THE USE OF RADON TRANSFORMS IN FULLY 3-DIMENSIONAL POSITRON VOLUME IMAGING - A FEASIBILITY STUDY

by

Christine J. Dykstra
B.Sc., Simon Fraser University, 1986

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
April 1989

APPROVAL

Name: Christine Joy Dykstra

Degree: Master of Science

Title of thesis: The Use of Radon Transforms in Fully 3-Dimensional Positron Volume Imaging - A Feasibility Study

Examining committee:

Chairman: Dr. Lou Hafer

Dr. R. Harrop, Professor, Senior Supervisor

Dr. A. Liestman, Associate Professor

Dr. M. S. Atkins, Assistant Professor

Dr. N. Cercone, External Examiner

Date Approved April 7 89

PARTIAL COPYRIGHT LICENSE

I hereby grant to Simon Fraser University the right to lend my thesis, project or extended essay (the title of which is shown below) to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users. I further agree that permission for multiple copying of this work for scholarly purposes may be granted by me or the Dean of Graduate Studies. It is understood that copying or publication of this work for financial gain shall not be allowed without my written permission.

Title of Thesis/Project/Extended Essay

The use of Radon Transforms in Fully 3-dimensional Positron Volume Imaging - A Feasibility Study.

Author: _____
(signature)

Christine Joy Dykstra
(name)

April 14, 1989
(date)

ABSTRACT

The purpose of this thesis is to assess the feasibility of using Radon transforms ((n-1) dimensional projections of an n dimensional object) in the reconstruction of good images from 3D positron emission data. To show such feasibility it is necessary to solve two problems. First it must be demonstrated that the Radon transform can be satisfactorily obtained from the emission data, and second, it must be shown that in some non-trivial case a good image can be reconstructed from the transform. In this paper, solutions are presented for both of these problems.

The solution to the first problem consists of a computer algorithm which credits a line (one unit of data) to each member of a set of predefined planes in which it is considered to lie (2D projection). The solution to the second problem requires a demonstration that it is theoretically correct to use Radon transforms and that the discretization errors introduced when the algorithm is implemented on a computer can be corrected. Examples of final images produced by these algorithms are shown, with and without these corrections, to support claims made in this thesis. The question as to whether the results are promising enough to continue work in this area is addressed and the direction of further work is discussed

ACKNOWLEDGEMENTS

I would like to thank a number of people who have put their knowledge and time into this thesis and helped make it better than it could have been without their assistance.

First, special recognition goes to my senior supervisor, Dr. Ronald Harrop, for his time, teaching, encouragement and much, much patience. Also many thanks to all the people working with the Tomograph Development Group at TRIUMF, in particular Dr. Joel Rogers, who have made many useful comments about this research.

Thanks to my friends who have given of their time, and expertise to put this thesis together.

Also, thanks to the members of the DIFTEB club who helped me finish on time, and of course, my parents without whose support this work may never have been started.

TABLE OF CONTENTS

Approval ii

Abstract iii

Acknowledgements iv

List of Figures vi

1. Introduction 1

2. Positron Volume Imaging 5

2.1 Introduction to 2D and 3D Image Reconstruction 5

2.2 2D Image Reconstruction 10

2.3 3D Image Reconstruction 14

2.4 Applications of PET and PVI 17

3. Theory of 3D Image Reconstruction 19

4. Implementation 27

4.1 Getting the Radon Transform 28

4.2 Filtering 33

4.3 Backprojection 38

5. Testing 39

5.1 Getting the Radon Transform 39

5.2 Filtered Backprojection 42

6. Discussion 49

Bibliography 51

A Inverse Fourier transform of the Hamming filter 53

B Calculation of k(0) 55

C Source code 57

LIST OF FIGURES

2.1. Single annihilation. 6

2.2. Single slice detection. 8

2.3. Multiple slice detection. 9

2.4. 3D detection. 10

2.5. One projection direction for 2D image reconstruction. 11

2.6. Data collection in 3 projection directions, and unfiltered backprojection. 13

2.7. Filtered backprojection. 14

2.8. 1D projections onto 2D space. 15

2.9. 2D projections onto 1D space. 17

3.1. 1D projection onto a u'=0 plane. 20

3.2. 2D projection onto the t"=u"=0 line. 22

4.1. The Hamming filter. 37

5.1. Reconstruction of sphere with 2^{nd} difference filter. 41

5.2. Reconstruction of sphere with Hamming filter. 41

5.3. Single ellipsoid, 36x36 with 2^{nd} difference filter. 46

5.4. Single ellipsoid, 36x36 with Hamming filter. 46

5.5. Single ellipsoid, 90x90 with 2^{nd} difference filter. 47

5.6. Single ellipsoid, 90x90 with Hamming filter. 47

5.7. Head phantom, 36x36 with 2^{nd} difference filter. 48

5.8. Head phantom, 90x90 with Hamming filter. 48

1. Introduction

Positron emission tomography (PET) is one of many imaging techniques used in medicine to obtain, in a non-invasive manner, information about the internal form or function of a subject from externally gathered data. The existence of some of these techniques, such as electroencephalography (EEG), X-ray, ultrasound and computer assisted tomography (CAT, CT or XCT), is common knowledge. Some lesser known imaging modalities are magnetic resonance imaging (MRI), single photon emission computed tomography (SPECT) and PET. The tomographic methods and MRI use a mathematical technique which takes the externally gathered data and uses it to reconstruct a cross section through the subject (hence the use of the word 'tomography', tomo=cut or slice). The usefulness of being able to view a cross section of a living subject by non-invasive means is without question, and the different techniques just mentioned provide varied kinds of information about the subject.

In CAT, X-ray beams are passed through the subject from many directions and detected on the opposite side. The intensity of the X-ray beam when it is detected on the far side is inversely related to the density of the material in-between. This information can be used to reconstruct a density map of the plane which was X-rayed. Such a map shows structures like bone, cartilage, water, air, etc. MRI also produces data about the density of an object. The density of hydrogen nuclei over a plane is measured and reconstructed, and the values of the hydrogen density are related to the density of the object over the reconstructed plane. Rather than measure density, SPECT and PET measure the distribution of a radioactive tracer which has been injected into the subject. This data is reconstructed into a 2-dimensional (2D) map of the distribution of the tracer over that

plane. SPECT and PET differ in the type of radioactive emitter used, SPECT uses a low energy photon emitter where PET uses a high energy positron emitter. The difference between these two methods of imaging primarily lies in the types of tracers that can be used, and in the hardware employed to collect the data.

The theoretical basis for reconstruction of the data into images for the above modalities is virtually the same [Deans 1983]. The difference between the modalities lies in the type of data measured and in the manner in which such data is measured, and also in the meaning attached to the final images.

In this work we are interested in reconstruction methods for fully 3D PET data, meaning data which has been collected from a wide section of the subject, rather than from a single slice. Since 3D positron emission data is neither obtained nor reconstructed in slices, the technique is properly called positron volume imaging (PVI) rather than PET The first step in reconstructing 3D images is collecting the data, the tomograph[1] must be capable of measuring data over a 3D section of the subject. Our work is designed for a new PVI machine currently being designed at TRIUMF. When the data has been obtained, there are two ways of reconstructing the image. The first, and most common, method is to use 1D projections of the data onto 2D spaces, filter over the 2D spaces and reconstruct the 3D object. The second, and equally valid, method is to use 2D projections of the object onto 1D spaces, filter over the 1D space and reconstruct. Note that in 2D image reconstruction there is no difference between these two methods. It is with this second, and less common method, that this thesis is concerned. Both methods will be discussed further in later chapters.

1. Unfortunately all positron emission machines are called tomographs, regardless of how they collect data. In this thesis the type of machine in question will be made clear by the context.

The set of 2D projections of a 3D function is the Radon transform [Radon 1917] of the function. Radon [1917] states that given the set of (n-1) dimensional projections of an n dimensional function, it is possible to reconstruct the original nD function. In PVI the nD function is the distribution of radioactivity in the object and the (n-1)D projections are obtained from the measured data. This method for reconstruction has been applied to many different forms of image reconstruction, including optics and electronmicroscopy, but it has not been applied to PVI. An unsuccessful attempt was made to apply Radon reconstruction to MRI [Shepp 1980]. The reasons why the attempt failed are important to this work and will be discussed in detail later.

We shall be specifically concerned with testing the feasibility of using Radon's transform in PVI reconstruction. There are two main problem to overcome before Radon transforms can be used. The first lies in the fact that the Radon transform is not directly measured by the tomograph and that the data must be converted to Radon transform form. The second is that of reconstructing good images from the transform. Both of these problems are considered and acceptable solutions found for them, indicating the feasibility of the method. Although the application of the theory in this paper is with positron emission data, it could be applied to any image reconstruction method where the data can be put into Radon transforms.

PVI cannot be considered common knowledge in Computing Science, so chapter 2 of this thesis will contain an introduction to the technique. It will also contain a discussion of the reasons for doing the 3D imaging and present some medical applications for the work. Chapter 3 provides the theoretical background for 3D imaging and proves that both methods of 3D image reconstruction are theoretically valid. The reason for investigating the feasibility of Radon transforms rather than using the more common 1D projection method is presented here as well. The conversion of the measured data to the Radon

transform and the implementation of the Radon reconstruction method is described in chapter 4 and the testing and results of the implementation are in chapter 5. Chapter 6 contains the conclusions and a discussion of further work to be done in this area.

2. Positron Volume Imaging

2.1. Introduction to 2D and 3D image reconstruction

The aim of positron emission imaging is to measure accurately the distribution of a radioactive tracer inside a 3D object. The radioactive tracer is composed of a positron emitter attached to some biological molecule like oxygen, water or glucose whose actions are of interest to a scientist or doctor. The tracer is injected into the object[2], where it follows the normal path taken by the biological molecule. As each radioactive molecule travels through the body, it emits one positron which travels a short distance, finds an electron and annihilates. The exact time at which the positron is emitted varies for each molecule, the overall rate is determined by the radioactive decay of the tracer. Each annihilation, called an **event**, produces 2 gamma rays which travel away from their point of production at 180° to one another (opposite directions along a straight line) and are detected by the tomograph detectors surrounding the object. A diagram showing one annihilation is given in figure 2.1. The positions in space where the gamma rays are detected define a line, called an **event line**, through the object, somewhere along which the annihilation, or event, took place. Knowing the position of the event line gives an indication of the position of the tracer molecule at the time the positron was emitted. Given enough event lines, it is possible to reconstruct the original distribution of the tracer molecules within the imaged object. Intuitively it seems that the number of detected

2. Note that in the following discussion the 'object' being imaged is assumed to be human and the region of interest is the brain. It is, of course, both possible and useful to image other regions, and while this may make a difference to the design and cost of the tomograph, it makes no difference to this discussion and no distinction will be made.

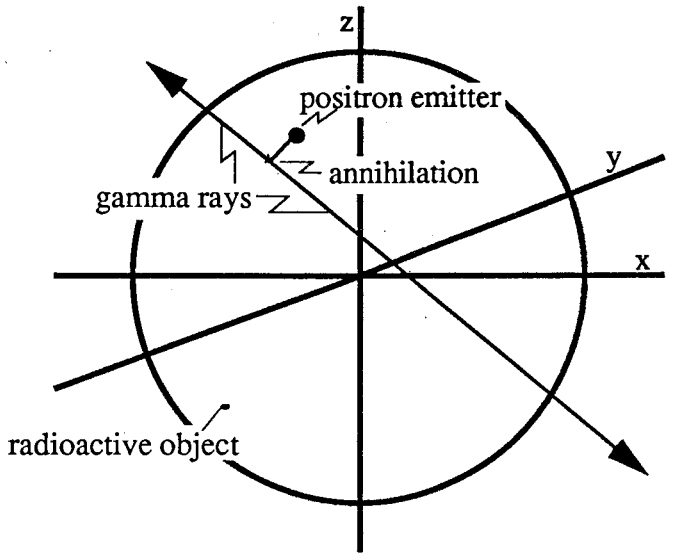


Figure 2.1:Diagram of a single annihilation (not to scale).

events must be large for this to be possible, and in fact this is the case.

