within a sector is repeated for all sectors). In order to implement the system on a digital computer the reconstruction area must be discretised into a finite number of pixels. Note

that the term pixel refers to an area of uniform intensity, which can be of any shape, ie. not only square.

Because the modified backprojection algorithm is not dependent upon the discretisation, the pixel geometry is arbitrary and can be chosen to suit the application. Two different sector pixel geometries are discussed in the next sections.

6.2.1.2 Equi-Radial Best-Fit Area Pixel Geometry

Many applications of tomographic reconstruction require the spatial resolution of the image to be almost uniform throughout the image (medical imaging is an example). Figure 6-5 shows a pixel geometry where the pixel area is kept fairly uniform with variation not exceeding plus or minus 0.5 times the desired pixel area. The radial increment is constant and the angular component is divided into an integer number of pixels such that the area of each pixel is as close to the desired area as possible.

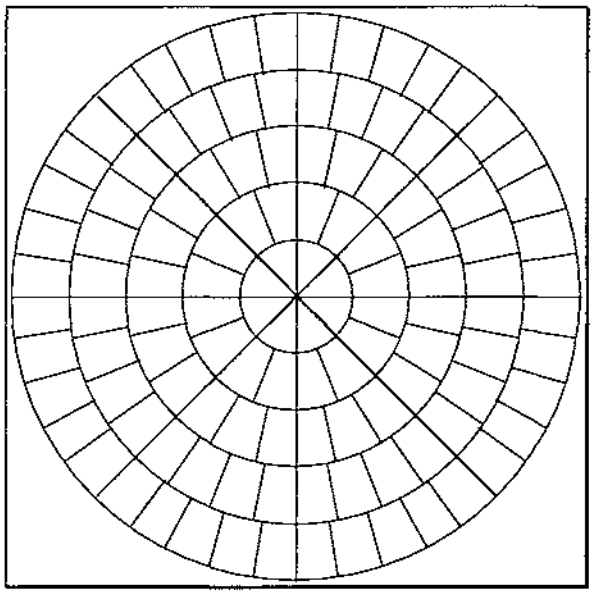


Figure 6-5. Reconstruction area discretised using Equi-radial Best-fit area.

The pixels are obviously not rectangular. However for pixels of reasonably small size they will be approximately rectangular. This approximation improves with increasing radius. Using this assumption in order to easily define the pixel centre of area, the discrete reconstruction equation for this pixel geometry is

$$f\left(\left[h-\tfrac{1}{2}\right]\Delta r, j\Delta\theta + g\Delta\phi'_h\right) = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t).$$
$$\delta_I\left(\left[h-\tfrac{1}{2}\right]\Delta r.\cos\left[(j-n)\Delta\theta + g\Delta\phi'_h\right] - m\Delta t\right) \qquad (6\text{-}9)$$
$$\Delta\phi'_h = \frac{\Delta\theta}{trunc\left(\frac{h\Delta r.\Delta\theta}{d}\right)+1}$$

where radial coordinate $h \geq 1$,
sector coordinate $0 \leq j < N$,
angular coordinate $1 \leq g \leq \text{trunc}\left(\frac{h\Delta r.\Delta\theta}{d}\right)+1$,
d is the maximum pixel width

By specifying the maximum allowable pixel width d, the depth at which the first segment division occurs can be controlled. A special case of this geometry is discussed below , where d is greater than the length of the segment at the outer radius ($d=h\Delta r.\Delta\theta$) and the coordinate g can be ignored.

6.2.1.3 Equi-Radial/Angular Pixel Geometry

Most current tomographic systems require a large number of projections, (>200) and consequently the number of sectors in the reconstruction area will also be large. It is reasonable then to discretise the rotational component in equation 6-8 from $j\Delta\theta+\phi'$ to $i\Delta\phi$ where $\Delta\phi = \Delta\theta$ (as the segment angle is the same as the projection increment angle), and the radial component from r to $(h-1/2)\Delta r$ where h is an integer greater then zero. An example of this form of pixel geometry is shown in figure 6-6 and the reconstruction equation is described by

$$f\left(\left[h-\tfrac{1}{2}\right]\Delta r, i\Delta\phi\right) = \frac{1}{2}\sum_{n=0}^{N-1}\sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t).$$
$$\delta_I\left(\left[h-\tfrac{1}{2}\right]\Delta r.\cos\left[(i-n)\Delta\theta\right] - m\Delta t\right) \qquad (6\text{-}10)$$

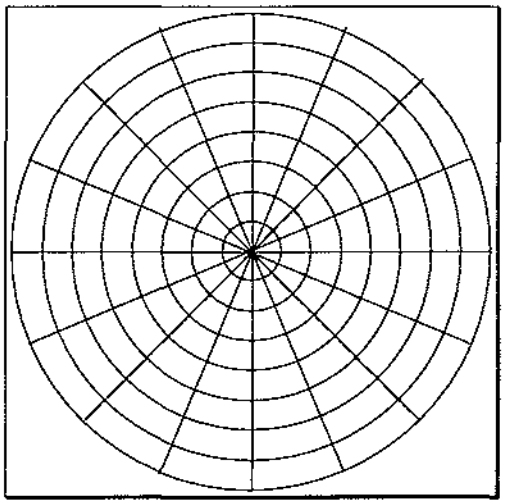


Figure 6-6. Reconstruction area discretised using Equi-radial\angular.

A major feature of this type of geometry is the apparent increase in resolution toward the centre (see following section on aliasing). This may be inappropriate for applications where the information is evenly distributed over the reconstruction area. Certain applications are well suited to this geometry. An example is the detection of the defect core within logs [Benson-Cooper, D.M. et al, 1982] where the majority of the defects are found within the central region of the log.

A simple extension of this pixel geometry involves changing the radial increment such that all pixels have equal area.

6.2.1.4 Aliasing

The discussion of reconstruction area discretisation would be incomplete without considering aliasing. During the backprojection process the pixels are effectively sampling the equi-spaced filtered X-ray data. The X-ray path width is defined by the hardware, and remains constant. Therefore the amount of aliasing that is likely to occur is dependent on the discretisation and interpolation methods selected.

According to the Nyquist criterion the signal must be sampled at twice its highest frequency if aliasing is to be avoided. This means that the maximum spatial dimension allowable for any pixel (without aliasing occurring) will be half the ray path width.

Consider the Equi-radial/angular reconstruction area in figure 6-7, with a single projection superimposed (although invisible) in the vertical direction.

The dots on the lines running horizontal represent the spatial sampling of the projection data by the pixels at different horizontal positions. The top half of the reconstruction area assumes that the interpolation method chosen takes into account all rays which intersect with the pixel (ie. linear, spline, or area), and the bottom half of the reconstruction area assumes that the interpolation method is nearest neighbour.

This simple example shows that the projection is sampled at a higher frequency in the centre of the reconstruction area (which is expected due to the increased number of pixels). Therefore aliasing is likely to increase with radial position when the reconstruction area is discretised using equi-radial/angular discretisation.

The sampling frequency is poorer where nearest neighbour interpolation is used (for example 2a has 6 samples whereas 2b has 4 samples), indicating an increase in the chance of aliasing occurring, and therefore a likely reduction in the resulting image quality. This demonstrates a trade-off between reduced computation due to simplistic interpolation methods and image quality.

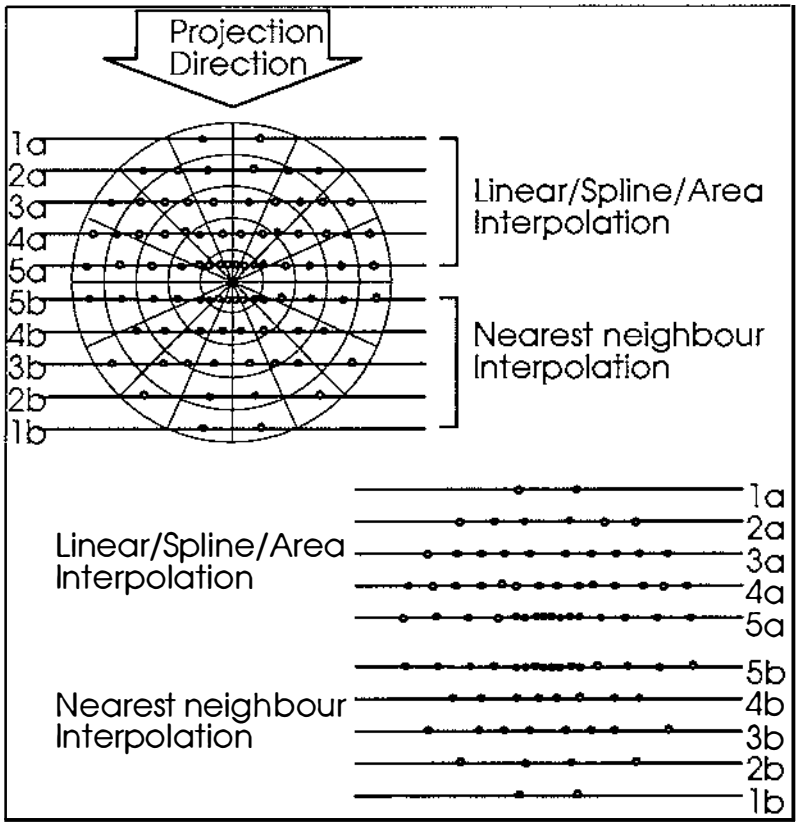


Figure 6-7. Spatial sampling of projection by pixels using Equi-radial/angular discretisation.

A similar exercise can be carried out using equi-radial best-fit angular discretisation, and the standard cartesian discretisation. The sampling frequency does not decrease with radial position with either of these methods, so the aliasing will be relatively consistent throughout the reconstruction area. However the trade-off between computation and image quality caused by the choice of interpolation will remain.

Note that the projection data may already contain aliased artefacts due to the discretisation of the physical space by the projections. It is common for the projection data to be filtered (low pass) by the data acquisition system to reduce the affect of aliasing in the projection data [Webb, S. 1988].

6.2.1.5 Reconstruction Interpolation

As mentioned previously, interpolation is required because the backprojection of any ray value will in general not coincide exactly with the discrete pixels. A simple method of interpolation is nearest neighbour. Using this method the interpolating mapping function is defined as

$$\delta_I(t) = \delta\left(\mathrm{rnd}\left[\frac{t}{\Delta t} \right] \right) \tag{6-11}$$

where rnd indicates the mathematical rounding operation.

A second simple method is linear interpolation, which is defined by

$$\delta_I(t) = \begin{cases} \left(1 - \dfrac{|t|}{\Delta t}\right), & |t| < \Delta t \\ 0, & |t| \geq \Delta t \end{cases} \tag{6-12}$$

Both of these interpolation methods are easy to implement and require little computation. However they are only approximations to the more exact interpolation method which uses the intersection area between the pixel of interest and each ray which has the following form

$$\delta_I(J,m) = \frac{\text{Intersection area of pixel J and ray } (m\Delta t)}{\text{Total area of pixel J}} \tag{6-13}$$

Note that J is a reference to the pixel whose coordinates depend upon the geometry used, as will the complexity of the intersection area calculation. It is unlikely that area interpolation would be calculated at run-time for any

high-speed reconstruction system because of the high computational cost. Rather it would need to be pre-calculated and stored for subsequent retrieval at run-time, increasing the storage requirements.

6.2.1.6 Posterior Interpolation

Certain applications do not require a visual display to determine the required information [Wallace, G. 1991]. However sometimes it is necessary to display an image, for example medical applications where the diagnosis is complex and requires human intervention. In such situations interpolation is required from the coordinate system used in the algorithm to a rectangular coordinate system for display on a monitor.

Once again a number of possible interpolation schemes exist, (eg. nearest neighbour, linear, spline) and those which require the least computation will in general introduce the largest aliasing effects. However for the purposes of high-speed implementation it is common to use nearest neighbour interpolation. To obtain the cartesian image f_c from the generated image f

$$f_c(x,y) = f\left(\text{round}[R.\cos\Phi], \text{round}[R.\sin\Phi]\right) \qquad (6\text{-}15)$$

where $R = (r\text{-}\tfrac{1}{2})\Delta r$ and $\Phi = j\Delta\theta + g\Delta\phi'$ for equi-radial best-fit area,

$R = (r\text{-}\tfrac{1}{2})\Delta r$ and $\Phi = i\Delta\theta$ for equi-radial/angular.

A literature search performed using the international literature databases confirmed the originality of the modified backprojection concept. The two databases queried were the engineering database (INSPEC) and the biomedical database (EMBASE), and a list of the search criteria and resulting document references is printed in appendix G.

6.2.2 Reduction of the Convolution Kernel

The mathematics of the reconstruction process specify a convolution kernel with twice the number of elements as there are ray values in a projection. However it is desirable for computational purposes to have a convolution kernel of only a few coefficients.

The following section shows how the kernel coefficients of the Ramachandran-Lakshminarayanan filter [Webb, S. 1988 pp115] transform

to a Chebychev economisation series [Notes from paper 60.201 Analysis, Dept. of Statistics, Massey University, New Zealand, 1986], and are therefore well suited to truncation to yield a convolution kernel of reduced size.

6.2.2.1 Optimum Kernel Coefficients Using Chebychev Minimisation

Approximating the continuous convolution function *h* from equation 4-24, the Ramachandran-Lakshminarayanan discrete convolution kernel is shown to be [Webb, S. 1988]

$$Q(t) = \tfrac{1}{4}P(t) - \frac{1}{\pi^2}\sum_{\substack{n \\ (t-n)\ \text{odd}}} \frac{P(t-n)}{(t-n)^2} \qquad (6\text{-}16)$$

where t is an integer indicating the ray number within the projection. This is only an approximation to the continuous convolution function since the continuous filter is of infinite extent in the frequency domain. Figure 6-7 shows the function in the time (space) domain.

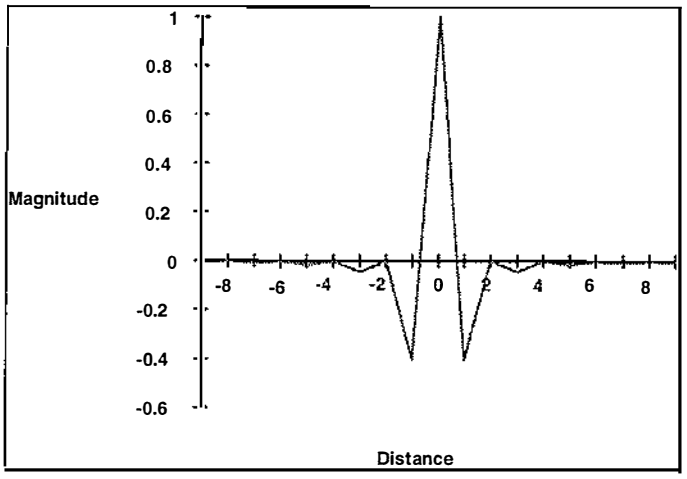


Figure 6-7. Ramachandran-Lakshminarayanan convolution function in the space domain.

To transform equation 6-16 into the discrete frequency domain it is first converted to the Z-domain (time delay) where

$$Q(Z) = \tfrac{1}{4}P(Z) - \frac{1}{\pi}\left(\ldots + \frac{P(Z)Z^{-5}}{25} + \frac{P(Z)Z^{-3}}{9} + \frac{P(Z)Z^{-1}}{1} + \frac{P(Z)Z}{1} + \frac{P(Z)Z^{3}}{9} + \ldots\right) \qquad (6\text{-}17)$$

Dividing through by P gives the Z-domain transfer function

$$\frac{Q(Z)}{P(Z)} = \frac{1}{4} - \ldots - \frac{Z^{-5}}{25\pi^2} - \frac{Z^{-3}}{9\pi^2} - \frac{Z^{-1}}{\pi^2} - \frac{Z^{1}}{\pi^2} - \frac{Z^{3}}{9\pi^2} - \frac{Z^{5}}{25\pi^2} - \ldots \qquad (6\text{-}18)$$

Substituting the frequency expression $Z=cos(wt)+j.sin(wt)$ gives

$$\begin{aligned}\frac{Q(Z)}{P(Z)} &= \frac{1}{4} - \ldots - \frac{\cos(-3wt)+j.\sin(-3wt)}{9\pi^2} - \frac{\cos(-wt)+j.\sin(-wt)}{\pi^2} \\ &\quad - \frac{\cos(wt)+j.\sin(wt)}{\pi^2} - \frac{\cos(3wt)+j.\sin(3wt)}{9\pi^2} - \ldots \\ &= \frac{1}{4} - \left(\frac{\cos(-wt)}{\pi^2} + \frac{\cos(wt)}{\pi^2}\right) - \left(\frac{j.\sin(-wt)}{\pi^2} + \frac{j.\sin(wt)}{\pi^2}\right) \\ &\quad - \left(\frac{\cos(-3wt)}{9\pi^2} + \frac{\cos(3wt)}{9\pi^2}\right) - \left(\frac{j.\sin(-3wt)}{9\pi^2} + \frac{j.\sin(3wt)}{9\pi^2}\right) - \ldots \end{aligned} \qquad (6\text{-}20)$$

The sine terms cancel and the cosine terms combine (since sine is an odd function and cosine is an even function) giving

$$\frac{Q(t)}{P(t)} = \frac{1}{4} - \frac{2.\cos(wt)}{\pi^2} - \frac{2.\cos(3wt)}{9\pi^2} - \frac{2.\cos(5wt)}{25\pi^2} - \ldots \qquad (6\text{-}21)$$

The coefficients of the discrete frequency terms cos(*nwt*) n=1,3,5,... are characteristic of those in a Chebychev polynomial series [Paper 60.201 Analysis course notes, Department of Mathematics and Statistics, 1986]. Characteristic of this series is the rapid descent of the coefficient magnitudes, which indicate the approximate error incurred by truncating from that point. The first 10 coefficients of the equation 6-21 to four decimal places are

0.2500, -0.2026, -0.0225, -0.0081, -0.0041,
-0.0025, -0.0017, -0.0012, -0.0009, -0.0007

6.2.2.2 Results and Discussions

A graph showing the maximum absolute error in the frequency domain incurred using different numbers of terms is shown in figure 6-8. Note that the maximum error incurred when truncated to three terms is only about 5 percent.

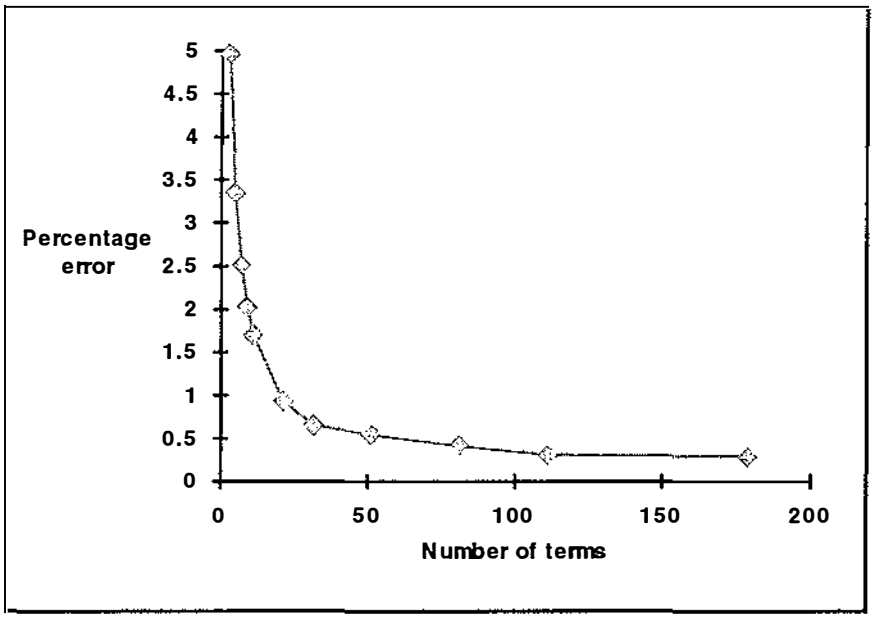


Figure 6-8. Graph of error incurred by truncating the discrete frequency series.

A phantom image representing a log cross-section was created on a cartesian grid containing 240 by 240 pixels. The phantom is shown in figure 6-9 and the intensity values are shown in figure 6-10.

Simulation X-ray data were computer generated using equation 3-2 giving 120 projections of parallel-ray data, where each projection contained 240 rays. (See appendix A for the computer program used to generate the projection data).

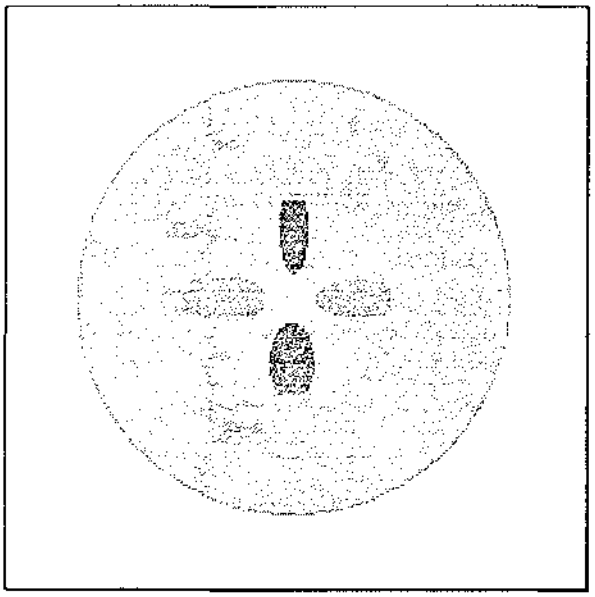


Figure 6-9. Phantom of log cross-section showing knots, sap pockets, and heartwood.

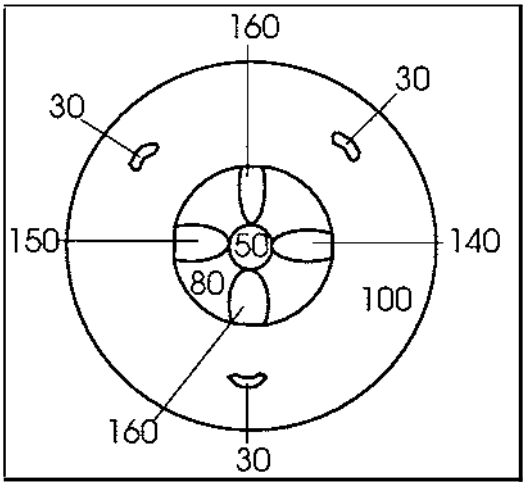


Figure 6-10. Relative density values of log phantom.

To examine the effect of reducing the convolution kernel on the reconstructed image, projection data were filtered using kernels of size 3, 7, 15, 31 and non-truncated. The computer program used to convolve the projection data is printed in appendix C.

Reconstructed images showing the effect of using a series truncated to 3, 7, 15, and 31 terms are shown in figures 6-11 to 6-14 respectively. For comparison figure 6-15 shows a reconstruction using the non-truncated series. The interpolation method used for the reconstructions was nearest neighbour. Intensity plots from the central line in the vertical axis are shown in figures 6-16 to 6-20 for the images in figures 6-11 to 6-15.

The images produced using the truncated convolution kernel show signs of blooming (excess brightness causing faded contrast) with the 3-term truncation shown in figure 6-11 being the most severe. However the high contrast features are clearly visible in all images. The intensity plots in figures 6-16 and 6-20 show both the blooming, and the high contrast changes clearly. The low contrast detail is hard to distinguish from figure 6-11 (three term kernel) although this improves as the number of terms is increased (figures 6-12 to 6-15). The appearance of rings in the images (noticable in figures 6-11 and 6-12) is due to the limitation of the printer, ie. when the intensity value slowly and evenly varies, the effects of digitisation using 256 grey-scales can be seen.

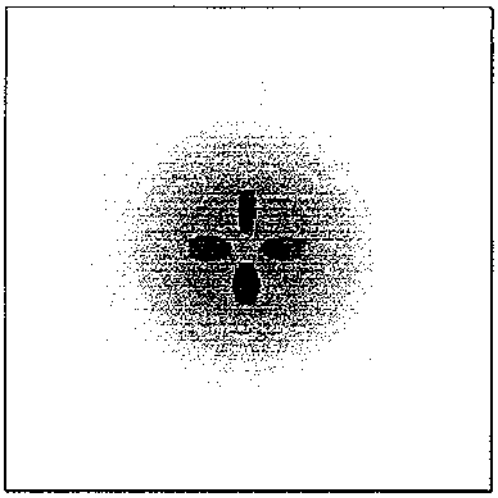


Figure 6-11. Reconstruction created with convolution kernel truncated to 3 terms

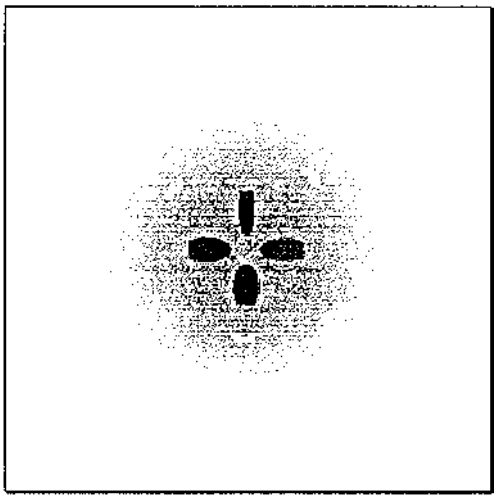


Figure 6-12. Reconstruction created with convolution kernel truncated to 7 terms

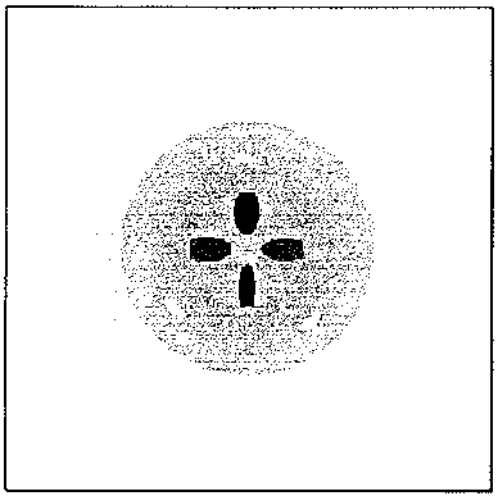


Figure 6-13. Reconstruction created with convolution kernel truncated to 15 terms

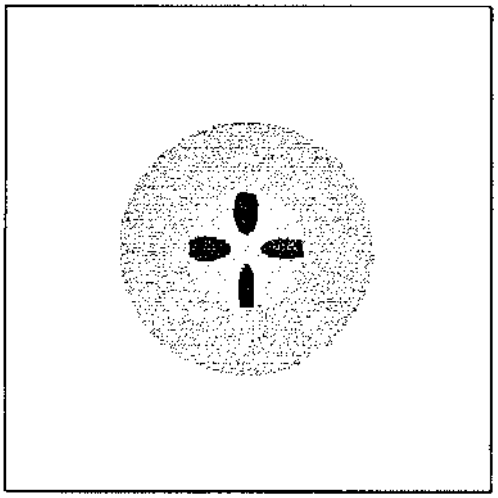


Figure 6-14. Reconstruction created with convolution kernel truncated to 31 terms

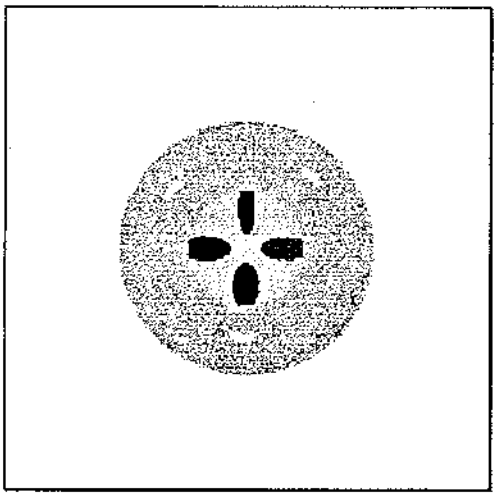


Figure 6-15. Reconstruction created with full convolution kernel.

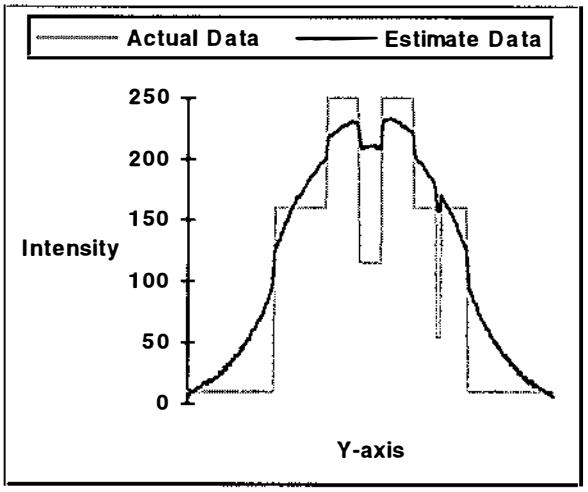


Figure 6-16. Intensity plot of image shown in figure 6-11 with 3 term convolution kernel. Line shown is centre of vertical axis.

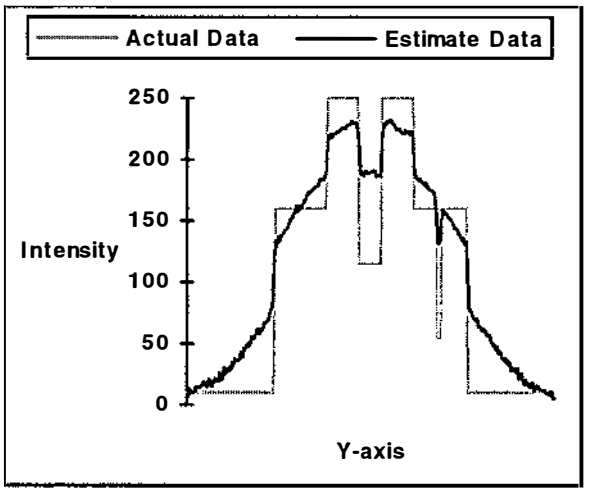


Figure 6-17. Intensity plot of image shown in figure 6-12 with 7 term convolution kernel. Line shown is centre of vertical axis.

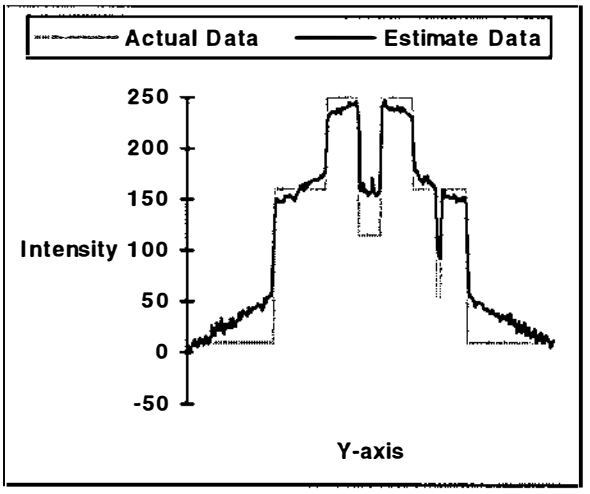


Figure 6-18. Intensity plot of image shown in figure 6-13 with 15 term convolution kernel. Line shown is centre of vertical axis.

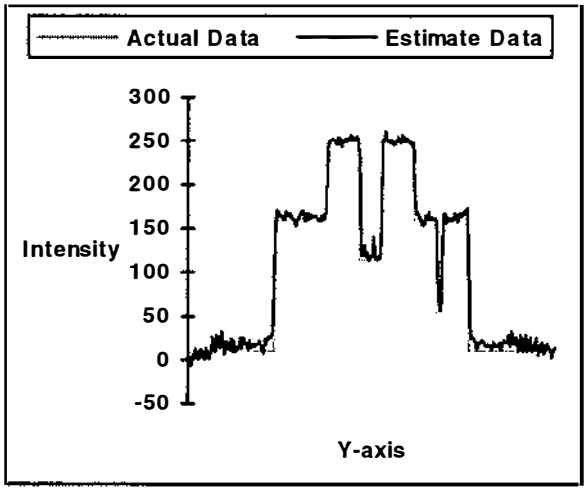


Figure 6-19. Intensity plot of image shown in figure 6-14 with 31 term convolution kernel. Line shown is centre of vertical axis.

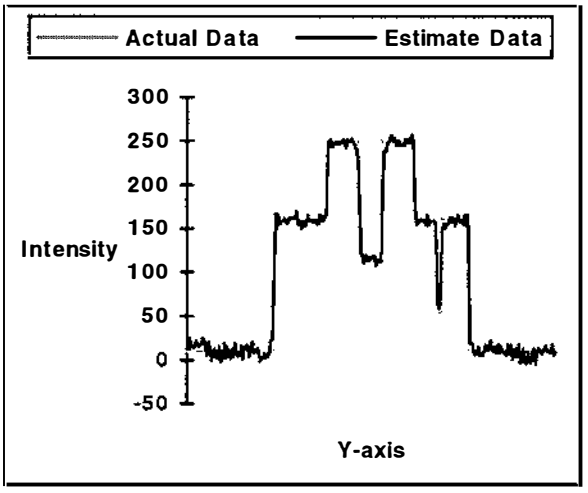


Figure 6-20. Intensity plot of image shown in figure 6-15 with full convolution kernel. Line shown is centre of vertical axis.

To examine the effect of noise on the truncated convolution kernel 5% uniform noise was added to the projection data used in the above reconstructions. The reconstructions using the noisy data are shown in figures 6-21 to 6-25, and the intensity graphs from the central vertical axis are shown in figures 6-26 to 6-30.

Table 6-1 shows the average absolute difference between the actual and estimate intensity values shown in figures 6-26 to 6-30. The reconstructions using the larger convolution kernel appear to suffer most from the

introduction of noise. This is because reducing the terms in the convolution kernel causes blurring (smoothing) which compensates for the introduction of noise. However this smoothing results in poorer overall image reconstruction quality which has been seen in figures 6-11 to 6-15.

Table 6-1. Average absolute difference between the actual and estimate intensity values for figures 6-16 to 6-20 and 6-26 to 6-30. The units are pixel intensity values relative to 256.

	Projection Data	
	Standard	Noisy
3-term kernel	31.61	34.46
7 -term kernel	27.02	33.54
15-term kernel	18.42	30.76
31-term kernel	7.65	29.10
Full kernel	6.22	27.8

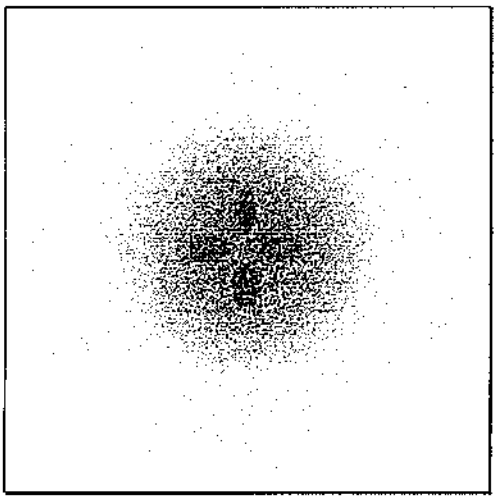


Figure 6-21. Reconstruction created with convolution kernel truncated to 3 terms and noisy projection data.

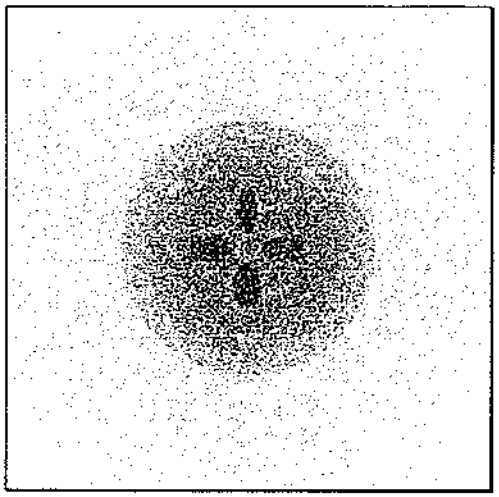


Figure 6-22. Reconstruction created with convolution kernel truncated to 7 terms and noisy projection data.

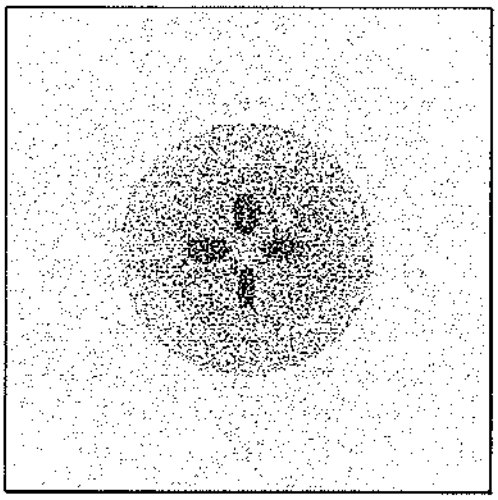


Figure 6-23. Reconstruction created with convolution kernel truncated to 15 terms and noisy projection data.

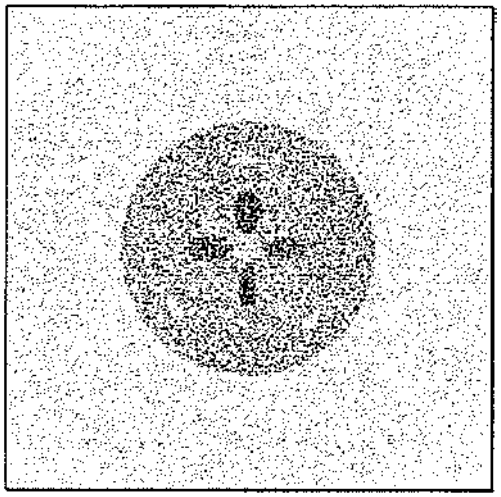


Figure 6-24. Reconstruction created with convolution kernel truncated to 31 terms and noisy projection data.

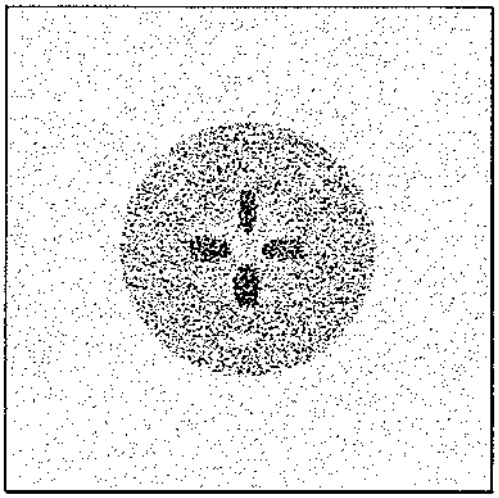


Figure 6-25. Reconstruction created with full convolution kernel and noisy projection data.