There are, of course, errors in the above simple statement. The positron annihilates a few mm away from the point of production, the gamma rays do not travel at exactly 180° from one another, some gamma rays are lost (attenuated) or scattered as they travel through tissue and the detectors themselves are not perfectly accurate in recording the position of the rays. To accentuate the problem further, it is impossible to enclose the object with detectors completely. Either the machine would have to be large enough to contain the entire body (theoretically as well as financially difficult) or else just the interesting section is surrounded by a ring of detectors. These considerations, and others, such as patient movement, cause a deterioration in the quality of the image and place a limit on the machine's resolution. Compensation for and/or correction of attenuation, scatter and detector error are made to the data before the image reconstruction work is started. In order to focus on the problems that the reconstruction algorithms must deal

with, the data used in this thesis is taken either from a perfect sampling of an object, that is, a sampling which contains no statistical noise, detection error, scatter or attenuation, or from a sampling which does contain statistical noise but not detection error, scatter or attenuation.

In PET (that is, imaging just a plane through the object) only events which lie in a plane are detected and used in reconstruction. These are the events which come in perpendicular to the axis of symmetry of the detector surface where the detector surface is a ring of detectors. In all commercial and most research machines, its resolution is partially determined by the size of the detectors, so the individual detectors in the ring are small, and the ring is very thin as shown in figure 2.2a. Since the entire object carries the radioactive tracer, and gamma rays may travel in any direction, most events are never detected. This is essentially wasted radiation dose to the subject. A further waste is the time taken to collect enough events for the reconstruction of an image which is not too badly affected by statistical noise.

More data can be collected if detector rings are placed side by side to form a stack. Events which lie in any plane imaged by a ring will be used. It is even possible to use events which are detected in adjacent rings, but not any detected in non-adjacent rings. Any gamma ray pair which is detected across two rings which are not adjacent, is rejected. Most tomographs are built like this, with the rings of detectors separated by lead septa which block any gamma rays which do not lie in or near the plane determined by a detector ring. Commonly these machines are called multiple slice machines [Evans, Harrop, Heywood, *et al* 1983]. With this hardware setup, the reconstructed images of each plane can be stacked up to form a 3D image, but this is not considered proper 3D imaging since the events used are still primarily those which are perpendicular to the axis of symmetry of the detector ring. With these tomographs, all gamma rays detected are assumed to have hit

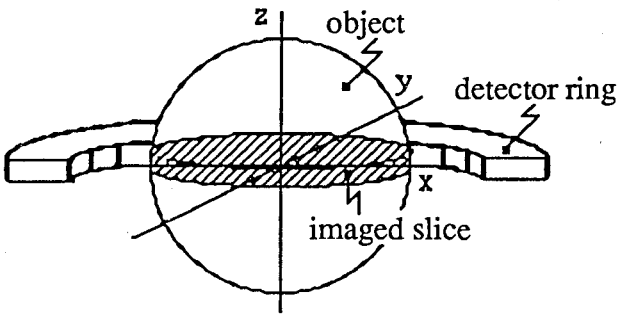


Figure 2.2a : Single slice detection.

Only the area of the object which is surrounded by the detectors can be imaged.

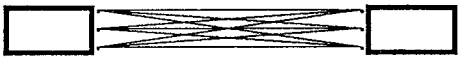


Figure 2.2b : Cross section of a slice.

Simplified diagram showing the angle at which events may be detected in a slice.

the centre of the detector, regardless of their actual position, so the detector size determines the lower limit of the resolution. The space requirements for the electronics associated with each detector places a limit on how small the detectors can get. A different problem with these machines is the uneven imaging over the thickness of each slice (a slice is only as thin as the detectors are small). Consider the cross section of one slice. Any source of radioactivity at the middle of a slice produces events which may vary over a range of angles and still be accepted into that slice and be credited to the centre of the slice. A radioactive source placed at the edge of a slice will emit far fewer gamma rays which are detected entirely within that slice. It would therefore not appear to be as radioactive as the source in the middle (see figure 2.3b). This creates a problem, called the partial volume effect. There is non-uniformity within slices along the axial direction.

In fully 3D imaging[3], all detected events are used. The image is reconstructed as a

3. For the remainder of this paper, fully 3D will be called simply 3D, the other methods will be referred to as single slice or multiple slice 2D reconstruction

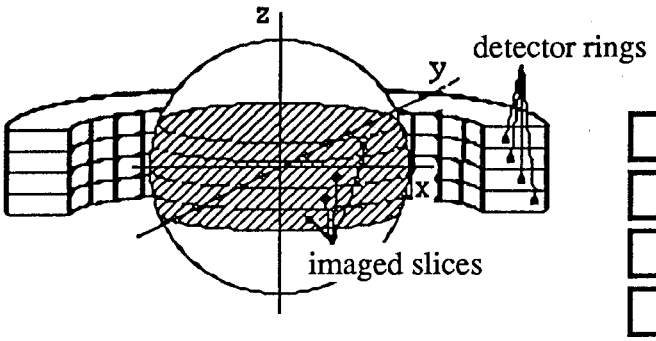


Figure 2.3a : Multiple slice detection.

A larger section of the object is imaged, but only those events which are detected in a slice are used.

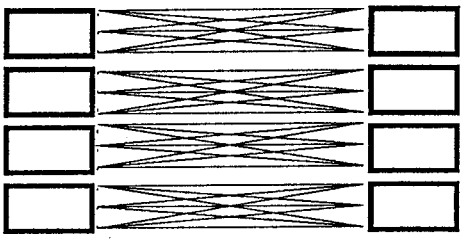


Figure 2.3b : Cross section.

Simplified diagram demonstrating the non-uniformity in the axial direction (along the z axis).

true 3D object rather than as a stack of 2D ones. Since no data is wasted, a better image can be reconstructed with the same amount of radiation dose to the subject or, alternately, the same image can be reconstructed with less dose to the person. 3D imaging can be accomplished in two different ways. One is to use a stack of detector rings without the septa, the other is to use large, position sensitive detectors (as in figure 2.4a). In the first method, resolution in the axial direction (along the z axis) is still limited by the width of the detector rings. The second method has an axial resolution which is equivalent to the resolution in the x and y directions and so does not have this problem. Our reconstruction method is designed for the PVI tomograph with large, position sensitive detectors which is currently being designed at TRIUMF [Rogers, Harrop, Coombes, *et al* 1989]. There are no commercial machines of this kind as yet.

Of course 3D imaging has its own set of problems. The main difficulty concerns the number of scattered events. A scattered event is one where at least one ray of the gamma ray pair changes course at least once before it is detected. The event line produced by a

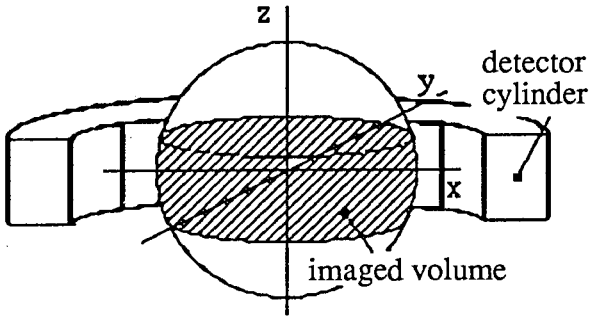


Figure 2.4a : 3D detection

All events where both gamma rays hit some detector are collected.

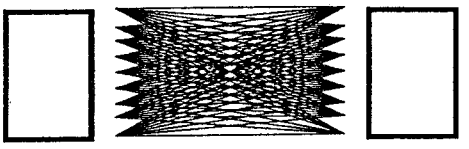


Figure 2.4b : Cross section

The object is imaged continuously and smoothly along the z axis.

scattered event is not the line along which the event occurred and therefore shouldn't be used in reconstruction. The problem lies in recognizing and correcting for, or removing, these events. In 2D imaging, only events which scatter inside the plane and which remain inside the plane will be collected by the detectors. This is not a large fraction of the total number of events detected and therefore is not a great problem. With 3D imaging, any gamma ray pair where both rays hit detectors is accepted. This is a large percentage of all events and is a significant problem. It is not, however, a reconstruction problem and is not part of this thesis except insofar as its existence needs to be noted and it need to be mentioned that there are empirical methods by which the problem can be handled, and better methods are being developed

2.2. 2D Image Reconstruction

Reconstruction can be considered to be an inverse transform operation applied to a

function. Suppose f(x,y) is the 2D distribution of tracer in the object. The set of event lines (measured data), F(θ,s), can be considered as a transform of the function f(x,y). Here (x,y) are the Cartesian coordinates for a point in 2D space, θ specifies the angle of a line through the origin and s is a position on that line. The line (θ,s) is the line at s which is perpendicular to θ. In order to get the original distribution back an inverse transform must be performed on the measured data; $f(x,y) = T^{-1}F(\theta,s)$. In PET the inverse transform is performed by reconstruction, of which the two main steps are filtering and backprojection.

Filtering is done to correct for errors introduced by the discreteness of the data collection and by the backprojection process. Backprojection is the method by which the image is reconstructed. These two can be done in any order, and though the filters would differ, the results would be same. Usually it is computationally easier to do the filtering on the transform before it is backprojected. In this discussion, backprojection will be

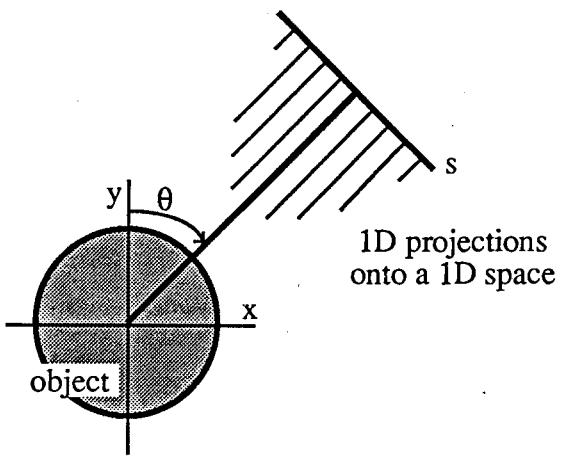


Figure 2.5 : One projection direction for 2D image reconstruction

explained first in order to make clear the errors introduced which are corrected by the filter.

In 2D reconstruction all the event lines, as they are detected, are binned into projections (which are set by the hardware). This is done by crediting all events detected along a line to that line[4]. The final value for the line represents the sum of all the activity inside the object over that line. This is a 1D projection of the distribution of the tracer over the plane being imaged. These projection lines vary over all directions from which data is collected. Each set of parallel projections (varying s for some fixed θ, as shown in figure 2.5) comprise one view of the plane. Each view is called a projection direction and is defined by the direction of the 1D projections.

To reconstruct the image, the values of the 1D projection in all projection directions are spread back over the image. The value of one projection is credited to all points in the reconstruction space which lie on that line. This is the problem with backprojection, as can be seen in figure 2.6. The collected data is indiscriminately credited to all points across the image where the event may have originated, including areas where there never was any radioactivity.This problem is handled by applying a filter to each view of the object (each projection direction), as in figure 2.7. The filter adds negative values to the data where it was previously zero so that, during backprojection, the positive values outside the object from one direction will be canceled by negative values from other directions. This, of course, only works where projections overlap, which brings in a different problem, the number of projection directions. If the object is viewed from too few directions the reconstruction will contain artifacts from uncanceled portions of the backprojected data. The size of the area around the object which properly cancels out increases as the number

4. Except in the theoretical portion of this paper, all points, lines and planes have 'thickness', due to the discreteness of the data and algorithms.

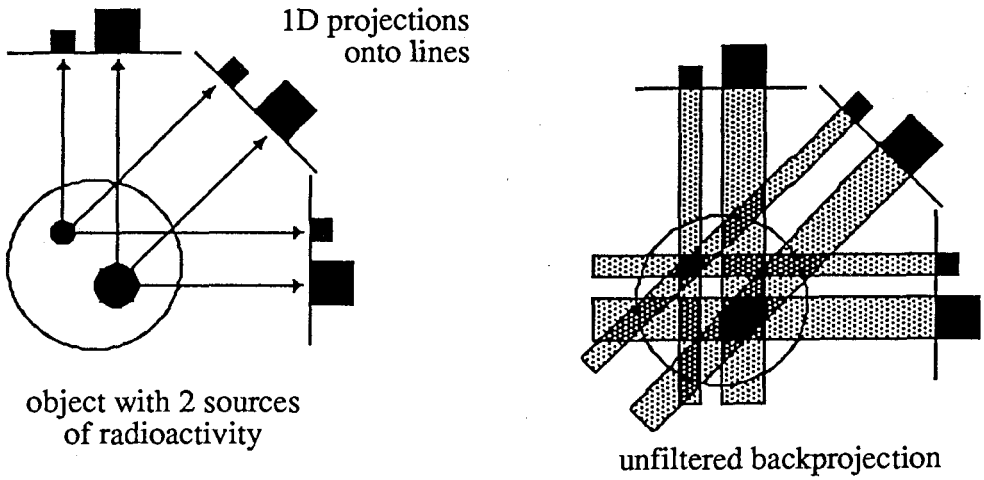


Figure 2.6 : Data collection in 3 projection directions, and unfiltered backprojection.

The error introduced by indiscriminately crediting all points which lie on a projection line with the value of the projection is clearly shown.

of projection directions increase.

Another problem with the number of projection directions is the inadequate sampling of the data [Barney 1988, Kak and Slaney 1988]. High frequency components are not sampled often enough to be correctly measured and can appear to be of a lower frequency. This problem is called aliasing and it also introduces artifacts into the image which can only be removed by better sampling. All frequencies above the Nyquist frequency (that is, the highest frequency which the data collection method can correctly sample) are known to be incorrect and are removed by introducing a cutoff window at a frequency less than or equal to the Nyquist frequency. Applying a sharp cutoff to the data introduces another error. It is called spectral leakage and it adds error to the reconstructed image [Chesler and Riederer 1975]. This effect is reduced by applying a function to the cutoff window so that the window approaches zero smoothly. By choosing a cutoff which is less than the Nyquist frequency the data can be smoothed so that a nicer looking image is produced, but this will cause a deterioration in the resolution and may obscure valid results.

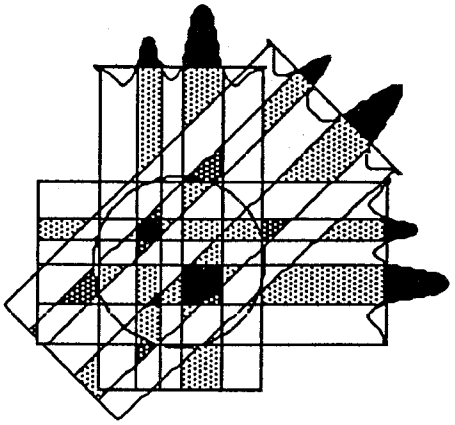


Figure 2.7 : Filtered backprojection.

When a filter is used on the data before backprojection, the negative and positive values cancel out around the radioactive sources. Note that in this diagram, only the positive values have been shaded in. Accumulations of negative values do affect the image but would only cause cause confusion here.

The cutoff window can be applied to the filter before the filter is convolved with the projection data so that the data need only be convolved with one filter rather than with both the filter and the cutoff window. Since convolution is a time consuming procedure, this is important for reducing the time necessary for image reconstruction.

2.3. 3D Image Reconstruction

The main difference between 2D and 3D reconstruction is just the addition of the extra dimension to the reconstruction process. This can be done in two ways. The first method is to collect the data into a set of 1D projections, as for 2D reconstruction, but then have the parallel projections vary over a plane (2D space) instead of a line [Kinahan, Harrop, Rogers and Johnson 1988]. A 1D projection is specified by (θ,ϕ,s,t), where (θ,ϕ)

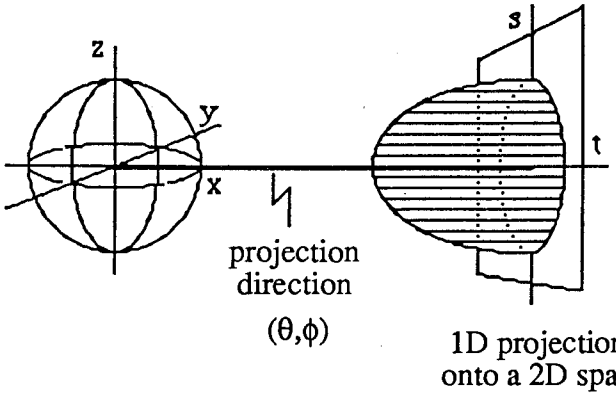


Figure 2.8 : 1D projections onto 2D space.

The set of 1D projections (θ=90, φ=0; s, t varying) are arranged as a set of parallel lines on a plane ("bed of nails"), as shown in the diagram.

defines the plane through the origin which is orthogonal to the line (θ,φ), and (s,t) defines a point on the plane (as in figure 2.8). θ is the angle between the line and the z axis and φ is the angle through the x-y plane. The line (θ,φ,s,t) is the line through the point (s,t) parallel to the line (θ,φ). The projection planes then vary over all directions (all θ,φ) from which data is collected. Each projection plane is a 2D view of the object and is considered to be one projection direction. The filtering is done on this 2D space and the filtered 1D projections are backprojected over the 3D reconstruction space. As with 2D, the value of the filtered 1D projection is credited to all points which lie on the line. This method has been successfully used to reconstruct 3D images.

Prior to this thesis the second method, the Radon transform method, had not been applied to positron emission data. Some researchers have mentioned the method, but not

done any actual work with it. It has been applied to MRI by Shepp [1980] and his work was used extensively as a basis against which this research was tested. This method differs from the above method in that the data is collected into 2D projections (planes) rather than 1D projections. Each plane is defined by (θ,ϕ,p), where (θ,ϕ) define a line through the origin and p is a point on that line. The plane is at point p, orthogonal to the line (θ,ϕ). 2D projections of a 3D object form the Radon transform of the 3D object. Each 2D projection represents the integral of the radioactivity in the object over a plane. The collected data is in the form of a 1D set of values associated with a stack of parallel planes (θ,ϕ constant, p varying) through the object varying over all directions (θ,ϕ) (Note that this is not just the directions from which the tomograph collects data, but all directions), as is diagrammed in figure 2.9. The projection direction is defined by the line (θ,ϕ) through the origin and the value of each plane is a point (p) on the line. The data is filtered along the line (1D), then backprojected. Since each value is associated with a plane rather than a line, the value is backprojected over that plane. All points in the image which lie on the plane are credited with the value of the plane.

The advantage of using Radon's method is expected to be in the fact that the filtering is done over one dimension. Filtering in object space involves convolution, which is very time consuming and reducing this should reduce the reconstruction time considerably. Another possible advantage is statistical in nature. With the Radon transform, all event lines occurring in a plane are summed to that plane, regardless of direction (there is no loss of information since any individual event gets credited to many planes). The 1D projections method splits these events into separate projection directions, which removes any smoothing effects obtained by summing the events.

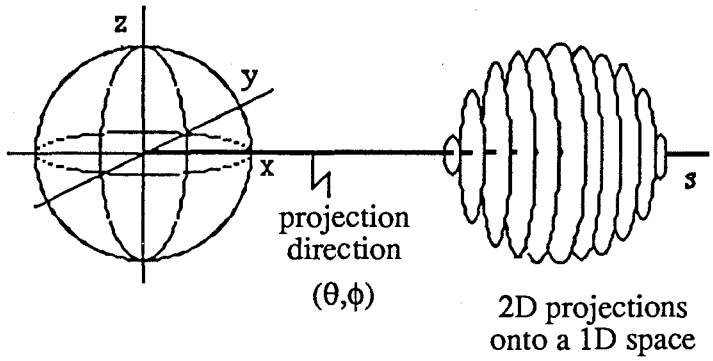


Figure 2.9 : 2D projections onto a 1D space.

One projection direction (θ=90, ϕ=0; varying s), that is, one 'stack' of Radon planes.

The error introduced by backprojection is the same as in 2D reconstruction, as is the solution. The effects of insufficient projection directions are also the same. A complete discussion of the theory of 3D reconstruction follows in the next chapter along with a proof of the equivalence of the two methods just described.

2.4. Applications of PET and PVI

The final image produced by PET is of interest in medical research and diagnosis because it is one of the few methods available for relatively non-invasive viewing of a living organ. Where modalities such as MRI, ultrasound and CAT show physical structure, and EEG and MEG show electrical activity, PET and SPECT can show metabolism. One particular application is the study of Alzheimer's disease [McGeer,

Kamo, Harrop, *et al* 1986], a condition where part of the brain stops functioning. Diagnosing this disease can be difficult since the symptoms can look like the symptoms of other problems such as depression, or other diseases such as Pick's disease. This study uses a FDG (flouro-deoxyglucose) tracer. FDG is basically a radioactive form of glucose. When it is injected into the body, the tracer goes to wherever glucose is needed. Since glucose is the sole form of nutrition for brain cells, some of it will go to the brain and be caught up in brain metabolism At this point the difference between the tracer and normal glucose becomes apparent, the tracer does not undergo complete glucose metabolism, it passes from the bloodstream into the cell and remains there without getting used by the cell for energy. Approximately 40 minutes after the FDG has been injected, most glucose has been taken up by the cells and is relatively stationary. An image of this tracer distribution shows which parts of the brain are active (and are therefore using glucose) and which parts aren't. Since the areas of the brain effected by Alzheimer's are different than those effected by Pick's, the PET image is capable of differentiating the two diseases [Kamo, *et al* 1987].

3. Theory of 3D Image Reconstruction

The two methods of 3D image reconstruction which were described in the previous chapter can be mathematically explained using Fourier transforms [Brooks and Di Chiro 1976]; let f(x,y,x) be the original distribution of radioactivity with some choice of origin O and rectangular Cartesian coordinate axes Ox, Oy, Oz. Let F(s,t,u) be the 3D Fourier transform of f, then by the definition of Fourier transform,

$$F(s,t,u) = \iiint_{-\infty}^{\infty} f(x,y,z)\, e^{-2\pi i(xs+yt+zu)}\, dx\, dy\, dz \ .$$

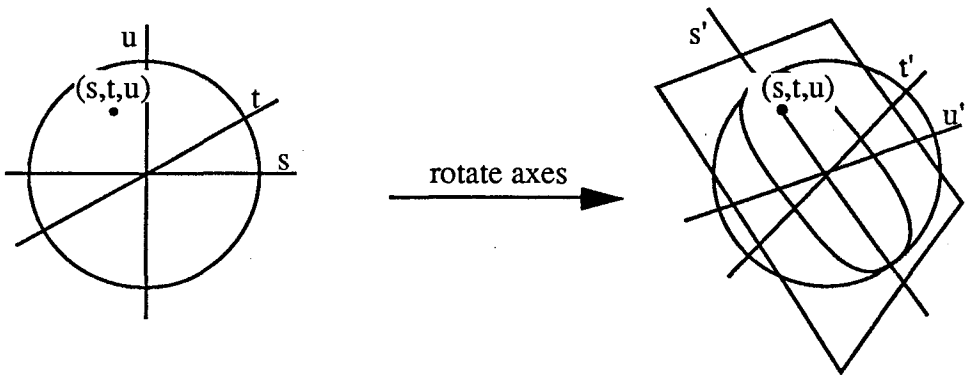
The origin of Fourier space is taken at O and the s,t,u axes are in the direction of the x,y,z axes respectively. Note that the Fourier transform is independent of the directions of the axes of the 3D space and is only dependent on the origin, that is, with fixed origin, it is a function of position in space, independent of choice of the axes.