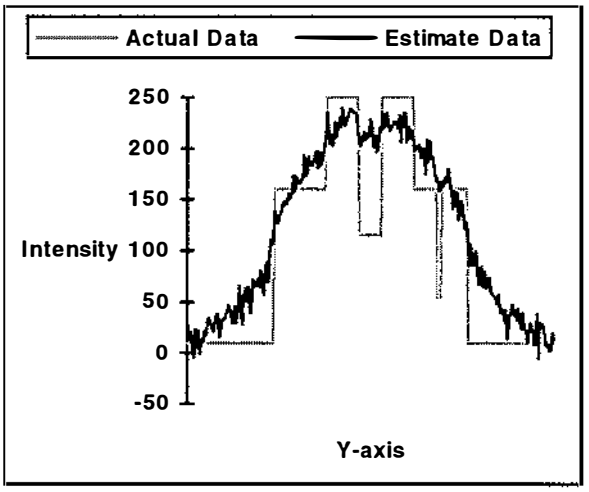


Figure 6-26. Intensity plot of image shown in figure 6-21 with 3 term convolution kernel and noisy projection data. Line shown is centre of vertical axis.

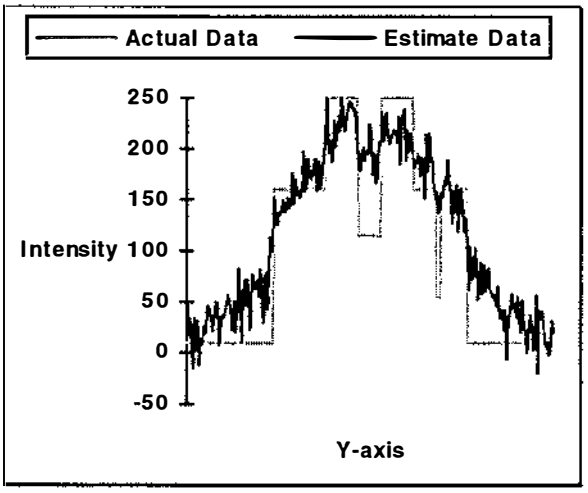


Figure 6-27. Intensity plot of image shown in figure 6-22 with 7 term convolution kernel and noisy projection data. Line shown is centre of vertical axis.

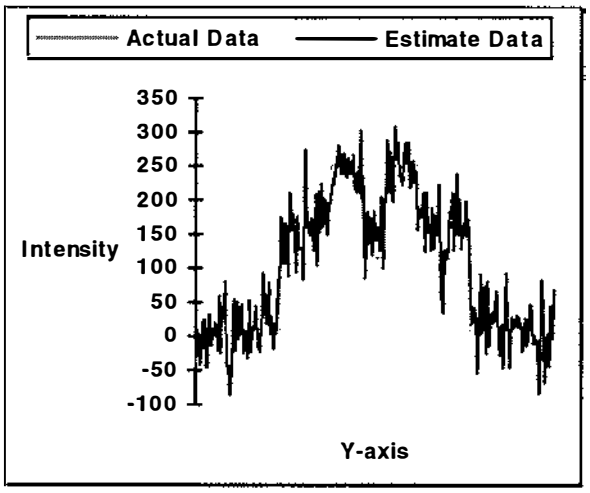


Figure 6-28. Intensity plot of image shown in figure 6-23 with 15 term convolution kernel and noisy projection data. Line shown is centre of vertical axis.

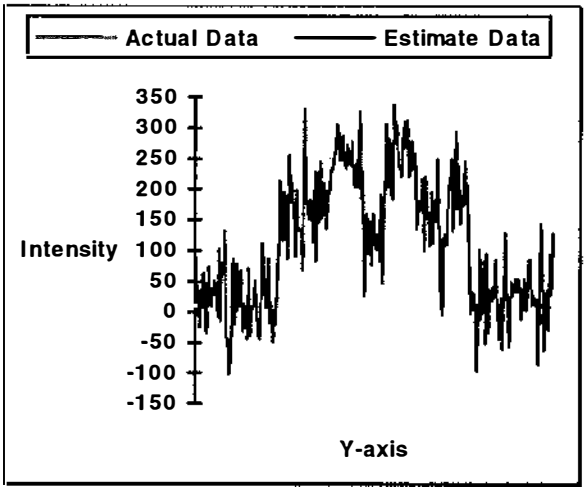


Figure 6-29. Intensity plot of image shown in figure 6-24 with 31 term convolution kernel and noisy projection data. Line shown is centre of vertical axis.

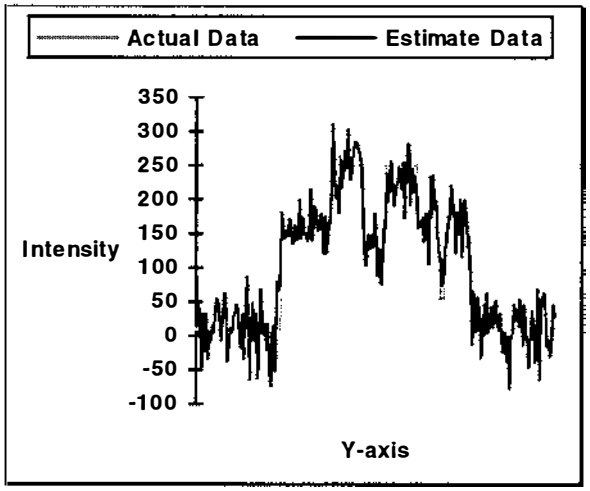


Figure 6-30. Intensity plot of image shown in figure 6-25 with full convolution kernel and noisy projection data. Line shown is centre of vertical axis.

Chapter 7

7 IMPLEMENTATIONS OF MODIFIED BACKPROJECTION

The previous chapter showed that the computation required by the modified backprojection algorithm in its simplest form is only addition. The number of hardware implementations possible is therefore large. This chapter shows a simple software implementation written using the PASCAL programming language, and uses this to compare the performance of the modified algorithm with the standard convolution backprojection algorithm.

A range of hardware implementations are discussed. The first is an inexpensive Gate Array system which requires little other than a field Programmable Gate Array component, and a number of RAM components. The second system uses the same schematic as the above but implements it using discrete components. The third system uses a systolic array architecture, and is discussed in relation to highly parallel processing systems.

The system described using the field Programmable Gate Array was implemented as far as the chip design (partitioning, placement, and routing), however a full working implementation was not produced because use of the target X-ray scanner was not available. Without the X-ray scanner for generating X-ray data at the correct speeds little would be gained by completing the hardware implementation.

The final section of this chapter is a conceptual model of an analogue system implementing the modified backprojection. Although a number of problems exist with the analogue implementation it is included because it was the stimulus for the modified backprojection concept.

7.1 SOFTWARE IMPLEMENTATION/SIMULATION

The main advantage of using the modified backprojection method is that it enables the use of high-speed look-up methods. The modified

backprojection equation (equation 6-8) can be rewritten to include the look-up table instead of the computational components.

Recognising that the rotation of the projection angle [(j-n)Δθ] is relative to the reconstruction area, the projection angle is kept constant and the reconstruction area is rotated. Equation 6-8 becomes

$$f\left(r_d, j, \phi_d'\right) = \sum_{n=0}^{N} f_n'\left(r_d, \mathrm{mod}\left[(j+n), N\right], \phi_d'\right) \tag{7-1}$$

where f' is defined by

$$f_n'(r_d, j', \phi_d') = \frac{1}{2} \sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t)\,\delta_I\left(r_d.\cos\left[j'\Delta\theta + \phi_d'\right] - m\Delta t\right) \tag{7-2}$$

The modulo arithmetic is required to ensure that the angle j' is within 0-2π. This operation was performed by the cosine function in equation 6-8.

Let the function $\zeta(r_d, j, \phi')$ return the value $r_d.\cos[j\Delta\theta+\phi_d']$ associated with any pixel (r_d, j, ϕ_d') for the first projection (n=0). Equation 7-2 can now be written as

$$f_n'(r_d, j', \phi_d') = \frac{1}{2} \sum_{m=-M}^{M} Q(n\Delta\theta, m\Delta t)\,\delta_I\left[\zeta\left(r_d, j', \phi_d'\right) - m\Delta t\right] \tag{7-3}$$

In this form the computation is greatly reduced, since the major computational component is pre-calculated and stored for look-up. Furthermore, by using nearest-neighbour interpolation for δ_I the only mathematical operation required is addition.

The reconstruction equations all describe the backprojection process on a pixel by pixel basis. In other words, the equations show that the value of a certain pixel can be calculated by summing all of the rays which pass through that pixel. However, to achieve high-speed reconstruction it is desirable to process the projection data on a ray-by-ray basis. This allows the reconstruction process to begin as soon as the first data value arrives.

Let the inverse of the function $\zeta(r_d, j, \phi_d')$ be the function $\psi(i,m)$ which returns a list of *i* pixels which intersect with the *m*th ray in the first projection (n=0), and a weighting indicating the intersection area between

the pixel and the ray. This extra parameter was implicit in the value of the inverse function $\zeta(r_d, j, \phi_d')$.

Let the *i*th pixel parameters (r, j, ϕ_d') of the function $\psi(i,m)$ be labelled ψ_r^i, ψ_j^i and ψ_ϕ^i respectively, and the intersection weighting as ψ_w^i. Incorporating the function $\psi(i,m)$ the reconstruction process can be described iteratively as follows:

Initial conditions

$$f^0\left(r_d, j, \phi_d'\right) = \underline{0} \tag{7-4}$$

for all projections *n*
  for all rays *m* in projection *n*
    for all pixels *i* associated with ray *m*

$$f^n\left(\psi_r^i, \psi_j^i, \psi_\phi^i\right) = f^{n-1}\left(\psi_r^i, \mathrm{mod}\left[\left(\psi_j^i + n\right)\Delta\theta, 2\pi\right], \psi_\phi^i\right) + \psi_w^i Q(n,m) \tag{7-5}$$

where the weighting factors are in the range $0 < \psi_w^i \leq 1$.

In practice the image function *f* will be stored and incremented using a digital computer. Consider the pixels stored as a rectangular array where the pixels from each segment are stored in some arbitrary but consistent fashion in a single row of the array. The x-dimension of the array will be the number of pixels per segment, and the y-dimension of the array will be the number of segments. Let these quantities be known as X and Y respectively, and let the pixels be referenced by (x,y) where the segment offset ψ_j^i becomes ψ_y^i and the radial and angular components within each segment ψ_ϕ^i ψ_r^i, are replaced by the single parameter ψ_X^i

Rotating the reconstruction area is achieved by circular shifting the array of pixels in the y-direction 1 row, since each row increment represents the segment angle $\Delta\theta$. Incorporating these changes into the algorithm described by equations 7-4 and 7-5 gives

Initial conditions

$$f^0(x,y) = \underline{0} \tag{7-6}$$

for all projections *n*
  for all rays *m* in projection *n*
    for all pixels *i* associated with ray *m*

$$f^n\left(\psi_X^i, \psi_Y^i\right) = f^{n-1}\left(\psi_X^i, \mathrm{mod}\left[\left(\psi_Y^i + n\right), X \times Y\right]\right) + \psi_w^i Q(n,m) \tag{7-7}$$

where $x = 1,2,3...X$
$y = 1,2,3...Y.$

The computation required for the backprojection process using nearest neighbour interpolation with the standard and modified methods without look-up tables can be seen from equations 6-3, 6-9, and 6-10, and is shown in table 7-1

Table 7-1. Computational requirements for different systems.

	Multiplication	Addition	cos/sin	round/ modulo
Standard	3×N×X×Y	2×N×X×Y	2×N×X×Y	N×X×Y
Equi-radial/angular	3×N×X×Y	2×N×X×Y	N×X×Y	2×N×X×Y
Equi-radial best-fit area	3×N×X×Y	3×N×X×Y	N×X×Y	2×N×X×Y
All methods using table look-up		N×X×Y		N×X×Y
Posterior interpolation for modified algorithms	2×N		2×N	2×N

Clearly the computational requirements are comparable for each of the methods, even when the posterior interpolation is considered. However the modified methods are best suited for implementation in the form of equations 7-6 and 7-7.

For implementation the function $\psi(i,m)$ would be pre-calculated and stored in the form of a look-up table indexed by i and m.

The above algorithm does not specify the pixel geometry or the interpolation method. Table 7-2 shows examples of the number of entries required by the look-up table for combinations of discretisation and interpolation. The results in the Area column were obtained experimentally using an image with 240 sectors and projection data containing 240 projections of 240 rays each. The number of look-up values depends on the number of projections, rays, sectors, pixels, and distribution of pixels within the sectors. However the values shown in table 7-2 are representative of a typical system.

When nearest-neighbour interpolation is used the weighting values ψ_w^i become unity, and can be ignored.

Table 7-2. Number of values required by look-up tables for different systems.

	Nearest-neighbour	Area
Standard	X×Y×N	X×Y×N×1.8
equi-radial/angular	X×Y	X×Y×2.4
equi-radial best-fit area	X×Y	X×Y×1.8

From table 7-2 it is clear that the modified backprojection method offers a significant reduction in the size of the look-up table, and if nearest neighbour interpolation is used that the number of weights required is also reduced by a factor of N.

The Pascal code used to perform equation 7-7 using nearest-neighbour interpolation is shown in figure 7-1. The two lines in bold typeface are the main processing instructions, at the center of the looped structure, showing (to those farmiliar with Pascal) the simplicity of the algorithm. The components of the program and their function are described below.

The array *AddressArray* is the lookup-table $\psi(i,m)$, and *index* is used to indicate the current look-up table value. Offset is the number of pixels in one segment X, and the convolved data are stored in the array pointed to by DataPtr.

The boolean AND operation provides the modular arithmetic for rotating the reconstruction area (Image^). The reconstruction area used in figure 7-1 contained 128 sectors, each containing 64 pixels, and the modulo arithmetic is based on the total number of pixels (128×64=8192). The Inc procedure adds the ray values to the appropriate pixels in the Image array.

The pixel geometry is not specified in the code and is dependent on the precalculated pixel values stored in the look-up table.

To modify the code for more elaborate interpolation schemes the added value would require a weighting from the look-up table.

```
for C0 := 0 to Views-1 do                                    {for all views}
begin
  index := 0;                                  {reset index into mapping function}
  for C1 := 0 to Rays-1 do                                   {for all rays in each view}
  begin
    Val:= round(DataPtr^[C0*views+C1]/RayLength[C1]);   {normalise Data}
    for C2 := index to index+NumAddress[C1]-1 do        {for all pixels in ray}
    begin
      Address:=(AddressArray[C2]+offset) AND 8191;
      Inc(Image^[Address],Val);
    end;
    Inc(index,NumAddress[C1]);
  end;
  offset := offset+Depth;
end;
```

Figure 7-1. Main section of Pascal code used to perform modified image reconstruction.

The following sections compare the performance of the modified backprojection algorithm with the standard backprojection algorithm. Pixel geometry combinations include both the equi-radial best-fit angular and equi-radial/angular modified geometries and the cartesian standard geometry. Interpolation methods compared are nearest-neighbour and area interpolation.

The area interpolation used by the modified methods has been approximated to within 1 percent because the pixels-ray intersection areas cannot be expressed using standard geometric interpretation.

The approximation is calculated by defining 100 points within each pixel separated by equal radial and angular increments. The ray intersecting each of the 100 points is then calculated akin to nearest neighbour interpolation, giving 100 values indicating the ray number intersecting with each point. By summing the points with equal values, the rays intersecting the pixel are determined and the area of intersection is provided by the summations to within 1 percent.

7.1.1 Standard Tests - Results and Discussion

The log phantom described in section 6.2.2.2 was reconstructed using all combinations of pixel geometry and interpolation mentioned above (ie. six combinations). The projection data were convolved using the full Ramachandran-Lakshminarayanan kernel for the following tests. The computer program used to convolve the data is printed in appendix C, and the reconstruction programs for the standard, equi-radial/angular, and equi-

radial best-fit area backprojection are printed in appendices D, E, and F respectively.

Figures 7-2, 7-3, and 7-4 show the results obtained using the cartesian, equi-radial/angular, and equi-radial best-fit angular pixel geometries respectively using nearest-neighbour interpolation.

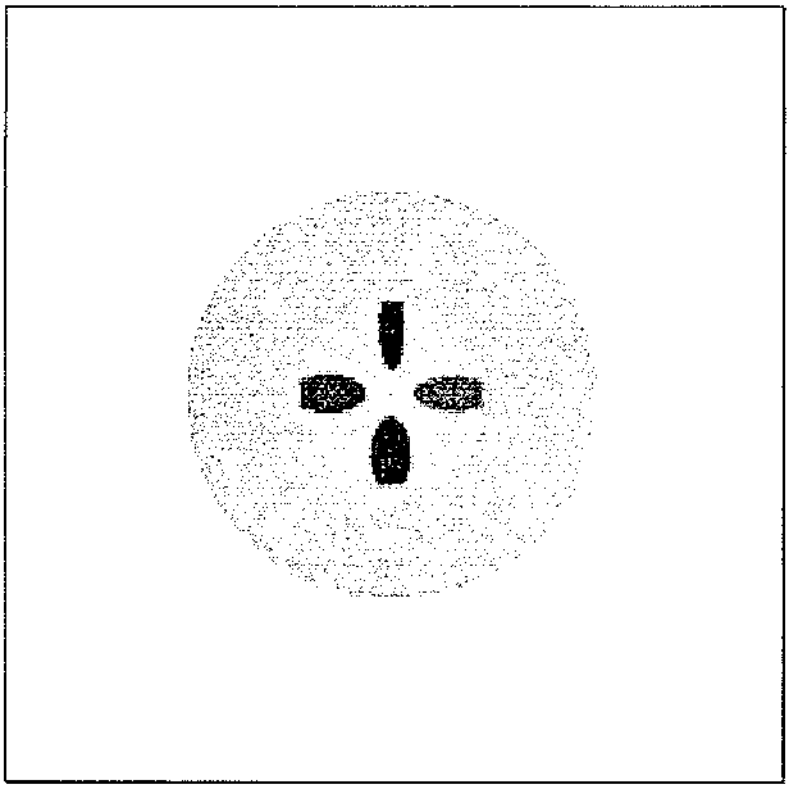


Figure 7-2. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation.

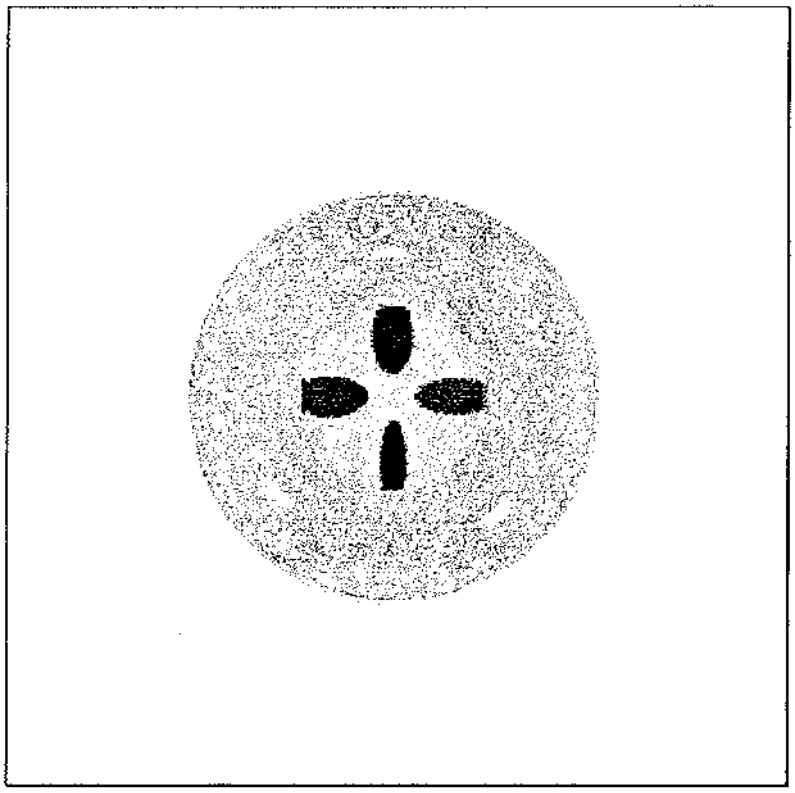


Figure 7-3. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation.

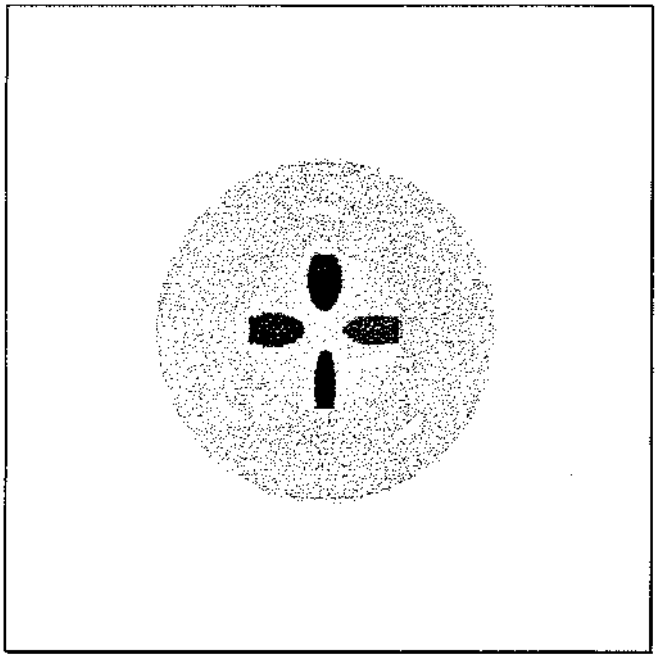


Figure 7-4. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation.

Figures 7-5, 7-6, and 7-7 show the results obtained using the cartesian, equi-radial/angular, and equi-radial best-fit angular pixel geometries respectively using area and approximate area interpolation.

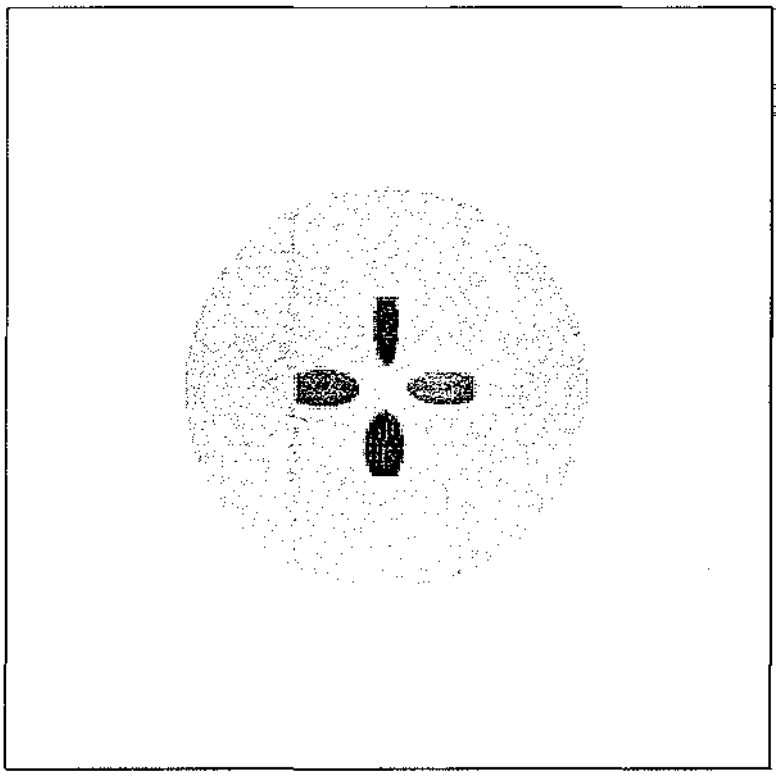


Figure 7-5. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with area interpolation.

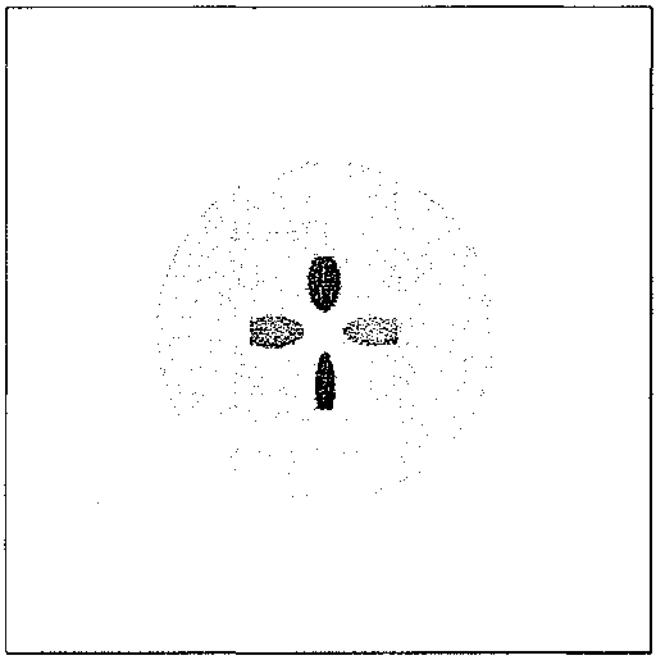


Figure 7-6. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation.

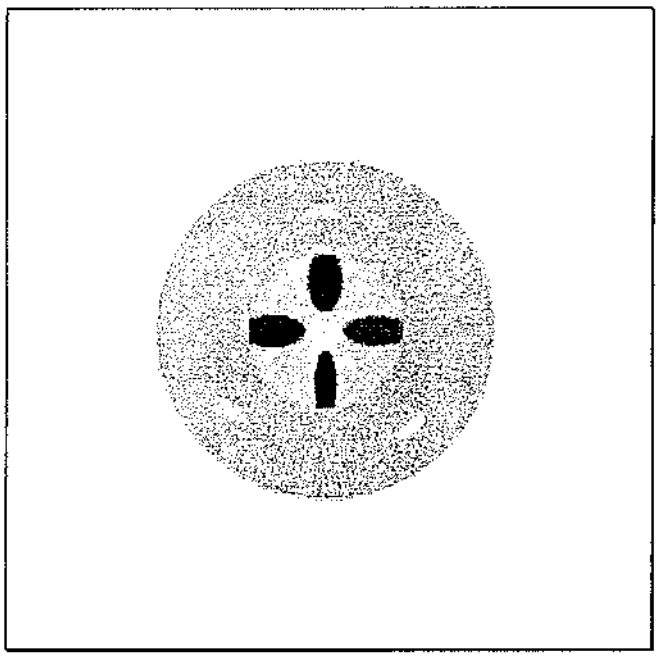


Figure 7-7. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation.

All of the output images give results which are clearly identifiable with the phantom. However the images generated using nearest neighbour interpolation (figures 7-2, 7-3 and 7-4) show slightly greater deviation from the phantom than those generated using area interpolation.

Figure 7-3 was produced by the equi-radial/angular modified backprojection method using nearest neighbour interpolation. This image shows rings of deviation at varying radial positions. These are due to the repetitiveness of the mapping function which introduces a bias toward a given ray position which will be repeated for every projection causing rings to occur.

The ring phenomena does not happen for the other methods because the bias introduced by the use of nearest neighbour interpolation is more evenly distributed throughout all ray positions. However figures 7-2 and 7-5 show a speckled pattern around the perimeter known as Moire artefacts [Jain A. K. 1989] and are caused by the finite sampling of the object being reconstructed.

It is commonly accepted that the true reconstruction area, being circular by nature of the physical measurement system, lies within the square reconstruction area when using cartesian reconstruction methods [Cho, Z. H. et al. 1990]. Therefore any information contained in the corners of the

cartesian reconstruction area can be taken as having lower precision, since only a portion of the projections contribute to the values in those areas. Figures 7-2 and 7-5 generated using cartesian reconstruction, contain the effects of this phenomenon in the corners.

The images created using the sector based reconstruction method have had to be re-sampled in order to display them on a cartesian monitor (discussed in section 6.2.1.6). Therefore it is useful to compare the actual pixel values, which represent the density values, of the images along a line of interest. The line of the *y-axis* was chosen because the pixels are equally spaced and equal in number for all output images.

All output images are automatically scaled to occupy the full range of allowable output values. The values output from the reconstruction programs are therefore relative rather then absolute and it is necessary for comparison to re-scale the output line data to provide a best fit with the actual phantom line.

Figures 7-8 to 7-13 show the central vertical reconstructed line values compared to the actual phantom line values for the images in figures 7-2 to 7-7. The average absolute difference between actual and estimate intensity values corresponding to figures 7-8 to 7-13 is shown in table 7-3. All of the results show that the reconstructed images have adequately determined the relative densities of the phantom. Table 7-3 indicates that the equi-radial/angular method produced better results then the other two methods. However the values are all very close, and it would be unfair to conclude that the equi-radial/angular method performs best based on these results.

Table 7-3 Average of the absolute differenct between actual and estimaged images for figures 7.8 to 7.13. The units are pixel intensity values relative to 256.

	Nearest Neighbour	Area
Equi-radial/angular	6.96	4.75
Equi-radial best-fit angular	7.66	5.89
Standard Backprojection	7.54	6.08

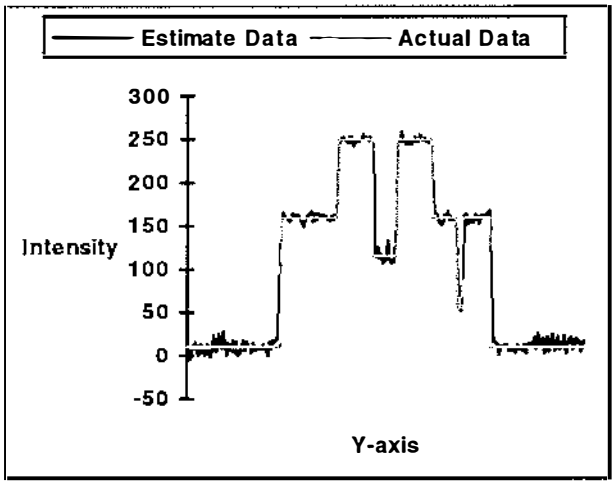


Figure 7-8. Standard Method with nearest-neighbour interpolation.

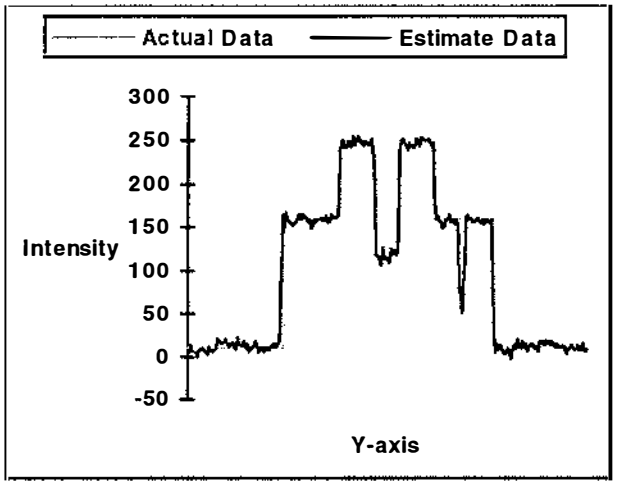


Figure 7-9. Equi-radial/angular with nearest-neighbour interpolation

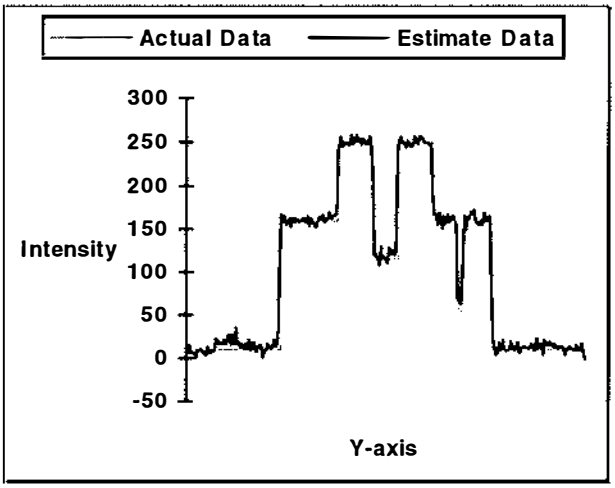


Figure 7-10. Equi-radial Best-fit area with nearest-neighbour interpolation.

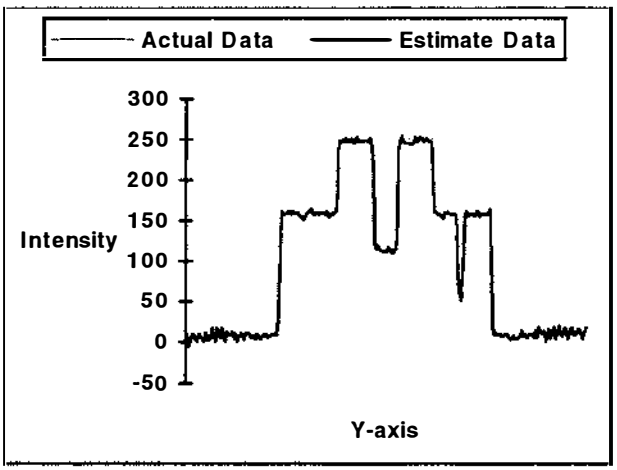


Figure 7-11. Standard Method with area interpolation.

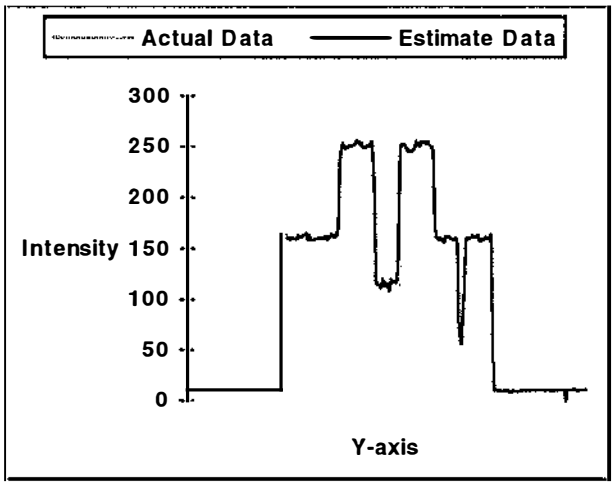


Figure 7-12. Equi-radial/angular with area interpolation

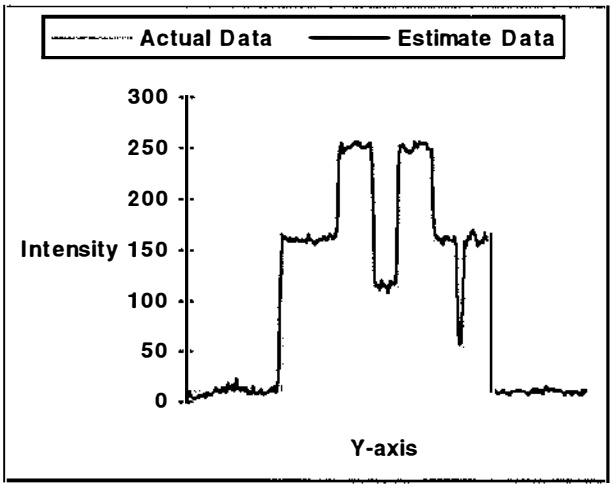


Figure 7-13. Equi-radial Best-fit area with area interpolation.

7.1.2. Noise Simulation Tests

The modifications to the standard backprojection algorithm do not change the way backprojection is achieved, only the platform the data is backprojected on to. Therefore the noise in the projection data should not have a greater or lesser effect on the reconstructed images when using the modified backprojection algorithm.

To confirm this three different levels of noise were added to the projection data. Uniform noise of plus or minus 1, 5, and 10 percent of the dynamic range of the projection data were added to the projection data. The data were then convolved using the full Ramachandran-Lakshminarayanan convolution filter discussed in section 6.2.2.

7.1.2.1. Results and Discussion

The reconstructed images created using the noisy data are shown in figures 7-14 to 7-31. From visual inspection it appears that both the standard and modified backprojection methods suffer reconstruction degradation to the same extent. (This is confirmed in the following section by inspection of the reconstructed values). At plus or minus 1 percent noise (figures 7-14 to 7-19) the images are clearly identifiable as the phantom. At plus or minus 5 percent noise (figures 7-20 to 7-25) the lower contrast information is not recognisable, and at plus or minus 10 percent (figures 7-26 to 7-31) only the highest contrast information is recognisable.

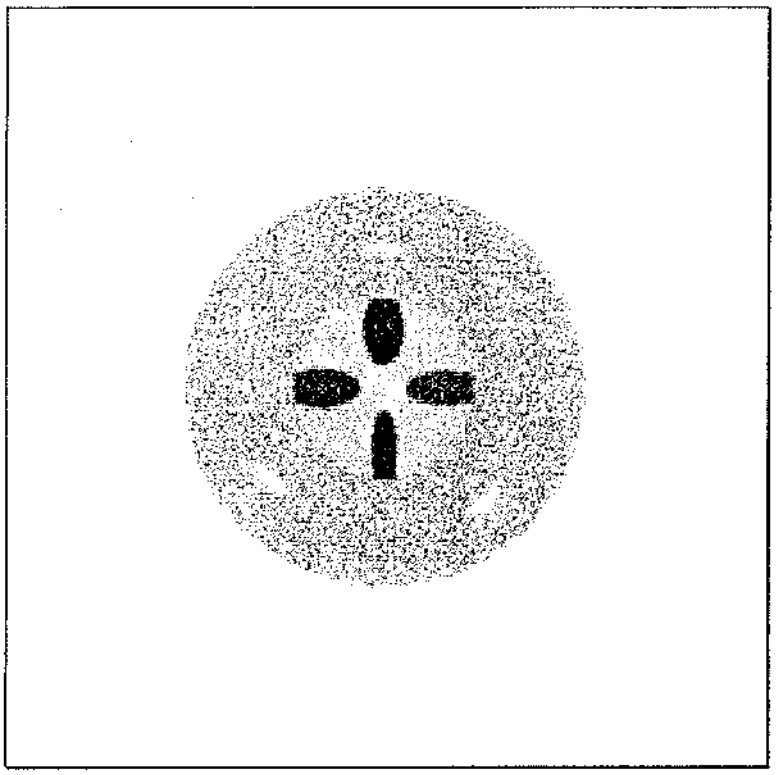


Figure 7-14. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 1 percent noise.