The function f is supposed to be a function of position, that is, determined at each point in space with a value not dependent on the rectangular coordinate system chosen to specify the point. F(s,t,u), the Fourier transform of f, will be a function of position (with fixed origin) if it can be shown that, given (x,y,z), (s,t,u), the value of $f(x,y,z)e^{-2\pi i(xs+yt+zu)}$ depends only on the positions (in the object and superimposed Fourier space) of (x,y,z), (s,t,u) and the origin (0,0,0). The required result is true since xs+yt+zu is the scalar product of (x,y,z) with (s,t,u) and depends only on the magnitudes of

the vectors (x,y,z) and (s,t,u) (the directed lines from the origin to (x,y,z) and (s,t,u), respectively) and the cosine of the angle between these two lines. The lengths of the lines and the cosine of the angle between them only depend of the positions of the points (x,y,z), (s,t,u) and the origin, and not on the coordinate system.

Since this is the case, given any point (s,t,u) in Fourier space, axes (x',y',z'), (s',t',u') can be chosen such that this point lies in the plane u'=0 (there are many possible choices of axes, one of which is shown in figure 3.1). If f' is the object function and F' is its Fourier transform referred to the new axes, then, for corresponding points (x,y,z), (x',y',z') and (s,t,u), (s',t',u'), f(x,y,z) = f'(x',y',z') and F(s,t,u) = F'(s',t',u'). Thus, for the given point, the following is true:

$$F(s,t,u) = F'(s',t',0) = \iiint_{-\infty}^{\infty} f'(x',y',z')\, e^{-2\pi i(x's'+y't'+0)}\, dx'dy'dz'$$



F(s,t,u) - the 3D Fourier transform of f(x,y,z)

F(s,t,u) = F'(s',t',u')

Figure 3.1 : 1D projection onto a u' = 0 plane.

It will be noticed that since u'=0 the exponent is no longer dependent on z'. Hence, for all points for which u'=0,

$$F(s,t,u) = \iint \left(\int f'(x',y',z')dz'\right) e^{-2\pi i(x's'+y't')}\, dx'dy' \ .$$

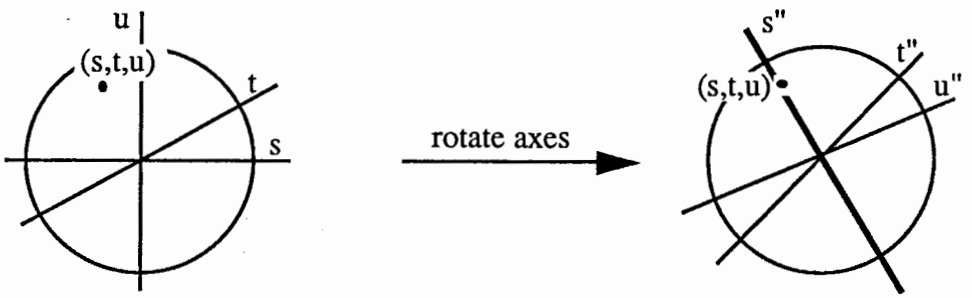
The term $\int f'(x',y',z')dz'$ is one 1D projection of the function f'(x',y',z') onto the x'-y' plane. Recall from the previous chapter that this is one possible form of the measured data. This means that the value of F(s,t,u) can be found by performing a 2D Fourier transform with respect to x', y' on the 1D projection data $(\int f'(x',y',z')dz')$. Thus, given any projection direction, the value of F(s,t,u) can be determined at all points on the plane through the origin perpendicular to that direction. If the projection directions define 1D projections which cover the entire object, then the 2D Fourier transform of all the projection planes will completely specify F(s,t,u). This is just the 'Fourier slice theorem'. Given the complete specification of F(s,t,u) for all (s,t,u), an inverse Fourier transform will give the original function f(x,y,z), which was the 3D distribution of radioactivity in the object. In order for the projection planes (planes perpendicular to the sets of parallel 1D projections) to cover the entire object, a complete set of projection directions over a half great circle around the object is required [Orlov 1975(a)]. Since the detectors form a ring around the object, the requirement is satisfied.

Suppose, in the above example, given the point (s,t,u), axes (x",y",z"), (s",t",u") are chosen so that the point lies on the line u"=0, t"=0 (figure 3.2). Then, with notation as above, for any point (s,t,u) on the line u"=t"=0 through the origin,

$$F(s,t,u) = F''(s'',0,0) = \iiint_{-\infty}^{\infty} f''(x'',y'',z'')\, e^{-2\pi i(x''s''+0+0)}\, dx''dy''dz''$$

that is, $$F(s,t,u) = \int \left(\iint f''(x'',y'',z'')dy''dz''\right) e^{-2\pi i(x''s'')}\, dx'' \ .$$

The term $\iint f''(x'',y'',z'')dy''dz''$ is a 2D projection of the function f"(x",y",z") onto the x" axis at the point (x",0,0). This corresponds to the other possible format for assembling the measured data (the Radon transform of the distribution of radioactivity). The equation shows that the Fourier transform along the x" axis is obtainable by applying a 1D Fourier transform to the set of 2D projections (constant θ,ϕ, varying p) at points along the line θ,ϕ. If all lines through the origin are considered, F(s,t,u) is known for all s,t,u, and the original function can be found by applying the inverse Fourier transform. To specify the Fourier transform at all points, the projection directions must cover a complete hemisphere [Orlov 1975(a)] (the implementation of this requirement is considered in chapter 4).



F(s,t,u) - the 3D Fourier transform of f(x,y,z)

F(s,t,u) = F"(s",t",u")

Figure 3.2 : 2D projection onto the t" = u" = 0 line.

In most actual implementations of the above two reconstruction methods, the required set of Fourier transforms on the projections, and the 3D inverse transform, are not actually calculated. Filtering and backprojection are equivalent, as the following shows [Shepp 1980], and are used instead.

Using the definition of inverse Fourier transform,

$$f(x,y,z) = \iiint_{-\infty}^{\infty} F(s,t,u)\, e^{2\pi i(xs+yt+zu)}\, dx\, dy\, dz \qquad (A)$$

let $f(x,y,z) \equiv g(\theta,\phi,p)$ and $F(s,t,u) \equiv G(\lambda,\mu,\rho)$ where (x,y,z) is a point in object space using Cartesian coordinates, (s,t,u) is the same point in Fourier space, also in Cartesian coordinates, and (θ,ϕ,p), (λ,μ,ρ) are the same point in polar coordinates in object and Fourier space respectively. Then

$$s = \rho \sin\lambda \cos\mu, \quad t = \rho \sin\lambda \sin\mu \quad \text{and} \quad u = \rho \cos\lambda .$$

Equation A becomes

$$f(x,y,z) = \int_{\mu=0}^{\pi} \int_{\lambda=0}^{\pi} \int_{-\infty}^{\infty} G(\lambda,\mu,\rho) e^{2\pi i(x\rho \sin\lambda \cos\mu + y\rho \sin\lambda \sin\mu + z\rho \cos\lambda)} \rho^2 \sin\lambda\, d\rho\, d\lambda\, d\mu \,. \quad (B)$$

Let $r_{\lambda,\mu}(p)$ be the Radon transform for the line through the origin for given values of λ and μ and let $R_{\lambda,\mu}(\rho)$ be the 1D Fourier transform for the line through the origin for given values of λ,μ. Then,

$$R_{\lambda,\mu}(\rho) = G(\lambda,\mu,\rho) \text{ all } \rho.$$

Hence B becomes

$$f(x,y,z) = \int_{\mu=0}^{\pi} \int_{\lambda=0}^{\pi} [\int_{-\infty}^{\infty} R_{\lambda,\mu}(\rho)\, e^{2\pi i\rho(x \sin\lambda \cos\mu + y \sin\lambda \sin\mu + z\cos\lambda)}\, \rho^2 d\rho] \sin\lambda\, d\lambda\, d\mu \quad . \text{(C)}$$

From our definition of $r_{\lambda,\mu}(p)$, $\quad r_{\lambda,\mu}(p) = \int_{-\infty}^{\infty} R_{\lambda,\mu}(\rho)\, e^{2\pi i\rho p}\, d\rho$

$$\frac{d^2}{dp^2} r_{\lambda,\mu}(p) = \int_{-\infty}^{\infty} (2\pi i\rho)^2\, R_{\lambda,\mu}(\rho)\, e^{2\pi i\rho p}\, d\rho$$

$$= -4\pi^2 \int_{-\infty}^{\infty} \rho^2\, R_{\lambda,\mu}(\rho)\, e^{2\pi i\rho p}\, d\rho$$

$$\therefore \quad \frac{1}{-4\pi^2} \frac{d^2}{dp^2} r_{\lambda,\mu}(p) = \int_{-\infty}^{\infty} \rho^2\, R_{\lambda,\mu}(\rho)\, e^{2\pi i\rho p}\, d\rho \quad .$$

For given x, y, z,λ, μ let

$$p = x \sin\lambda \cos\mu + y \sin\lambda \sin\mu + z \cos\lambda \ .$$

Then the bracketed integral in C equals

$$-\frac{1}{4\pi^2}\frac{d^2 r_{\lambda,\mu}(p)}{dp^2} \ .$$

Thus $$f(x,y,z) = \frac{-1}{4\pi^2}\int_{\mu=0}^{\pi}\int_{\lambda=0}^{\pi}\frac{d^2 r_{\lambda,\mu}(p)}{dp^2}\sin\lambda \, d\lambda \, d\mu \ .$$

The partial second derivative is the filter for the measured data and the double integral is the backprojection operation. The computer implementation of the above equation is discussed in section 4.2.

The theoretical work shows what may be one of the main advantages of using Radon transforms, namely that the 1D Fourier transform of all the projection lines is required instead of a 2D Fourier transform of all the projection planes. When put into the filtering and backprojection form, the 2D Fourier transform becomes a 2D convolution, which involves considerably more operations than the 1D convolution which arises from the 1D Fourier transform.

On the other hand, and this might be considered a disadvantage of the Radon method, PVI does not measure a 2D projection directly. It does not, of course, exactly measure 1D projections either, but the data is almost in that form and it requires much less handling to get the data into 1D projections than into 2D projections. The exact method used to obtain the Radon transform is discussed in the next chapter along with the actual implementation used to do the filtering and backprojection.

4. Implementation

In chapter 3 it was shown that, in theory, Radon transforms are a valid method for PVI image reconstruction. The next step in this feasibility study is to do a computer implementation of the theory in a discretized form, and show that it can be used to produce usable images. To implement and test the theory, it will be assumed that the data is collected with a complete sphere of detectors. This arrangement is necessary because the theory assumes that all projection directions of the Radon transform can be weighted equally in image reconstruction. What this means is that the values of the 2D projections over all directions should depend only on the amount of radioactivity in the projection and not on any aspect of the detector arrangement. If a cylindrical arrangement of detectors were used, this requirement would not be easily satisfied [Orlov 1975(b)] and some non-trivial corrections would have to be made during reconstruction. Although any final form of a reconstruction method will have to work on data collected by some form of a detector cylinder (the only practical arrangement), it was considered appropriate for this study to just test the discretized form of the theory, without correction for detector arrangement, and decide from these results if continued work on the Radon method is justified. Since the data used to test the reconstruction algorithm is generated by software, there is no difficulty in making the tests with a spherical detector arrangement.

Some consideration has been given to possible solutions for the problem of implementing Radon reconstruction with the detector ring, and it is felt that methods the same as, or similar to, those used to correct this problem with 1D projections [Kinahan and Rogers 1989] could be used on 2D projections.

When testing the discrete form of the Radon theory, the first implementation problem is in getting the Radon transform from the PVI data. Recall from chapter 2 that each event is specified by the line on which it occurred. These event lines have to be put into 2D projections. The second problem is, obviously enough, the reconstruction of the image. The following two sections present the detailed solutions to the problems of getting the Radon transform and of performing the filtering and backprojection operations.

4.1. Getting the Radon Transform

The Radon transform must be defined for projection directions over an entire hemisphere. The theoretical reasons for this were presented in chapter 3, the practical implication is that the 2D projections along a line must be known for all lines through a 3D object. In this work the Radon transform is defined as $r(\theta,\phi,p)$ where $0° \leq \theta,\phi < 180°$ and $-20 \leq p \leq 20$ (where the distance is measured in cm). This defines the transform over the entire area inside a 20 cm radius sphere of detectors, a size which is appropriate for a head tomograph. Each θ,ϕ defines a line through the origin and p is the directed distance from the origin (in cm) of a point along that line. The value of $r(\theta,\phi,p)$ is the integral of all radioactivity over the plane defined by that point.

In order to obtain a representation of the Radon transform from real PVI data, each event line must be credited to all members of a set of predefined planes in which it can be considered to lie. Each of these planes is considered to be one point in the Radon transform. When all the event lines have been credited to the set of planes, each plane will have a value which represents the integral of the radioactivity in the plane.

The first decision which had to be made in this section, was what the set of pre-defined planes would look like. It was decided to vary θ over 180° and ϕ over 180°, defining the hemisphere of projection directions, and to sample each projection from -20 to 20 cm. This covers the entire 3D area over which the Radon transform is non-zero (it is assumed that the object fits inside the 20 cm radius sphere of detectors). Testing was done with the angular sampling at every 5° or every 2° (0:175 by 5 or 0:178 by 2) and with projection sampling at every 5 mm or 2 mm.

The next decision to be made was how to credit event lines to the Radon planes, that is, how to decide when a line was to be considered close enough to a plane to be considered to lie in it (recall that all planes have thickness). It is desirable to maintain data consistency along a stack of planes (along one projection direction). This means that a point source of radioactivity should appear equally bright no matter where it is located in a stack of planes. Essentially this is the same problem as was mentioned in chapter 2, the partial volume effect present in multi-slice tomographs. Since the set of Radon planes is not defined by the hardware (unlike the slices imaged with a 2D tomograph) it is possible to choose a method of crediting events to planes which will avoid this.

If the criterion for crediting an event line to a plane were that the line must entirely lie within the plane, then the partial volume effect would be a problem just as with 2D tomographs, since this is how events are credited to slices in a 2D tomograph. To avoid its effect, it was decided to credit a plane with any event line provided it was such that its midpoint was close enough to the centre of the plane, and the angle (α) between the event line and the plane was less than some maximum. For this work the distance of the centre of the line from the centre of the plane is at most 1 mm and α is at most 0.1°, where the

planes were defined as being 2 mm apart, so any event line which has α within 0.1° of any plane in some stack will be within 1 mm of one and only one plane in that stack. The following inequality was used to test for α ,

$$\left| \frac{(x_2-x_1)\sin q \cos\phi}{r} + \frac{(y_2-y_1)\sin\theta\sin\phi}{r} + \frac{(z_2-z_1)\cos\theta}{r} \right| \leq \cos(90-\alpha)$$

where $(x_1,y_1,z_1)(x_2,y_2,z_2)$ are the endpoints of the event line and r is equal to the length of the event line ($r = [(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2]^{1/2}$). This inequality uses the scalar product of the direction cosines of the event line and of the line (θ,ϕ) (normal of the plane) to find the cosine of the angle between them. If the angle is between $90-\alpha$ and $90+\alpha$, then the event line is considered to lie in a plane with the normal (θ,ϕ).

Since the decision about which planes get credited with which event lines is only dependent on the 'nearness' of an event line to a plane, an event line may get credited to a group of adjacent planes. Because the criterion used to test if a line belongs in a plane is 'nearness', the result depends only on the relative positions of the line and the plane. It should be observed that an event is <u>not</u> credited to some fixed number of planes. If that were the case, then the number of events credited to any one plane would be dependent on the number of planes nearby, which would result in inconsistent data sampling. This is due to the fact that it is not possible to uniformly tessellate a solid sphere. We are simulating a

sampling of Radon space, so it is important to ensure that the value obtained for the sample at a point is independent of the presence or absence of other planes in the sampling set.

The most obvious method of finding all the planes that an event line should be credited to, is to take the event line and check it against the (θ,ϕ) of all stacks of parallel planes. If the direction of line is within α of the plane through the origin, then it will be within 1 mm of some plane in that stack. Clearly, this method would work, but it is very time consuming. A method was found where, given a line and ϕ for a stack of planes, it was possible to find directly all values of θ for which planes might be credited by the line. Although this did help speed up the running time of the program considerably, getting the Radon transform was still a time consuming process. At this point in the feasibility study it was considered inappropriate to spend a lot of time looking for an optimal method to generate the Radon transform once a method was found which was reasonably satisfactory. Nevertheless, some effort was put into the design of a faster method. A lookup table was generated so that, given an event line, it was possible to look up all the planes in which that line lay. Such a table need only be generated once. It can then be used on all data sets. To keep the table down to a manageable size, the directions of all event lines were rounded to the nearest degree in θ and ϕ, and, to reduce it still further, the algorithm made use of the following two facts;

1 If an event line is rotated round the z axis through an angle which is an integer number of degrees, then the θ's and p's of the Radon planes associated with the line will remain the same; the ϕ's will vary by the amount of rotation.

2 If an event line is translated through some distance without change of direction, then the θ's and ϕ's of the Radon planes will remain the same and the p's will vary by some amount less than or equal to the amount of translation.

It follows from these facts that the θ of an event line is all that is necessary to find the stack of Radon planes appropriate for any event lines; given θ of an event line, look up the Radon planes for the event (θ,0,0) and then rotate the planes by ϕ of the event line and directly calculate p. This method gives considerable improvement in computation time over directly finding all planes for every event line, but the rounding of the event line directions to the nearest degree has an effect on the number of events credited to planes.

This problem is caused by the uneven tessellation of the sphere. Radon planes which have normals with θ near 0° or 180° (called the polar planes) are more crowded, so the area over which events are binned to a polar plane is much smaller than the area from which events are binned to an equatorial plane (θ of the normal near 90°). The effect is that fewer events get credited to a polar plane than to an equatorial plane even when there is actually the same amount of radiation in both planes. The number of event lines credited to the planes varies as a function of sin θ and α (the maximum permitted angle between the event line and a containing plane). It would be both necessary and possible to correct for this if the table look up method should be implemented. Since it seems likely that hardware would be used to implement some, if not all, of the Radon reconstruction work, no further work has been done in pursuing the relative advantages and disadvantages of these

two methods, now that they have been shown to be feasible.

4.2. Filtering

Once the data has been put into the Radon transform, the next step is to filter it. The data could be backprojected before it is filtered (using a different filter), with the same results, but it is easier to implement the filter on the transform rather than on the image.

The theoretical form for reconstruction from the Radon transform, as derived in chapter 3 is

$$f(x,y,z) = \frac{-1}{4\pi^2}\int_0^{\pi}\int_0^{\pi} r''(\theta,\phi,p)\sin\theta \; d\theta \, d\phi \; .$$

The simplest way to implement the inverse transform on real data is to sample the object at discrete intervals and apply the 2^{nd} difference filter (approximation to 2^{nd} derivative). This is the approach used by Shepp [1980]. His discrete formula is

$$\bar{f}(x,y,z) = \frac{-1}{4mn}\sum_{0}^{n-1}\sum_{0}^{m-1}(2r(\theta,\phi,p) - r(\theta,\phi,p-1) - r(\theta,\phi,p+1))\sin\theta$$

where n and m are the number of projection directions in ϕ and θ respectively. The 2^{nd}

derivative filter only accounts for the errors introduced by backprojection since the theoretical form of the transform uses infinitely many projection directions and infinitely many samples along each direction, that is, continuous projection directions and continuous sampling along each direction. When the formula is discretized on real data, the filter must be changed to handle the new errors caused by finite sampling. These changes include the introduction of a cutoff window to remove high frequency noise caused by inadequate sampling and a raised cosine function to smooth the cutoff window and remove statistical noise. The 2^{nd} difference filter doesn't do either of these operations and Shepp got poor images. He concluded that his reconstructions could not be used in a clinical setting and, since this thesis is concerned with the possibility of eventually using Radon transforms in clinical settings, Shepp's work has been repeated to verify his results and to compare them with images obtained using the filter being suggested.

The filter designed here for use with Radon transforms [Kinahan 1988] was originally developed in Fourier space since the components of the filter are easier to understand in that context. The first part of the filter is a cutoff window. This removes all frequencies above the Nyquist frequency , which was determined as follows; the function is considered as sampled every 1 unit of length, therefore the shortest wavelength that is correctly sample is 2 units, and the highest frequency correctly measured is 1/2 per unit. This is the Nyquist frequency (f_n). It is possible to remove more frequencies by using a smaller cutoff which would take out more statistical noise and give a nicer looking image, but would degrade the resolution . For the testing in this work, f_n was taken as 0.5 per unit length. The second part of the filter is a raised cosine function. This smooths the data, removing statistical variations (provided that the statistical noise is within some reasonable limit) and provides a smooth approach to the cutoff window.

The exact form of the filter is derived as follows [Brooks and Di Chiro 1976, Chesler and Riederer 1975]. Given the reconstruction formula:

$$f(x,y,z) = \frac{-1}{4\pi^2}\int_0^{\pi}\int_0^{\pi} r''(\theta,\phi,p)\sin\theta\ d\theta\ d\phi$$

where f(x,y,z) is the original function; r(θ,ϕ,p) is the known Radon transform of f(x,y,z) and r"(θ,ϕ,p) is the (partial) second derivative of r(θ,ϕ,p) with respect to p, we try to determine a 1D convolution filter h such that r" = r*h (where ' * ' denotes convolution) that is, such that

$$f(x,y,z) = \frac{-1}{4\pi^2}\int_0^{\pi}\int_0^{\pi} (r(\theta,\phi,p)*h(p))\sin\theta\ d\theta\ d\phi\ .$$

From the definition of Fourier transform and differentiation under the integral sign, it can be shown that, for any integer $n \geq 0$,

$$\frac{d^n x(p)}{dp^n} = F^{-1}[(2\pi i f)^n X(f)]$$

where F denotes 1D Fourier transform and $X(f) = F[x(p)]$.