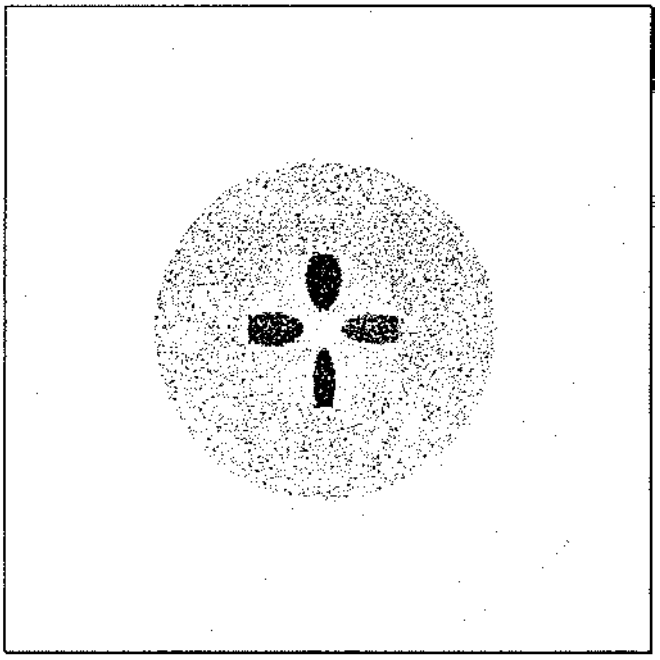


Figure 7-15. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 1 percent noise.

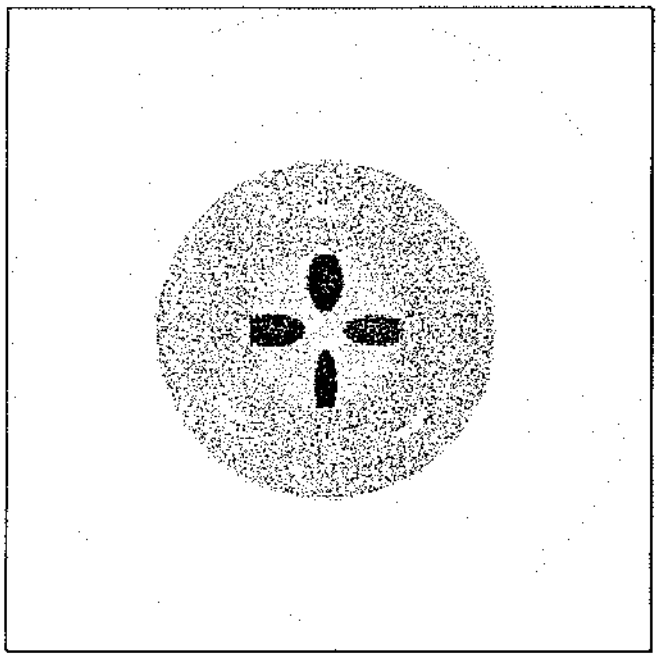


Figure 7-16. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 1 percent noise.

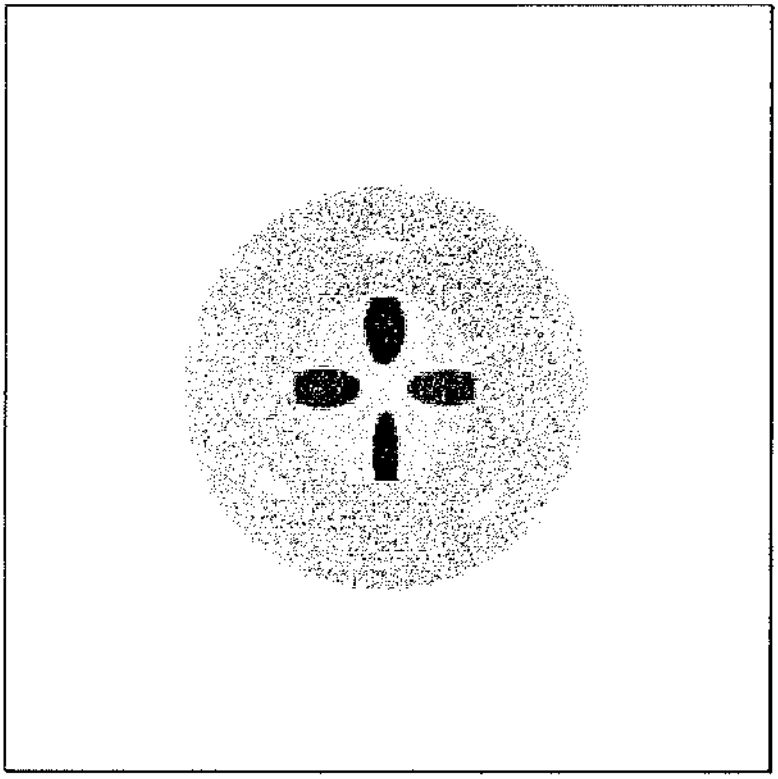


Figure 7-17. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with area interpolation and 1 percent noise.

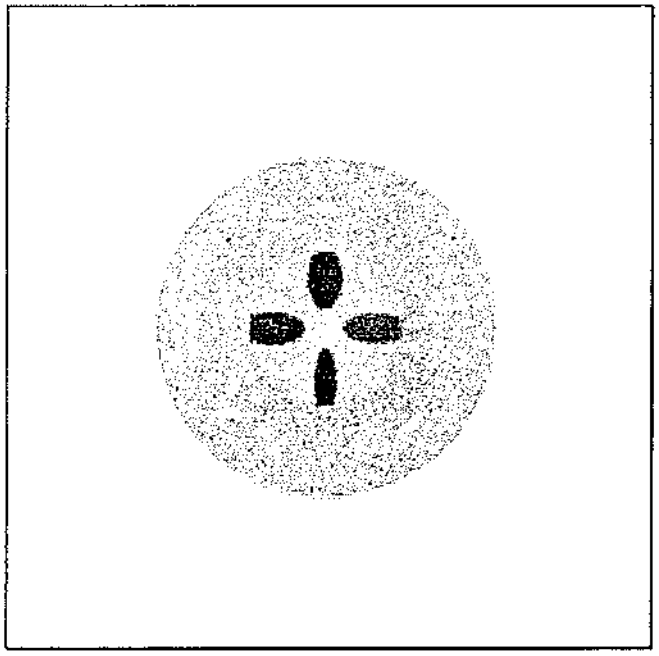


Figure 7-18. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 1 percent noise.

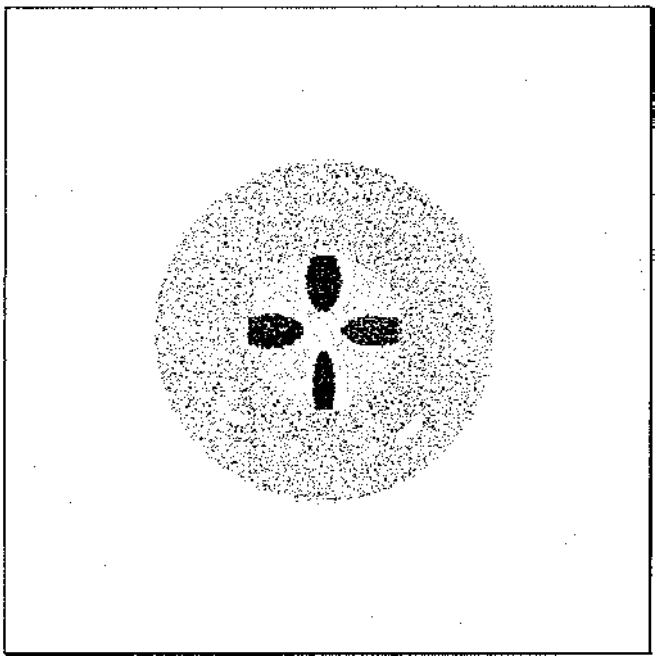


Figure 7-19. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 1 percent noise.

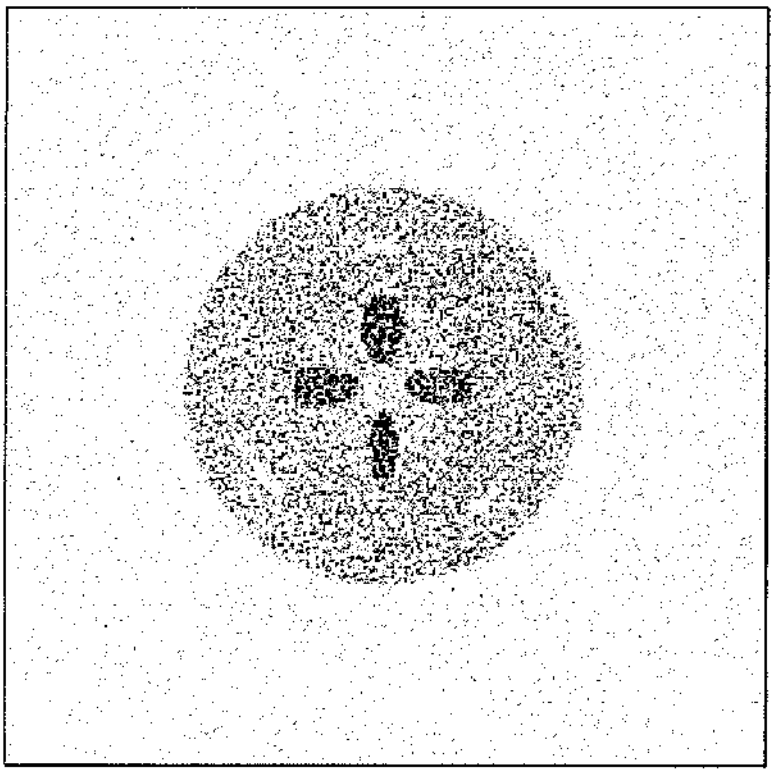


Figure 7-20. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 5 percent noise.

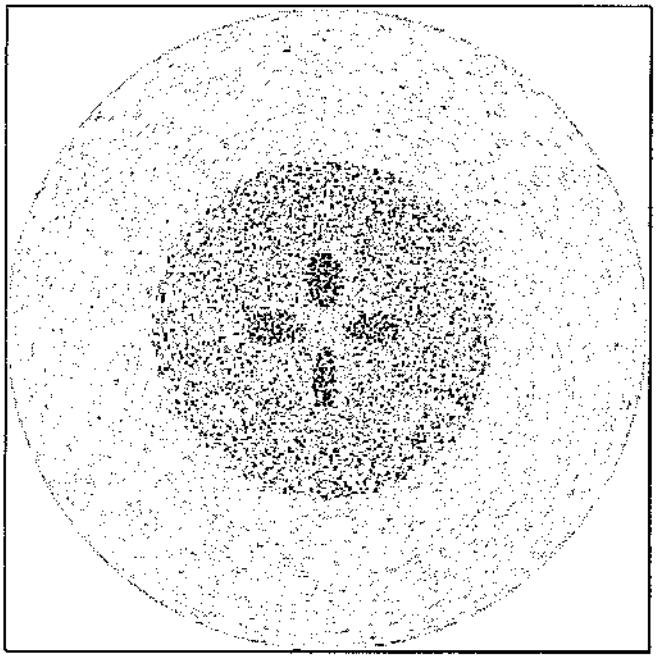


Figure 7-21. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

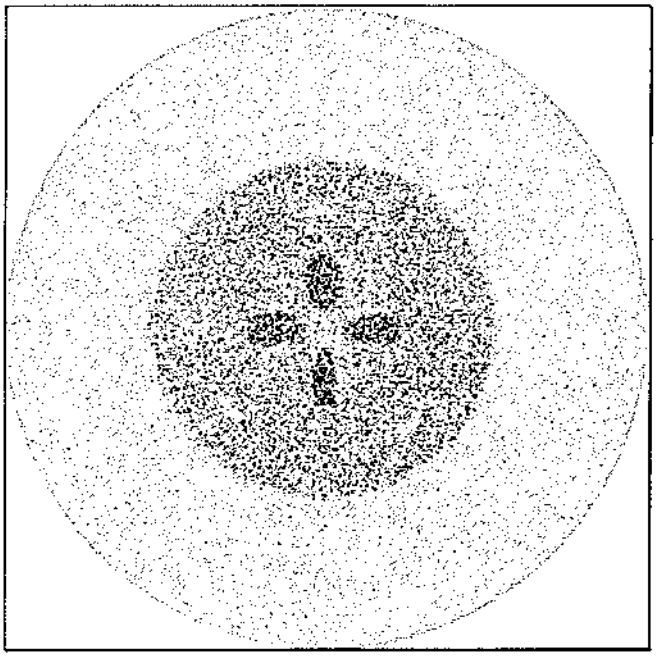


Figure 7-22. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

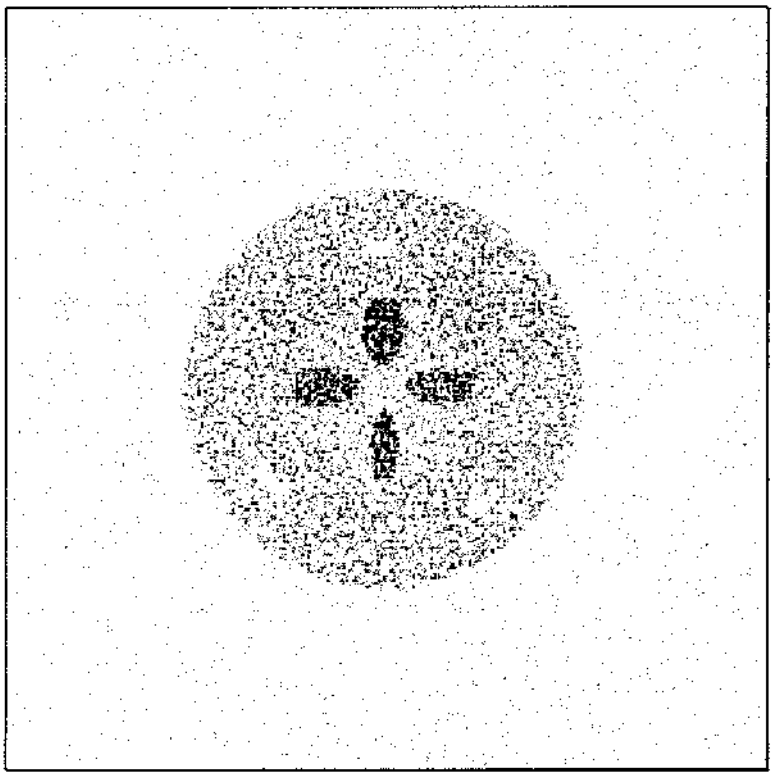


Figure 7-23. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with area interpolation and 5 percent noise.

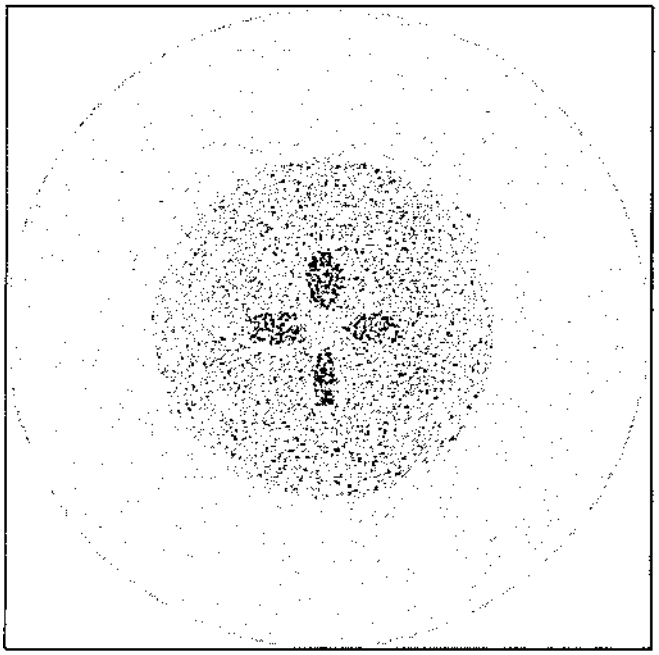


Figure 7-24. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 5 percent noise.

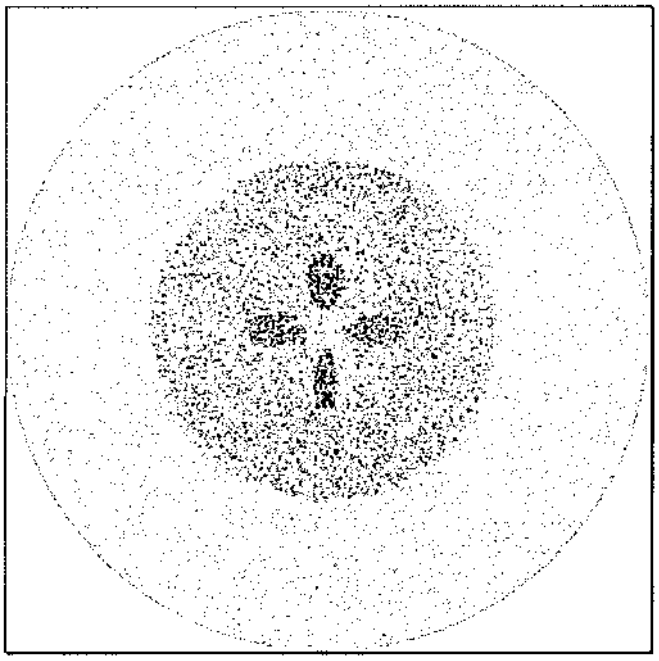


Figure 7-25. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 5 percent noise.

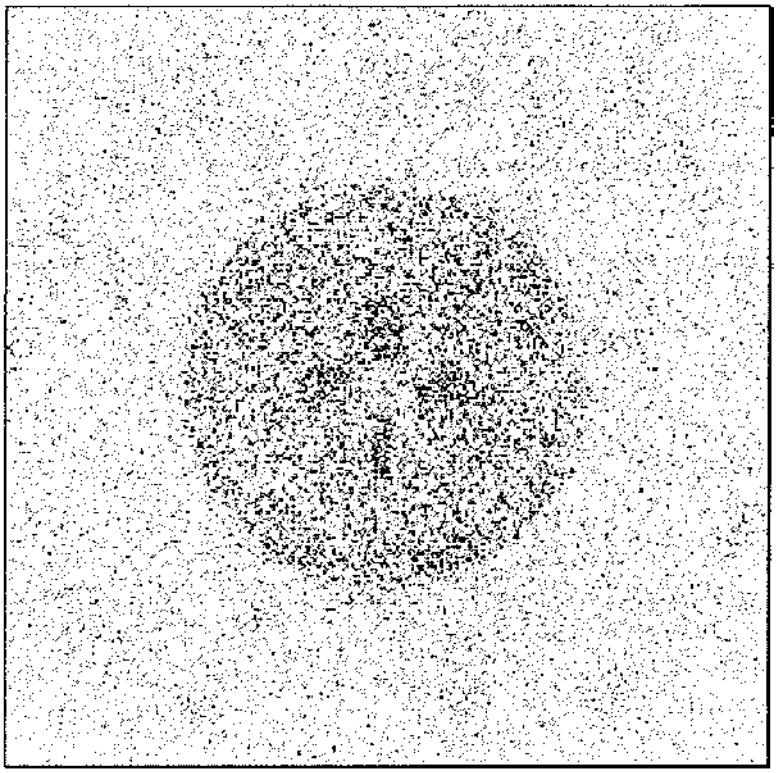


Figure 7-26. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 10 percent noise.

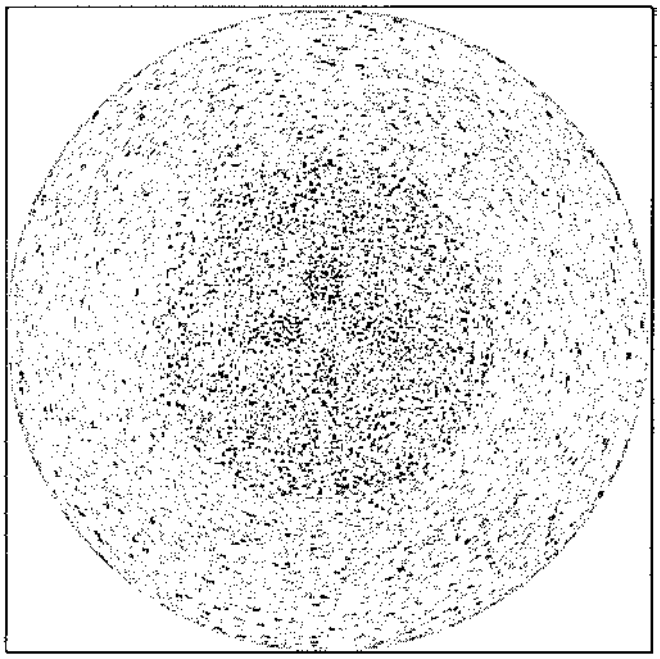


Figure 7-27. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 10 percent noise.

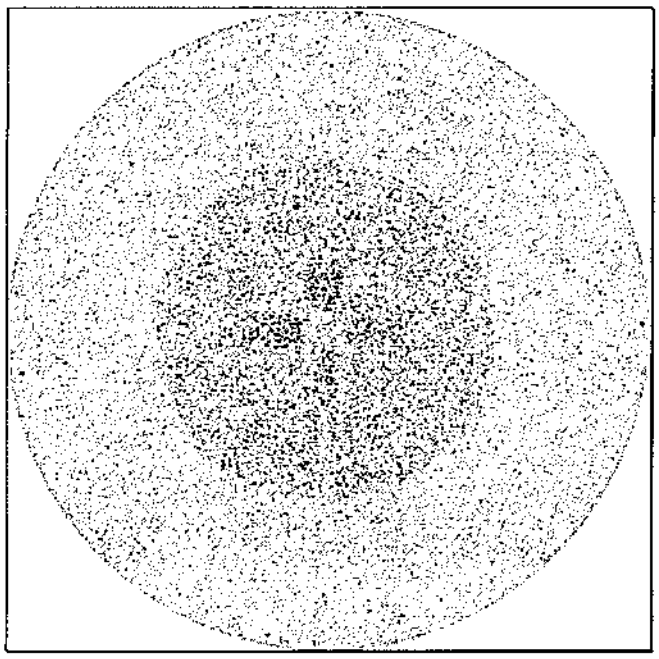


Figure 7-28. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 10 percent noise.

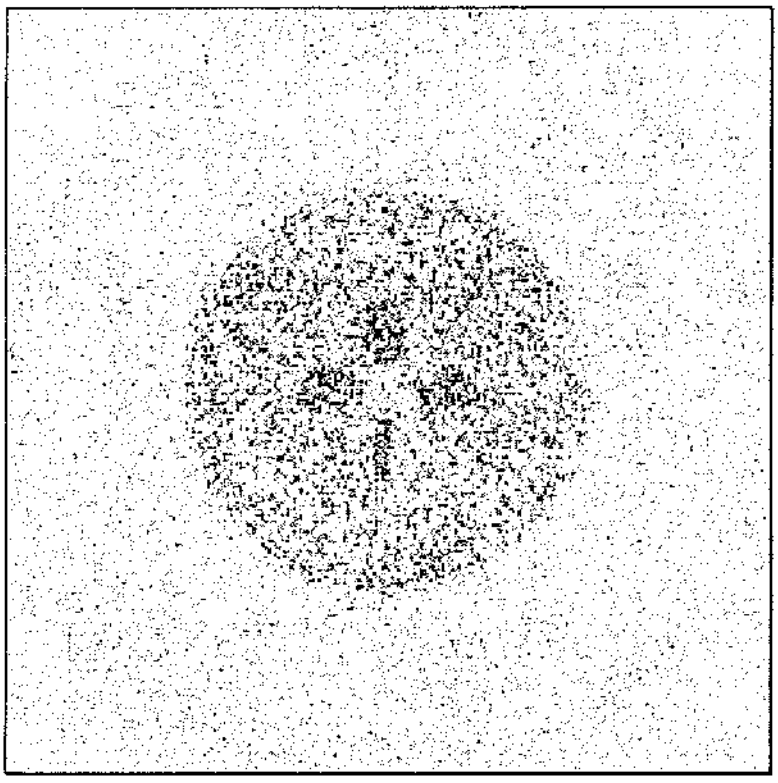


Figure 7-29. Reconstruction of log phantom using standard backprojection on cartesian reconstruction area with area interpolation and 10 percent noise.

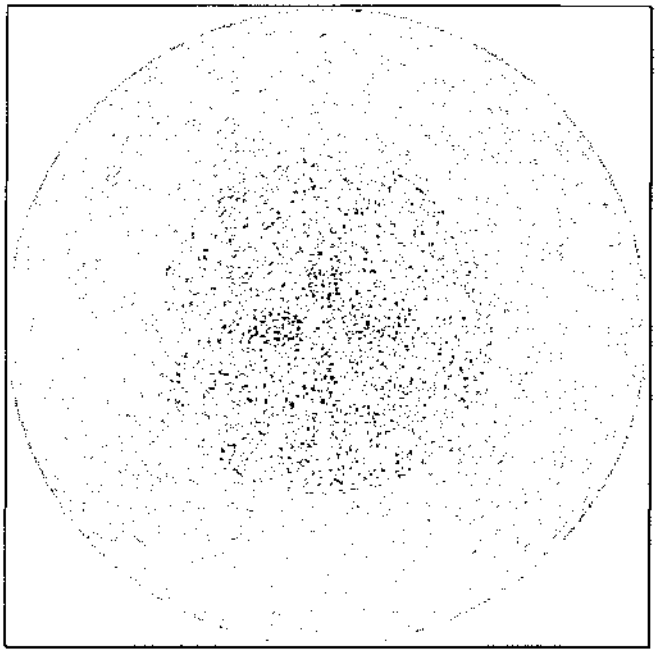


Figure 7-30. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 10 percent noise.

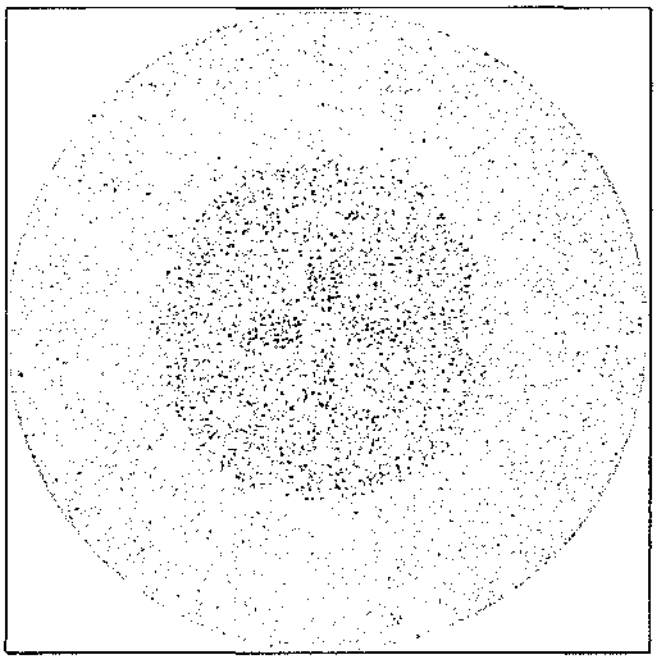


Figure 7-31. Reconstruction of log phantom using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 10 percent noise.

The intensity values along the y-axis of each of the above images are shown in figures 7-32 to 7-49 for value comparison. Table 7-4 shows the average magnitude of intensity deviation for the pixels depicted in the following figures, and the difference per percent of noise added. From table 7-4 the effect of noise follows a similar pattern for every reconstruction method, where the largest deviation per percent occurs for the 1 percent noise, followed by the smallest deviation for the 5 percent noise, and a value between the first two for the 10 percent noise. This pattern may have occurred because each method uses the same three sets of projection data, and the random fluctuations may have resulted in an outcome favoring the projection set containing 1 percent noise. The results and the graphs shown in the figures do however appear to confirm the conclusions obtained by visual comparison, that the effect of noise is similar for both the standard and modified backprojection methods, with the standard backprojection method performing slightly better in many instances, and equi-radial best-fit area performing slightly worse in most instances.

Table 7-4 Average of the absolute difference between actual and estimated images for figures 7.32 to 7.49. The units are pixel intensity values relative to 256.

	Nearest Neighbour			Area		
	1 %	5 %	10 %	1 %	5 %	10 %
Equi-radial/angular	11.98	58.62	119.48	8.38	31.82	68.39
per percent	11.98	11.72	11.95	8.38	6.36	6.84
Equi-radial best-fit angular	12.48	51.15	124.29	10.21	40.00	96.91
per percent	12.48	10.23	12.43	10.21	8.00	9.69
Standard Backprojection	10.43	29.15	64.75	8.94	24.22	61.98
per percent	10.43	5.83	6.48	8.94	4.84	6.20

The actual noise levels are not meaningful since the projection data has not been filtered before reconstruction to reduce the effects of noise, as would usually be the case. The reconstructed images serve only as a comparison and are not representative of realistic reconstructions.

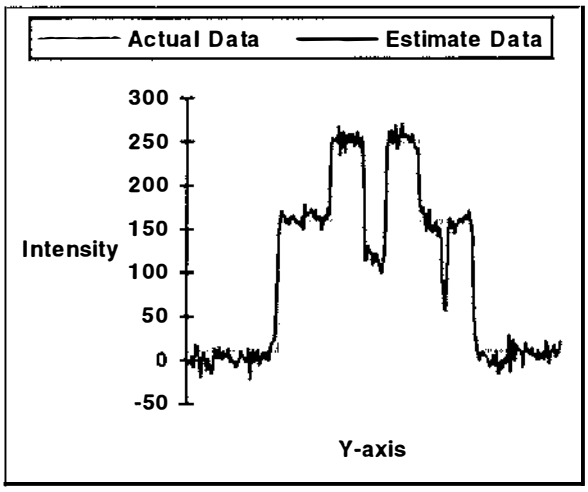


Figure 7-32. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 1 percent noise.

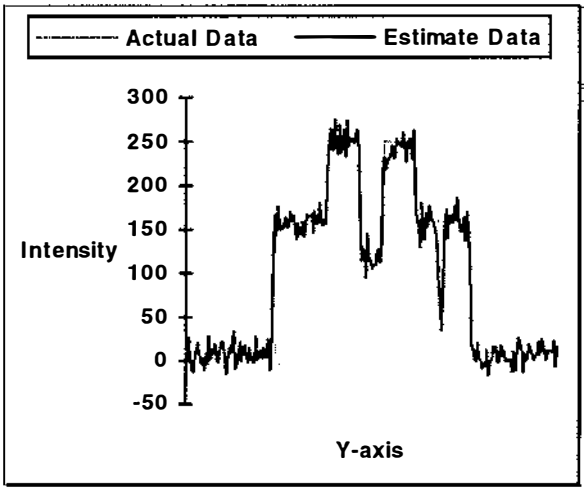


Figure 7-33. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 1 percent noise.

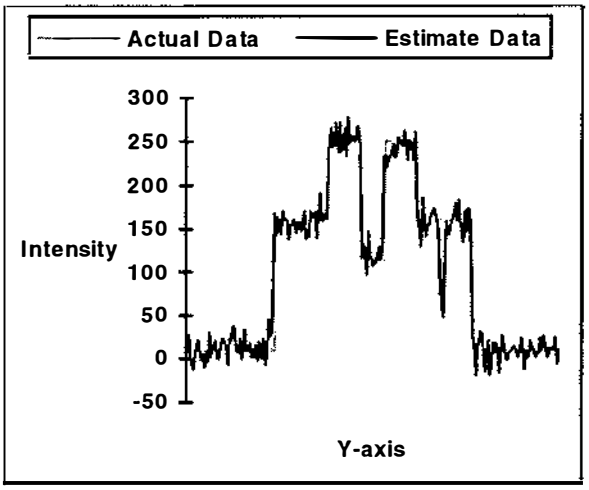


Figure 7-34. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 1 percent noise.

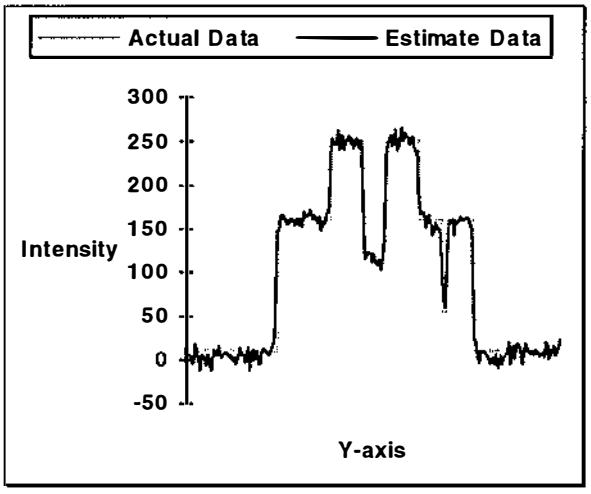


Figure 7-35. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with area interpolation and 1 percent noise.

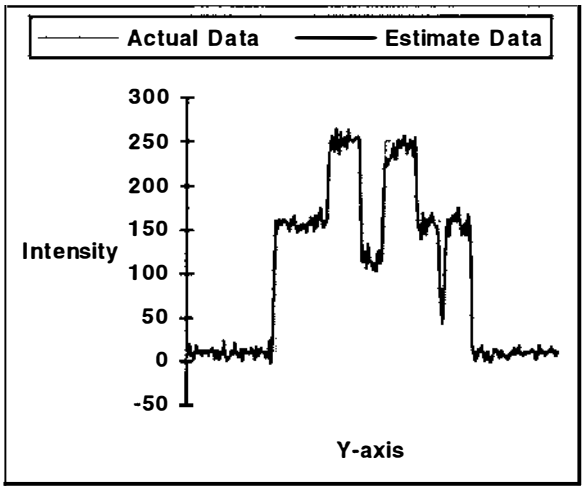


Figure 7-36. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 1 percent noise.

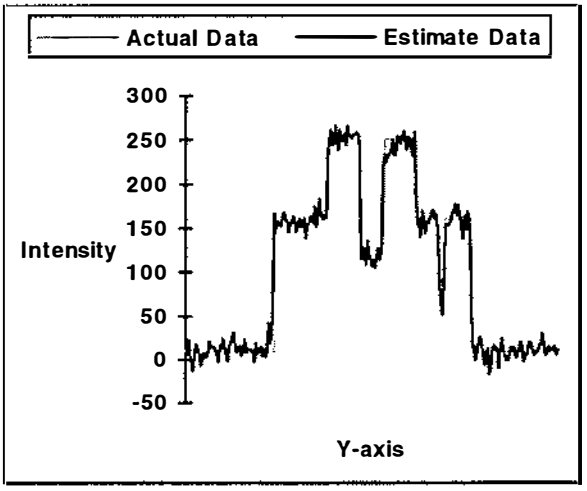


Figure 7-37. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 1 percent noise.

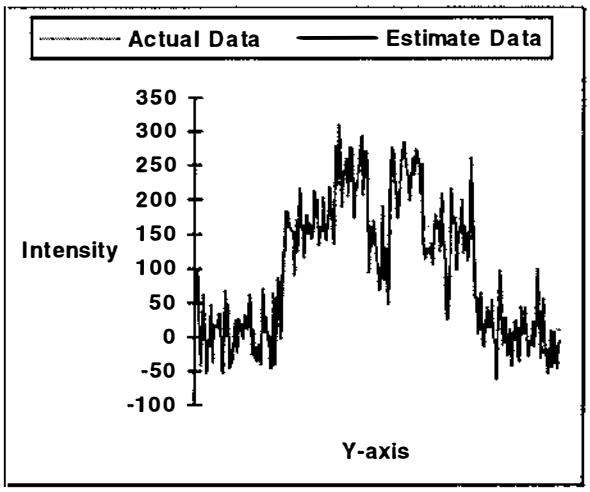


Figure 7-38. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 5 percent noise.

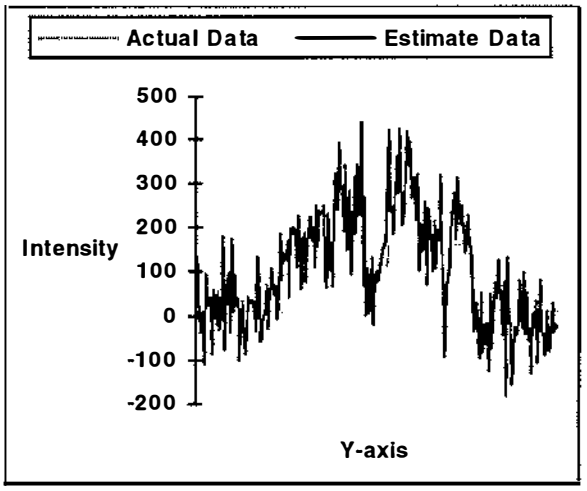


Figure 7-39. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

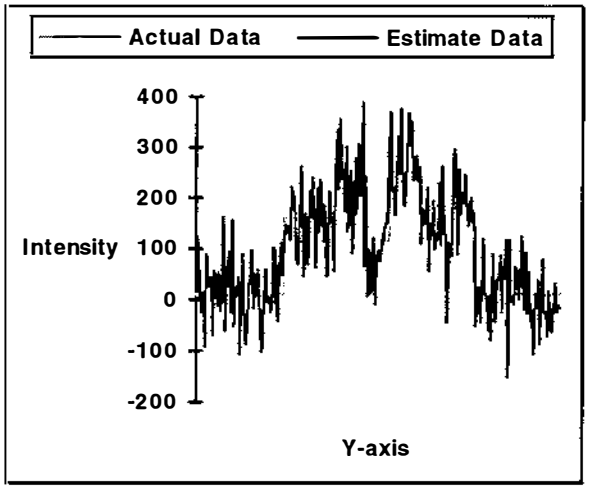


Figure 7-40. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

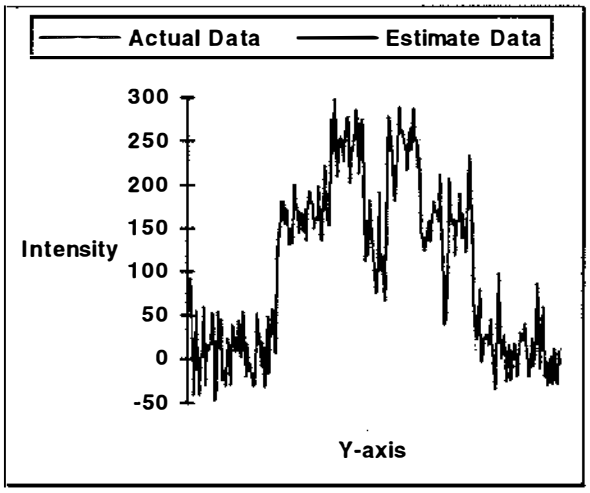


Figure 7-41. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with area interpolation and 5 percent noise.