In particular, $x''(p) = F^{-1}[-4\pi^2 f^2 X(f)]$. Replacing x"(p) with r"(θ,ϕ,p) gives

$$r''(\theta,\phi,p) = F^{-1}[-4\pi^2 f^2 F(r(\theta,\phi,p)]$$

$$= F^{-1}[-4\pi^2 f^2] * F^{-1}[F[r(\theta,\phi,p)]]$$

$$= F^{-1}[-4\pi^2 f^2] * r(\theta,\phi,p)$$

where $[-4\pi^2 f^2]$ is the Fourier transform of h(p). In this form, the need for the cutoff window can clearly be seen. For large values of f (that is, the high frequencies which are known to be incorrect), the value of $-4\pi^2 f^2$ will be large. This will amplify the high frequency noise and introduce error into the reconstruction. Adding a cutoff window at the Nyquist frequency suppresses this noise amplification. However, the sharp edge introduced by the window function in frequency space, will bring a ringing effect into the data in object space [Barney 1988, Chesler and Riederer 1975], unless the window is made to approach zero smoothly.

The smoothed version of r''(p) is given by $F^{-1}[-4\pi^2 f^2 W(f)] * r(\theta,\phi,p)$. The window function (W(f)) chosen for this work is the Hamming window, which has a cutoff at the Nyquist frequency (f_n) and uses a raised cosine function to smooth the cutoff. It is given by

$$W(f) = \frac{1}{2}[\,1 + \cos(\frac{\pi f}{f_n})]\,\mathrm{rect}(\frac{f}{2f_n}) \quad , \quad \text{where} \quad \mathrm{rect}\,(x) = \begin{cases} 1 \text{ if } |x| \le 1/2 \\ 0 \text{ otherwise} \end{cases} .$$

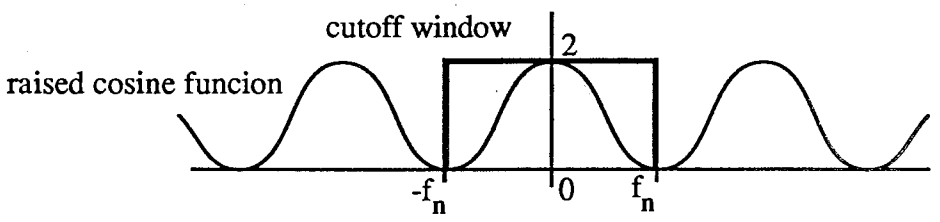


FIgure 4.1: the Hamming window

So the 1D Radon convolution filter (called the Hamming filter for the remainder of this paper) in frequency space is:

$$H(f) = -2\pi^2 f^2\,[1 + \cos(\frac{\pi f}{f_n})]\,\mathrm{rect}\,(\frac{f}{2f_n}) \quad .$$

It was decided to reconstruct the data in object space, so this filter (which is in Fourier space) must be inverse transformed. The final form is

$$h(p) = F^{-1}[H(f)] = -2\pi^2\,(k(p) + \frac{1}{2}k(p - \frac{1}{2f_n}) + \frac{1}{2}k(p + \frac{1}{2f_n}))$$

where $$k(p) = \frac{f_n}{\pi^2 p}\cos(2\pi f_n p) + \frac{2\pi^2 f_n{}^2 p^2 - 1}{2\pi^3 p^3}\sin(2\pi f_n p) \quad .$$

The details of the inverse transform is given in Appendix A, and the method of calculation of k(0) is given in Appendix B.

This filter is convolved with the 2D projection in each projection direction and the results are backprojected over a 100x100x100 image space.

4.3. Backprojection

The data for each point in the Radon transform is collected from a plane through the object and therefore must be backprojected back over that plane. This is accomplished by crediting each voxel in the image with a contribution from each stack of Radon planes. This method is called voxel-driven backprojection. To avoid interpolating between planes for every voxel, the values along a projection direction are interpolated before they are backprojected. Once backprojection is done the reconstruction process is complete.

5. Testing

The main work in testing the Radon transform method is divided into two sections. The first section is the testing of the algorithm which places the PVI data into the Radon planes and the second is the testing of the reconstruction filters. The primary consideration during testing was to ascertain the correctness of the algorithms developed, not to show that they were the most efficient possible. It would be premature to do any fine tuning before the feasibility of Radon reconstruction in clinical work has been shown.

5.1. Getting the Radon Transform

To test this portion of the work, simulation data was acquired from a Monte Carlo simulation program (PHANTOM). This program simulates the detection of events from some radioactive object by a tomograph with some arrangement of detectors. It is possible to define any detector configuration and any phantom (the radioactive object) over a wide range of possibilities. Although the program is capable of simulating scatter and attenuation, these were not necessary for this thesis. In order to test the Radon reconstruction method properly, only those problems which the reconstruction method is expected to handle should exist in the data, namely statistical noise and inadequate sampling.

PHANTOM was used to produce event lines for a centred, 10 cm radius sphere of uniform radioactivity inside a 20 cm radius sphere of detectors. Three million events were collected, all of which were 'perfect' (a 'perfect' event is one which did not scatter or

attenuate). The molecule which emitted the positron was exactly on the event line, and the detectors recorded the event perfectly. These events were run through two programs; DIRECT which directly finds all planes in which each event line lies, and TABLE which uses a look up table to find which planes are credited with an event. In both programs 36x36x81 Radon planes were used; 36x36 projection directions (0:175 by 5° in θ and ϕ) and 81 samples along a projection (-20:20 by 0.5 cm in p). The Radon transform of the uniformly radioactive sphere looks like a parabola along each projection direction (the areas of the intersections of a stack of parallel planes with a sphere, plotted against the position of the intersection of the plane with the corresponding direction line, is a parabolic function). Only three million events were used for this test, so the results had some statistical variation, but both programs produced a parabola along each projection direction. The results of the two programs were not identical, due to the sin θ and α factors from the extra binning in TABLE where each event is rounded to the nearest degree in θ and ϕ in order to look up the Radon planes in the table.

The result of DIRECT was further tested by running the reconstruction program on it; the reconstruction showed a centred sphere of uniform activity (see figures 5.1&5.2) as desired.

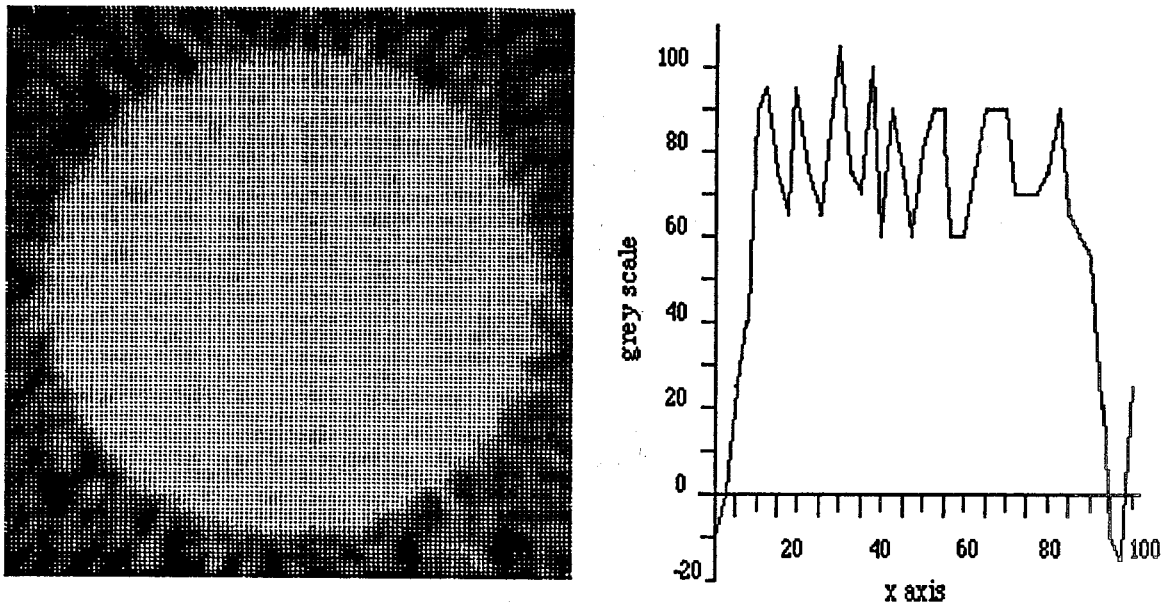


Figure 5.1: Reconstruction of a centred sphere of uniform radioactivity with 3 million events using the 2nd difference filter. The slice is taken at the z=0 plane; the line is taken along the x axis.
Due to the requirements of the image display routines, all reconstructed images were scaled to a maximum value of 125, so the plots of lines through the images show only relative, not absolute, values. Also, the plotted values were rounded to the nearest 5.

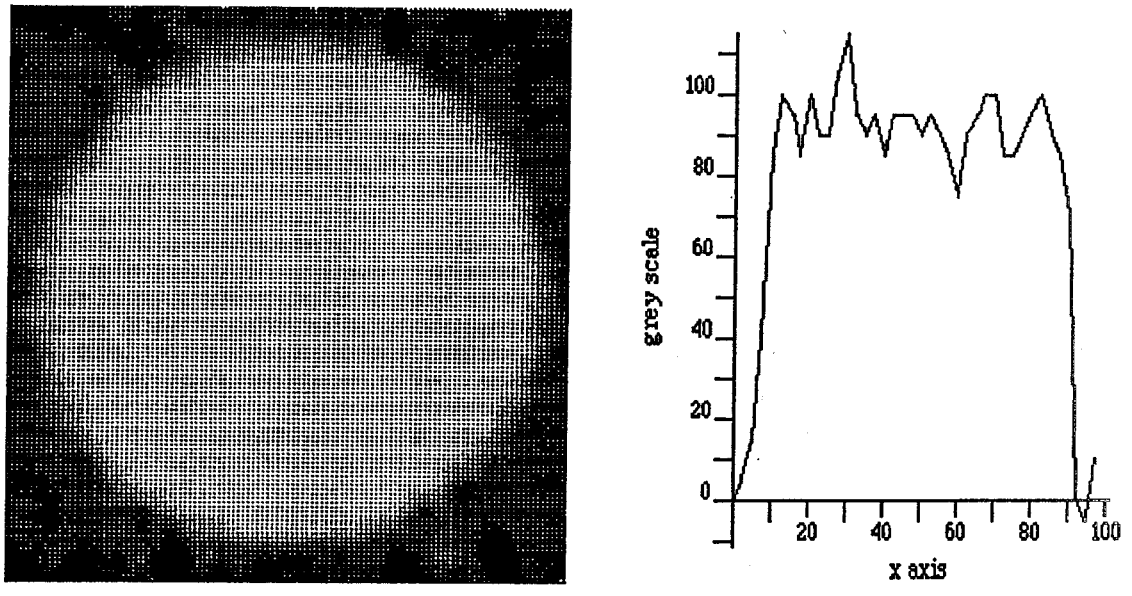


Figure 5.2: Reconstruction of the centred sphere with 3 million events using the Hamming filter. The slice is taken at the z=0 plane; the line is taken along the x axis

5.2. Filtered Backprojection

The spherical shape of the phantom is particularly suitable for testing the effects of statistical noise. Because the Radon transform along any projection direction is a parabola, and the 2^{nd} difference of a parabolic function is the same as the 2^{nd} derivative, and is constant (which means the difference filter is exactly correct for this function), the reconstruction of the sphere using the 2^{nd} difference filter is not affected by inadequate sampling. Since the PHANTOM data did not contain scatter or attenuation, the reconstructed images will only show the effects of the smoothing function (raised cosine) of the Hamming filter. Three million events are not sufficient to get a good image of any object much more complicated than a simple geometric shape, like a sphere, but the image does show very clearly statistical noise which is what is required for this test. Figure 5.1 shows a reconstruction with the 2^{nd} difference filter. There is considerable noise in the image (differentiation and the 2^{nd} difference function actually enhance noise), but the general shape is clear. The Hamming filter derived in the last chapter was used to reconstruct figure 5.2. The image is much smoother.