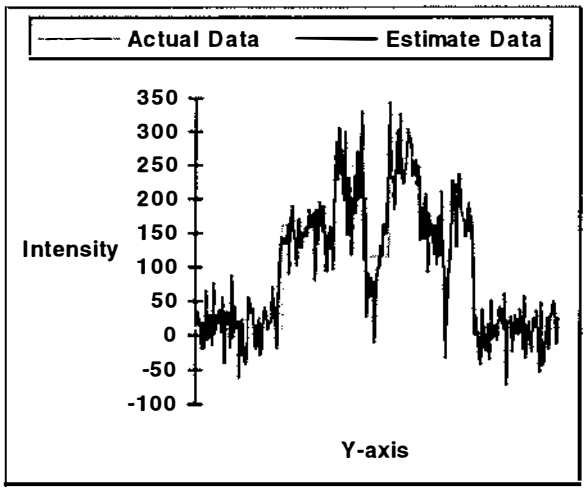


Figure 7-42. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 5 percent noise.

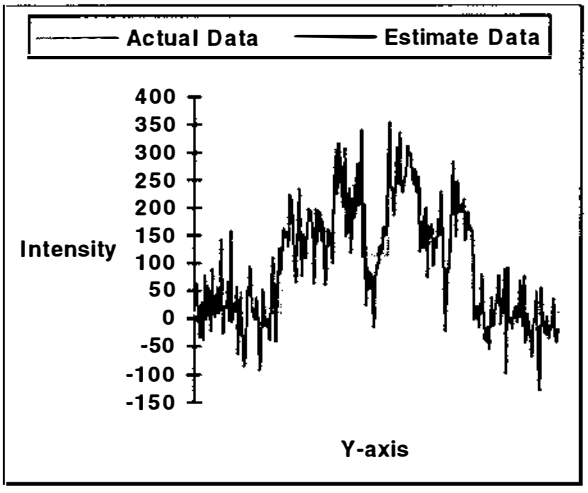


Figure 7-43. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 5 percent noise.

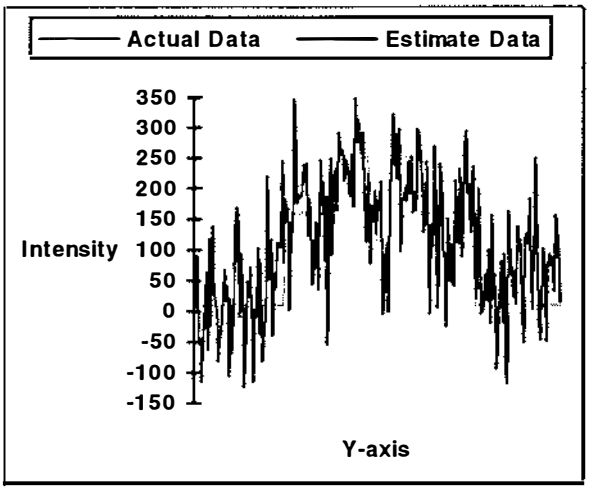


Figure 7-44. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with nearest-neighbour interpolation and 10 percent noise.

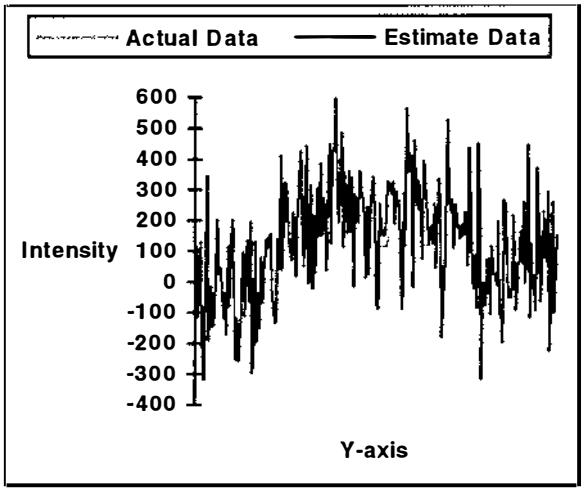


Figure 7-45. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 10 percent noise.

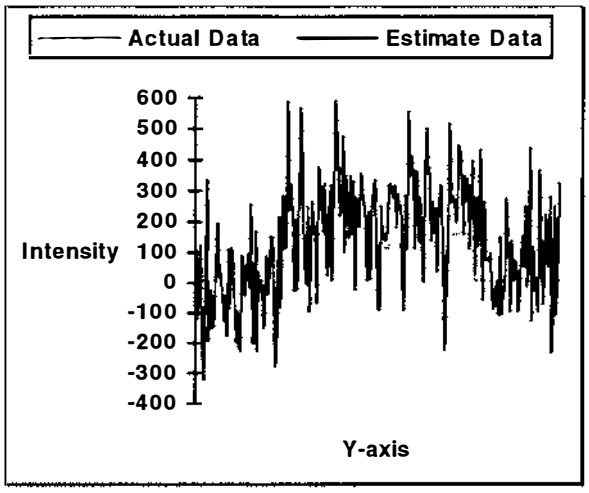


Figure 7-46. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with nearest-neighbour interpolation and 10 percent noise.

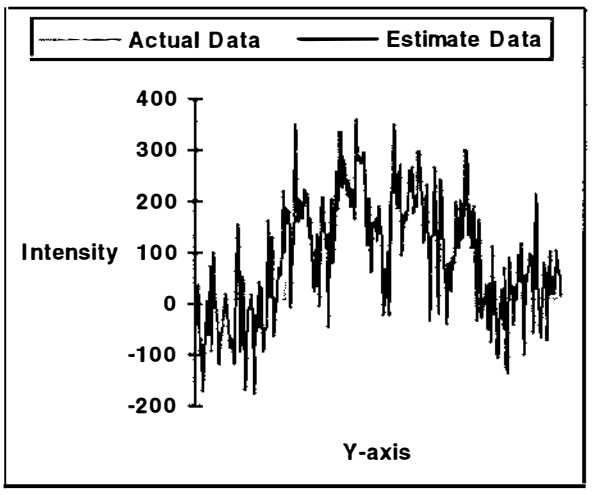


Figure 7-47. Y-axis of log phantom reconstruction using standard backprojection on cartesian reconstruction area with area interpolation and 10 percent noise.

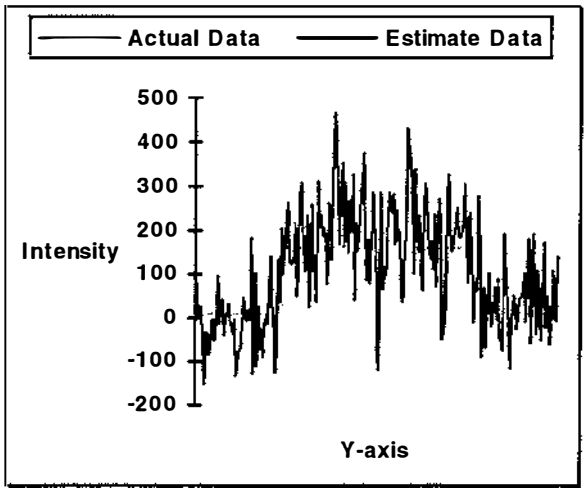


Figure 7-48. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with approximate area interpolation and 10 percent noise.

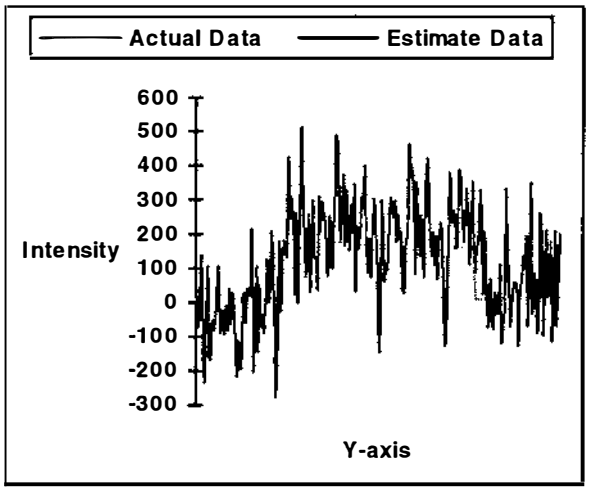


Figure 7-49. Y-axis of log phantom reconstruction using modified backprojection on equi-radial best-fit angular reconstruction area with approximate area interpolation and 10 percent noise.

7.1.3 Reduced Convolution Kernel Tests

Having observed the image quality obtained when using both the modified backprojection method and the reduced convolution kernel separately this section shows how the two concepts can be applied sequentially to produce images of comparable quality to those obtained using the standard backprojection method with the reduced convolution kernel.

Projection data convolved using 15 terms was used to reconstruct images of the log-phantom detailed in section 6.2.2.2. Figures 7-50 and 7-51 show the results from the equi-radial/angular and equi-radial/best-fit area modified backprojection methods using nearest-neighbour interpolation respectively. The image detail is easily definable and comparable to the result obtained using the standard backprojection method (figure 6-13). The y-axis plots shown in figures 7-52 and 7-53 show the presence of blooming as expected.

Figures 7-54 and 7-55 show the results obtained when 5 percent uniform noise is added to the projection data before convolving, and again the respective y-axis plots are shown in figures 7-56 and 7-57. Once again the results are comparable with those obtained using the standard backprojection method (figure 6-23).

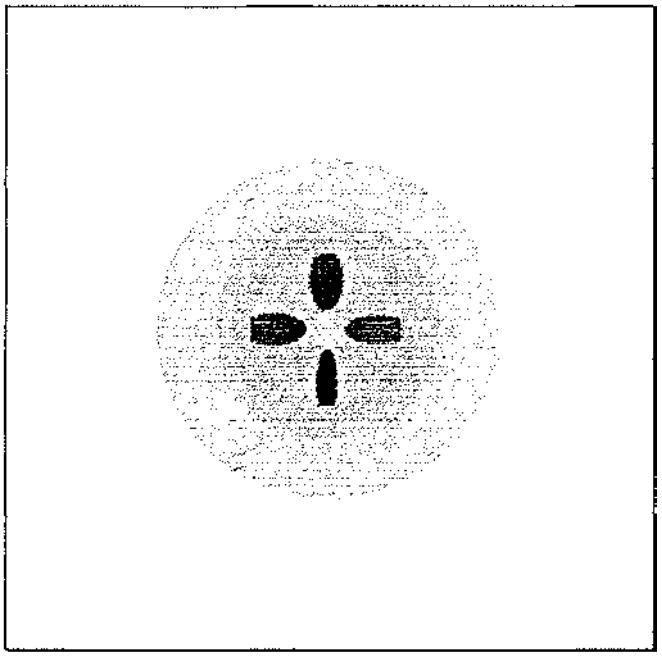


Figure 7-50. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation.

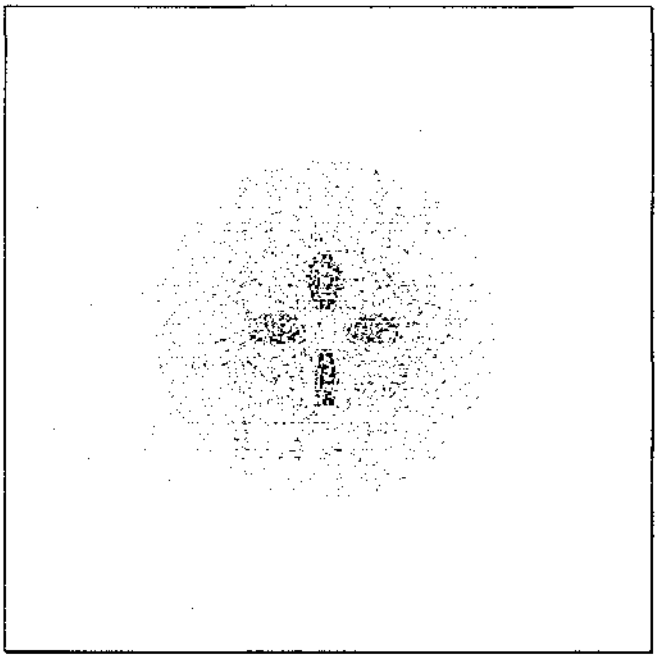


Figure 7-51. Reconstruction of log phantom using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

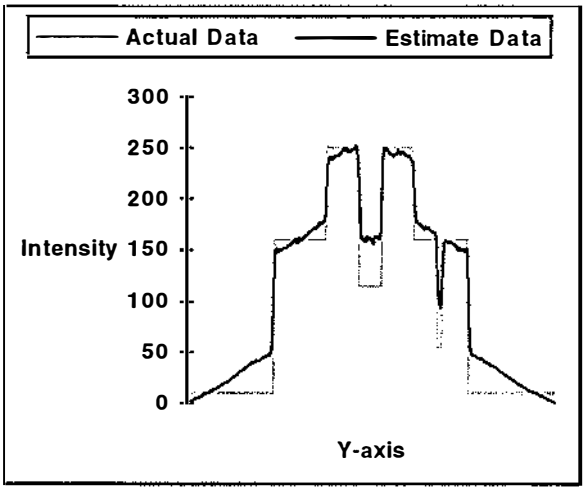


Figure 7-52. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation.

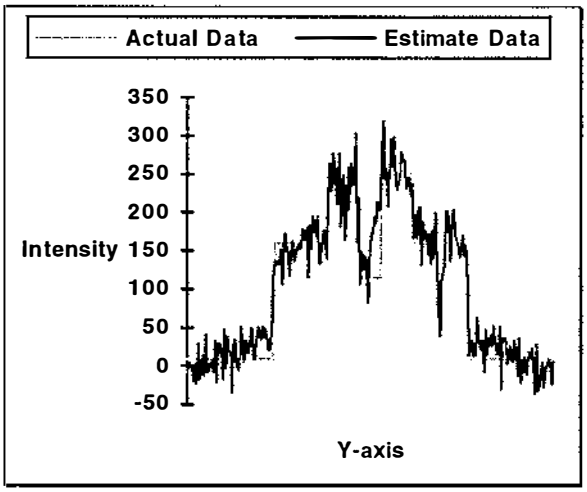


Figure 7-53. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

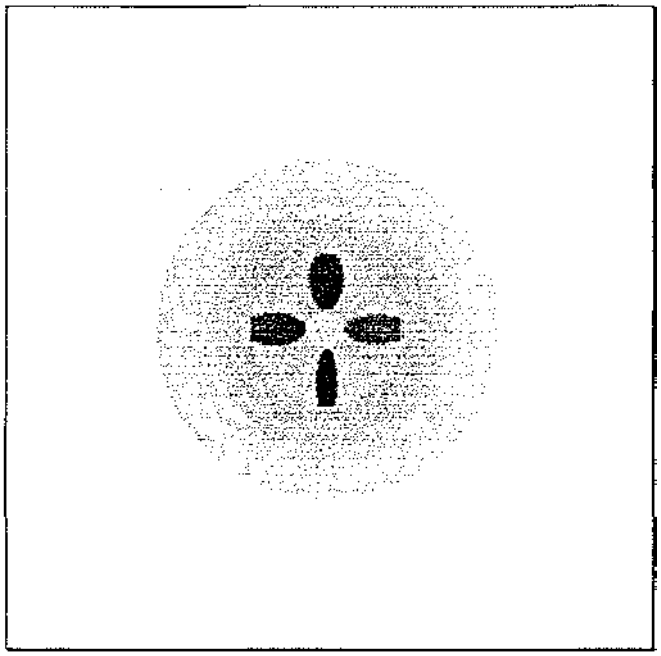


Figure 7-54. Reconstruction of log phantom using modified backprojection on equi-radial/best-fit angular reconstruction area with nearest-neighbour interpolation.

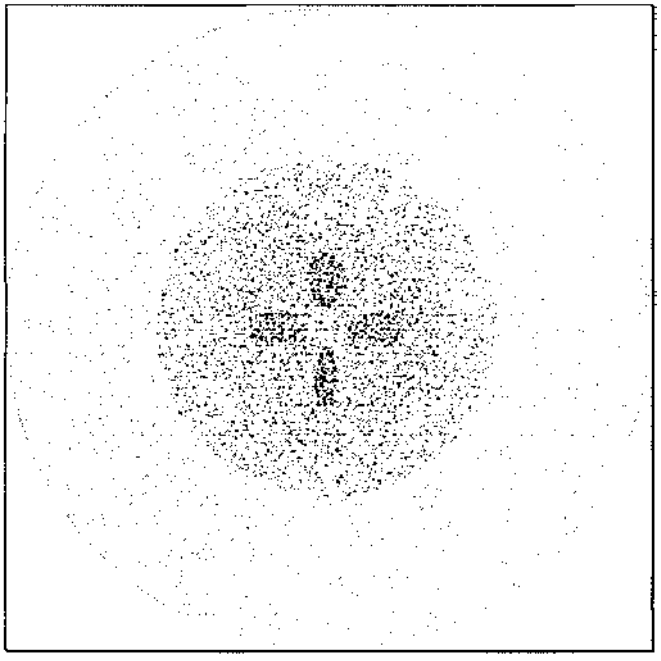


Figure 7-55. Reconstruction of log phantom using modified backprojection on equi-radial/best-fit angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

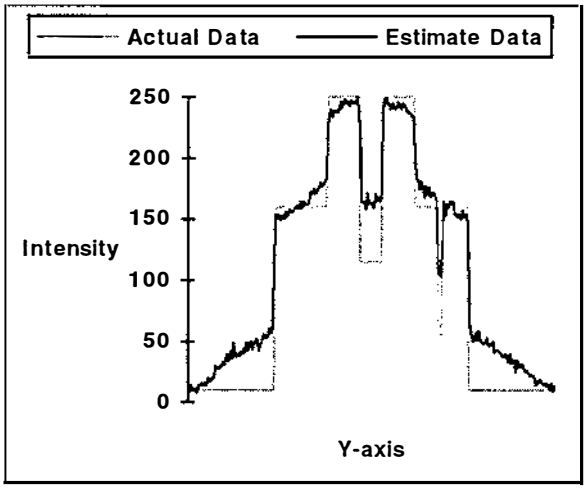


Figure 7-56. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/best-fit angular reconstruction area with nearest-neighbour interpolation.

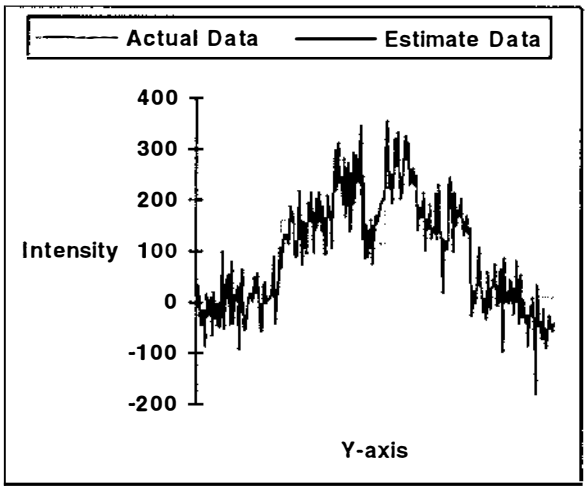


Figure 7-57. Y-axis of log phantom reconstruction using modified backprojection on equi-radial/best-fit angular reconstruction area with nearest-neighbour interpolation and 5 percent noise.

7.2 DIGITAL HARDWARE IMPLEMENTATIONS

It has been mentioned that many hardware implementations are capable of performing the modified backprojection algorithm, because only addition is required. In this section three different hardware implementations are applied to the algorithm. The implementations range from inexpensive to expensive. The first two implement the same schematic design, although the first is a dual-chip solution using a field Programmable Gate Array, and the second is built from discrete hardware components. The third implementation employs a systolic array architecture and is therefore well suited to parallel processing. To establish the algorithm as a systolic array it is rearranged so as to result in a number of individual processing blocks.

7.2.1 Schematic Design

The data is processed in the same way as described by the algorithms in section 7-1. From equations 7-6 and 7-7 it can be seen that the following components are required:

1. Storage for the Image array (f)
2. Storage for the pixel address look-up table (ψ)
3. Storage for the pixel per ray table (I)
4. Segment/Projection counter (n)
5. Ray counter (m)
6. Pixel per ray counter (i)
7. Modulo addition
8. Addition

The schematic design showing these elements is shown in figure 7-58.

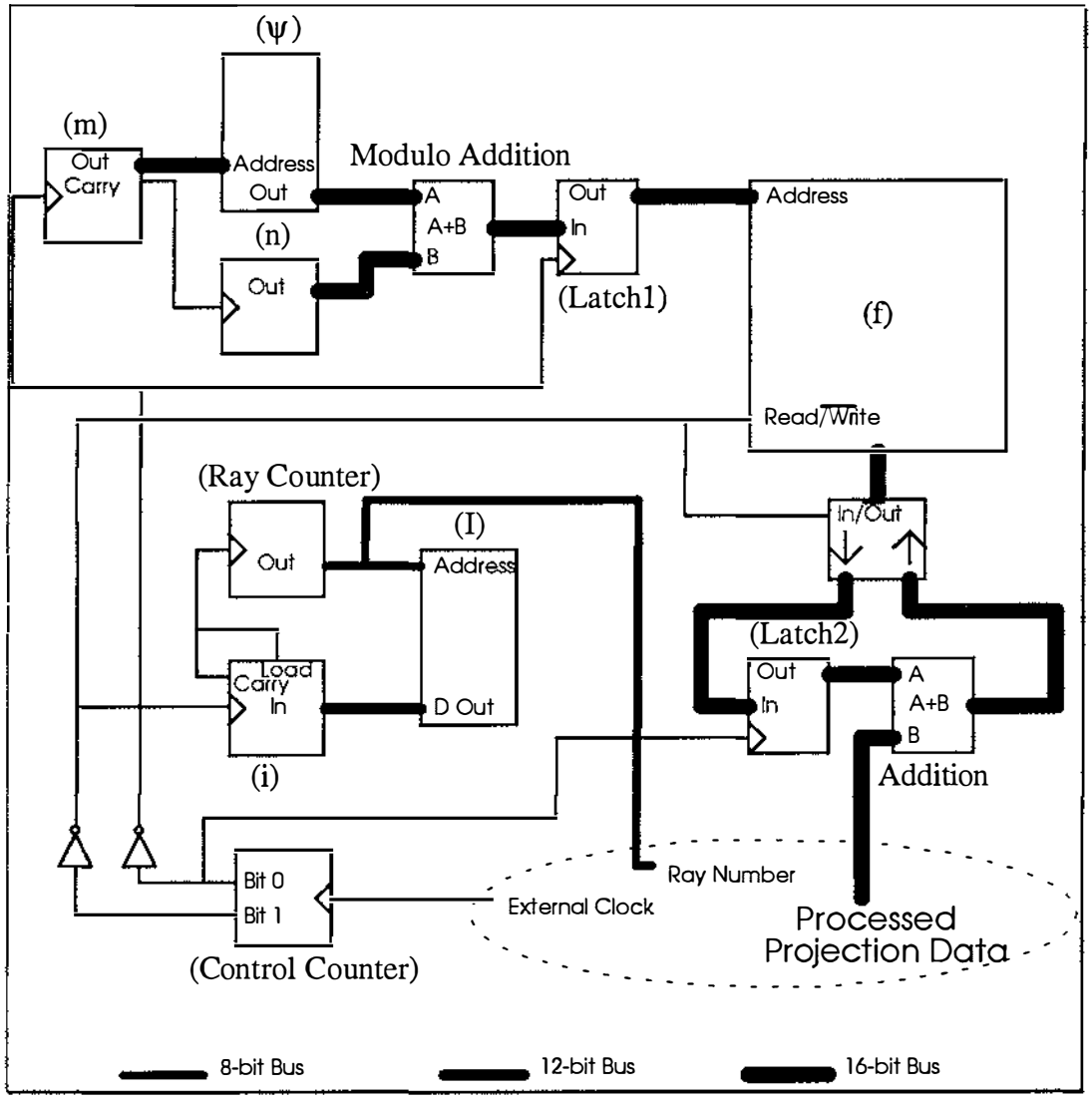


Figure 7-58. Schematic design of modified backprojection hardware.

The dotted line represents the external system and includes the convolved (filtered) projection data, and the external clock. The convolved ray value is passed into the system where it is added to a number of pixels by the ADDER (Addition). The pixels to be added to are determined by the look-up table (ψ) and the current projection number (n). The values from ψ and n are added together using the ADDER (Modulo Addition). The modulo arithmetic is achieved by disregarding the ADDER output bits Q_n where $2^n \geq C$. Note that C is the total number of pixels in the reconstruction area X× Y. Using this simple method of modulo arithmetic imposes a restriction on the size of the image f(x,y) where X×Y must equal a power of 2. The pixel calculation is latched (Latch1) so that the succeeding pixel calculation may be performed while the latched pixel is being processed.

The value of the calculated pixel is read from the reconstruction area (f) into a latch (Latch2). The bi-directional port is then toggled into the Out state (active low) where the pixel value incremented by the convolved ray value is written back into the reconstruction area (f).

The number of pixels to be incremented for any given ray depends upon the path of the ray. For a pre-determined system configuration these values can be pre-calculated and stored in a look-up table (I). The n-bit counter (i) initially reads the value from the look-up table (I). For the above schematic this value is 2^n - the number of pixels to process for the current ray. This value is incremented until the carry out bit is activated, at which point the above counter (m) is incremented and the next value from the look-up table (I) is loaded into the counter (i). The output from the counter (m) is used to indicate the required ray number to the external system.

The use of a 16-bit bus is arbitrary and assumes that the reconstruction area is equal to or less then 256^2 pixels. The ray number is never greater than 256 for such a system so an 8-bit bus is sufficient. However the number of pixels intersecting any ray may be much greater than 256, and a 12-bit bus is required.

To perform the operations of the system at high speed parallel processing is employed. A 2-bit Johnson counter is used to provide four divisions per processed pixel. To determine the approximate timing of the system some timing requirements are estimated for the components in the schematic.

1.	Memory Read/Write cycle time (High-speed RAM)	16	ns
2.	Latch	8	ns
3.	ADDER (16-bit)	16	ns
4.	Counter increment (16-bit)	24	ns

Based on these requirements timing diagrams for the operations of the schematic are shown in figures 7-59 to 7-61

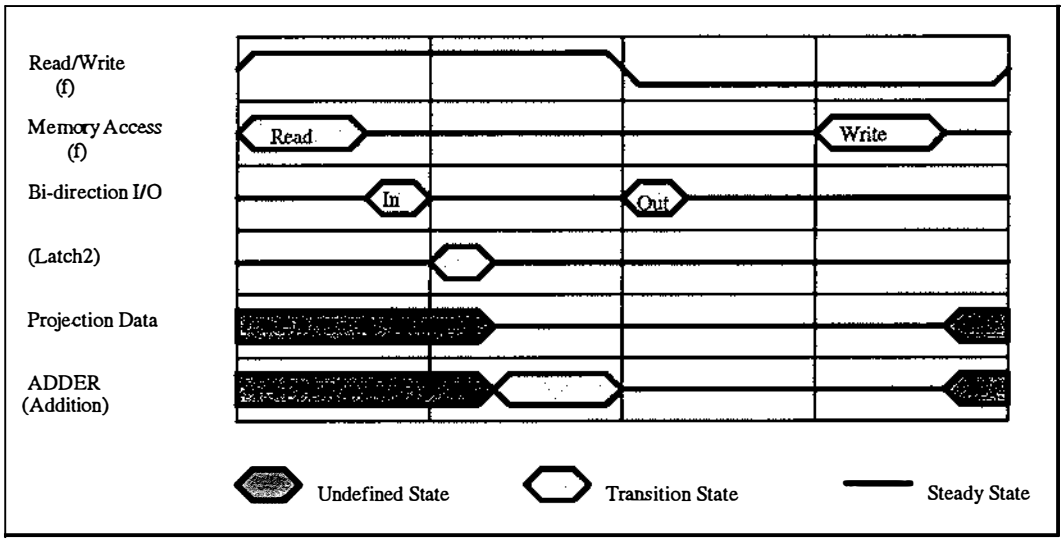


Figure 7-59. Pixel value incremented by projection data.

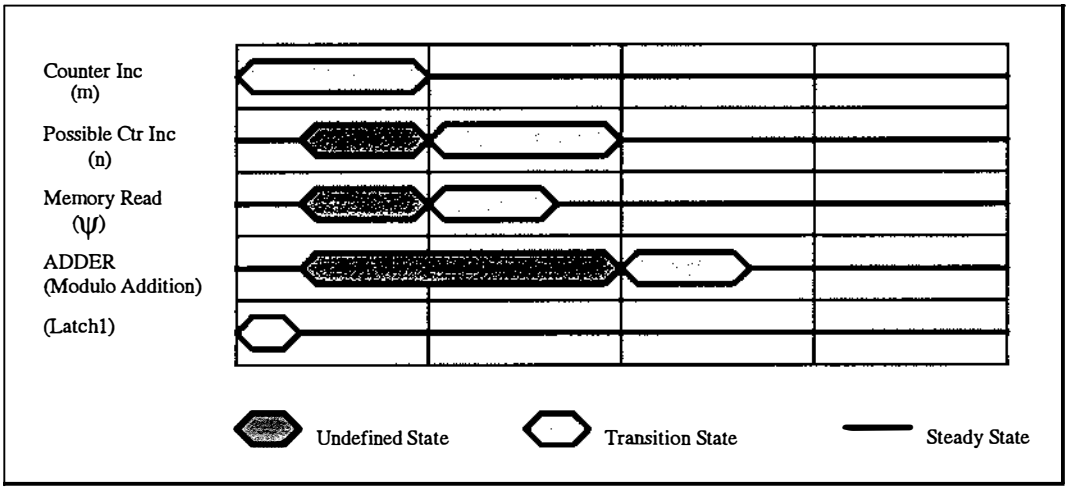


Figure 7-60. Pixel address calculated from look-up table (ψ) and segment number (*n*).

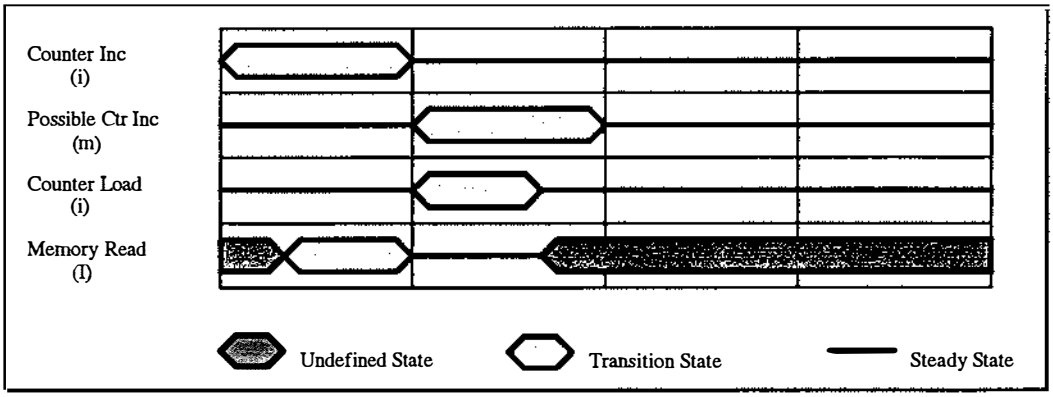


Figure 7-61. Ray number derived from look-up table (I) and pixel counter (i).

The analysis of timing given does not include path delays caused by routing, and is only intended as an initial estimate for determining the control timing for the clocked signals and control pins.

The first timing diagram (figure 7-59) contains the longest sequential process, and is therefore likely to be the bottleneck of the overall process.

The timing for the pixel address calculation (figure 7-60) shows that the latch (Latch1) is clocked at the same time as the counter (m) is incremented, indicating that this section has been pipelined without using latches by using the internal propagation of each individual component (note that the latch is edge triggered). It is risky doing this because propagation times vary between components of the same type, and hence a large margin has been allowed for in figure 7-60. Given that the pixel increment section is likely to be the bottleneck it may be unnecessary to use the pipeline in this section.

The third timing diagram (figure 7-61) shows that the counter (i) loads the value from memory before the new memory address has propagated through to the RAM. This means that the values must be offset by one location in the RAM. The reason for doing this is once again speed, since the load can be performed immediately instead of waiting for the load signal to propagate through the counter (m), into the address port of the RAM, and waiting for the output value from the RAM. This also may not be necessary because the bottleneck is in the pixel increment section. However it simplifies the control of this section, as the control signal comes from the counter (i) rather then deriving a control line from the Johnson counter (see the schematic in figure 7-58).

In order for the pixel address to be calculated in time for the pixel increment section, the timing of the pixel address calculation section (figure 7-60) needs to begin on the fourth division. This ensures that the pixel address is latched by the beginning of the first division. The logic for the control signals is therefore given by

	a b
Johnson counter:	0 0
	0 1
	1 1
	1 0

Latch1	=	/a
Latch2	=	a
Counter clock (*m*)	=	/a
Counter clock (*i*)	=	/b
Read/Write	=	/b
Input/Output	=	/b

These control signals are shown in the schematic from figure 7-58.

7.2.1.1 Field Programmable Gate Array (FPGA)

A XILINX 4000 series Field Programmable Gate Array was used to implement the schematic design described in section 7.2. The FPGA contains a number of combinatorial logic blocks (CLB's) which can be programmed to either calculate a boolean expression using pre-programmed boolean logic, or act as a memory cell for high-speed storage and retrieval

Combining several logic blocks together using pre-defined Hard Macros (ie. they have been designed to perform at optimum speed on the chip) allows many useful higher-level functions to be performed. These include accumulator, adder/subtracter, Bi-directional I/O pads with buffers, comparators, universal counters, data-registers (latches), Programmable Read-Only-Memories, Static Random-Access Memories, and others.

The FPGA's can either be programmed at the CLB level, or at a schematic level using a conversion program to convert the schematic to a CLB design. If the entry level is the schematic design then a number of conversion programs are required before the system can be downloaded onto the chip. These include

1. Schematic to netlist translator. This program creates a file containing the details of the schematic in Xilinx netlist format.

2. Hard macro module synthesis. The netlist file is read and details of the hard macros and their attributes are extracted. The data type information is propagated throughout the design, the hard macro modules are synthesised, architecture specific optimisation is performed, and files containing information about the hard macros and memory modules are created. The hard macro file is in Xilinx netlist format.

3. Memory generation. The file containing information about the memory modules is read, and a netlist file is created for the memory modules.

4. Partition, Placement, and Routing. The initial netlist file is merged with the hard macro and memory module netlist files to partition the operations into the required CLB's. The CLB's are then placed onto the chip, and auto routing is performed.

5. Makebits. The resulting file is converted into a bitstream which can be downloaded onto the PGA.

Using the schematic entry level the schematic design from figure 7-58 was programmed into the schematic design package. This is shown in figure 7-62.

Three memory blocks are specified by the schematic design of figure 7-58. However the smallest memory block (I) is contained within the chip in figure 7-62. This reduces the cost of the system, and the number of pins required. The input/output requirements of the design are

Inputs		
	External Clock	1-bit
	Image RAM input	16-bit
	AddressArray RAM input	16-bit
	Projection data input	16-bit
Outputs		
	Image RAM read/write	1-bit
	Image RAM output	16-bit
	AddressArray RAM output	16-bit

The total number of pins required is therefore 82, which is well under the maximum number of input/output pins (240) provided in the 4000 series FPGA.

Two control lines (nets) are marked critical (indicated by a *c* beside the line in figure 7-58). These nets are the latch clock for Latch1, and the read/write line going to the external RAM. Marking a net as critical causes the auto-routing program to minimise the delay along that net. Latch1 is marked critical to ensure that the non-latched pipeline of the pixel address calculation (see timing diagram, figure 7-60) does not suffer from extreme skew, causing the pipeline to be ineffective. The read/write line is marked

critical to minimise the delay in the bottleneck path caused by the external RAM.

The first four conversion programs were executed on the schematic, and the resulting chip layout is shown in figure 7-63. Although it is difficult to ascertain much from the figure, it does show that most of the CLB's (indicated by small square boxes) in the chip have been used.

The program allows worst-case timing analysis to be performed. This is useful in determining the maximum speed of operation for the FPGA chip. A detailed description of the worst-case timing for the main sections of the design is given in appendix H, and is summarised below (the actual output file contained 687 pages).

Pixel increment section	
Input pin (f) to latch input (Latch2)	16.3 ns
Latch input (Latch2) through ADDER to output pin (f)	40.8 ns
Input pin (projection data) to output pin (f)	41.2 ns
Address calculation section	
Clock output (Control counter) to clock input (n)	38.7 ns
Clock input (n) to Clock output (n)	34.0 ns
Clock output (n) through ADDER to latch input (Latch1)	25.1 ns
Latch input (Latch1) to output pin (f)	20.8 ns
Clock input (m) to output pin (ψ)	20.7 ns
Input pin (ψ) to latch input (Latch1)	26.3 ns
Ray calculation section	
Clock output (Control counter) to clock input (m)	34.7 ns
Clock input (m) through PROM to data input (i)	44.1 ns

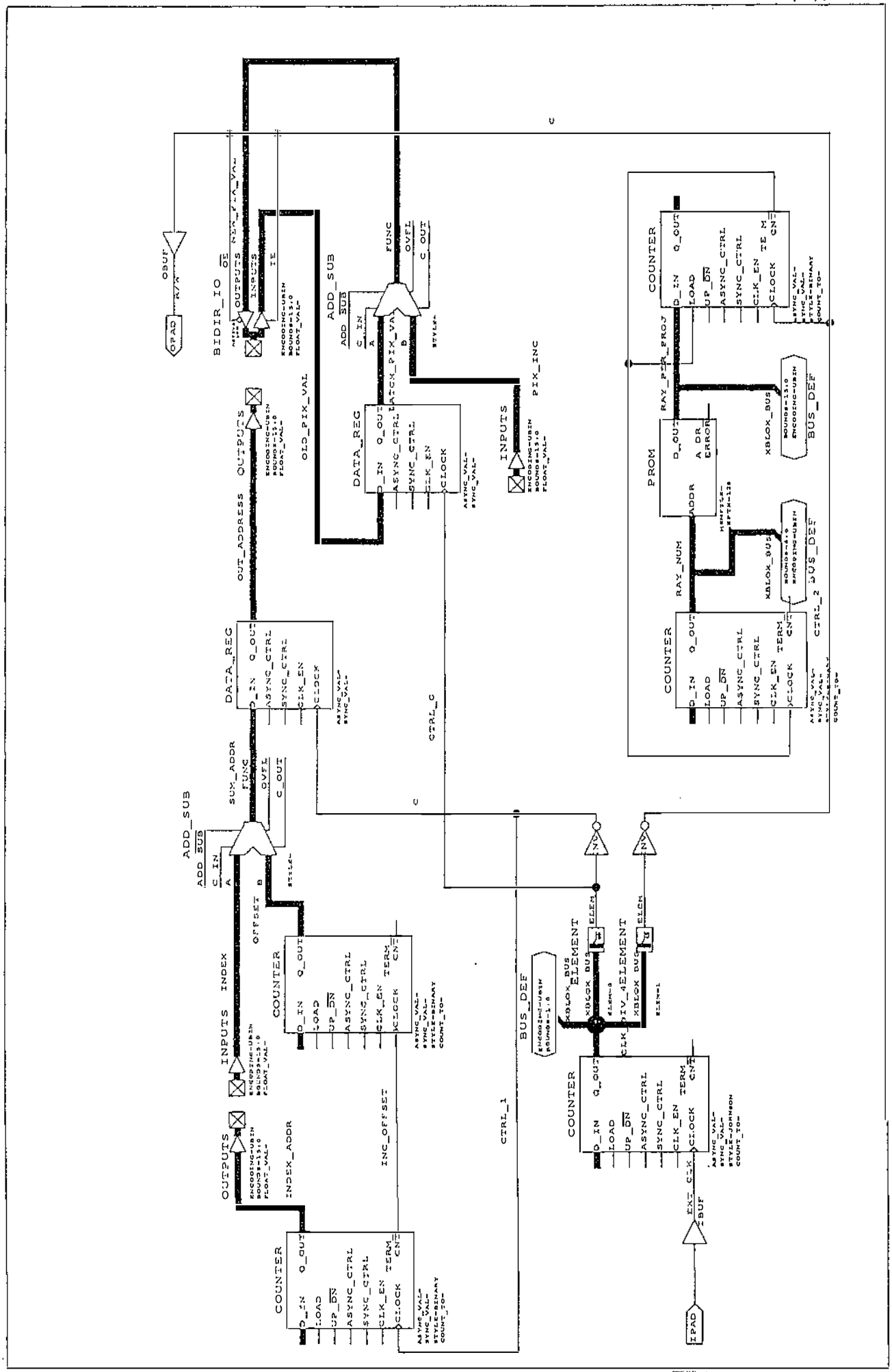
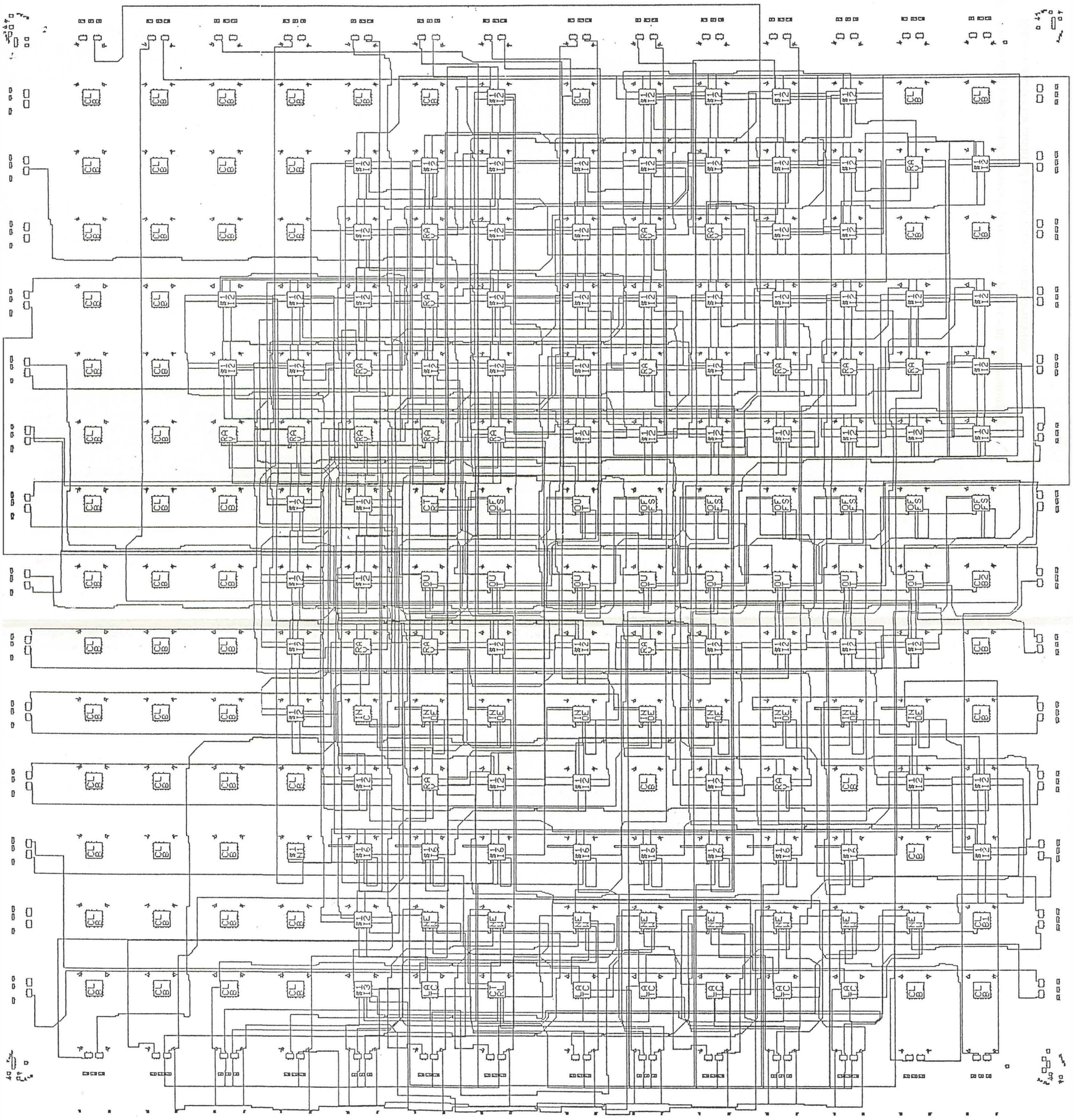


Figure 7-62. Schematic design of modified backprojection algorithm for FPGA implementation.

Figure 7-63. FPGA chip layout after automatic partitioning, placement, and routing by Xilinx software for modified backprojection implementation.



Substituting these times into the timing diagrams shown in figures 7-59 to 7-53, and assuming that the time for a memory read/write is 15 ns maximum [IDT71981S/L Integrated Device Technology 1987] shows that the minimum time for one quarter cycle is determined by the memory read (f) and pin to latch (Latch2) times, giving a quarter cycle time of 31.3 ns (15ns + 16.3ns). This is shown in figure 7-64.

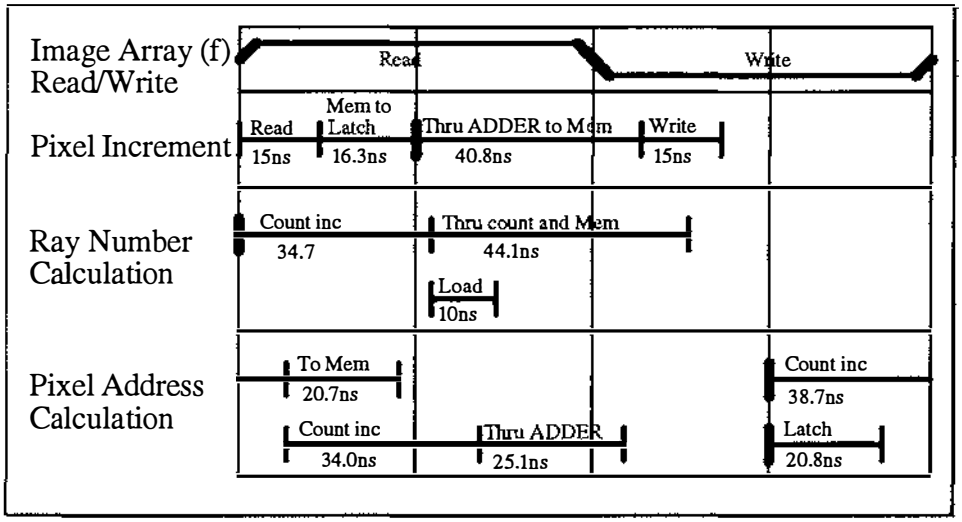


Figure 7-64. Timing of FPGA system using worst case delays.

At this rate one pixel is processed every 125.2 ns. For a system with a 100^2 pixel reconstruction area, and 100^2 ray values, the number of pixels to process per image is 1 million. Therefore the maximum number of image reconstructions per second using the schematic design of figure 7-58 is about 8.

It may be possible to improve the performance by specifying different critical paths in light of the timing information. It may also be desirable to redesign the schematic to reduce the bottleneck. However this shows that about 8 images of 100^2 pixels can be reconstructed per second using inexpensive hardware. The construction process is also extremely simple as only three chips are required (the PGA and two RAM chips).

7.2.1.2 Discrete Hardware Components

Although the Field Programmable Gate Array implementation simplifies design and construction, it does so at the cost of the reconstruction speed. Building the hardware from discrete components increases the complexity of construction, but may significantly reduce the processing time for each reconstruction.

Fast Schottky technology is used in the following discussion for the latch, counter, and addition components. The data sheets for the components mentioned can be found in [Integrated Device Technology 1987]. This technology has been superseded by a number of faster technologies (eg. Emitter Coupled Logic (ECL) which is up to twice as fast). However this merely reinforces the possible improvements in speed from the Field Programmable Gate Array.

For high-speed 16-bit addition, four 4-bit arithmetic logic units (74F181 ALU) are used with a carry look-ahead generator (74F182). The ALUs generate two signals which are connected to the carry look-ahead generator. These signals are called *carry propagate*, and *carry generate*, and require a maximum of 7.5 nanoseconds (typ. 5.7 ns) to propagate through the ALU. The carry look-ahead generator uses the two signals from each of the ALUs and calculates all of the *carry in* signals for the ALUs (as opposed to the ripple method where the *carry out* signal from one ALU is passed to the *carry in* input of the next ALU). The maximum propagation time required for the carry look-ahead generator is 7.5 nanoseconds (typ. 4.0 ns). The *carry in* signal then propagates through the ALU to the outputs giving the result of the addition. The maximum ALU propagation time from the *carry in* signal to the stable outputs is 8.5 nanoseconds (typ. 6.6 ns). The configuration described here is shown in the data book [Integrated Device Technology 1987, pp 6-193].

The maximum time to perform the 16-bit addition is 23.5 nanoseconds (typ. 16.3 ns), given by the total of the three propagation times above.

The 16-bit counters are created using four 4-bit binary counters (74F191). For the fastest speed the four counters are connected as a synchronous 4-stage counter with common clock and parallel gated terminal count (see data book [Integrated Device Technology 1987, pp 6-203]. These counters are quoted as having a minimum maximum counting frequency of 100 MHz (typ. 125 MHz), giving a maximum count increment time of 10.0 nanoseconds (typ. 8.0 ns). The maximum propagation time from the asynchronous parallel load signal to the stable outputs is 11.5 nanoseconds (typ. 8.0 ns).

The 16-bit latch is created from two 8-bit transparent latches (74F373). The maximum propagation delay from the clock signal to stable outputs is 8.5 nanoseconds (typ. 6.5 ns). The 74F373 latches have 3-state outputs and are

used for the bi-directional input/output. A tri-state latch is required only for the output (write) direction, as the input data is latched by Latch2.

The memory (I) was distributed withinin the array for the FPGA implementation, and three memory blocks are now required instead of two. The timing analysis will assume that the memory chosen has a memory access and cycle time of 15 nanoseconds. The IDT71981S/L 16K by 4-bit static RAM complies with these speed requirements. However a smaller chip may be used for the (I) look-up table since it contains only a few values.

The delay times caused by the capacitive and inductive effects of the connection lines is dependent on the layout of the board. However since the number of chips is relatively small it should be possible to minimise the delay on the time critical paths, and delays caused by PCB wiring interconnections are assumed to be negligible compared to the propagation delay of the components.

Substituting the above times into the timing diagrams of figures 7-59 to 7-61 shows that the minimum time for two quarter cycles is given by the latch (Latch2), add (Addition), and Bi-directional output components and is equal to 40.5 nanoseconds. This is shown in figure 7-65.

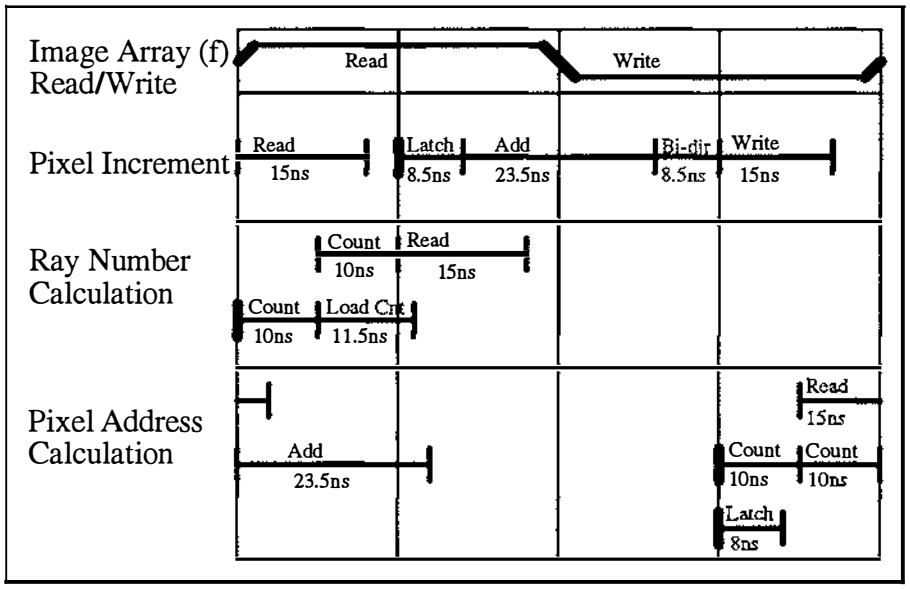


Figure 7-65. Timing of system using discrete components and worst-case timing.

The bottleneck of the system is once again the pixel increment section. The amount of idle time is greatly reduced when discrete hardware components are used because the delays caused by signals traveling across the silicon substrate, through complex networks of switching transistors, is eliminated and replaced by high speed signal transfer via copper wire or PCB (printed circult board) track.

Using the timing in figure 7-65 one pixel is processed every 81.0 ns. For a system with a 100^2 pixel reconstruction area, and 100^2 ray values, the maximum number of image reconstructions per second about 12.5.

All of the timing for the discrete hardware components has been based upon worst case propagation delay times. However since the components are inexpensive it is possible to buy multiple components, test the actual propagation delays, and select the fastest components. By doing this it is likely that the timing will be better represented by the typical propagation delay values. Figure 7-66 shows the same analysis done using typical delay values from the data book [Integrated Device Technology 1987].

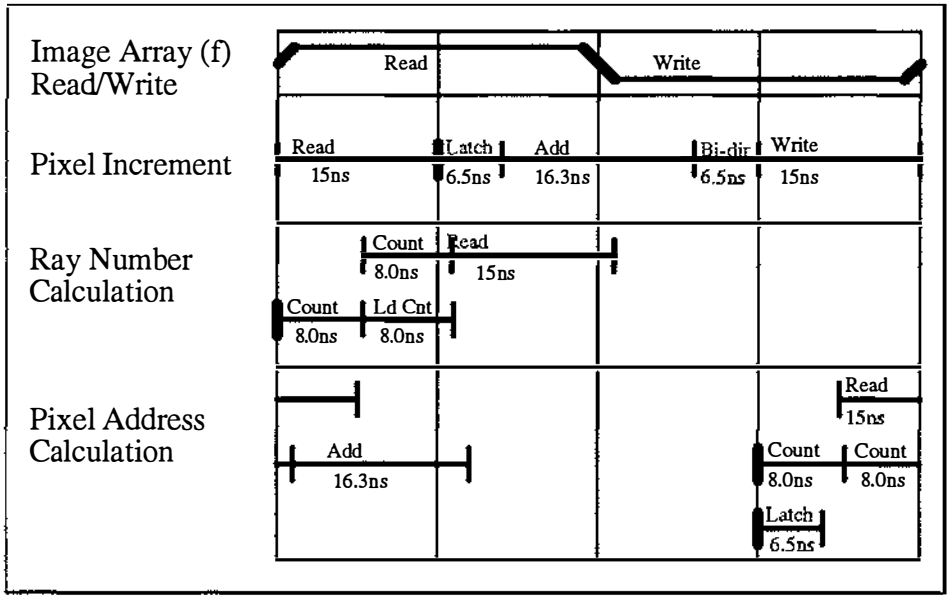


Figure 7-66. Timing of system using discrete components and typical timing.

The minimum time for a quarter cycle is now 15 nanoseconds determined by the memory read and write times. The resulting reconstruction rate is nearly 17 images per second.

7.2.2 Systolic Array Architecture

To separate the modified backprojection algorithm from section 7.1 into a large number of cells suitable for systolic array organisation, each pixel (x,y) becomes an individual process. From equation 7-7 each process involves selecting the correct filtered projection value Q for a given pixel (x,y) determined by the look-up table ψ. After the first projection, the pixel values f(x,y) are circular shifted in the y-direction to achieve rotation of the reconstruction area.

To implement this process each cell must contain

1) Storage for the image cell f(x,y)
2) Storage for the filtered ray value $Q(n\Delta\theta,m\Delta t)$
3) Storage for the pre-calculated ray selection value ψ_w for the pixel (x,y)
4) A counter m to increment the ray number ($m\Delta t$)
5) A comparator to determine whether the ray number equals the pre-calculated ray-selection value ψ_w.
6) An adder to add the appropriate ray values to the image cell

Figure 7-67 shows a block diagram of the hardware for a single block of the systolic array.

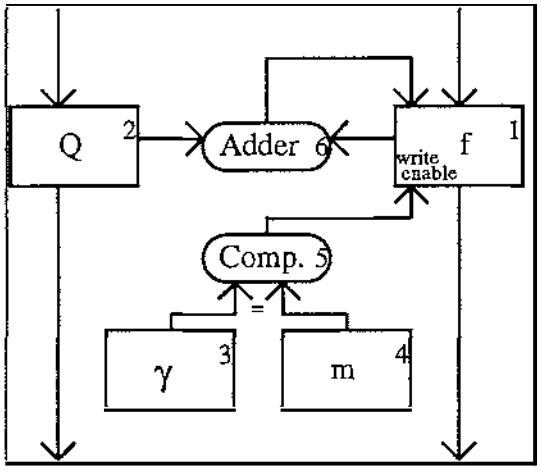


Figure 7-67. Block diagram of single processing cell of systolic array.

The filtered data Q is passed through the array by the block interconnections at regular synchronised time intervals. Each filtered ray value is passed into

the storage cell Q for one cycle only, whereas the pixel values remain in the storage cell f for M cycles (1 projection duration).

Figure 7-68 shows how these blocks are arranged in the systolic array.

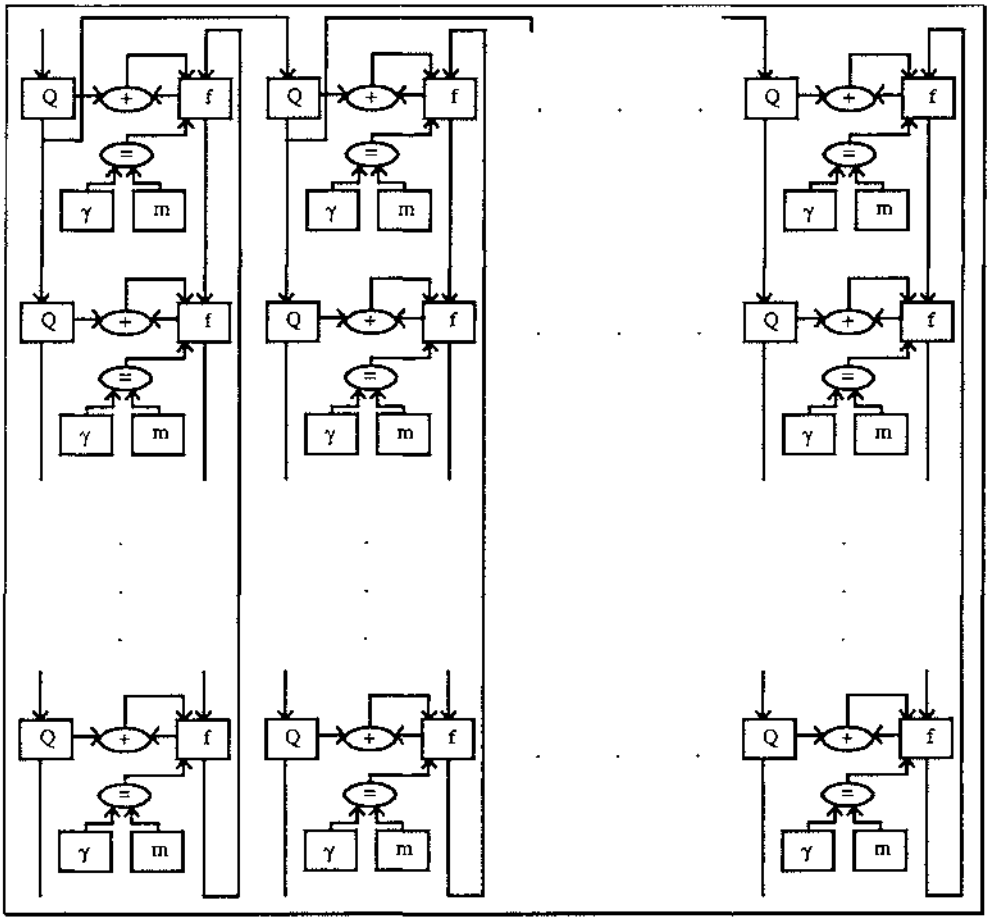


Figure 7-68. Interconnection of processing blocks in a systolic array architecture.

Note from figure 7-68 that there are three variations on the basic cell.

1) The top left cell is the only cell to receive filtered data Q from outside the system. The data is passed to the cell below, and the cell to the right.

2) The top row of cells receive ray data from the cell to the left, and pass the ray value to the right, as well as down to the cell below.

3) The bottom row of cells do not pass the ray data on. However the pixel value f(x,y) is passed to the top cell giving the circular shifting of the reconstruction area.

Because of the cell interconnections the data is *rippled* through the array from top left to bottom right moving one cell per cycle. Processing is parallel going from bottom left to top right (since all diagonal cells process the same ray value Q), and pipelined going from top left to bottom right. All projection data moves through the array in unison, so flip-flops can be used to shift the ray data. However since the pixel values f(x,y) move only when the cell has processed an entire projection, a temporary storage unit f' is required to store the old f value before the new f value is written. Without this the old f value would be overwritten before being shifted to the cell below. The cell layout now becomes

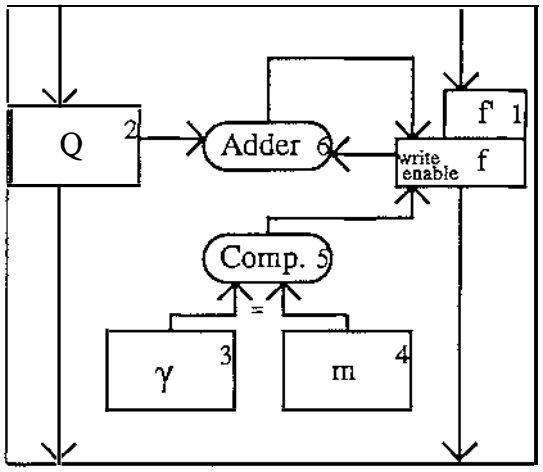


Figure 7-69. Single processing cell showing temporary storage of pixel value at f'.

Note that Y is equal to the number of projections N, so the top cell at position x will be ready to process the first ray of projection n while the bottom cell at position x is processing the first ray of projection n-1. Therefore it is possible for the array of cells to be processing multiple rays from multiple projections at the same time.

7.2.2.1 Discussion of Systolic Array Implementation

The systolic array design shown in figure 7-68 is well suited to parallel processing on any level from a few processors to a VLSI application specific integrated circuit (ASIC). A problem encountered when designing parallel applications is the bottleneck created by the communication between processors. By dividing the array in figure 7-68 into columns where each processor processes any number of adjacent columns the communication between processors is a single one-way channel. For

example a 100^2 array could be processed by ten transputers by dividing the array into ten sections each containing 10×100 cells.

To achieve the fastest implementation of this design a processor per cell would be required. To implement this using transputers or discrete hardware components would be unrealistic because of the size and cost of the system. The processing components in each cell are simple and therefore it may be possible to implement the system on a VLSI ASIC.

One limiting factor of this design is the size of each cell in terms of transistors on a VLSI circuit design. Assuming 12-bit data, and up to 256^2 pixels, the number of storage bits for a single cell would be 60 (12[Q] + 24[f] + 16[γ] + 8[m]). To store a single bit of information using static RAM 8 transistors are required, and about 18 transistors are required for each bit of the Adder. This resulting in about 1000 transistors per cell.

Current state of the art custom designed chips contain approximately 4 million transistors, giving about 4000 cells per chip. It is therefore necessary to divide the array up into multiple chips so that each chip is feasible to implement. As mentioned above this is best done by dividing the array into a number of columns because the flow of communication is primarily vertical, and only 12 output pins and 12 input pins would be required for the data transfer between chips.

A system containing 100^2 cells implemented using this vertical segmentation would require 3 chips.

To give some idea of reconstruction speed let the cycle time for one cell be the same as the pixel processing time for the system using discrete hardware components in section 7.2.1.2 which is 60 nanoseconds. Note that the actual time would be significantly less then this because each cell would be grouped tightly on the silicon substrate reducing the path delays.

The total processing is estimated by the time to propagate the projection data through the array. If the reconstruction area contains 100^2 pixels then the projection data typically contains 100 projections of 100 rays per projection. Therefore the total processing time for a single image is 0.6 milliseconds ($100^2 \times 60$ ns).

Running continuously, this system would reconstruct over 1600 images per second. However the images need to be output from the system, during which time the system is not reconstructing images.

The pixel values f(x,y) are moved through the array in the vertical direction at a rate of 1 shift every projection and are completed after they have shifted through Y cells. It would be desirable to shift the pixels out of the array at the same rate as the data Q is moved through the array, ie. where X (cells in the x-direction) = Y (cells in the y-direction) = N (number of projections) = M (number of rays per projection), then X pixels would be shifted out every N rays (1 projection). The ray data moves through the array as a diagonal wave, so connecting the diagonal pixel values (going from bottom left to top right) to a shift register means that data processed to the same level can be moved out of the array at the same rate as the data is moved into the array.

To ensure that the data in the non-diagonal cells does not get altered a projection set containing null values must be propagated through the array. This simple arrangement causes the rate of reconstruction to be halved, since a null data set is processed after every real data set. For the system with 100^2 pixels, and 100 projections the reconstruction rate would still be over 800 reconstructions per second.

Table 7-4 shows a list of reconstruction scenarios, and their approximate requirements and performance details assuming a cycle time of 60 nanoseconds, and a null projection data set to read out the image.

Table 7-4. Reconstruction hardware required and estimated performance using systolic array architecture.

Size of Reconstruction Area	Number of VLSI ASICs	Reconstruction Rate Images/second
100^2	3	833.3
256^2	17	127.1
512^2	66	31.8
1024^2	263	7.9

7.2.3 Implementations For Divergent-Ray Projection Data

Section 4.2.3 showed that incorporating divergent-ray projection data into the mathematics of the convolution backprojection process altered the convolution kernel, and introduced a weighting factor into the backprojection process based on the pixel distance from the X-ray source.

To incorporate divergent-ray projection data into the backprojection hardware described above, the filtered ray value must be multiplied by a weight for every pixel. To maintain fast speeds the weight would be pre-calculated and stored, adding another storage component to the design, and a high speed multiplier would be required.

A high-speed multiplier chip which may be suitable is the HMU16/883 16× 16-bit CMOS parallel multiplier [Harris Semiconductor Digital Signal Processing Data Book, 1991, pp 2-13] capable of performing a multiplication of two 16-bit numbers within 45 nanoseconds.

The Xilinx 4000 series Field Programmable Gate Arrays do not facilitate multiply functions, so an external multiplier chip would be required. This would increase the number of input and output pins required by the FPGA, increase the complexity of the construction, and reduce the rate of reconstruction possible. Also the pre-calculated weight array would be as large as the image array, requiring separate off-FPGA storage.

The implementation using the discrete hardware components would suffer similar drawbacks as the FPGA implementation.

For the systolic array implementation an additional weight cell would be required per cell, where the weight would remain stationary as does the ray-selection value. A multiplier would also be required for every cell, significantly increasing the complexity of the design.

Clearly the introduction of a multiplier and another storage unit is undesirable for all of the above systems, as it increases the cost and complexity of the hardware, and reduces the performance. A better solution may be to re-bin the divergent-ray projection data as described in section 4.2.3.1 to generate parallel projection data.

7.2.4 Reduced Convolution Kernel

The convolution process involves repetitive multiplication and addition operations. When the convolution kernel is large, for instance the 512 coefficients required to exactly convolve 256 samples of projection data, the hardware required to perform the calculations becomes large and expensive. However, if the size of the convolution kernel is small, for instance the 15 coefficients required to estimate the 256 sample convolution in section 6.2.2, small compact digital signal processing devices (DSPs) can be used.

The convolution process adapts well to the power of parallel/pipelined processing so that the time required for an output value (constituting a number of multiplications and additions) is the same as the time required for a single multiply and addition operation.

Consider a number of data samples X_{-n} , ... X_{-1} ,X_0 , X_1 , X_2 , ... X_n and a number of coefficients C_{-m} , ... C_{-1} , C_0 , C_1, C_2, ... C_m which result in a number of convolved output values Y_{-n} , ... Y_{-1} , Y_0 , Y_1, Y_2, ... Y_n after being convolved together. A single output value is calculated using

$$Y_n = ... + X_{n-2} \times C_{-2} + X_{n-1} \times C_{-1} + X_n \times C_0 + X_{n+1} \times C_1 + X_{n+2} \times C_2 + ... \quad (7\text{-}8)$$

To show how the convolution process is performed using parallel/pipelined techniques consider sample data X passed to 5 processors P_0 .. P_4 in parallel, and 5 convolution coefficients piped through the 5 processors in a circular fashion. This is shown in Table 7-5.

Table 7-5. Parallel/pipelined convolution operation performed using 5 processors.

Step	P_0	P_1	P_2	P_3	P_4	Output
0	$C_2 \times X_0$	0	0	0	0	-
1	$C_1 \times X_1$	$C_2 \times X_1$	0	0	0	-
2	$C_0 \times X_2$	$C_1 \times X_2$	$C_2 \times X_2$	0	0	-
3	$C_{-1} \times X_3$	$C_0 \times X_3$	$C_1 \times X_3$	$C_2 \times X_3$	0	-
4	$C_{-2} \times X_4$	$C_{-1} \times X_4$	$C_0 \times X_4$	$C_1 \times X_4$	$C_2 \times X_4$	$P_0(Y_2)$
5	$C_2 \times X_5$	$C_{-2} \times X_5$	$C_{-1} \times X_5$	$C_0 \times X_5$	$C_1 \times X_5$	$P_1(Y_3)$
6	$C_1 \times X_6$	$C_2 \times X_6$	$C_{-2} \times X_6$	$C_{-1} \times X_6$	$C_0 \times X_6$	$P_2(Y_4)$
7	$C_0 \times X_7$	$C_1 \times X_7$	$C_2 \times X_7$	$C_{-2} \times X_7$	$C_{-1} \times X_7$	$P_3(Y_5)$
8	$C_{-1} \times X_8$	$C_0 \times X_8$	$C_1 \times X_8$	$C2 \times X_8$	$C_{-2} \times X_8$	$P_4(Y_6)$
9	$C_{-2} \times X_9$	$C_{-1} \times X_9$	$C_0 \times X_9$	$C1 \times X_9$	$C_2 \times X_9$	$P_0(Y_7)$

The HSP43881 digital signal processing device [Harris semiconductor, 1991] is ideally suited to this parallel/pipelined task. The HSP43881 chip contains eight processing cells, which perform the multiply-accumulate operation shown above, and a multiplexed output stage to poll the resulting values as they become available.

A simplified representation of the HSP43881 is shown in figure 7-70. The data input bus (X) and coefficient buses (C) are all 8-bits wide, and the output bus (Y) is 26-bits wide. The bus sizes have not been shown in figure 7-70 to simplify the diagram.

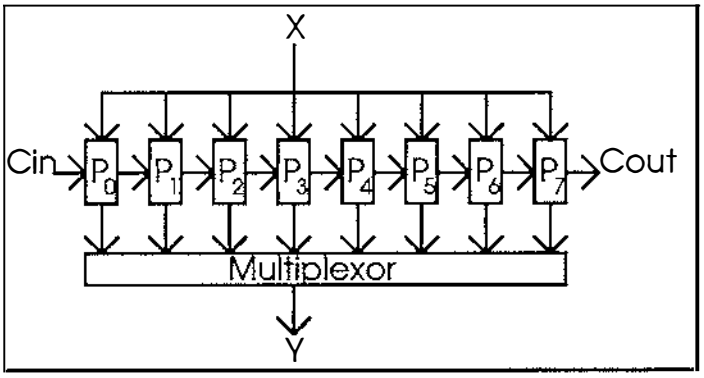


Figure 7-70. A simplified representation of the HSP43881 DSP chip

To process over eight coefficients a number of the chips can be cascaded, allowing convolution kernels with over 1000 coefficients if required. Figure 7-71 shows two HSP43881 chips cascaded with the coefficient output joined to the coefficient input providing the mathematical functionality required for a 16-coefficient convolution operation.

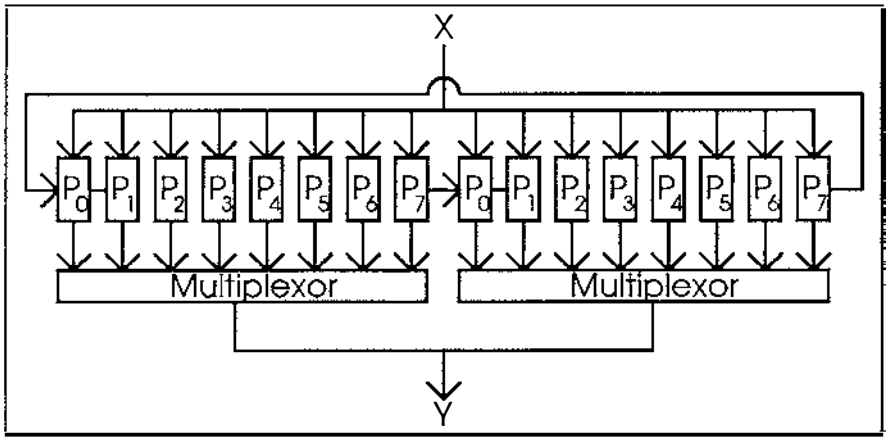


Figure 7-71. Two HSP43881 DSP chips cascaded to provide a parallel/pipelined convolution operation.

Note that the output stage of the HSP43881 chip has a 3-state buffer so that a number of output busses can be connected to one common output bus without the need for additional hardware.

The number of data samples required to produce for an image of 240^2 pixels using the convolution backprojection method (such as those shown in section 6.2.2.2) is typically $240 \times 120 = 28800$. Table 7-5 showed that a convolved output value is calculated in a single multiply/add step (not taking into consideration the switching and propagation time for the multiplexer output stage) using the above hardware configuration. Therefore to perform a 16-coefficient convolution operation in real-time (ie. convolving 25 complete data sets each second), the above hardware is required to perform $28800 \times 25 = 720000$ steps every second, ie 720 kHz. The maximum step time for HSP43881 DSP chip is 30 MHz including propagation and switching delays.

The HSP43881 DSP chip provides a simple solution for convolution of the sampled projection data in real-time.

7.3 COMPOSITE ANALOGUE/DIGITAL HARDWARE IMPLEMENTATION

This section is included not as a practical implementation, but to show the reader the ideas which prompted the modified backprojection algorithms described previously.

Consider the setup shown in figure 7-72 where an oscilloscope driven by a digital address calculation system (DACS) is used to display projection data from the data acquisition system (DAS), and the light from the projection data at each position is integrated using a charge coupled device (CCD) cell.

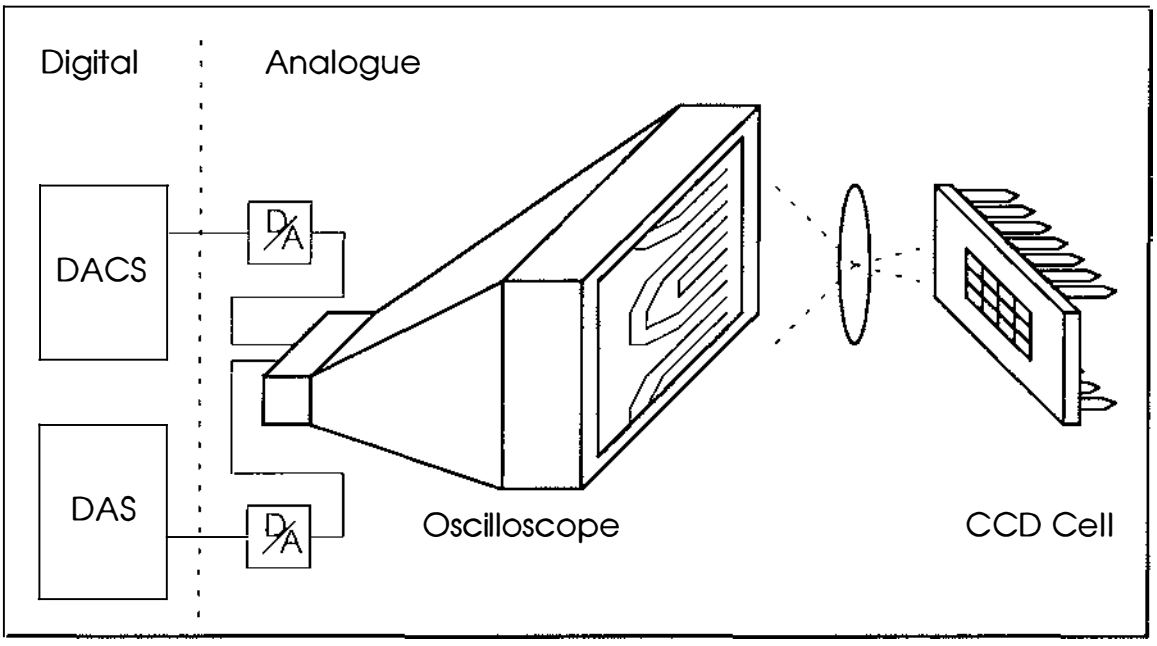


Figure 7-72. Setup of composite analogue/digital backprojection system.

The CCD cell represents the transformed reconstruction area where increments in the x and y directions indicate radial and angular increments respectively. An optical lens is used to focus the light from the oscilloscope display onto the CCD cell.