To test the sampling problem in the reconstruction, a program, PERFECT was written which generates perfect Radon transforms for any arrangement of generalized ellipsoids, which represent solid areas of uniform activity. The term 'perfect Radon transform' means that the program calculates the exact areas of intersection of the predefined Radon planes with the ellipsoids. The value of the Radon plane is exact, but the transform is sampled at a finite number of points. This means the data has no statistical noise and no attenuation or scatter. The only problem which could cause artifacts in the

final image is the inadequate sampling of the 3D distribution of radioactivity (the arrangement of ellipsoids). The exact formula for the calculation of area of intersection of an ellipsoid with a plane was taken directly from Shepp [1980] after its correctness had been verified.

The PERFECT program was first used to calculate the Radon transform for a single, centred ellipsoid (semiaxes a = 4 cm, b = 2 cm and c = 3 cm), with the axes aligned with the system co-ordinate axes. Two different sets of predefined planes were used as the sampling points, one had 36x36x201 points (0:175 by 5° in θ and ϕ; -20:20 by 0.2 cm in p), the other had 90x90x201 points (0:178 by 2° in θ and ϕ; same in p). Each of these two sets were reconstructed twice, once with the 2^{nd} difference filter and once with the Hamming filter. The images of these reconstructions, shown in figures 5.3 to 5.6, are from a slice through the centre (z = 0), the line used in the plot is the y axis.

The first reconstructed image (figure 5.3) shows clearly, even with this single ellipsoid, the reconstruction artifacts that Shepp found with his multiple ellipsoid phantom. The effect of using the cutoff at the Nyquist frequency and some smoothing effects are shown in figure 5.4. The star-like effects are still very visible but the noise around the ellipsoid has been removed. In particular the two very low spots directly above and below the ellipsoid are gone. Figure 5.5 shows the effect of using more projection directions with the 2^{nd} difference filter, the star shaped artifacts are almost entirely gone, but the spectral leakage artifacts can still be seen. There are no leakage artifacts in the figure 5.6, reconstructed with the Hamming filter and the 90x90 projection directions, further the star artifacts are much reduced. The artifacts which are still present in figure 5.6 would likely be acceptable for clinical applications and if better reconstructions are required, they can

be obtained by using a still finer sampling of the function.

The lack of a correct discretized form of the filter in Shepp's MRI work generated a lot of artifacts in his reconstructions. Since the amplitudes of the artifacts are related to the density of the object which caused them, the high density objects in Shepp's phantom caused artifacts with high values throughout his images. In clinical settings these large errors could not be tolerated because they cannot be distinguished from real information. Using a corrected filter, the Hamming filter, it would appear that we are able to remove these errors.

In order to demonstrate further that the Hamming filter can take out all serious imaging artifacts it is necessary to reconstruct a more complicated phantom. To do this, a reduced version of Shepp's head phantom was used, namely the following (lengths in mm).

Head Part	Ellipsoid centre	Semiaxes Length and Orientation	Grey Scale
1 skull	(0,0,0)	(72,96,13) (1,0,0)(0,1,0)(0,0,1)	2.0
2. nose	(0,114,-20)	(13,34,17) (1,0,0)(0,55,-84)(0,84,55)	1.5
3. left tumor	(-8,-61,38)	(5,2,2) (1,0,0)(0,1,0)(0,0,1)	0.01
4. centre tumor	(0,-61,38)	(2,2,5) (1,0,0)(0,1,0)(0,0,1)	0.01
5. right tumor	(6,-61,38)	(2,5,2) (1,0,0)(0,1,0)(0,0,1)	0.01
6. left ventricle	(-22,0,38)	(16,41,38) (-95,-31,0)(-31,95,0)(0,0,1)	-0.02
7. right ventricle	(22,0,38)	(11,31,25) (95,-31,0)(31,95,0)(0,0,1)	-0.02

The grey scale values correspond to hydrogen density values in Shepp's work and to

levels of radioactivity in ours. Where the ellipsoids overlap, the grey scale values are summed. These seven ellipsoids were considered enough to test the filter properly without complicating the issue too much.

Two reconstructions were done on this data, one with 36x36x201 and the 2^{nd} difference filter (figure 5.7), and the other with 90x90x201 and the Hamming filter (figure 5.8). Both figures show the reconstruction through the plane at $z = 3.8$ cm. The ventricles and tumors are in this plane, but the nose is not. Figure 5.7 shows how the effect of the nose makes itself felt throughout the entire object, the ventricles can be seen but the artifacts are as big or bigger than the real objects. The effects of the nose have been almost completely eliminated in figure 5.8, but the tumors are still obscured. The loss of the tumors is due to the lack of contrast between them and their background (0.3%). The data must be scaled in order to display it and these small differences are lost. If an extremely accurate image is displayed, the tumors do show up. However, in clinical work, the regions of interest tend to be larger, and have greater contrast, so extremely accurate images are unnecessary.

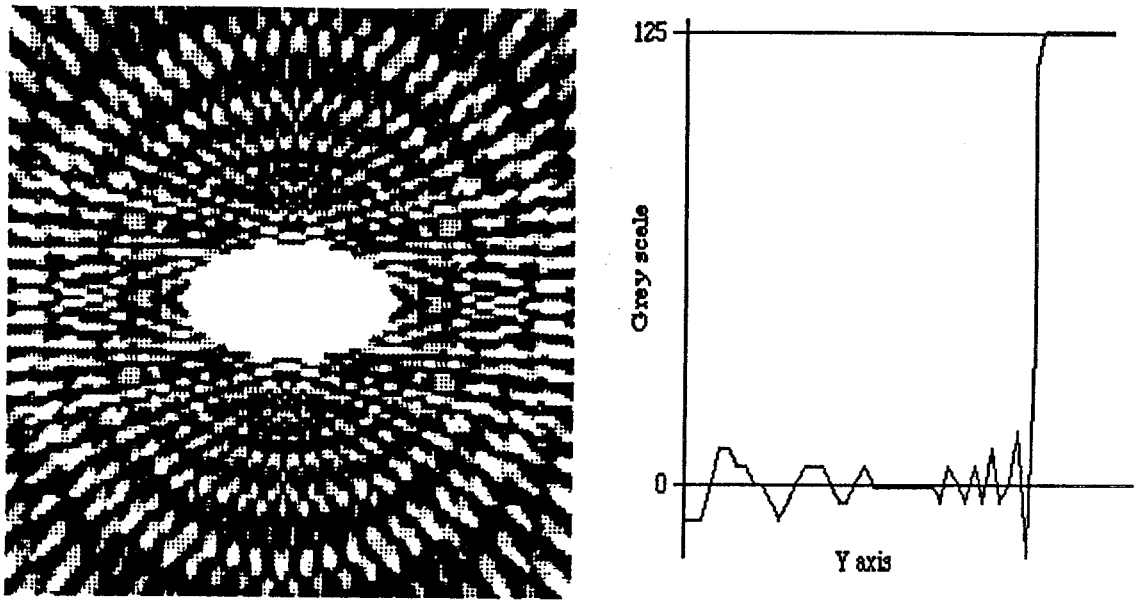


Figure 5.3 : Single ellipsoid, 36x36 with 2nd difference filter.
The reconstructed slice is taken on the x-y plane, the line is the first half of the values along the Y axis (since the image is symmetric, no information is missing).

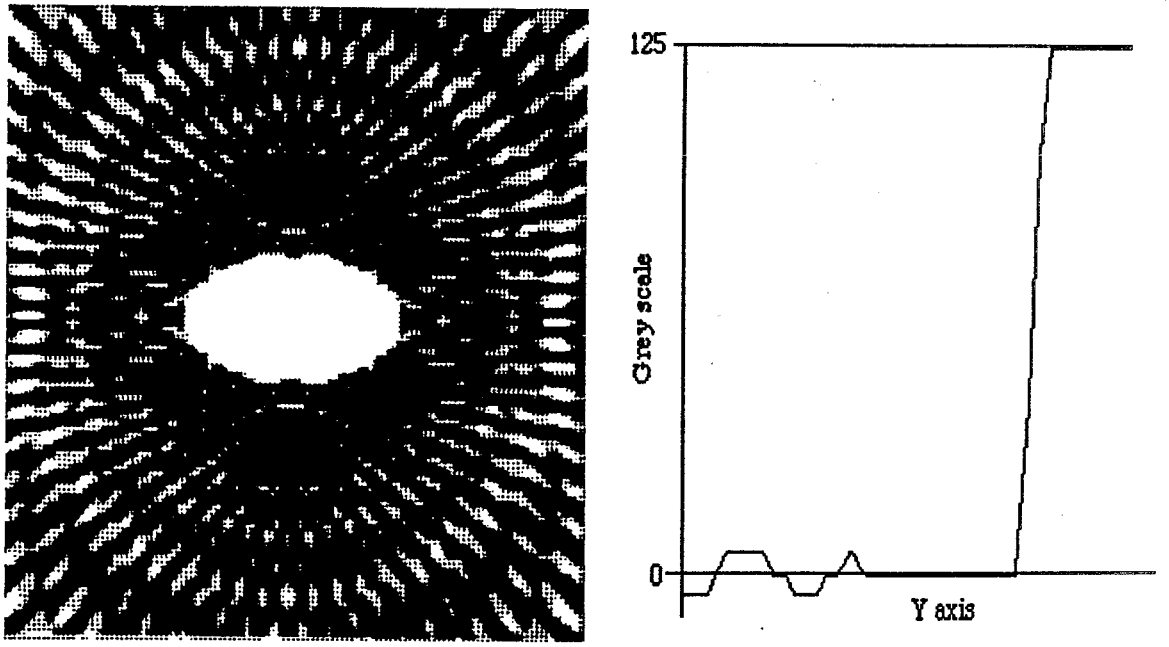


Figure 5.4 : Single ellipsoid, 36x36 with the Hamming filter.
The plane and slice are the same as in figure 5.3. Note that, although there is still significant noise at the edges, the Hamming filter has removed much of it from around the ellipsoid.
The slices shown of all single ellipsoid reconstructions have been displayed with just 3 grey scale values to bring out the artifacts; negative values are black, zero values are grey and positive values are white.

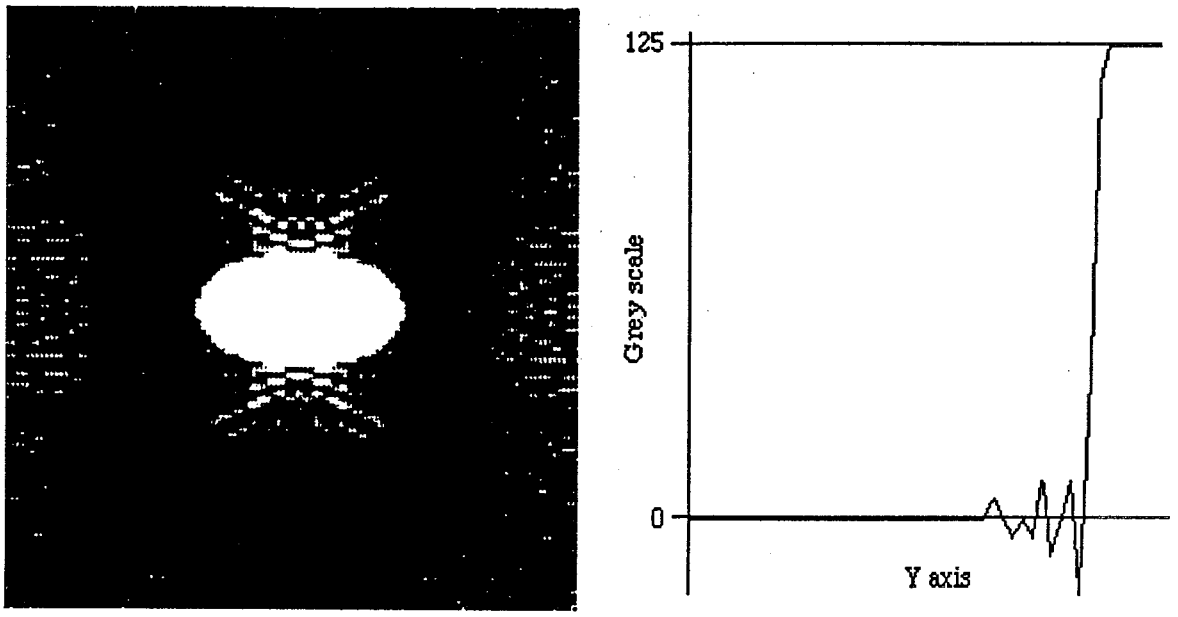


Figure 5.5: Single ellipsoid, reconstructed with 90x90x201 sampling points and the 2nd difference filter.
The outside of the ellipsoid is much smoother, due to the increased sampling, but the area near its edge is noisy.