The filtered projection data are converted into an analogue data stream and used to modulate the light intensity (z axis) of the oscilloscope. A digital address calculation system is used to calculate the position of the scope beam so that the correct pixels are illuminated. This signal is converted into analogue and fed into the x and y axis controls of the scope.

It is undesirable for the oscilloscope to emit light while the scope is moving between pixel locations because the light falls on pixels which are not part

of the ray being processed. Therefore the z-axis is multiplied by a clock oscillating at a frequency equal to the rate at which the x,y-axis is being changed (figure 7-73).

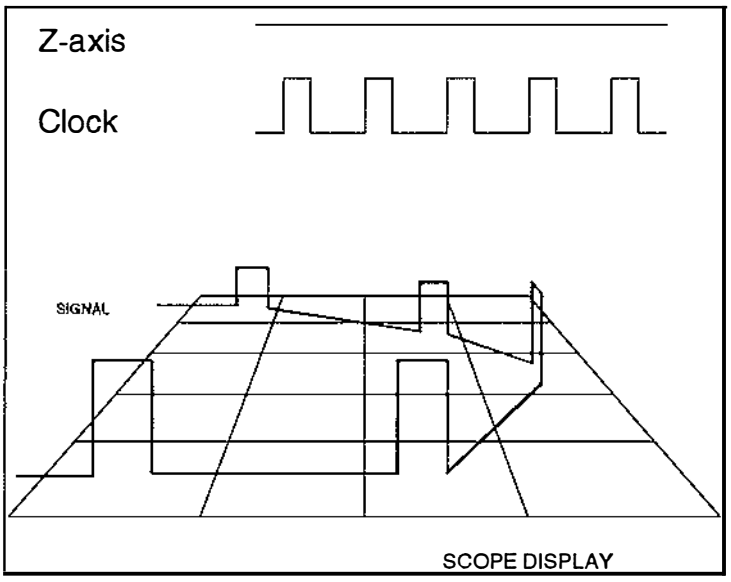


Figure 7-73. Z-axis multiplied by an oscillator.

The light from each projection is integrated by the CCD cell representing the summation of the ray contributions for a given pixel (x,y). After all of the projections have been processed by the oscilloscope the array can be read out, converted to digital, and interpolated into the cartesian domain for display by a computer.

The address calculation system is similar to those used in the digital implementations of section 7.2.1. A look-up table contains the scope locations of the ray paths for the first projection. An offset is added to these values for subsequent rotated projections.

The main problems in realising the above backprojection unit are

1. High cost. The cost of an oscilloscope, and CCD cell capable of operating at fast speeds is high.

2. Ghosting. The persistence of the phosphor on the oscilloscope can only be chosen to within certain limits, and a certain amount of ghosting is inevitable.

3. Critical light interface. The interface between the oscilloscope and the CCD cell is critical. Any diffraction or deflection results in degradation of the image quality.

4. Restricted by digital section. The digital address calculation system will restrict the maximum speed of the system.

5. Difficult construction. Constructing the system would be extremely difficult because the image quality is critically dependant upon the relative position of the oscilloscope, lens and CCD cell. This means that any vibration would cause degradation of image quality.

In view of the above problems associated with the composite analogue/digital backprojection unit, it is recommended that one of the purely digital methods described in section 7.2.1 be used for implementation of the modified backprojection algorithm.

Chapter 8

8 CONCLUSIONS

Advances in technology have opened up many new areas for application. One example which is specific to New Zealand is the detection of the defect core in felled logs. New Zealand is one of the few countries where trees are pruned as they grow, creating long lengths of knot free timber from the pruned stumps outwards. The central core containing the pruned branches and sap pockets caused by the pruning is known as the defect core. It is advantageous to both the log grower and the buyer to know the whereabouts of the defect core so that the log can yield the maximum amount of clear defect free wood.

Many technologies exist for non-destructive examination of internal structures. However only X-ray transmission tomography is well suited to defect detection in logs. The physical factors which hinder the other imaging methods include the large size of the cross-section being imaged, the problems associated with directly coupling the sensors to the log, the low water content in the logs, and the complexity of the associated reconstruction algorithms.

X-ray tomography has been in use for 17 years [Webb, S 1988], and many methods of image reconstruction have been developed. The most commonly used method is the convolution and backprojection method, where the X-ray data is filtered in the spatial domain by means of convolution, and backprojected over the reconstruction area. Algebraic reconstruction methods are used in preference to the convolution backprojection method for many industrial applications where X-ray data acquisition is restricted due to physical constraints. However the application in question does not suffer from physical limitations. Maximum entropy methods are suggested for producing results with reduced noise (maximum order). However this is achieved at the expense of

computational complexity, which results in longer reconstruction times which is not favourable for the log-defect application.

For some applications the data acquisition can be performed at real-time speeds, and the data stored for off-line image reconstruction at a later time. However other applications require real-time data acquisition, and real-time image reconstruction. The log defect problem is one such application.

A few real-time reconstruction systems exist, although they are all expensive and complex to construct. The author has shown how it is possible to produce an inexpensive reconstruction system by reducing the complexity of the computation.

The process of convolution is computationally expensive when using a full width convolution kernel. However the Ramachandran-Lakshminarayanan convolution filter exhibits the property of Chebychev or mini-max economisation, which means that the convolution kernel coefficients may be truncated at any point, and still yield the best approximation to the full kernel. Results carried out by the author in this thesis show that the convolution kernel can be truncated to only a few terms and still produce images showing most of the relatively high-contrast information. The density of the defects within the defect core is significantly higher than the surrounding clear-wood, so it may be possible to significantly reduce the size of the convolution kernel without losing the information required to detect the defects from the surrounding clear-wood.

The author offers a digital hardware design for the convolution process suitable for convolving projection data with a truncated convolution kernel. The design is extremely simple, utilising for the main part two digital signal processing chips cascaded to form a highly parallel/pipelined convolver.

As an alternative to reducing the amount of data used, which causes reduction in the image quality, the author suggests modifying the mathematical processes underlying backprojection to achieve high-speed reconstruction without loss of image quality. The standard backprojection process uses rectangular pixels in the reconstruction space. However the data acquisition system is inherently circular, causing complexities in the mathematical mapping of the rays to the reconstruction area. By dividing the reconstruction area into sectors of equal size, where the sector angle is equal to the projection angular increment, a polar type coordinate system

employing a radial component, a sector index, and an angular component within the sector, can be used to reduce the complexity of the backprojection mapping.

Using parallel ray data, and nearest neighbour interpolation, the modified backprojection process is well suited to fast table look-up methods. As a result only addition is required, and many high-speed low-cost implementations are possible.

If fan-beam ray data is used then the data can be re-binned to provide parallel ray data. Although the mathematical overheads incurred by re-binning are small in comparison with the reconstruction process, it requires all of the projection data for the image before re-binning can begin. If this is not practical then each ray value must be multiplied by an additional weight before backprojecting to any pixel, and a weighting factor must be added to the convolution kernel. This significantly increases the mathematical complexity of the modified algorithm, although it is still less then the standard convolution backprojection algorithm.

The author offers three different implementations of the modified backprojection algorithm, ranging from simple and inexpensive to not yet possible with the current level of technology. The first implementation employs a Field Programmable Gate Array for all operations except storage of large arrays, which require separate high-speed random access memory. This design is estimated as being capable of reconstructing about 8 images (100^2 pixels) per second, and it is inexpensive and simple to construct (only 3 chips).

The second implementation uses the same schematic design as the Field Programmable Gate Array, although discrete hardware components are used. The main advantage is the reduction in propagation delay between functional blocks. This design is estimated at reconstructing at 12 images per second using worst case timing analysis, and about 17 images (100^2 pixels) per second using typical timing. However the increase in speed is traded off against the complexity of the hardware construction process.

The third implementation requires the modified backprojection algorithm to be rearranged into a systolic array architecture. In this form the design is well suited to parallel processing. Clearly the performance is determined by the number of processors and the power of those processors. At its limit,

where one cell is implemented by one simple processor block (with all 100^2 processor blocks on one VLSI ASIC) the processing times are estimated at 833 images (100^2 pixels) per second. Unfortunately with current technology, it is not possible to build a single chip device with enough transistors to implement this.

Any of the implementations discussed would produce images of suitable quality for the detection of the defect core within felled logs, and the second two implementations have reconstruction rates which are likely to be fast enough for the defect detection application.

8.1 Future Work

This thesis discusses only two reconstruction area discretisations suitable for the modified backprojection method. Investigation into other discretisations may uncover discretisations with properties useful for different applications.

The reconstructions produced using area interpolation were of noticeably better quality then those using nearest neighbour interpolation. However a multiplication operator is required for the latter, complicating the simplicity of the former which uses only addition. If a more elaborate interpolation scheme could be introduced without the added complexity of multiplication the image quality would be increased.

The systolic array architecture can be broken into a number of linked processing units where each unit processes a number of columns. The processing within each unit is likely to be sequential, and more efficient unit processing methods could be developed to further increase the rate of reconstruction for composite parallel/sequential systems.

References

Baringer, W.B., Brodersen, R.W., and Petkovic, D. (1991) Computer Vision Hardware using the Radon Transform. Proceedings of the IEEE, Vol.5, pp508-513.

Benson-Cooper, D.M. and Knowles, R.L. and Thomson, F.J. and Cown, D.J. (1982). Computed tomographic scanning of the detection of defects within logs. FRI Bulletin No.8, Forest Research Institute, New Zealand.

Birks, J.B. (1964). The theory and practice of Scintillation Counting. London : Pergamon Press Ltd.

Brown, J. G. (1966). X-rays and their applications. London : ILIFFE Books Ltd.

Browne, J.A. and Holmes, T.J. (1992) Developments with Maximum Likelihood X-ray Computed Tomography. IEEE Transactions on Medical Imaging, Vol.11 (No.1), pp40-52.

Censor, Y. (March 1983). Finite Series-Expansion Reconstruction Methods. Proceedings of the IEEE, Vol.71 (No.3), pp351-355.

Cho, Z.H. and Chen, C.M. and Lee, S.Y. (1990). Incremental Algorithm - A New Fast Backprojection Scheme for Parallel Beam Geometries. IEEE Transactions on Medical Imaging, Vol.9 (No.2), pp207-217.

Dusaussoy, N.J. and Abdou, I.E. (1991). The Extended MENT Algorithm: A Maximum Entropy Type Algorithm Using Prior Knowledge for Computerized Tomography. IEEE Transactions on Signal Processing, Vol.39 (No.5), pp1164-1180.

Eggermont, P.P.B. and Herman, G.T. and Lent, A. (1981). Iterative algorithms for large partitioned linear systems, with application to image reconstruction. Linear Algebra and Its Applications, Vol.34, pp37-67.

Garden, K. and Mnyama, D. (1987). Timber grading improved by computed tomography. New Electronics Magazine, February edition, pp37-40.

Gmitro, A.F. and Tresp, V. and Gindi, G.R. (1990). Videographic Tomography-Part I:

Reconstruction with Parrallel-Beam Projection Data. IEEE Transactions on Medical Imaging, Vol.9 (No.4), pp366-375.

Goutis, C.E. and Drossos, S.N. (1984). Fast iterative algorithm for reconstruction from divergent-ray projections. Proceedings of the IEE Pt.E, Vol.131 (No.3), pp89-96.

Harris Semiconductor, (1991). Digital Signal Processing Data Book.

Helgason, S. (1980). The Radon Transform. Boston, U.S.A.: Birkhauser.

Integrated Device Technology, (1987). High Speed CMOS Data Book.

Kak, A.C. (1988). Principles of Computerized Tomographic Imaging. New York, U.S.A.: IEEE Press.

Kingswood, N. and Dagless, E.L. and Belchamber, R.M. and Betteridge, D. and Lilley, T. and Roberts, J.D.M. (1986). Image Reconstruction using the Transputer. IEE Proceedings, Vol.133 (Pt.E, No.3), pp139-144.

Kreyszig, E. (1983). Advanced Engineering Mathematics (fifth edition). U.S.A. : John Wiley & Sons, Inc.

Larsen, A. (1992). Export market opportunities for radiata pine solid wood products. Notes from 1992 Lower North Island Wood Processors' Conference. Ministry of Forestry, Palmerston North, New Zealand, pp 22-57.

Lewitt, R.M. (1983). Reconstruction Algorithms: Transform Methods. Proceedings of the IEEE, Vol.71 (No.3), pp390-408.

Levack, H. (1986). Forestry Handbook. Institute of Foresters (Inc.), New Zealand. Notes from 1992 Lower North Island Wood Processors' Conference. Ministry of Forestry, Palmerston North, New Zealand, pp 58-68.

Matej, S. (1990). New Hybrid spline-linear interpolation for the fast CT and MR imaging. Proceedings of the European Signal Processing Conference, Vol.2 (No.5), pp909-912.

Matej, S. and Bajla, I. (1990). A High-Speed Reconstruction from Projections Using Direct Fourier Method with Optimized Parameters - An Experimental Analysis. IEEE Transactions on Medical Imaging, Vol.9 (N o.4), pp421-429.

McConchie, D. (1992). The changing Resource - New Crop Radiata Pine. Notes from 1992 Lower North Island Wood Processors' Conference. Ministry of Forestry, Palmerston North, New Zealand, pp 58-68.

Ministry of Forestry (1991). New Zealand Forestry Statistics 1991 Wellington, New Zealand.

Roder, F.L. and Scheinman, E. and Magnuson, P. (1986). High Speed CT Scanning of Logs. Imatron Inc.

Tabei, M. and Ueda, M. (1990). Backprojection with Fourier Series Expansion and FFT Proceedings of the IEEE, February, pp 1853-1856.

Wallace, G. (1991). PortaCAT : An introduction to a new system for quantitative assessment of power pole integrity. Report by Nuclear Sciences Group. DSIR, Lower Hutt, New Zealand.

Webb, S. (1988). The Physics of Medical Imaging. Philadelphia, U.S.A.: IOP Publishers.

Wylie, R. E. J. and Allan, G. P. S. and Frost, I. A. and McKenzie, G. R. (1984). Radiata Pine - A basis for selection of trees for pruning and thinning. New Zealand Forest Services, Wellington, New Zealand.

Bibliography

Andersen, A.H. (1989). Algebraic Reconstruction in CT from Limited Views. IEEE Transactions on Medical Imaging, Vol.8 (No.1), pp50-55.

Axel, L. and Herman, G.T. and Roberts, D.A. and Dougherty, L. (1990). Linogram Reconstruction for Magnetic Resonance Imaging (MRI). IEEE Transactions on Medical Imaging, Vol.9 (No.4), pp447-449.

Burden, R.L. and Faires, J.D. (1985). Numerical Analysis (3rd ed.). Boston, U.S.A.: Prindle, Weber and Schmidt Publishers.

Challis, R.E. and Kitney, R.I. (1990). Time-domain methods (Part 1 of 4 in Biomedical signal processing). Medical and Biological Engineering and Computing, Vol.28, pp509-524.

Challis, R.E. and Kitney, R.I. (1991). The frequency transforms and their inter-relationships (Part 2 of 4 in Biomedical signal processing). Medical and Biological Engineering and Computing, Vol.29, pp1-17.

Challis, R.E. and Kitney, R.I. (1991). The power spectrum and coherence function (Part 3 of 4 in Biomedical signal processing). Medical and Biological Engineering and Computing, Vol.29, pp225-241.

Crawford, C.R. (1991). CT Filtration Aliasing Artifacts. IEEE Transactions on Medical Imaging, Vol.10 (No.1), pp99-102.

Donohue, K.D. and Saniie, J. (1989). A Scanning and Sampling Scheme for Computationally Efficient Algorithms of Computer Tomography. IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol.37 (No.3), pp402-414.

Dudeon, M. (1984). Multi Dimensional Digital Signal Processing. London: Prentice Hall.

Garden, K. (1984). An Overview of Computed Tomography. Thesis from University of Canterbury, New Zealand.

Glick, S.J. and King, M.A. and Penny, B.C. (1989). Characterisation of the Modulation Transfer Function of Discrete Filtered Backprojection. IEEE Transactions on Medical Imaging, Vol.8 (No.2), pp203-213.

Golub, G.H. and Van Loan, C.F. (1983). Matrix Computations. Baltimore, Maryland 21218: Johns Hopkins University Press.

Herman, G.T. (1980). Image Reconstruction from Projections. New York, U.S.A.: Academic Press.

Herman, G.T. and Lent, A. (1976). Iterative Reconstruction Algorithms. Computer, Biological and Medical, Vol.6, pp273-294.

Herman, G.T. and Lent, A. and Lutz, P.H. (1978) Relaxation methods for image reconstruction. Communications ACM, Vol.21, pp152-158.

Jaffe, J.S. (1990). Limited Angle Reconstruction Using Stabilized Algorithms. IEEE Transactions on Medical Imaging, Vol.9 (No.3), pp338-344.

Jain, A.K. (1989). Fundamentals of Digital Image Processing. U.S.A.: Prentice Hall.

Lange, K. (1990). Convergance of EM Image Reconstruction Algorithms with Gibbs Smoothing. IEEE Transactions on Medical Imaging, Vol.9 (No.4), pp439-445.

Lewitt, R.M. (1977). Contributions to Image Reconstruction. Thesis from University of Canterbury, New Zealand.

Louis, A.K. (March 1983). Mathematical Problems of Computerised Tomography. Proceedings of the IEEE, Vol.71 (No.3), pp379-389.

Llacer, J. and Veklerov, E. (1989). Feasible Images and Practical Stopping Rules for Iterative Algorithms in Emission Tomography. IEEE Transactions on Medical Imaging, Vol.8 (No.2), pp186-193.

Macovski, A. (1983). Physical Problems of Computerised Tomography. Proceedings of the IEEE, Vol.71 (No.3), pp373-378.

Manglos, S.H. and Thomas, F.D. and Capone R.B. (1991). Attenuation Compensation of Cone Beam SPECT Images Using Maximum Likelihood Reconstruction. IEEE Transactions on Medical Imaging, Vol.10 (No.1), pp66-73.

McCready, V.R., Leach, M. and Ell, P.J. (1987). Functional Studies using NMR. Great Britain: Springer-Verlag.

McIlroy, C.D. and Linggard, R. and Monteith, W. (1984). Hardware for Real-time Image Processing. IEE Proceedings, Vol.131 (Pt.E, No.6).

Monteith, W. and Chaplin, R.I. and Hodgson, R.M. (1988). High Speed X-ray Image Reconstruction for Log Evaluation. Interim Report, Department of Technology, Massey University, New Zealand.

Nunez, J. and Llancer, J. (1990). A Fast Bayesian Reconstruction Algorithm for Emission Tomography with Entropy Prior Converging to Feasible Images. IEEE Transactions on Medical Imaging, Vol.9 (No.2), pp159-171.

Ollinger, J.M. (1990). Iterative Reconstruction-Reprojection and the Expectation-Maximization Algorithm. IEEE Transactions on Medical Imaging, Vol.9 (No.1), pp94-98.

Oskoui, P. and Stark, H. (1989). A Comparitive Study of Three Reconstruction Methods for a Limited-View Computer Tomography Problem. IEEE Transactions on Medical Imaging, Vol.8 (No.1), pp43-49.

Paudler, W.W. (1987). Nuclear Magnetic Resonance, general concepts and applications. New York, U.S.A.: John Wiley and Sons, Inc.

Peng, H. and Stark, H. (1989). One-Step Image Reconstruction from Incomplete Data in Computer Tomography. IEEE Teansactions on Medical Imaging, Vol.8 (No.1), pp16-31.

Politte, D.G. and Snyder, D.L. (1991). Corrections for Accidental Coincidences and Attenuation in Maximum-Likelihood Image Reconstruction for Positron-Emission Tomography. IEEE Transactions on Medical Imaging, Vol.10 (No.1), pp82-89.

Tollner, E.W. and Davis, J.W. and Verma, B.P. (1989). Managing Errors with X-ray Computed Tomography (X-ray CT) When Measuring Physical Properties. Transactions of the American Society of Agricultural Engineers, Vol.32 (No.3), pp1090-1096.

APPENDIX A

(Pascal program for projection data generation)

```
Program ParallelProjections;
{generates parallel projection data given a digital image}

const
  Size          = 240;
  Rays          = 240;
  Projections   = 120;   {half of 240 since using parrallel projections}

type
  Image = array[0..Size-1,0..Size-1] of byte;
  Proj  = array[0..Projections-1,0..Rays-1] of word;

var
  ImagePtr : ^Image;
  ProjPtr  : ^Proj;

Procedure Init;
{initialize pointers to projection data and image data}
  begin
    new(ImagePtr);
    new(ProjPtr);
  end; {*Init*}

Procedure ReadImage;
{read in image from disk. Nb file stored as either binary or text.}
  var
    IFName:string;
    ImFile:file;
    c1,c2:integer;
  begin
    {$I-}           { disable IO checking so IOResult can be used }
    Write('Enter Phantom Image : ');
    Readln(IFName);
    Assign(ImFile, IFName);
    Reset(ImFile, Size*Size);
    if (IOResult <> 0) then Exit;   { Quit & return problem in Result }
    BlockRead(ImFile, ImagePtr^, 1);
    {$I+}
    if (IOResult <> 0) then Exit; { Quit & return problem in Result }
    Close(ImFile);
  end; {*ReadImage*}


Procedure WriteProj;
{write projection data to disk as binary file.}
  var
    OFName:string;
    OpFile:file;
  begin
    {$I-}           { disable IO checking so IOResult can be used }
{    Write('Enter Projection File : ');
    Readln(OFName);}
    OFName := 'phant1.prj';
    Assign(OpFile, OFName);
```

```
   Rewrite(OpFile, Rays*Projections);
   if (IOResult <> 0) then Exit;   { Quit & return problem in Result }
   BlockWrite(OpFile, ProjPtr^, 2); {2 used because words not bytes}
   {$I+}
   if (IOResult <> 0) then Exit; { Quit & return problem in Result }
   Close(OpFile);
  end; {*WriteProj*}

function modulus(a,b:real):real;
{perform modulo arithmetic on real numbers. Nb not supported in Turbo Pascal 5.5}
begin
  while (a>b) do a:=a-b;
  modulus := a;
end;

function max(a,b:real):real;
{return maximum of two numbers}
begin
  if a>b then max:=a else max:=b;
end;

Procedure CalcProj;
{calculate the projection data. The procedure calculates the contribution of each pixel to the}
{appropriate ray in the current projection. After the projection is complete, the process is repeated}
{for the next projection}
  var
   TempProj:array[0..Rays-1] of Real;
   Theta,T,X,Y,Z:Real;
   C1,C2,C3:integer;
   PixWidth,A:Real;
   X1,X2,hyp,B,PW,Temp:real;
   Maxi,Mini,A1,A2,Area:real;

 begin
  PixWidth := sqrt(2)/Size;
  PW := Pixwidth/(2/rays);
  Hyp := PW*Sqrt(2)/2;

  {An Array of Real is required to add the projection contributions}
  {Then they are rounded into the Word storage array}
   for C3 := 0 to Projections-1 do  {* Theta *}
   begin
    for C1 := 0 to Rays-1 do TempProj[C1] := 0;   {Reset Array}
    Write(C3,' : ');
    Theta := C3 * pi/Projections;
    for C1 := 0 to Size-1 do  {* X *}
     for C2 := 0 to Size-1 do  {* Y *}
     begin
      X := (C1+0.5-size/2)/(size/2) * Sqrt(2)/2;   {From -0.707 to 0.707}
      Y := (C2+0.5-size/2)/(size/2) * Sqrt(2)/2;   {From -0.707 to 0.707}
      T := X*Cos(Theta) + Y*Sin(Theta);
      T := T + 1;                    {Shift from -1 -> +1  to  0 -> 2}
      Z := T/(2/Rays);    {gives ray number in continuous eg. ray 7.34}
      A := frac(Z);
```

```
    {Next section calculates contribution area of square pixels}
        X1 := Hyp*abs(cos(Theta+pi/4));
        X2 := Hyp*abs(cos(Theta-Pi/4));
        if X2<X1 then begin Temp:=X1; X1:=X2; X2:=Temp; end;
        if ((modulus(theta+pi/4,pi)>0) and  {calcs max ray length in pixel}
           (modulus(theta+pi/4,pi)<pi/2)) then
            B := PW/abs(cos(theta))
        else B := PW/abs(sin(theta));
        Area := sqr(PW);
        if (A+X2 > 1) then Mini := 1 else Mini := A+X2;
        if (A-X2 < 0) then Maxi := 0 else Maxi := A-X2;
        A1:=0; A2:=0;
        if (Maxi=0) then begin
          if (X2-X1)<>0 then
            A1:=((X2-max(A,X1))/(X2-X1))*B/2 * (X2-max(A,X1));
          if (A<X1) then A1:=A1+B*(X1-A);
          TempProj[Trunc(Z)-1]  :=TempProj[Trunc(Z)-1]+
                        ImagePtr^[C1,C2]*(A1/Area);
        end;
        if (Mini=1) then begin
          if (X2-X1)<>0 then
            A2:=((X2-max((1-A),X1))/(X2-X1))*B/2 * (X2-max((1-A),X1));
          if ((1-A)<X1) then A2:=A2+B*(X1-(1-A));
          TempProj[Trunc(Z)+1]  :=TempProj[Trunc(Z)+1]+
                         ImagePtr^[C1,C2]*(A2/Area);
        end;
        TempProj[Trunc(Z)]  :=TempProj[Trunc(Z)]+
                        ImagePtr^[C1,C2]*((Area-A1-A2)/Area);
      end; {*For C2&C1*}
     {Now scale Reals and put into Word array}
     for C2 := 0 to Rays-1 do ProjPtr^[C3,C2]:=round(TempProj[C2]);
     for C2 := 0 to Rays-1 do write(ProjPtr^[C3,C2],' ');
     writeln;
   end; {*For C3*}
  end; {*CalcProj*}

Procedure Terminate;
{dispose of the pointers formally}
  begin
    dispose(ImagePtr);
    dispose(ProjPtr);
  end; {*Terminate*}

begin {*Main*}
  Init;
  readimage;
  CalcProj;
  WriteProj;
  Terminate;
end.  {*Main*}
```

APPENDIX B

(Pascal program for image reconstruction)

(using Algebraic Reconstruction Technique)

```
program iterative_art;
{Reconstructs an image using the Algebraic Reconstruction Technique}
{Relaxation is optional}
uses
 graph,dos,crt;

const
 size          = 101;  {size of image array}
 lambda        = 1.0;  {value for changing sensitivity of image maintenance}
 radius        = 2.0;  {radius of X-ray source when fan-beam data used}
 max_rays      = 256;
 blank         = 9999; {used as an indicator for no real value}

 strlen = 12;     {length of one unit in output file}
 deci = 8;     {number of decimal places for data}

type
 ImageArrayRow =array[trunc(-size/2)..trunc(size/2)] of byte;
 ImageArray = array[trunc(-size/2)..trunc(size/2)] of ImageArrayRow;
 ImageArrayPtr=^ImageArray;
 pro_data = record
        angle:real;
        pro:array[-trunc(max_rays/2)..trunc(max_rays/2)] of integer;
        end;
 cell = record
      x:real;  {defines distance from centre of cell to ray}
      x1,y1:integer;  {picture element reference}
      end;

var
 est: ImageArrayPtr;      {estimate image}
 estpro1,                 {projection data through estimate}
 actpro1,                 {projection through actual image}
 respro1:pro_data;        {= actpro - estpro}
 ray_spread:real;         {angular spread of rays}
 num_rays:integer;        {number of rays within the angular spread}
 D_alpha:real;            {angle between rays - calculated from spread & num_rays}
 alpha:real;              {angle between ray and beta = ray_num * D_alpha}
 pix_width:real;
 nfile1,nfile2,
 opfname:string;
 graphmode,
 graphdriver:  integer;
 count1,count2,count3,
 itt_num,                 {number of itterations performed}
 itt_sep: integer;        {number of itterations between actual picture changes}
 angle:    real;          {for random spread projections}
 beta:real;               {angle of fan}
 init_ang,                {initial projection angle}
 temp:real;
 ang_disp:real;           {angular displacement between projections}
 lf,cr:char;
 act_moving:char;         {indicates if the actual picture file is changing}
```

```
function tan(ang:real):real;
begin
  tan := sin(ang)/cos(ang);
end;

Procedure ReadEstFile(fname:string; num:integer);
{Read the current estimate file from the disk}
var
  c1:integer;
  s1:string;
  ImFile    : file;
begin
  str(num,s1);
  fname := fname + '.' + s1;
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, FName);
  Reset(ImFile, SizeOf(ImageArrayRow));
  BlockRead(ImFile, Est^, Size);
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
  if (num mod itt_sep) <> 0 then erase(ImFile);
end;  { LoadImages }

procedure readactfile(fname:string; num:integer);
{read the actual file, simulating the object being reconstructed, from the disk}
var
  c1:integer;
  s1:string;
  ImFile : file;
begin
  if act_moving = 'Y' then
  begin
    str(num,s1);
    fname := fname + '.' + s1;
  end;
  writeln;  writeln('Reading in actual picture from ',fname);
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, FName);
  Reset(ImFile, SizeOf(ImageArrayRow));
  BlockRead(ImFile, Est^, Size);   { Read Half image from file }
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;


procedure writefile(fname:string; num:integer);
{Write the current estimate file to the disk}
var c1:integer;
    s1:string;
    ImFile : file;
begin
  str(num,s1);
```

```
  fname := fname + '.' + s1;
  writeln;  writeln('Writing out estimate picture to ',fname);
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, FName);
  Rewrite(ImFile, SizeOf(ImageArrayRow));
  BlockWrite(ImFile, Est^, Size);   { Read Half image from file }
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;

procedure initialise;
{obtain from the user all of the details required to perform the reconstruction, including}
{file name of actual object, file name to write estimate object, number of iterations, angle}
{between iterations, the value of the relaxation parameter (lambda) if relaxation is required, etc.}
var
  dum:char;
  dummy,name:string;
  i1,i2,t:integer;
begin
graphmode := detect;
graphdriver := detect;
clrscr;
writeln('****************************************');
writeln('******              Iterative ART            ******');
writeln('****************************************');
lf := chr(010);
cr := chr(013);
writeln('Lambda is ',lambda:5:2);
pix_width := 1/(sqrt(2)*size/2);
if trunc(size/2)=(size/2) then
begin  writeln('Size must be odd!! **FATAL ERROR**'); halt; end;
{Size is restricted to odd values so that the central ray does not traverse between two columns of pixels)
write('Please enter file containing initial estimate picture (no ext.) : ');
readln(nfile1);
write('Enter file with initial actual picture (no ext. if changing): ');
readln(nfile2);
write('Does the actual picture change (Y/N): ');
readln(act_moving);
act_moving := upcase(act_moving);
if act_moving = 'Y' then
  begin
    write('How many itterations per input image change : ');
    readln(itt_sep);
  end else itt_sep := 1;
write('Please enter the angle of the first projection : ');
readln(init_ang);
write('Please enter the angular displacement between projections : ');
readln(ang_disp);
write('How many itterations would you like performed : ');
readln(itt_num);
{write('Please enter initial Lambda : ');
readln(lambda);
write('Does Lambda change (Y/N): ');
```

```
readln(dum);
if upcase(dum) = 'Y' then
  begin
    write('Please enter increment per itteration : ');
    readln(itt_lambda);
  end else itt_lambda := 1;
}
write('Please enter the output file name (no extension allowed!) : ');
readln(opfname);
write('What is the angular spread of rays :');
readln(ray_spread);
repeat
  write('Please enter the number of rays per projection :');
  readln(num_rays);
  if num_rays > max_rays then writeln('Current maximum rays is ',max_rays);
until num_rays <= max_rays;
D_alpha := ray_spread/(num_rays-1);
writeln('**Initialising picture array**');
new(est);
for i1:= round(-size/2) to round(size/2) do
  for i2 := round(-size/2) to round(size/2) do
    est^[i1,i2] := 0;
end;


function deg(angle:real):real;
{returns an angles value in radians}
begin
        deg := angle * PI / 180;
end;

function pct(X,Y:integer; T:boolean):integer;
{translate the image array if the angle is not between 45 and 135 degrees}
{so that the same procedures may be used for all projection angles}
begin
  if T then pct := est^[Y,X]
  else pct := est^[X,Y];
end;

procedure calc_ray(beta:real; alpha:real; var proval:integer);
{calculate an estimate of the current ray given the current estimage}
{This procedure moves vertically across the image array, calculating the position of the ray}
{in row n based on the abgle and position of the ray in row n-1.}
{
var
  ang,m,c,c2,x,x2,y,h1,h2:real;  {for doing line calculations}
  box:cell;    {the cell overlay containing floating line info}
  tran:boolean;   {indicates whether to transpose line or not}
  intg:real;      {stores integral value before passing to proval}
begin
  proval := 0;
  intg := 0;
  tran := false;
  ang := beta-alpha;
```

```
 if ((ang<>0) and (ang<>90) and
    (ang<>180) and (ang <> 270)) then   {angle ok to do calcs}
 begin
  m := tan(deg(ang));
  c := radius*sin(deg(beta))-tan(deg(ang))*radius*cos(deg(beta));
  if abs(m) < 1 then
  begin
   tran := true;
   m := 1/m;
   c := radius*cos(deg(beta))-radius*sin(deg(beta))/tan(deg(ang));
   ang := arctan(m)*(180/pi);
  end;
{Find first point on grid where ray hits and set box cell to just before it}
  c2 := -sin(deg(45));   {=0.707 ie. y-intercept of bottom line of grid}
  x := (c2-c)/m;
  if abs(x)<((size/2)*pix_width) then  {ray hits bottom line of grid}
  begin
   box.x1 := round(x/pix_width);
   box.y1 := trunc(-size/2);
   box.x := x - box.x1*pix_width;    {see page 3 of attached notes}
  end else
  begin  {doesn't hit bottom}
   x2 := -(m/abs(m))*sin(deg(45));  {x-intercept of side of grid where
                              ray is lowest.  See page 3}
   y := m*x2 + c;
   if abs(y)<=((size/2)*pix_width) then  {ray hits grid at side}
   begin  {hits side of grid}
    box.y1 := round(y/pix_width);
    c2 := box.y1*pix_width - pix_width/2;
    x := (c2-c)/m;
    box.x1 := round(x/pix_width);
    box.x := x - box.x1*pix_width;
   end   {hits side of grid}
   else exit;
  end;   {dosen't hit bottom}
{Now to calculate the line segment values and traverse the ray}
 repeat
  h1 := abs((m/abs(m))*pix_width/2 - box.x)/abs(cos(deg(ang)));
        {calculating hypot of triangles in cell}
  h2 := abs(pix_width/sin(deg(ang)));  {see page 4 for details}
  if h1>h2 then {ray doesn't cross boundary. Purposly left out =}
  begin
   intg := intg + h2*pct(box.x1,box.y1,tran);
   box.x := box.x + pix_width/tan(deg(ang));
   box.y1 := box.y1 + 1;
  end else
  begin  {ray crosses pixel boundary}
   intg := intg + h1*pct(box.x1,box.y1,tran) +
         (h2-h1)*pct((box.x1+round(m/abs(m))),box.y1,tran);
   box.x := box.x + pix_width/tan(deg(ang)) - (m/abs(m))*pix_width;
   box.x1 := box.x1 + round(m/abs(m));
   box.y1 := box.y1 + 1;
  end;   {ray crosses pixel boundary}
 until ((box.y1>trunc(size/2))or(box.x1*(m/abs(m))>trunc(size/2)));
```

```
        {ie. ray has passed out top of grid}
  end  {if angle is ok}  else
  begin  {gradient is infinite or zero}
    if ((ang=0) or (ang=180)) then
    begin  {gradient is zero}
      x := radius*sin(deg(beta));
      tran := true;
    end    {gradient is zero}
      else x := radius*cos(deg(beta));
    box.x1 := round(x/pix_width);
    box.y1 := trunc(-size/2);
    repeat
      intg := intg + pix_width*pct(box.x1,box.y1,tran);
      box.y1 := box.y1 + 1;
    until box.y1>trunc(size/2);
  end;   {gradient is infiniteor zero}
  proval := round(intg);
end;    {end procedure}


procedure spread_back(res:pro_data);
{Backproject the residual projection data over the image array}

var
  sdash,hdash:real;
  pin,b,c,Y,X,temp:integer;
  a,angle1,angle2:real;

begin
  angle1:=(res.angle)*pi/180;
  for Y := -trunc(size/2) to trunc(size/2) do
  begin
   for X := -trunc(size/2) to trunc(size/2) do
   begin
    hdash := X*pix_width*cos(angle1) + Y*pix_width*sin(angle1);
    sdash := -X*pix_width*sin(angle1) + Y*pix_width*cos(angle1);
    if (sdash <> 0) then
      begin
        angle2 := (arctan(sdash/(radius-hdash)));
        sdash:=sdash+(sin(angle2)/cos(angle2))*hdash;
      end;
    a := arctan(sdash/radius)*(180/pi)/D_alpha;
    if a>0 then begin b := trunc(a); c := b + 1; end
          else begin c := trunc(a); b := c - 1; end;
    if ((a< trunc(num_rays/2)) and (a> -trunc(num_rays/2))) then
    pin:=round(((res.pro[b]*(c-a)+res.pro[c]*(a-b))/pix_width)/(size/lambda))
    else pin := 0;
    temp:=est^[X,Y]+pin;
    if temp < 0 then temp := 0;
    if temp > 252 then temp := 252;
    est^[X,Y] := temp;
   end;  {for kk}
  end;
end;
```

```
begin {MAIN PROGRAM}
 initialise;
 for count3 := 1 to itt_num do
 begin
  readactfile(nfile2,((count3-1) div itt_sep)+1);
  estpro1.angle:= (init_ang+(count3-1)*ang_disp)-
     trunc((init_ang+(count3-1)*ang_disp)/360)*360;   {calculate the angle of the current projection}
  actpro1.angle := estpro1.angle;
  respro1.angle := estpro1.angle;

  for count1 := -trunc(num_rays/2) to trunc(num_rays/2) do
  begin
   temp := count1*D_alpha;
   calc_ray(actpro1.angle,temp,actpro1.pro[count1]);  {calculate the simulated X-ray data}
  end;
  readestfile(nfile1,count3-1);
  for count1 := -trunc(num_rays/2) to trunc(num_rays/2) do
  begin
   temp := count1*D_alpha;
                {calculate the estimate projection data based on the current estimated image}
   calc_ray(estpro1.angle,temp,estpro1.pro[count1]);
   respro1.pro[count1] := actpro1.pro[count1]-estpro1.pro[count1];  {calculate the resudial projection}
  end;
  spread_back(respro1);  {spreads residual into residual image}
  writefile(opfname,count3);
  nfile1 := opfname;
 end; {count3}
 dispose(est);
end.
```

APPENDIX C

(Pascal program for projection data convolution)

```
program convolution;
{Inputs projection data, and convolves with the Ramachandran-Lakshminarayanan filter}
{To truncate the terms in the kernel simply remove the double quotes from the for loop}
{in the Convolve procedure}
uses
  graph,
  DOS,
  CRT;

const
  projfname = 'phant1.prj';          {file containing projection data}
  oputfname = 'ph1T3.cnv';                  {file for writing convolved projection data}
  E = 0.0;                                  {set to 0.0 for Ramachandran-Lakshminarayanan filter}
  RealConv=true;                    {true for full kernal convolution}
  OkWrite=true;                     {Ok to write output to disk}
  Rays = 240;
  Views = 120;                      {half of 240 since parrallel data}
  dx=60/rays*3.1415/180;

type
  projarray = array[0..views-1,0..Rays-1] of word;
  ConvArray = array[0..views-1,0..rays-1] of integer;

var
  qq        :array[-250..250] of real;
  proj      :^ProjArray;
  Cproj     :^ConvArray;
  h,s,m,hun:word;

Procedure ReadIn;
{read in projection data from disk}
var
  ImFile:file;
begin
  {$I-}             { disable IO checking so IOResult can be used }
  Assign(ImFile, projfname);
  Reset(ImFile, Rays*Views);
  if (IOResult <> 0) then Exit;   { Quit & return problem in Result }
  BlockRead(ImFile, Proj^, 2);   { Read Half image from file }
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;

procedure filter;
{This procedure is used to pre-filter the projection data when using fan-beam projections}
var i,j:integer;
begin
for i := 0 to views-1 do
  for j := 0 to rays-1 do
    Proj^[i,j]:=round(Proj^[i,j]*cos((j-(rays/2))*(60/rays)*pi/180));
end;
```

```
Procedure WriteFile;
{write the convolved projection data to disk}
var
  i1:word;
  ImFile:file;
begin
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, OputFName);
  Rewrite(ImFile, Rays*Views);
  BlockWrite(ImFile, CProj^, 2);   {2 since word sized data}
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;

Procedure Init;
{initialise the graphics screen for displaying the data}
{and the pointers to the projection data}
var
  GraphDriver,GraphMode:integer;
begin
  GraphDriver:=9;
  GraphMode  :=2;
  InitGraph(GraphDriver,GraphMode,'');
  New(Proj);
  New(CProj);
end;

procedure Convolve;
{generates the convolution function, and then convolves it with the projection data}
var
  c1,c2,p,q1,mm,mdash:integer;
  pc,q:real;
  ofile2:text;
  dummy:string;
begin
Moveto(0,120);  lineto(5,120);  moveto(0,300);
{**Next section evaluates the convolution function (qq)**}
for c1:=-rays to rays do qq[c1]:=0;
{for c1 := -3 to 3 do }                {this line is uncommented for reduced kernel convolution}
begin
  if c1 = 0 then
  qq[c1]:=(3-2*E)/(12*sqr(dx))
    else if (c1 mod 2) = 0 then qq[c1] := -E/(sqr(pi)*(sqr(sin(dx*c1))))
      else qq[c1] := -(1-E)/(sqr(pi)*(sqr(sin(dx*c1))));
  qq[c1]:=qq[c1]/25;
  lineto(c1+rays,300-round(qq[c1]));
end;
moveto(0,0);
{**Next section evaluates the convolved projections**}
for P := 0 to Views-1 do
begin
  for mdash := 0 to Rays-1 do
  begin
```

```
    pc := 0;
    for mm := 0 to Rays-1 do
    begin
      q1 := mdash - mm;
      q := qq[q1];
      pc := pc + q*proj^[P,mm];
    end;  {for mm}
    pc := pc * dx;
    cproj^[P,mdash] := trunc(pc*5);
  end;   {mdash}
  lineto(2,p);
end;  {p}
if OkWrite then WriteFile;
end;  {initialise}


Procedure Display;
{Display the projection and convolved projection to the screen}
var
  C1,C2:integer;
  Ch:char;
begin
C2:=0;
repeat
  ClearDevice;
  MoveTo(0,350);
  setcolor(3);
  for c1:=0 to Rays-1 do
    LineTo(C1,350-trunc(Proj^[c2,c1]/256));
  MoveTo(0,350);
  SetColor(2);
  for c1:=0 to Rays-1 do
    LineTo(C1,350-trunc(CProj^[c2,c1]/256));
repeat until keypressed;
ch:=readkey;
if ((c2>Views-2) and (ch<>'q')) then ch:='d';
if ch=' ' then c2:=c2+1;
until(ch='q');
end;

begin {Main}
  writeln('Reading in projection data');
  Init;
  readin;
{  filter;}     {only filter fan beam data}
  if RealConv then Convolve;
  Display;
  Dispose(Proj);
  Dispose(CProj);
end.
```

APPENDIX D

(Pascal program for image reconstruction)

(using convolution backprojection method)

```
program backprojection;

{This program reads convolved projection data from disk, and reconstructs an estimate image}
{The process is standard backprojection}

uses
  CRT;

const
  projfname      = 'data.tmp';                    {projection data file name}
  oputfname      = 'test.pct';                    {output image file name}
  num_rays       = 128;
  num_proj       = 128;
  side_length    = 127;                           {image is 127 by 127 pixels}
  dx             = 1/((num_rays-1)/2);            {nb dy and ds equal dx}
  dtheta         = (180/num_proj)*3.1415927/180;  {angle between projections}
  miK            = -trunc((side_length-1)/2);     {image processed from -63 to 63 = 127 pixels}
  K              = trunc((side_length-1)/2);
  miL            = miK;
  L              = K;

type
  projarray = array[1..200,1..15] of real;

var
  theta:array[1..15]of real;
  qq:array[-181..181] of real;
  ofile:text;
  proj:projarray;
  kk,ll,nn:integer;
  fb,sdash,pin,a,pc,s,I,q : real;
  b,c,mdash,mm,q1:integer;

procedure initialise;
{open output file for writing and set up array of projection angles}
var
  c1:integer;
begin
clrscr;
assign(ofile,oputfname);
rewrite(ofile);
for c1 := 1 to 15 do theta[c1] := (c1-1)*dtheta;
writeln('Starting Reconstruction');
writeln('**********************');
end;  {initialise}

procedure readin;
{read projection data from disk}
var ifile:text;
    c1,c2:integer;
    r:real;
begin
assign(ifile,projfname);
reset(ifile);
```

```
for c2 := 1 to num_proj do
 for c1 := 1 to num_rays do
 begin
  read(ifile,r);
  proj[c1,c2] := 255-r;
 end;    {for c1 and c2}
close(ifile);
writeln('** Finished Loading Projection data **');
end; {readin}

begin    {Main}
{This program calculates the pixel values one at a time by summing }
{the appropriate ray contributions from each projection}
initialise;
readin;
for ll := mil to l do          {pixels in x direction}
begin
 for kk := mik to k do      {pixels in y direction
 begin
  fb := 0;
  for nn := 1 to num_proj do
  begin
   sdash := kk*dx*cos(theta[nn]) + ll*dx*sin(theta[nn]);
   pin := 0;
   a := sdash/dx;
   if a>0 then begin b := trunc(a); c := b + 1; end
        else begin c := trunc(a); b := c - 1; end;
   for mdash := b to c do
   begin
    pc := proj[(mdash+trunc(num_rays/2)+1),nn]*dx;
    s := sdash-mdash*dx;
    if abs(s)>=dx then I := 0
     else I := (1/dx)*(1-(abs(s)/dx));
    pin := pin + pc*I;
   end;  {for mdash}
   pin := pin * dx;
   fb := fb + pin;
  end; {for nn}
  fb := fb * dtheta;
  write(ofile,fb,' ');
 end;  {for kk}
 write(ofile,chr(13),chr(10));
writeln('Row ',ll,' completed');
end;  {for ll}
write(ofile,chr(13),chr(10));
close(ofile);
end.
```

APPENDIX E

(Pascal program for image reconstruction)

(using equi-radial/angular modified backprojection method)

```
Program SectorReconstruct;        {Equi-radial/angular}
{Reconstructs an image using a sector approach rather then the standard}
{cartesian approach. Also using a new high speed concept}
{approximates area contributions by using 100 points per pixel}
{set radial and angular discretisation}

USES
  Dos,TPEArray;

const
  IFile          ='ph1n10.cnv';          {convolved projection data file name}
  OFile          ='ph11an10.pct';        {estimate image output file name}
  Depth          = 120;                  {number of pixels in each sector}
  Around         = 240;                  {number of sectors}
  Rays           = 240;
  Views          = 120;
  LoadFromDisk   =true;

Type
  Data           = Array[0..Rays*Views-1] of Integer;
  ImageOut       = Array[0..Around*Depth-1] of Byte;

var
  Image          : TpArray;                    {Image Reconstruction Space}
  AddressArray   : TpArray;                    {Ray pixel intersections}
  PercentArray   : TpArray;                    {Area of intersections}
  CurrentAddr    : Array[0..Rays-1] of Integer;  {used with AddressArray & Percent}
  RayLength      : Array[0..Rays-1] of Real;   {Length of ray through space}
  DataPtr        : ^Data;                      {Convolved Data}
  OutArray       : ^ImageOut;

Procedure Init;
{initialize image, and contribution arrays}
  var c:integer;
begin
  MakeA(Image,1,Depth*Around,sizeOf(Real));
  MakeA(AddressArray,Rays,10*Depth,sizeof(Word));
  MakeA(PercentArray,Rays,10*Depth,sizeof(Word));
  New(DataPtr );
  c:=0;
  ClearA(AddressArray,c,FastInit);
  ClearA(PercentArray,c,FastInit);
  ClearA(Image,c,FastInit);
  for c:=0 to rays-1 do CurrentAddr[C]:=-1;
end;

Procedure Terminate;
  var c:integer;
begin
  Dispose(DataPtr);
  DisposeA(Image);
  DisposeA(AddressArray);
  DisposeA(PercentArray);
end;
```

```
function GetA(A:TpArray; R,C:Word):Word;
var X:integer;
begin
  Reta(A,R,C,X);
  GetA := X;
end;

Procedure SetVal(A:TpArray; R,C:Word; V:Word);
{set a value of the array stored in extended memory}
begin
  SetA(A,R,C,V);
end;

function GetAR(A:TpArray; R,C:Word):Real;
{get a value from an array stored in extended memory}
var X:Real;
begin
  Reta(A,R,C,X);
  GetAR := X;
end;

Procedure SetValR(A:TpArray; R,C:Word; V:Real);
begin
  SetA(A,R,C,V);
end;


Procedure ReadData;
{read projection data from disk}
var
  ImFile:file;
  c1:integer;
begin
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, IFile);
  Reset(ImFile, Rays*Views);
  if (IOResult <> 0) then Exit;   { Quit & return problem in Result }
  BlockRead(ImFile, DataPtr^, 2);   { Read Half image from file }
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;


Procedure WriteFile;
{write image data to disk after scaling values between 0 and 255}
var
  i1,i2:word;
  ImFile:file;
  R,min,max,scale:Real;
begin
  New(OutArray);
  min := 999999;
```

```
  max := 0;
  For I1 := 0 to Around-1 do
   For I2 := 0 to Depth-1 do
   begin
    R:=GetAR(Image,0,I1*Depth+I2);
    if min>R then Min := R;
    if Max<R then Max := R;
   end;
  writeln('Max = ',max:8:1);
  writeln('Min = ',min:8:1);
  scale := 255/(Max-Min);
  For I1 := 0 to Around-1 do
   For I2 := 0 to Depth-1 do
   begin
    R:=GetAR(Image,0,I1*Depth+I2);
    OutArray^[I1*Depth+I2]:=round((R-min)*scale);
   end;
  writeln;  writeln('Writing out estimate picture to test.pct');
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, OFile);
  Rewrite(ImFile, Depth*Around);
  BlockWrite(ImFile, OutArray^, 1);
  {$I+}
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;


Procedure CalculateAddresses;
{generate a look-up table containing the list of pixel-ray intersection areas.}
var
  C1,C2,rr,OO,ZZ:integer;
  r,O,Z,T1:real;
begin
  if LoadFromDisk then
  begin
   LoadA(AddressArray,'Phant1_1.Add');
   LoadA(PercentArray,'Phant1_1.Per');
   for C1 := 0 to rays-1 do
   begin
    c2 := 10*depth-1;
    while ((GetA(PercentArray,C1,C2) = 0) and (C2>-1)) do C2:=C2-1;
    CurrentAddr[C1]:=C2;
   end;
  end else
  begin
   writeln('Please Wait 5 seconds while I generate the Pixel Addresses Table');
   {**GENERATE ADDRESS TABLE**}
   for C1 := 0 to Around-1 do
   begin
    writeln(c1,'/',Around);
    for C2 := 0 to Depth-1 do
    begin
     for rr :=0 to 9 do
```

```
        for OO := 0 to 9 do
        begin
         r := (C2+(rr+0.5)/10)/depth;
         O := 2*pi * (C1+(OO+0.5)/10)/around;
         Z := r*cos(O) + 1;
         ZZ:= trunc(Z/(2/rays));
         if ((CurrentAddr[ZZ]=-1) or
         ((GetA(AddressArray,ZZ,CurrentAddr[ZZ]) <>C1*Depth+C2))) then
         begin
          Inc(CurrentAddr[ZZ]);
          SetVal(AddressArray,ZZ,CurrentAddr[ZZ],C1*Depth+C2);
          SetVal(PercentArray,ZZ,CurrentAddr[ZZ],0);
         end;
         SetVal(PercentArray,ZZ,CurrentAddr[ZZ],
          GetA(PercentArray,ZZ,CurrentAddr[ZZ])+1);
        end;
       end;
     end;
    writeln('Writing Lookup Table to Disk');
    StoreA(AddressArray,'Phant1_1.add');
    StoreA(PercentArray,'Phant1_1.Per');
  end;
  for C1 := 0 to Rays-1 do
   RayLength[C1] := 2*Sqrt(1-Sqr((C1-Rays/2+0.5)/(rays/2))); {ray length}
end;


Procedure ReconstructImage;
{using the look-up table, reconstruct the image using the modified backprojection method}
var
 C0,C1,C2:word;
 h,m,s,hun,offset,TwoPi:word;
 Address:longint;
 Temp1,Temp2,val:real;
begin
 TwoPi := Depth*Around;
 Writeln('Resetting Image Array');
 GetTime(h,m,s,hun);
 writeln(s,':',hun);
 offset := 0;
 for C0 := 0 to Views-1 do    {for all views 0 to pi}
 begin
 writeln('View = ',C0);
  for C1 := 0 to Rays-1 do          {for all rays in each view}
  begin
   Val:= DataPtr^[C0*Rays+C1]{/RayLength[C1]};
   for C2 := 0 to CurrentAddr[C1] do
   begin
    Address:=(GetA(AddressArray,C1,C2)+offset) mod TwoPi;
    Temp1:=GetAR(Image,0,Address);
    Temp2:=Val*GetA(PercentArray,C1,C2)/100;
    SetValR(Image,0,Address,Temp1+Temp2);
   end;
  end;
```

```
    offset := offset+Depth;
  end;
end;


{********* MAIN *********}
begin
{  Clrscr;}
  Init;
  ReadData;
  CalculateAddresses;
  ReconstructImage;
  WriteFile;
  Terminate;
end.
```

APPENDIX F

(Pascal program for image reconstruction)

(using equi-radial best-fit area modified backprojection method)

```
Program SectorReconstruct; {Equi-radial best-fit area}
{Reconstructs an image using a sector approach rather then the standard}
{cartesian approach. Also using a new high speed concept}
{approximates area contributions by using 100 points per pixel}
{set radial discretisation, best fit angular discretisation within segments}

USES
  Dos,TPEArray;

const
  IFile          ='phant1.cnv';
  OFile          ='Ph12a.pct';
  Depth          = 120;
  Around         = 240;
  Rays           = 240;
  Views          = 120;
  LoadFromDisk   =true;
  OutPixNum      = 5;     {Specify pixel width based on number of pixels}
                          {across outer pixel. 5 gives 2 pixels from 0.2 out}

Type
  Data    = Array[0..Rays*Views-1] of Integer;

var
  Image          : TpArray;                    {Image Reconstruction Space}
  OutArray       : TpArray;
  AddressArray   : TpArray;                    {Ray pixel intersections}
  PercentArray   : TpArray;                    {Area of intersections}
  CurrentAddr    : Array[0..Rays-1] of LongInt;{used with AddressArray & Percent}
  RayLength      : Array[0..Rays-1] of Real;   {Length of ray through space}
  DataPtr        : ^Data;                      {Convolved Data}
  g              :array[1..Depth] of Byte;             {gives number of pixels per width}
  gg             :array[0..Depth] of longint;  {cumulative version if g}


Procedure Init;
{Calculate values used later in the program}
  var c:longint;
     O,Z:Real;
begin
  O:=2*pi/Around;  {segment angle}
  Z:=O/OutPixNum;  {largest allowable width of pixel}
  gg[0] :=0;
  for C := 1 to Depth do
  begin
    g[C] := trunc((O*C/Depth)/Z)+1;  {calc number of pixels at each depth}
    gg[C]:= gg[C-1]+g[C];
  end;
{initialise the arrays in extended memory}
  MakeA(Image,Around,gg[Depth],sizeOf(Real));
  MakeA(OutArray,Around,gg[depth],sizeOf(Byte));
  MakeA(AddressArray,Rays,9*Depth,sizeof(LongInt));
  MakeA(PercentArray,Rays,9*Depth,sizeof(LongInt));
  New(DataPtr );
```

```
  c:=0;
  ClearA(AddressArray,c,FastInit);
  ClearA(PercentArray,c,FastInit);
  ClearA(Image,c,FastInit);
  for c:=0 to rays-1 do CurrentAddr[C]:=-1;
end;


Procedure Terminate;
  var c:longint;
begin
  Dispose(DataPtr);
  DisposeA(Image);
  DisposeA(AddressArray);
  DisposeA(PercentArray);
end;

function GetA(A:TpArray; R,C:Word):LongInt;
var X:LongInt;
begin
  Reta(A,R,C,X);
  GetA := X;
end;

Procedure SetVal(A:TpArray; R,C:Word; V:LongInt);
begin
  SetA(A,R,C,V);
end;

function GetAR(A:TpArray; R,C:Word):Real;
var X:Real;
begin
  Reta(A,R,C,X);
  GetAR := X;
end;

Procedure SetValR(A:TpArray; R,C:Word; V:Real);
begin
  SetA(A,R,C,V);
end;


Procedure ReadData;
{read convolved projection data from disk}
var
  ImFile:file;
  c1:longint;
begin
  {$I-}              { disable IO checking so IOResult can be used }
  Assign(ImFile, IFile);
  Reset(ImFile, Rays*Views);
  if (IOResult <> 0) then Exit;   { Quit & return problem in Result }
  BlockRead(ImFile, DataPtr^, 2);   { Read Half image from file }
  {$I+}
```

```
  if (IOResult <> 0) then Exit; { Quit & return problem in Result }
  Close(ImFile);
end;


Procedure WriteFile;
{write estimaged image array to disk after scaling between 0 and 255}
var
  i1,i2,i3:longint;
  ImFile:file;
  R,min,max,scale:Real;
begin
  min := 999999;
  max := 0;
  For I1 := 0 to Around-1 do
    For I2 := 0 to gg[Depth]-1 do
    begin
      R:=GetAR(Image,i1,I2);
      if min>R then Min := R;
      if Max<R then Max := R;
    end;
  writeln('Max = ',max:8:1);
  writeln('Min = ',min:8:1);
  scale := 255/(Max-Min);
  For I1 := 0 to Around-1 do
    For I2 := 0 to gg[Depth]-1 do
    begin
      R:=GetAR(Image,I1,I2);
      SetVal(OutArray,I1,I2,round((R-min)*scale));
    end;
  StoreA(OutArray,Ofile);
end;


Procedure CalculateAddresses;
{calculate pixel-ray intersection areas}
var
  C1,C2,C3,c4,rr,OO,ZZ:LongInt;
  r,O,Z,T1:real;
  Temp,Temp2:LongInt;

begin
  if LoadFromDisk then
  begin
    Writeln('Loading from disk');
    LoadA(AddressArray,'Phant1_2.Add');
    LoadA(PercentArray,'Phant1_2.Per');
    for C1 := 0 to rays-1 do
    begin
      c2 := 9*depth-1;
      while ((GetA(PercentArray,C1,C2)=0) and (C2>-1)) do C2:=C2-1;
      CurrentAddr[C1]:=C2;
    end;
  end else
```

```
 begin
  writeln('Please Wait 5 seconds while I generate the Pixel Addresses Table');
  {**GENERATE ADDRESS TABLE**}

  for C1 := 0 to Around-1 do
  begin
   writeln(c1,'/',Around);

   for C2 := 0 to Depth-1 do
   begin
    for C3 := 1 to g[C2+1] do
    begin
     for rr :=0 to 9 do
      for OO := 0 to 9 do
      begin
       r := (C2+(rr+0.5)/10)/depth;
       O := 2*pi * (C1+((C3-1)+(OO+0.5)/10)/g[C2+1])/around;
       Z := r*cos(O) + 1;
       ZZ:= trunc(Z/(2/rays));

       if (CurrentAddr[ZZ]=-1) then
       begin
        Inc(CurrentAddr[ZZ]);
        Temp2:=LongInt(C1*gg[Depth]+gg[C2]+(C3-1));
        SetVal(AddressArray,ZZ,CurrentAddr[ZZ],Temp2);
        SetVal(PercentArray,ZZ,CurrentAddr[ZZ],0);
       end else
       begin
        Temp:=GetA(AddressArray,ZZ,CurrentAddr[ZZ]);
        if (Temp <> C1*gg[Depth]+gg[C2]+(C3-1)) then
        begin
         Inc(CurrentAddr[ZZ]);
        Temp2:=LongInt(C1*gg[Depth]+gg[C2]+(C3-1));
        SetVal(AddressArray,ZZ,CurrentAddr[ZZ],Temp2);
        SetVal(PercentArray,ZZ,CurrentAddr[ZZ],0);
        end;
       end;

       SetVal(PercentArray,ZZ,CurrentAddr[ZZ],
        GetA(PercentArray,ZZ,CurrentAddr[ZZ])+1);
      end;
     end;
    end;
   end;
  writeln('Writing Lookup Table to Disk');
  StoreA(AddressArray,'Phant1_2.Add');
  StoreA(PercentArray,'Phant1_2.Per');
 end;
 for C1 := 0 to Rays-1 do
  RayLength[C1] := 2*Sqrt(1-Sqr((C1-Rays/2+0.5)/(rays/2))); {ray length}

end;
```

```
Procedure ReconstructImage;
{use look-up table to reconstruct image using modified backprojection method}
var
  C0,C1,C2:LongInt;
  h,m,s,hun:word;
  offset,TwoPi:LongInt;
  Address,Temp3,Temp4,Temp5:longint;
  Temp1,Temp2,val:real;
begin
  Randomize;
  TwoPi := gg[Depth]*Around;
  Writeln('Resetting Image Array');
  GetTime(h,m,s,hun);
  writeln(s,':',hun);
  offset := 0;
  for C0 := 0 to Views-1 do    {for all views 0 to pi}
  begin
  writeln('View = ',C0);
    for C1 := 0 to Rays-1 do          {for all rays in each view}
    begin
      Val:= DataPtr^[C0*Rays+C1]{/RayLength[C1]};
      for C2 := 0 to CurrentAddr[C1] do
      begin
        Temp5 :=(GetA(AddressArray,C1,C2)+offset);
        Address:= Temp5 mod TwoPi;
        Temp3 :=Trunc(Address/gg[depth]);
        Temp4 :=Address mod gg[depth];
        Temp1:=GetAR(Image,Temp3,Temp4);
        Temp2:=Val*GetA(PercentArray,C1,C2)/100;
        SetValR(Image,Temp3,Temp4,Temp1+Temp2);
      end;
    end;
    offset := offset+gg[Depth];
  end;
  GetTime(h,m,s,hun);
  writeln(s,':',hun);
end;


{********* MAIN *********}
begin
  Init;
  ReadData;
  CalculateAddresses;
  ReconstructImage;
  WriteFile;
  Terminate;
end.
```

APPENDIX G

(International literature database search, details and results)

Database Search 16/7/92 (= 850 NZ)

Inspec (Engineering)

nb. underlined abstract numbers were sent for on-line.

```
      Set   Items  Description
      ---   -----  -----------
? s cc=B7510B

      S1     6752  CC=B7510B  Radiation & radioactivity applications
? s cc=A8770E

      S2    13299  CC=A8770E  Diagnostic methods & instrumentation
? s computerised tomography/df

      S3     4012  COMPUTERISED TOMOGRAPHY/DF  (July 1978)
? ss pixel?(1w)transformation? or conformal()mapping or
geometric()transformatio
      S4     7383  PIXEL?
      S5    58983  TRANSFORMATION?
      S6        3  PIXEL?(1W)TRANSFORMATION?
      S7     4741  CONFORMAL
      S8    12284  MAPPING
      S9      554  CONFORMAL(W)MAPPING
     S10    14185  GEOMETRIC
     S11    58983  TRANSFORMATION?
     S12      112  GEOMETRIC(W)TRANSFORMATION?
     S13      669  PIXEL?(1W)TRANSFORMATION? OR CONFORMAL()MAPPING OR
                   GEOMETRIC()TRANSFORMATION?
? s backprojection

     S14      313  BACKPROJECTION
?  c 1 and 14 and 13 and 3

             6752  1
              313  14
              669  13
             4012  3
     S15        0  1 AND 14 AND 13 AND 3
? c 14 and 13

              313  14
              669  13
     S16        1  14 AND 13
? t 16/6/all)


      ---   -----  -----------
? t 6/6/all

 6/6/1
03624376   INSPEC Abstract Number: A90070344
 Title: Optical distortion due to geomagnetism in quantitative angiography

 6/6/2
03319124   INSPEC Abstract Number: B89014968, C89013415
 Title: New image processing accelerator for advanced algorithms

 6/6/3
02176370   INSPEC Abstract Number: B84006477, C84006238
  Title:  An  advanced  method  of  data base conversion in a communication
processing node
? c 3 and 14
```

Pixel Transformation

```
             4012  3
              313  14
      S17     207  3 AND 14
? ss fast or high()speed? or real()time?

      S18 . 51812  FAST
      S19  371347  HIGH
      S20   74159  SPEED?
      S21   27235  HIGH(W)SPEED?
      S22   62154  REAL
      S23  339238  TIME?
      S24   28189  REAL(W)TIME?
      S25  102181  FAST OR HIGH()SPEED? OR REAL()TIME?
? c 17 and 25

              207  17
           102181  25
      S26      27  17 AND 25
? t 26/6/all
```

Computerized Tomography . Backprojection . (Fast + high speed + real time)

```
 26/6/1
04163878   INSPEC Abstract Number:  A9213-8710-005,  B9207-6140C-103,
C9207-125U-093
   Title:  Backprojection  by  upsampled  Fourier  series  expansion  and
interpolated FFT

 26/6/2
04043231   INSPEC Abstract Number: A9202-8760J-002, C9201-7330-128
   Title: Three-dimensional helical-scan computed tomography using cone-beam
projections

 26/6/3
04033892   INSPEC Abstract Number: C9201-5260B-026
   Title: Development  and  implementation  of  real-time ultrasound process
tomography using a transputer network

 26/6/4
03986816   INSPEC Abstract Number: A91130602, C91065689
   Title:  Cone  beam  single  photon emission computed tomography using two
orbits

 26/6/5
03936626   INSPEC Abstract Number: A91089238, B91051701, C91047256
   Title:  New  hybrid  spline-linear  interpolation  for the fast CT and MR
imaging

 26/6/6
03907567   INSPEC Abstract Number: A91073777, C91043380
   Title: Real-time parallel processing in industrial flow measurement using
transputer arrays

 26/6/7
03875681   INSPEC Abstract Number: A91056566, B91033548, C91026449
   Title: Backprojection with Fourier series expansion and FFT

 26/6/8
03851951   INSPEC Abstract Number: A91050542
   Title:  Image  reconstruction  for  a  3D PET system using a minimum norm
```

constraint

26/6/9
03846385 INSPEC Abstract Number: A91050359
Title: A high-speed reconstruction from projections using direct Fourier method with optimized parameters-an experimental analysis

26/6/10
03766441 INSPEC Abstract Number: A91005880, B90080173, C91005730
Title: A transputer implemented algorithm for electrical impedance tomography

26/6/11
03727574 INSPEC Abstract Number: B90063768, C90063675
Title: Backprojection with Fourier series expansion and FFT

26/6/12
03722382 INSPEC Abstract Number: A90134805
Title: Incremental algorithm-a new fast backprojection scheme for parallel beam geometries

26/6/13
03623875 INSPEC Abstract Number: A90070338, C90036415
Title: A comparison of least squares and maximum likelihood reconstruction methods for Poisson sources

26/6/14
03573700 INSPEC Abstract Number: A90039764, B90018977, C90022408
Title: Simulations for parallel processing of ultrasound reflection-mode tomography with applications to two-phase flow measurement

26/6/15
03475649 INSPEC Abstract Number: A89130596
Title: Improvement of quantification in SPECT studies by scatter and attenuation compensation

26/6/16
03454302 INSPEC Abstract Number: A89106774, B89063577, C89052870
Title: High speed reprojection and its application (in CT)

26/6/17
03189512 INSPEC Abstract Number: A88103440, B88054808, C88047708
Title: Real-time parallel tomographic ultrasound imaging using transputers

26/6/18
02709470 INSPEC Abstract Number: A86087277, B86046401
Title: Ultrasound reflection-mode tomography using fan-shaped-beam insonification

26/6/19
02658184 INSPEC Abstract Number: A86053036, B86032304, C86025714
Title: Fast non-recursive reconstruction algorithm for limited data (PET)

26/6/20
02626151 INSPEC Abstract Number: A86036937, B86024414
Title: Computed tomography videography: an electrooptical system for video-rate image reconstruction

26/6/21
02576212 INSPEC Abstract Number: A85126696, B86005926

Title: Ultrasound reflection tomography of cylindrical rods

26/6/22
02538803 INSPEC Abstract Number: A85117316, C85047256
Title: A new digital tomosynthesis method with less artifacts for angiography

26/6/23
02481246 INSPEC Abstract Number: A85077702, B85042925, C85033274
Title: A real-time TOFPET slice-backproject engine employing dual AM2V116 microprocessors

26/6/24
02477794 INSPEC Abstract Number: A85073607, C85034637
Title: New implementations of fan-beam X-ray reconstruction filters

26/6/25
02136088 INSPEC Abstract Number: A83110237, B83057666, C83040171
Title: A high-speed computerized tomography image reconstruction using direct two-dimensional Fourier transform method

26/6/26
02116666 INSPEC Abstract Number: A83097585
Title: Implementation of real-time image reconstruction (positron emission tomography)

26/6/27
02090958 INSPEC Abstract Number: A83053836, B83032117
Title: Some signal processing aspects of time-of-flight positron emission tomography (TOFPET) system implementation

? s computer assisted tomography/df
basic (Biomedical)
S1 34020 COMPUTER ASSISTED TOMOGRAPHY/DF
? ss pixel?(1w)transformation? or conformal()mapping or geometric()transformatio
S2 646 PIXEL?
S3 23873 TRANSFORMATION?
S4 0 PIXEL?(1W)TRANSFORMATION?
S5 46 CONFORMAL
S6 15996 MAPPING
S7 5 CONFORMAL(W)MAPPING
S8 2614 GEOMETRIC
S9 23873 TRANSFORMATION?
S10 1 GEOMETRIC(W)TRANSFORMATION?
S11 6 PIXEL?(1W)TRANSFORMATION? OR CONFORMAL()MAPPING OR GEOMETRIC()TRANSFORMATION?
? s backprojection

S12 67 BACKPROJECTION
? ss fast or high()speed? or real()time?

S13 16463 FAST
S14 235485 HIGH
S15 7084 SPEED?
S16 1080 HIGH(W)SPEED?
S17 7111 REAL
S18 211691 TIME?
S19 2940 REAL(W)TIME?
S20 20334 FAST OR HIGH()SPEED? OR REAL()TIME?

? c 1 and 11

```
           34028  1
               6  11
     S21       0  1 AND 11
? c 1 and 12

           34028  1
              67  12
     S22      10  1 AND 12
? t 22/6/all
```

Computer Assisted Tomography. Backprojection

22/6/1
6480780 EMBASE No: 92164644
Truncation artifact suppression in cone-beam radionuclide transmission CT using maximum likelihood techniques: Evaluation with human subjects

22/6/2
6246330 EMBASE No: 91276802
Cone-beam transmission computed tomography for nonuniform attenuation compensation of SPECT images

22/6/3
7971522 EMBASE No: 91001470
Computed tomography scanning with simultaneous patient translation

22/6/4
7124127 EMBASE No: 88121022
Noise characteristics for cone beam collimators: A comparison with parallel hole collimator

22/6/5
7055806 EMBASE No: 88070709
Reconstruction for fan beam with an angular-dependent displaced center-of-rotation

22/6/6
6998754 EMBASE No: 87135415
The noise power spectrum of CT images

22/6/7
6148803 EMBASE No: 86143863
Simulation of CT reconstruction artifacts associated with multiple-rotation fan-beam data collection

22/6/8
6098211 EMBASE No: 86093271
Accuracy of area measurements made from MR images compared with computed tomography

22/6/9
6066893 EMBASE No: 86061953
Computed tomography videography: An electrooptical system for video-rate image reconstruction

22/6/10
6066889 EMBASE No: 86061949
Algorithms for limited-view computed tomography: An anotated bibliography and a challenge

03319124/5

03319124 INSPEC Abstract Number: B89014968, C89013415

Title: New image processing accelerator for advanced algorithms

Author(s): Dawson, B.; Lou, T.; Poullain, E.

Author Affiliation: Imaging Technol. Inc., Woburn, MA, USA

Conference Title: Electronic Imaging '88. International Electronic Imaging Exposition and Conference. Advance Printing of Paper Summaries p.377-82 vol.1

Publisher: Inst. Graphic Commun, Boston, MA, USA

Publication Date: 1988 Country of Publication: USA 2 vol. xxix+950 pp.

Conference Sponsor: Diagnostic Imaging Magazine; ESD: Electron. Syst. Design Magazine; EP & P 88 Magazine; Adv. Imaging Magazine

Conference Date: 28-31 March 1988 Conference Location: Anaheim, CA, USA

Language: English Document Type: Conference Paper (PA)

Treatment: New Developments (N); Practical (P); Product Review (R)

Abstract: The series 150 Image Processor from Imaging Technology Incorporated provides real-time, pipelined processing for many standard image processing algorithms. These include point or pixel transformations, operations on one or more images (addition, subtraction, averaging), convolutions and histogram and feature extraction. The Series 150 Image Processing Accelerator (IPA-150) is fast, inexpensive, easy to program in high-level languages, has a variety of data types, and addresses a large amount of memory easily. This makes the IPA-150 ideal for many advanced image processing algorithms. The IPA-150 combines a Weitek super-microcomputer, floating-point processor, hardware to assist image processing, and large image memories and temporary storage on a single VME board. (0 Refs)

Descriptors: computerised picture processing; digital signal processing chips; pipeline processing; real-time systems

Identifiers: image operations; Imaging Technology Incorporated; real-time, pipelined processing; pixel transformations; convolutions; histogram; feature extraction; Series 150 Image Processing Accelerator; data types; IPA-150; image processing algorithms; Weitek super-microcomputer; floating-point processor; large image memories; temporary storage; VME board

Class Codes: B1265F (Microprocessors and microcomputers); C5260B (Computer vision and picture processing); C5130 (Microprocessor chips)

03936626 INSPEC Abstract Number: A91089238, B91051701, C91047256

Title: New hybrid spline-linear interpolation for the fast CT and MR imaging

Author(s): Matej, S.

Author Affiliation: Inst. of Measure. & Measuring Eng., Slovak Acad. of Sci., Bratislava, Czechoslovakia

Conference Title: Signal Processing V. Theories and Applications. Proceedings of EUSIPCO-90, Fifth European Signal Processing Conference p.909-12 vol.2

Editor(s): Torres, L.; Masgrau, E.; Lagunas, M.A.

Publisher: Elsevier, Amsterdam, Netherlands

Publication Date: 1990 Country of Publication: Netherlands 3 vol. lviii+2034 pp.

ISBN: 0 444 88636 2

Conference Sponsor: CIDEM; CIRIT; IBM; et al

Conference Date: 18-21 Sept. 1990 Conference Location: Barcelona, Spain

Language: English Document Type: Conference Paper (PA)

Treatment: Theoretical (T); Experimental (X)

Abstract: Modern tomographic applications like imaging of 3-D objects or dynamic processes require reconstruction techniques of extra high speed. One of the most promising reconstruction methods is the direct fourier method (DFM). Provided an appropriate reconstruction parameters (discretization parameters, filter type, 2-D IFT dimension, interpolation type) are used, the DFM offers results of the same quality as the classical convolution, backprojection method yet in considerable shorter time. The most important reconstruction parameter is the accuracy of spectrum data interpolation. The interpolation proposed leads to fast image reconstruction of high quality. This fact is demonstrated by means of simulation. (10 Refs)

Descriptors: biomedical NMR; computerised picture processing; computerised tomography; fast Fourier transforms; interpolation; magnetic resonance spectroscopy; splines (mathematics)

Identifiers: magnetic resonance imaging; computerised tomography; FFT; image reconstruction; hybrid spline-linear interpolation; tomographic applications; imaging; 3-D objects; spectrum data interpolation

Class Codes: A0650D (Data gathering, processing, and recording, data displays including digital techniques); A8770E (Diagnostic methods and instrumentation); A0758 (Magnetic resonance spectrometers, auxiliary instruments and techniques); A0260 (Numerical approximation and analysis); B6140C (Optical information processing); B0290F (Interpolation and function approximation); B02902 (Other numerical methods); B7510B (Radiation and radioactivity applications); C5260B (Computer vision and picture processing); C4130 (Interpolation and function approximation); C7330 (Biology and medicine); C4190 (Other numerical methods)

03875681 INSPEC Abstract Number: A91056566, B91033548, C91026449

Title: Backprojection with Fourier series expansion and FFT

Author(s): Tabei, M.; Ueda, M.

Author Affiliation: Tokyo Inst. of Technol., Yokohama, Japan

Conference Title: ICASSP 90. 1990 International Conference on Acoustics, Speech and Signal Processing (Cat. No.90CH2847-2) p.1853-6 vol.4

Publisher: IEEE, New York, NY, USA

Publication Date: 1990 Country of Publication: USA 5 vol. 2970 pp.

U.S. Copyright Clearance Center Code: CH2847-2/90/0000-1853$01.00

Conference Sponsor: IEEE

Conference Date: 3-6 April 1990 Conference Location: Albuquerque, NM, USA

Language: English Document Type: Conference Paper (PA)

Treatment: Theoretical (T); Experimental (X)

Abstract: A computerized tomography reconstruction technique with the fast Fourier transform (FFT) is discussed. A fast backprojection method through the use of interpolated FFT is presented. An approach to interpolating and backprojecting the convolved projections onto the image frame is proposed. First, the Fourier series expansion of the convolved projection is calculated. It is then projected onto a rectangular grid in the frequency domain, using the aliasing-free interpolation of FFT bins. The total amount of computation in this procedure for a 512*512 image is 1/6 of the conventional backprojection method with linear interpolation. This technique also allows arbitrary control of the frequency characteristics. (9 Refs)

Descriptors: computerised picture processing; computerised tomography; fast Fourier transforms; series (mathematics)

Identifiers: frequency characteristics control; Fourier series expansion; computerized tomography reconstruction; fast Fourier transform; fast backprojection method; interpolated FFT; convolved projections; image frame ; frequency domain; aliasing-free interpolation

Class Codes: A8760J (Corpuscular radiation and radioisotopes); A8770E (Diagnostic methods and instrumentation); B6140C (Optical information processing); B7510B (Radiation and radioactivity applications); B0290Z (Other numerical methods); C1250 (Pattern recognition); C5260B (Computer vision and picture processing); C4190 (Other numerical methods); C7330 (Biology and medicine)

03846385 INSPEC Abstract Number: A91050359

Title: A high-speed reconstruction from projections using direct Fourier method with optimized parameters-an experimental analysis

Author(s): Matej, S.; Bajla, I.

Author Affiliation: Inst. of Meas. Sci., Slovak Acad. of Sci., Bratislava, Czechoslovakia

Journal: IEEE Transactions on Medical Imaging vol.9, no.4 p.421-9

Publication Date: Dec. 1990 Country of Publication: USA

CODEN: ITMID4 ISSN: 0278-0062

U.S. Copyright Clearance Center Code: 0278-0062/90/1200-0421$01.00

Language: English Document Type: Journal Paper (JP)

Treatment: Theoretical (T)

Abstract: Some issues of the direct Fourier method (DFM) implementation are discussed. A hybrid spline-linear interpolation for the DFM is proposed. The results of comprehensive simulation research are presented. The following reconstruction problems and parameters are emphasized: interpolation, increasing the radial density of the polar raster, filtering, the 2-D inverse Fourier transformation dimension, and considering the cases of noiseless and noisy input data. For the a priori prescribed resolution of the reconstructed image, values of reconstruction parameters have been determined which are optimal with regard to reconstruction quality and computation cost. The computational requirements of the DFM algorithm which correspond to distinct interpolation schemes are compared to one another for CT and MR tomography, respectively. The estimations obtained are compared to computational characteristics of the convolution backprojection method. (19 Refs)

Descriptors: computerised tomography; Fourier transforms

Identifiers: noiseless input data; high-speed reconstruction from projections; direct Fourier method; optimized parameters; hybrid spline-linear interpolation; interpolation; radial density; polar raster; filtering; noisy input data; computational characteristics; convolution backprojection method

Class Codes: A8710 (General, theoretical, and mathematical biophysics); A8770E (Diagnostic methods and instrumentation)

03722382 INSPEC Abstract Number: A90134805

Title: Incremental algorithm-a new fast backprojection scheme for parallel beam geometries

Author(s): Cho, Z.H.; Chen, C.M.; Lee, S.

Author Affiliation: Dept. of Radiol. Sci., California Univ., Irvine, CA, USA

Journal: IEEE Transactions on Medical Imaging vol.9, no.2 p.207-17

Publication Date: June 1990 Country of Publication: USA

CODEN: ITMID4 ISSN: 0278-0062

U.S. Copyright Clearance Center Code: 0278-0062/90/0600-0207$01.00

Language: English Document Type: Journal Paper (JP)

Treatment: Theoretical (T)

Abstract: A fast backprojection scheme for parallel beam geometries is proposed. Known as the incremental algorithm, it performs backprojection on a ray-by-ray (beam-by-beam) basis rather than the pixel-by-pixel backprojection in the conventional algorithm. By restructuring a

conventional backprojection algorithm, the interdependency of pixel computations (position and value) is transformed to a set of incremental relations for a beam, where a beam is a set of pixels enclosed by two adjacent rays in 2-D computed tomography (CT), and a set of voxels enclosed by four adjacent rays in 3-D CT. To minimize the overhead of searching for the next pixels, a searching flow technique has been developed to implement the first-order and second-order incremental relations for 2-D and 3-D CTs, respectively. The values of all the pixels in each beam (except the first pixel) are computed with additions only, the key idea of the proposed backprojection scheme. The incremental algorithm has been implemented on two different machines and compared to B.F. Shepp and L.A. Logan's (1974) algorithm. The present implementation results show the superiority of this approach over the conventional algorithm. (9 Refs)

Descriptors: computerised tomography

Identifiers: 2D computed tomography; fast backprojection scheme; parallel beam geometries; pixel-by-pixel backprojection; voxels; searching flow technique

Class Codes: A8760J (Corpuscular radiation and radioisotopes); A8770E (Diagnostic methods and instrumentation); A8710 (General, theoretical, and mathematical biophysics)

03454308 INSPEC Abstract Number: A89106774, B89063577, C89052870

Title: High speed reprojection and its application (in CT)

Author(s): Crawford, C.R.; Colsher, J.G.; Pelc, N.J.; Lonn, A.H.R.

Author Affiliation: GE Med. Syst., Milwaukee, WI, USA

Journal: Proceedings of the SPIE - The International Society for Optical Engineering vol.914, pt.A p.311-18

Publication Date: 1988 Country of Publication: USA

CODEN: PSISDG ISSN: 0277-786X

Conference Title: Medical Imaging II

Conference Sponsor: SPIE

Conference Date: 31 Jan.-5 Feb. 1988 Conference Location: Newport Beach, CA, USA

Language: English Document Type: Conference Paper (PA); Journal Paper (JP)

Treatment: Theoretical (T)

Abstract: Reprojection is the process by which projections are produced from an image such that, if these projections are filtered and backprojected, they yield the original image. A method is presented that enables an unmodified backprojector to be used as a reprojector. Because backprojectors are designed to exploit the parallelism in the backprojection algorithm, the time required to obtain reprojections is significantly reduced. Another method, based on the Fourier Slice Theorem, is presented that enables a general purpose array processor to be used as a high speed reprojector. It is also shown that the parameters of the reprojection algorithm can be adjusted to decrease significantly the time required to perform an application that uses reprojection. Finally, two applications of reprojection in computed tomography are discussed. (12 Refs)

Descriptors: computerised picture processing; computerised tomography; filtering and prediction theory; medical diagnostic computing

Identifiers: high speed reprojection; image reconstruction; filtered images; unmodified backprojector; Fourier Slice Theorem; general purpose array processor; reprojection algorithm; computed tomography

Class Codes: A4230V (Image processing and restoration); A0650D (Data gathering, processing, and recording, data displays including digital techniques); A8760J (Corpuscular radiation and radioisotopes); A8770E (Diagnostic methods and instrumentation); B6140C (Optical information processing); B7510B (Radiation and radioactivity applications); C1250 (

Pattern recognition); C1260 (Information theory); C5260B (Computer vision and picture processing); C7330 (Biology and medicine)

02626151 INSPEC Abstract Number: A86036937, B86024414
Title: Computed tomography videography: an electrooptical system for video-rate image reconstruction
Author(s): Gmitro, A.F.; Gindi, G.R.
Author Affiliation: Dept. of Diagnostic Radiol., Yale Univ. Sch. of Med., New Haven, CT, USA
Journal: Applied Optics vol.24, no.23 p.4040-5
Publication Date: 1 Dec. 1985 Country of Publication: USA
CODEN: APOPAI ISSN: 0003-6935
U.S. Copyright Clearance Center Code: 0003-6935/85/234040-06$02.00/0
Conference Title: Meeting on Industrial Applications of Computed Tomography and NMR Imaging
Conference Date: 13-14 Aug. 1984 Conference Location: Hecla Island, Man., Canada
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: A system for reconstruction of computed tomography (CT) images in a video frame time (1/30 sec) is described. The system implements the filtered backprojection algorithm using an electrooptic filtering technique, an optical backprojection, and a video detector. When coupled to a real-time system for acquiring projection data, the combined system realizes what the authors term CT videography. (10 Refs)
Descriptors: computerised picture processing; computerised tomography; optical information processing; signal processing equipment; television applications
Identifiers: computed tomography videography; CT videography; optical processor; electrooptical system; video-rate image reconstruction; video frame time; filtered backprojection algorithm; electrooptic filtering technique; optical backprojection; video detector; real-time system
Class Codes: A0650D (Data gathering, processing, and recording, data displays including digital techniques); A0785 (X-ray, gamma-ray instruments and techniques); A4230V (Image processing and restoration); A8760J (Corpuscular radiation and radioisotopes); B6140C (Optical information processing); B6430J (Applications of television systems); B7210X (Other instrumentation and measurement systems); B7220 (Signal processing and conditioning equipment and techniques); B7510B (Radiation and radioactivity applications)

02538803 INSPEC Abstract Number: A85117316, C85047256
Title: A new digital tomosynthesis method with less artifacts for angiography
Author(s): Haaker, P.; Klotz, E.; Koppe, R.; Linde, R.; Moller, H.
Author Affiliation: Philips GmbH Forschungslab. Hamburg, West Germany
Journal: Medical Physics vol.12, no.4 p.431-6
Publication Date: July-Aug. 1985 Country of Publication: USA
CODEN: MPHYA6 ISSN: 0094-2405
U.S. Copyright Clearance Center Code: 0094-2405/85/040431-06$01.20
Language: English Document Type: Journal Paper (JP)
Treatment: Theoretical (T); Experimental (X)
Abstract: A new nonlinear reconstruction method for tomosynthesis is described. This method is suited for 'dilute' objects, i.e. objects in which most of the voxels have negligibly small absorption. Images of blood vessels filled with contrast material approximate this condition if the background is subtracted. The technique has been tested experimentally

using a wire phantom and a prepared human heart. The results show significantly less artifacts than the well-known backprojection. It is possible to get diagnostic image quality with a few projections. The reconstruction algorithm can be realized with dedicated real-time hardware. (23 Refs)
Descriptors: blood; cardiology; computerised tomography
Identifiers: digital tomosynthesis method; artifacts; angiography; blood vessels; contrast material; wire phantom; human heart; reconstruction algorithm
Class Codes: A8760J (Corpuscular radiation and radioisotopes); A8770E (Diagnostic methods and instrumentation); C7330 (Biology and medicine)

02481246 INSPEC Abstract Number: A85077702, B85042925, C85033274
Title: A real-time TOFPET slice-backproject engine employing dual AM29116 microprocessors
Author(s): Hartz, R.; Bristow, D.; Mullani, N.
Author Affiliation: Texas Univ. Med. Sch., Houston, TX, USA
Journal: IEEE Transactions on Nuclear Science vol.NS-32, no.1 p. 839-4
Publication Date: Feb. 1985 Country of Publication: USA
CODEN: IETNAE ISSN: 0018-9499
U.S. Copyright Clearance Center Code: 0018-9499/85/0200-0839$01.00
Conference Title: 1984 Nuclear Science Symposium and 1984 Symposium on Nuclear Power Systems
Conference Sponsor: IEEE; Oak Ridge Nat. Lab.; NASA; NBS; EPRI; US DOE
Conference Date: 31 Oct.-2 Nov. 1984 Conference Location: Orlando, FL, USA
Language: English Document Type: Conference Paper (PA); Journal Paper (JP)
Treatment: Practical (P)
Abstract: Positron emission tomographic (PET) cameras produce a large volume of data per study. The use of time-of-flight (TOF) information adds a level of complexity and increases the amount of data to be processed. Tomographic backprojection for both conventional and TOFPET systems is data intensive, and without special hardware can require an unacceptable amount of time. The authors have designed, constructed, and are using a prototype slice-backproject engine that has improved the throughput by a factor of 30. The design is centered on two microprogrammable 16-bit processors and a 16*16 combinational multiplier with an instruction rate of 150 to 200 ns. The engine architecture as it relates to the backproject algorithm is described along with design features that allow it to be used as a general processor. The use of two processing elements permits shared and concurrent operations to be performed at the microcode level. Finally, the engine is reviewed in the context of a system architecture employing multiple engines to backproject several slices in parallel. (0 Refs)
Descriptors: biomedical equipment; computerised tomography; data acquisition; data reduction; real-time systems
Identifiers: positron emission tomography; real-time; TOFPET; slice-backproject engine; dual AM29116 microprocessors; cameras; complexity; backprojection; architecture; design features
Class Codes: A8770E (Diagnostic methods and instrumentation); B7210G (Data acquisition systems); B7510B (Radiation and radioactivity applications); C5520 (Data acquisition equipment and techniques)

02477794 INSPEC Abstract Number: A85073607, C85034637
Title: New implementations of fan-beam X-ray reconstruction filters
Author(s): Jonckheere, E.A.; Truong, T.K.; Reed, I.S.; Kwoh, Y.S.
Author Affiliation: Dept. of Radiol., Memorial Hospital Med. Center, Long

Beach, CA, USA
Conference Title: Frontiers of Engineering and Computing in Health Care - 1984. Proceedings - Sixth Annual Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No. 84CH2058-6) p.289-95
Editor(s): Semmlow, J.L.; Welkowitz, W.
Publisher: IEEE, New York, NY, USA
Publication Date: 1984 Country of Publication: USA viii+857 pp.
U.S. Copyright Clearance Center Code: CH2058-6/84/0000-0289$01.00
Conference Sponsor: IEEE
Conference Date: 15-17 Sept. 1984 Conference Location: Los Angeles, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P); Theoretical (T)
Abstract: The authors show how the modified convolution/backprojection reconstruction procedure outlined by E.A. Jonckheere et al. (1984) can be implemented with one FFT convolution per projection datum. Central to this paper is the observation that a convolution/backprojection implementation of a reconstruction is obtained if one is able to separate the variables of a certain operator kernel. (5 Refs)
Descriptors: computerised tomography; fast Fourier transforms
Identifiers: computerised tomography; fan-beam X-ray reconstruction filters; modified convolution/backprojection reconstruction procedure; one FFT convolution per projection datum; operator kernel
Class Codes: A0230 (Function theory, analysis); A8760J (Corpuscular radiation and radioisotopes); A8770E (Diagnostic methods and instrumentation); C4190 (Other numerical methods); C7330 (Biology and medicine)

02136088 INSPEC Abstract Number: A83110237, B83057666, C83040171
Title: A high-speed computerized tomography image reconstruction using direct two-dimensional Fourier transform method
Author(s): Niki, N.; Mizutani, T.; Takahashi, Y.; Inouye, T.
Author Affiliation: Faculty of Engng., Tokushima Univ., Tokushima, Japan

Journal: Transactions of the Institute of Electronics and Communication Engineers of Japan, Part D vol.J66D, no.6 p.707-14
Publication Date: June 1983 Country of Publication: Japan
CODEN: DTGDA3 ISSN: 0374-468X
Language: Japanese Document Type: Journal Paper (JP)
Treatment: Applications (A); New Developments (N); Practical (P)
Abstract: A practical high-speed image reconstruction algorithm based on direct two-dimensional Fourier transform method is developed. It eliminates low image quality resulting from the previous direct two-dimensional Fourier transform method. The key point of the new algorithm is the elaborate interpolations of the coordinate transform in the Fourier domain. The program based on the new algorithm is developed and is compared with that based on the conventional filtered backprojection method. The result reveals that the processing time is only one tenth of the latter and the quality of the obtained images are almost the same. (11 Refs)
Descriptors: computerised picture processing; computerised tomography; interpolation
Identifiers: computerized tomography; image reconstruction; two-dimensional Fourier transform; image quality; interpolations; coordinate transform; processing time
Class Codes: A8770E (Diagnostic methods and instrumentation); B6140C (Optical information processing); B7510B (Radiation and radioactivity applications); C5260 (Digital signal processing); C7330 (Biology and medicine)

7055806 EMBASE No: 88070709

Reconstruction for fan beam with an angular-dependent displaced center-of-rotation

Crawford C.R.; Gullberg G.T.; Tsui B.M.W.

Applied Science Laboratory, Medical Systems Business Group, General Electric Company, Milwaukee, WI 53201 USA

MED. PHYS. (USA) , 1988, 15/1 (67-71) CODEN: MPHYA ISSN: 0094-2405

LANGUAGES: English

SUBFILES: 014; 027

A convolutional backprojection algorithm is derived for a fan beam geometry that has an angular-dependent displacement in its center-of-rotation from the midline of the fan beam. In both x-ray computed tomography and single photon emission computed tomography, misalignment can occur when the mechanical center-of-rotation is not colinear with midline of the fan beam. In some cases the shift in the center-of-rotation is constant for every angle, whereas, in other cases it varies with angular position. Standard reconstruction algorithms, which directly filter and backproject the fan beam data without rebinning into parallel beam geometry, have been derived for a geometry having its center-of-rotation at the midline of the fan beam. However, in the case of any misalignment of the center-of-rotation, if these conventional reconstruction algorithms are used to reconstruct the fan beam projections, structured artifacts and a loss of resolution will result. Simulations are performed that illustrate these artifacts and demonstrate how the new algorithm corrects for this misalignment. A method for estimating the parameters of the fan beam geometry, including the angular-dependent shift in the center-of-rotation, is also described.

EMTAGS:
Diagnosis 0140; Automation and computers 0530; Theoretical study 0110; Apparatus, equipment and supplies 0510

MEDICAL DESCRIPTORS:
*computer assisted tomography--diagnosis--di
algorithm; x ray; photon emitter

6398754 EMBASE No: 87135415

The noise power spectrum of CT images

Kijewski M.F.; Judy P.F.

Department of Radiology, Harvard Medical School, Boston, MA 02115 USA

PHYS. MED. BIOL. (UK) , 1987, 32/5 (565-575) CODEN: PHMBA

LANGUAGES: ENGLISH SUMMARY LANGUAGES: FRENCH; GERMAN

An expression for the noise power spectrum of images reconstructed by the discrete filtered backprojection algorithm has been derived. The formulation explicitly includes sampling within the projections, angular sampling, and the two-dimensional sampling implicit in the discrete representation of the image. The effects of interpolation are also considered. Noise power spectra predicted by this analysis differ from those predicted using continuous theory in two respects: they are rotationally asymmetric, and they do not approach zero at zero frequency. Both of these properties can be attributed to two-dimensional aliasing due to pixel sampling. The predictions were confirmed by measurement of noise power spectra of both simulated images and images from a commercial x-ray transmission CT scanner.

EMTAGS:
Priority journal (0007); Diagnosis (0140); Methodology (0130); Human (0888)

; Automation and computers (0530)
DESCRIPTORS:
*computer assisted tomography (0229332); *noise (0033436)
algorithm (0156846)

IDENTIFIERS: noise power spectrum
SECTION HEADINGS:
  02709070000 BIOPHYSICS AND BIOENGINEERING/ COMPUTER APPLICATIONS/
              Radiology
  01405010000 RADIOLOGY/ RADIODIAGNOSIS/ Techniques and apparatus

6148803 EMBASE No: 86143863
  Simulation of CT reconstruction artifacts associated with multiple-rotation fan-beam data collection
  Jelinek J.; Overton T.R.
  Department of Applied Sciences in Medicine, University of Alberta, Edmonton, Alta. T6G 2G3 CANADA
  COMPUT. RADIOL. (USA) , 1986, 10/1 (23-36) CODEN: CRADD
  LANGUAGES: ENGLISH
  Artifacts in reconstructed image caused by the errors in projection data depend on the source of the error itself, on the data collection mode and on the reconstruction algorithm. The effect of the multiple-rotation fan-beam data collection modes and corresponding recorded parallel convolution-backprojection algorithms on reconstruction artifacts of this type is analysed.

EMTAGS:
Priority journal (0007); Diagnosis (0140); Automation and computers (0530); Methodology (0130); Nonhuman (0777); Nonbiological model (0503)
DESCRIPTORS:
*computer assisted tomography (0229332); *image reconstruction (0487989); *artifact (0003710)
IDENTIFIERS: fan beam geometry; simulation of artifacts
SECTION HEADINGS:
  01405010000 RADIOLOGY/ RADIODIAGNOSIS/ Techniques and apparatus
  02709070000 BIOPHYSICS AND BIOENGINEERING/ COMPUTER APPLICATIONS/
              Radiology
  02706170000 /SPECIALIZED INSTRUMENTATION/ Radiology

6066889 EMBASE No: 86061949
  Algorithms for limited-view computed tomography: An anotated bibliography and a challenge
  Rangayyan R.; Prakash Dhawan A.; Gordon R.
  University of Calgary, Department of Electrical Engineering, Calgary T2N 1N4 CANADA
  APPL. OPT. (USA) , 1985, 24/23 (4000-4012) CODEN: APOPA
  LANGUAGES: ENGLISH
  In many applications of computed tomography, it may not be possible to acquire projection data at all angles, as required by the most commonly used algorithm of convolution backprojection. In such a limited-data situation, we face an ill-posed problem in attempting to reconstruct an image from an incomplete set of projections. Many techniques have been proposed to tackle this situation, employing diverse theories such as signal recovery, image restoration, constrained deconvolution, and constrained optimization, as well as novel schemes such as iterative object-dependent algorithms incorporating a priori knowledge and use of multispectral radiation. We present an overview of such techniques and offer a challenge to all readers to reconstruct images from a set of

APPENDIX H

(Timing analysis from Field Programmable Gate Array software)

PIXEL INCREMENT

LOGICAL PATH CRITICAL ELEMENT DELAY CUMULATIVE

From: BLK PIX_VAL14 PAD to C5.I2 : 3.0ns (3.0ns)
Thru: NET PIX_VAL14 C5.I2 to TBUF_R2C0.2.I : 0.0ns (3.0ns)
Thru: BLK PIX_VAL14 TBUF_R2C0.2.I to TBUF_R2C0.2.O : 1.2ns (4.2ns)
Thru: NET PIX_VAL14 TBUF_R2C0.2.O to CLB_R10C1.F2 : 7.6ns (11.8ns)
To: BLK _PIX_VAL7 CLB_R10C1.F2 to SETUP : 4.5ns (16.3ns)

From: BLK _PIX_VAL3 CLOCK to CLB_R9C1.XQ : 3.0ns (3.0ns)
Thru: NET _PIX_VAL3 CLB_R9C1.XQ to CLB_R12C2.G4 : 3.1ns (6.1ns)
Thru: BLK _PIX_VAL2 CLB_R12C2.G4 to CLB_R12C2.COUT : 6.0ns (12.1ns)
Thru: NET M/Carry_3 CLB_R12C2.COUT to CLB_R11C2.CIN : 0.0ns (12.1ns)
Thru: BLK _PIX_VAL4 CLB_R11C2.CIN to CLB_R11C2.COUT : 1.5ns (13.6ns)
Thru: NET M/Carry_5 CLB_R11C2.COUT to CLB_R10C2.CIN : 0.0ns (13.6ns)
Thru: BLK _PIX_VAL6 CLB_R10C2.CIN to CLB_R10C2.COUT : 1.5ns (15.1ns)
Thru: NET M/Carry_7 CLB_R10C2.COUT to CLB_R9C2.CIN : 0.0ns (15.1ns)
Thru: BLK _PIX_VAL8 CLB_R9C2.CIN to CLB_R9C2.COUT : 1.5ns (16.6ns)
Thru: NET M/Carry_9 CLB_R9C2.COUT to CLB_R8C2.CIN : 0.0ns (16.6ns)
Thru: BLK PIX_VAL10 CLB_R8C2.CIN to CLB_R8C2.COUT : 1.5ns (18.1ns)
Thru: NET /Carry_11 CLB_R8C2.COUT to CLB_R7C2.CIN : 0.0ns (18.1ns)
Thru: BLK PIX_VAL12 CLB_R7C2.CIN to CLB_R7C2.COUT : 1.5ns (19.6ns)
Thru: NET /Carry_13 CLB_R7C2.COUT to CLB_R6C2.CIN : 0.0ns (19.6ns)
Thru: BLK PIX_VAL14 CLB_R6C2.CIN to CLB_R6C2.Y : 7.5ns (27.1ns)
Thru: NET PIX_VAL15 CLB_R6C2.Y to A3.O : 3.7ns (30.8ns)
To: BLK PIX_VAL15 A3.O to PAD : 10.0ns (40.8ns)

From: BLK /PIX_INC0 PAD to E14.I1 : 3.0ns (3.0ns)
Thru: NET PIX_INC0 E14.I1 to CLB_R13C2.F2 : 1.9ns (4.9ns)
Thru: BLK _PIX_VAL0 CLB_R13C2.F2 to CLB_R13C2.COUT : 6.0ns (10.9ns)
Thru: NET M/Carry_1 CLB_R13C2.COUT to CLB_R12C2.CIN : 0.0ns (10.9ns)
Thru: BLK _PIX_VAL2 CLB_R12C2.CIN to CLB_R12C2.COUT : 1.5ns (12.4ns)
Thru: NET M/Carry_3 CLB_R12C2.COUT to CLB_R11C2.CIN : 0.0ns (12.4ns)
Thru: BLK _PIX_VAL4 CLB_R11C2.CIN to CLB_R11C2.COUT : 1.5ns (13.9ns)
Thru: NET M/Carry_5 CLB_R11C2.COUT to CLB_R10C2.CIN : 0.0ns (13.9ns)
Thru: BLK _PIX_VAL6 CLB_R10C2.CIN to CLB_R10C2.COUT : 1.5ns (15.4ns)
Thru: NET M/Carry_7 CLB_R10C2.COUT to CLB_R9C2.CIN : 0.0ns (15.4ns)
Thru: BLK _PIX_VAL8 CLB_R9C2.CIN to CLB_R9C2.COUT : 1.5ns (16.9ns)
Thru: NET M/Carry_9 CLB_R9C2.COUT to CLB_R8C2.CIN : 0.0ns (16.9ns)
Thru: BLK PIX_VAL10 CLB_R8C2.CIN to CLB_R8C2.COUT : 1.5ns (18.4ns)
Thru: NET /Carry_11 CLB_R8C2.COUT to CLB_R7C2.CIN : 0.0ns (18.5ns)
Thru: BLK PIX_VAL12 CLB_R7C2.CIN to CLB_R7C2.COUT : 1.5ns (20.0ns)
Thru: NET /Carry_13 CLB_R7C2.COUT to CLB_R6C2.CIN : 0.0ns (20.0ns)
Thru: BLK PIX_VAL14 CLB_R6C2.CIN to CLB_R6C2.Y : 7.5ns (27.5ns)
Thru: NET PIX_VAL15 CLB_R6C2.Y to A3.O : 3.7ns (31.2ns)
To: BLK PIX_VAL15 A3.O to PAD : 10.0ns (41.2ns)

ADDRESS CALCULATION

LOGICAL PATH CRITICAL ELEMENT DELAY CUMULATIVE
---------- ---------- ----------
Source clock net : "CTRL_1" (Rising edge)
Worst case skew from clock origin "RAY_PER_PROJ10.XQ" is 21.6ns.
From: BLK DEX_ADDR6 CLOCK to CLB_R10C5.XQ : 3.0ns (3.0ns)
Thru: NET DEX_ADDR6 CLB_R10C5.XQ to A1.O : 7.7ns (10.7ns)
To: BLK DEX_ADDR6 A1.O to PAD : 10.0ns (20.7ns)

Source clock net : "CTRL_1" (Rising edge)
Worst case skew from clock origin "RAY_PER_PROJ10.XQ" is 21.6ns.
From: BLK DEX_ADDR0 CLOCK to CLB_R13C5.YQ : 3.0ns (3.0ns)
Thru: NET DEX_ADDR1 CLB_R13C5.YQ to CLB_R13C5.G4 : 1.4ns (4.4ns)
Thru: BLK DEX_ADDR0 CLB_R13C5.G4 to CLB_R13C5.COUT : 6.0ns (10.4ns)
Thru: NET M/Carry_1 CLB_R13C5.COUT to CLB_R12C5.CIN : 0.0ns (10.5ns)
Thru: BLK DEX_ADDR2 CLB_R12C5.CIN to CLB_R12C5.COUT : 1.5ns (12.0ns)
Thru: NET M/Carry_3 CLB_R12C5.COUT to CLB_R11C5.CIN : 0.0ns (12.0ns)
Thru: BLK DEX_ADDR4 CLB_R11C5.CIN to CLB_R11C5.COUT : 1.5ns (13.5ns)
Thru: NET M/Carry_5 CLB_R11C5.COUT to CLB_R10C5.CIN : 0.0ns (13.5ns)
Thru: BLK DEX_ADDR6 CLB_R10C5.CIN to CLB_R10C5.COUT : 1.5ns (15.0ns)
Thru: NET M/Carry_7 CLB_R10C5.COUT to CLB_R9C5.CIN : 0.0ns (15.0ns)
Thru: BLK DEX_ADDR8 CLB_R9C5.CIN to CLB_R9C5.COUT : 1.5ns (16.5ns)
Thru: NET M/Carry_9 CLB_R9C5.COUT to CLB_R8C5.CIN : 0.0ns (16.5ns)
Thru: BLK EX_ADDR10 CLB_R8C5.CIN to CLB_R8C5.COUT : 1.5ns (18.0ns)
Thru: NET /Carry_11 CLB_R8C5.COUT to CLB_R7C5.CIN : 0.0ns (18.0ns)
Thru: BLK EX_ADDR12 CLB_R7C5.CIN to CLB_R7C5.COUT : 1.5ns (19.5ns)
Thru: NET /Carry_13 CLB_R7C5.COUT to CLB_R6C5.CIN : 0.0ns (19.5ns)
Thru: BLK EX_ADDR14 CLB_R6C5.CIN to CLB_R6C5.COUT : 1.5ns (21.0ns)
Thru: NET /Carry_15 CLB_R6C5.COUT to CLB_R5C5.CIN : 0.0ns (21.0ns)
Thru: BLK NC_OFFSET CLB_R5C5.CIN to CLB_R5C5.X : 6.0ns (27.0ns)
Thru: NET NC_OFFSET CLB_R5C5.X to CLB_R13C8.K : 11.7ns (38.7ns)
To: BLK OFFSET2 CLB_R13C8.K to CLOCK INPUT : 0.0ns (38.7ns)

Source clock net : "INC_OFFSET" (Rising edge)
From: BLK OFFSET0 CLOCK to CLB_R14C8.YQ : 3.0ns (3.0ns)
Thru: NET OFFSET1 CLB_R14C8.YQ to CLB_R13C7.G1 : 2.5ns (5.5ns)
Thru: BLK _ADDRESS0 CLB_R13C7.G1 to CLB_R13C7.COUT : 6.0ns (11.5ns)
Thru: NET M/Carry_1 CLB_R13C7.COUT to CLB_R12C7.CIN : 0.0ns (11.5ns)
Thru: BLK _ADDRESS2 CLB_R12C7.CIN to CLB_R12C7.COUT : 1.5ns (13.0ns)
Thru: NET M/Carry_3 CLB_R12C7.COUT to CLB_R11C7.CIN : 0.0ns (13.0ns)
Thru: BLK _ADDRESS4 CLB_R11C7.CIN to CLB_R11C7.COUT : 1.5ns (14.5ns)
Thru: NET M/Carry_5 CLB_R11C7.COUT to CLB_R10C7.CIN : 0.0ns (14.6ns)
Thru: BLK _ADDRESS6 CLB_R10C7.CIN to CLB_R10C7.COUT : 1.5ns (16.1ns)
Thru: NET M/Carry_7 CLB_R10C7.COUT to CLB_R9C7.CIN : 0.0ns (16.1ns)
Thru: BLK _ADDRESS8 CLB_R9C7.CIN to CLB_R9C7.COUT : 1.5ns (17.6ns)
Thru: NET M/Carry_9 CLB_R9C7.COUT to CLB_R8C7.CIN : 0.0ns (17.6ns)
Thru: BLK ADDRESS10 CLB_R8C7.CIN to CLB_R8C7.COUT : 1.5ns (19.1ns)
Thru: NET /Carry_11 CLB_R8C7.COUT to CLB_R7C7.CIN : 0.0ns (19.1ns)
Thru: BLK ADDRESS12 CLB_R7C7.CIN to CLB_R7C7.COUT : 1.5ns (20.6ns)

Thru: NET /Carry_13 CLB_R7C7.COUT to CLB_R6C7.CIN : 0.0ns (20.6ns)
To: BLK ADDRESS14 CLB_R6C7.CIN to SETUP : 4.5ns (25.1ns)

From: BLK 81/INDEX7 PAD to R10.I2 : 3.0ns (3.0ns)
Thru: NET INDEX7_1 R10.I2 to CLB_R10C9.C4 : 2.9ns (5.9ns)
Thru: BLK OM/BANK31 CLB_R10C9.C4 to CLB_R10C9.YQ : 3.5ns (9.4ns)
Thru: NET INDEX7_2 CLB_R10C9.YQ to CLB_R10C7.G4 : 1.9ns (11.2ns)
Thru: BLK _ADDRESS6 CLB_R10C7.G4 to CLB_R10C7.COUT : 6.0ns (17.2ns)
Thru: NET M/Carry_7 CLB_R10C7.COUT to CLB_R9C7.CIN : 0.0ns (17.2ns)
Thru: BLK _ADDRESS8 CLB_R9C7.CIN to CLB_R9C7.COUT : 1.5ns (18.7ns)
Thru: NET M/Carry_9 CLB_R9C7.COUT to CLB_R8C7.CIN : 0.0ns (18.8ns)
Thru: BLK ADDRESS10 CLB_R8C7.CIN to CLB_R8C7.COUT : 1.5ns (20.3ns)
Thru: NET /Carry_11 CLB_R8C7.COUT to CLB_R7C7.CIN : 0.0ns (20.3ns)
Thru: BLK ADDRESS12 CLB_R7C7.CIN to CLB_R7C7.COUT : 1.5ns (21.8ns)
Thru: NET /Carry_13 CLB_R7C7.COUT to CLB_R6C7.CIN : 0.0ns (21.8ns)
To: BLK ADDRESS14 CLB_R6C7.CIN to SETUP : 4.5ns (26.3ns)

From: BLK _ADDRESS4 CLOCK to CLB_R11C7.XQ : 3.0ns (3.0ns)
Thru: NET _ADDRESS4 CLB_R11C7.XQ to P4.O : 7.8ns (10.8ns)
To: BLK _ADDRESS4 P4.O to PAD : 10.0ns (20.8ns)

RAY NUMBER CALCULATION

LOGICAL PATH CRITICAL ELEMENT DELAY CUMULATIVE

Source clock net : "CTRL_2" (Rising edge)
From: BLK I63_HM/q0 CLOCK to CLB_R12C3.XQ : 3.0ns (3.0ns)
Thru: NET I63_HM/q0 CLB_R12C3.XQ to CLB_R12C3.F1 : 1.3ns (4.3ns)
Thru: BLK I63_HM/q0 CLB_R12C3.F1 to CLB_R12C3.COUT : 6.0ns (10.3ns)
Thru: NET M/Carry_1 CLB_R12C3.COUT to CLB_R11C3.CIN : 0.0ns (10.3ns)
Thru: BLK I63_HM/q2 CLB_R11C3.CIN to CLB_R11C3.COUT : 1.5ns (11.8ns)
Thru: NET M/Carry_3 CLB_R11C3.COUT to CLB_R10C3.CIN : 0.0ns (11.8ns)
Thru: BLK I63_HM/q4 CLB_R10C3.CIN to CLB_R10C3.COUT : 1.5ns (13.3ns)
Thru: NET M/Carry_5 CLB_R10C3.COUT to CLB_R9C3.CIN : 0.0ns (13.3ns)
Thru: BLK I63_HM/q6 CLB_R9C3.CIN to CLB_R9C3.COUT : 1.5ns (14.8ns)
Thru: NET M/Carry_7 CLB_R9C3.COUT to CLB_R8C3.CIN : 0.0ns (14.8ns)
Thru: BLK I63_HM/q8 CLB_R8C3.CIN to CLB_R8C3.COUT : 1.5ns (16.3ns)
Thru: NET M/Carry_9 CLB_R8C3.COUT to CLB_R7C3.CIN : 0.0ns (16.3ns)
Thru: BLK 63_HM/q10 CLB_R7C3.CIN to CLB_R7C3.COUT : 1.5ns (17.8ns)
Thru: NET /Carry_11 CLB_R7C3.COUT to CLB_R6C3.CIN : 0.0ns (17.9ns)
Thru: BLK 63_HM/q12 CLB_R6C3.CIN to CLB_R6C3.COUT : 1.5ns (19.4ns)
Thru: NET /Carry_13 CLB_R6C3.COUT to CLB_R5C3.CIN : 0.0ns (19.4ns)
Thru: BLK 63_HM/q14 CLB_R5C3.CIN to CLB_R5C3.COUT : 1.5ns (20.9ns)
Thru: NET /Carry_15 CLB_R5C3.COUT to CLB_R4C3.CIN : 0.0ns (20.9ns)
Thru: BLK $1N146 CLB_R4C3.CIN to CLB_R4C3.X : 6.0ns (26.9ns)
Thru: NET $1N146 CLB_R4C3.X to CLB_R8C3.G3 : 3.3ns (30.2ns)
To: BLK I63_HM/q8 CLB_R8C3.G3 to SETUP : 4.5ns (34.7ns)

Source clock net : "$1N146" (Rising edge)
From: BLK RAY_NUM2 CLOCK to CLB_R5C9.YQ : 3.0ns (3.0ns)
Thru: NET RAY_NUM3 CLB_R5C9.YQ to CLB_R12C6.F2 : 17.6ns (20.6ns)
Thru: BLK OM/BANK09 CLB_R12C6.F2 to CLB_R12C6.X : 6.0ns (26.6ns)
Thru: NET OM/BANK09 CLB_R12C6.X to CLB_R9C6.F4 : 3.4ns (29.9ns)
Thru: BLK PER_PROJ9 CLB_R9C6.F4 to CLB_R9C6.Y : 6.0ns (35.9ns)
Thru: NET PER_PROJ9 CLB_R9C6.Y to CLB_R8C3.G1 : 3.7ns (39.6ns)
To: BLK I63_HM/q8 CLB_R8C3.G1 to SETUP : 4.5ns (44.1ns)