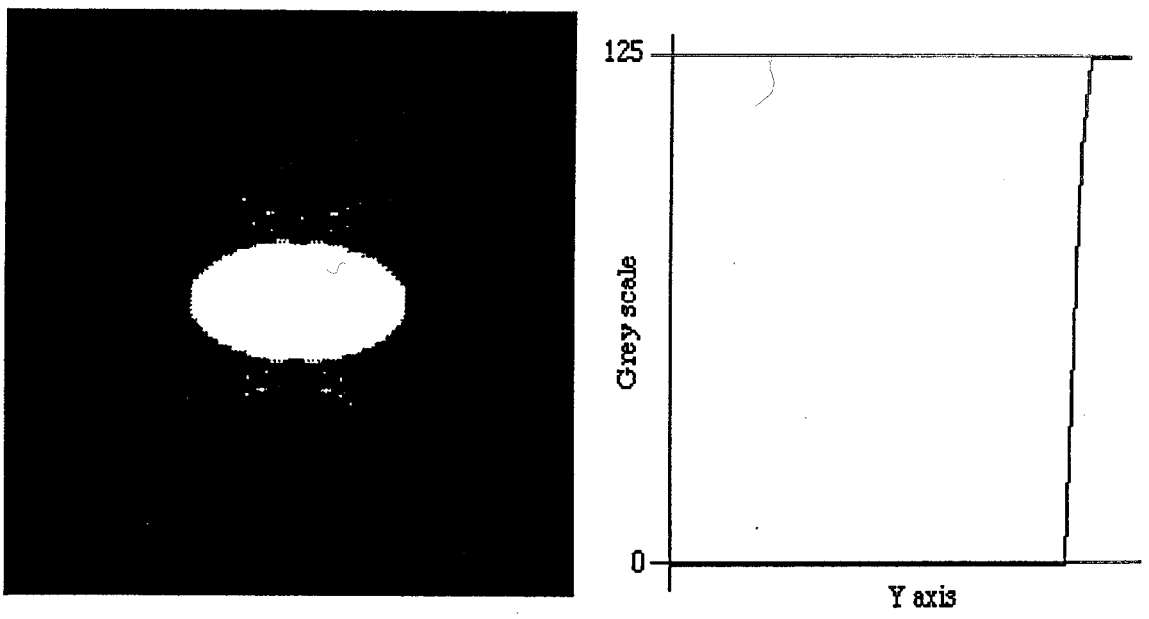


Figure 5.6: Reconstruction of the single ellipsoid with 90x90x201 sampling points and the Hamming filter.

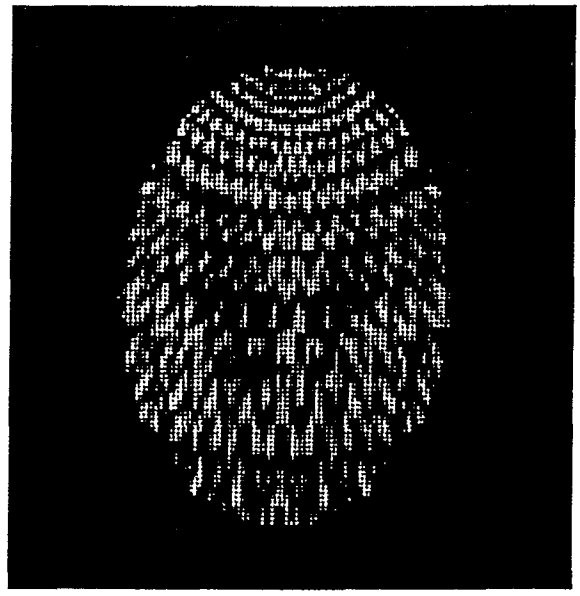


Figure 5.7: Reconstruction of 7 ellipsoids from Shepp's head phantom using 36x36x201 sampling points. The reconstructed plane is at z = 3.8 cm, which contains the ventricles and the tumors, but not the nose.
Almost all of the artifacts in this image are caused by the nose and its large grey scale value.

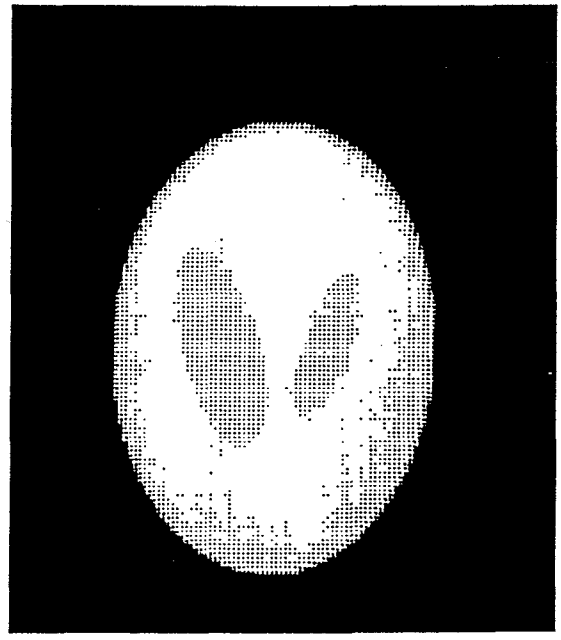


Figure 5.8: 7 ellipsoids from Shepp's head phantom, reconstructed with 90x90x201 sampling points, the reconstructed plane is at z = 3.8 cm.
Both the reconstructions of 7 ellipsoids were displayed with a shifted grey scale which emphasized small differences at the top 10% of the image values.

6. Discussion

The aim of this thesis was to test the feasibility of using Radon transforms for fully 3D PVI image reconstruction. The concepts of image reconstruction have been discussed and the underlying theory presented. The bulk of the work has been first, to make sure that PVI data can be converted satisfactorily into Radon transform form and second, to use a method of reconstruction based on the transform to reconstruct an image which is good enough to be used in a clinical setting. Both of these targets have been satisfactorily obtained for a special case of detector arrangement (spherical detectors).

A solution to the first problem was pursued until it was determined that a method did exist which would produce a satisfactory Radon transform. In fact, two methods were found which may be used to convert the data. Continued work in fine tuning these methods is reserved until the extension to this work is done and a method is developed for reconstruction from data collected by cylindrical detectors. There is little use in putting effort toward perfecting any particular portion of the Radon reconstruction work until the whole method is known to be feasible.

A full and complete solution to the second problem has been defined, encoded, tested and presented in the thesis. The findings of this work is that, at this stage it would appear to be feasible to use Radon transforms for fully 3D PVI reconstruction. There could also be an application of this technique, with the Hamming filter, in MRI[5].

5. MRI is capable of measuring the Radon transform directly and need not translate the measured data before reconstruction.

Due to the spherical detector arrangement, no final statement can be made as to whether or not this method could be used in a real tomograph in a clinical setting. In order to determine this, it will be necessary to do the next step, which is to attempt to solve the problem of imaging with a cylindrical arrangement of detectors. Some thought has been given to this step and it is considered that the solution used in 1D projection reconstruction will also be applicable to the 2D projection method. The method is fully described in Kinahan and Rogers [1989]. The basic idea is to do two reconstructions, the first pass uses only the 1D projections which are not affected by the detector arrangement, that is, only those events which are perpendicular to the axial direction of the detector surface. A reconstruction is performed on this data and a low statistics image is produced, this information is then used with all detected events to do a second, high statistics image. This method would also offer some advantages for scatter and attenuation correction, since the low statistics image could be used to find where scatter and attenuation would have occurred and use this to help to correct all the data for the second pass reconstruction. This is particularly important because the potential of PVI imaging depends heavily on the success of the scatter correction technique. It is realized that, to use this method with 2D projections there will be a translation in dimension and there would be some theoretical difficulty with using the restricted data set for the first pass reconstruction. An extension to the thesis, to include the cylindrical detector arrangement, would be far from trivial, but, due to the good results obtained so far, it seems that it would be well worthwhile to pursue such an extended feasibility study.

BIBLIOGRAPHY

J.S.Barney, **The fast Fourier transform**, student paper, Simon Fraser University, 1988.

R.A.Brooks and G.Di Chiro, **Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging**, Phys. Med. Biol., 1976, 21(5), pp 689-732.

D.A.Chesler and S.J.Riederer, **Ripple suppression during reconstruction in transverse tomography**, Phys. Med. Biol, 1975, 20(4), pp 632-636.

S.R.Deans, The Radon transform and some of its applications, John Wiley & Sons, Toronto, 1983.

B.Evans, R.Harrop, D.Haywood, J.Mackintosh, R.W.Moore, B.D.Pate, J.G.Rogers, T.J.Ruth, C.Sayre, H.Sprenger, N.Van Oers and Y.X.Guang, **Engineering developments in the UBC-TRIUMF modified PET VI positron emission tomograph**, IEEE Trans. on Nuclear Science, 1983, NS 30, pp 707-710.

A.C.Kak and M.Slaney, **Principles of computerized tomographic imaging**, IEEE Press, 1988, pp 177-201.

H.Kamo, P.L.McGeer, R.Harrop, E.G.McGeer, D.B.Calne, W.R.W.Martin and B.D.Pate, **Positron emission tomography and histopathology in Pick's disease**, Neurology, 1987, 37, pp 439-445.

P.E.Kinahan, **Radon 1D convolution filtering**, private communication, 1988.

P.E.Kinahan, R.Harrop, J.G.Rogers and R.R.Johnson, **Three-dimensional reconstruction in object space**, IEEE Trans. on Nuclear Science. 1988, 35, pp 635-638.

P.E.Kinahan and J.G.Rogers, **Analytic 3D image reconstruction using all detected events**, IEEE Trans. on Nuclear Science, 1989, 36(1), pp 964-068.

P.L.McGeer, H.Kamo, R.Harrop, D.B.K.Li, H.Tuokko, E.G.McGeer, M.J.Adam, W.Ammann, B.L.Beattie, D.B.Calne, W.R.W.Martin, B.D.Pate, J.G.Rogers, T.J.Ruth, C.I.Sayre and A.J.Stoessl, **Positron emission tomography inpatients with clinically diagnosed Alzheimer's disease**, Can. Med. Assoc. J., 1986, 134, pp 597-606.

S.S.Orlov, **Theory of three dimensional reconstruction. I. Conditions for a complete set of projections**, Sov. Phys. Crystallogr., 1975(a), 20(3), pp 312-314.

S.S.Orlov, **Theory of three dimensional reconstruction. II. The recovery operator**, Sov. Phys. Crystallogr., 1975(b), 20(4), pp 429-433.

J.Radon, **Ueber die bestimmung von funktionen durch ihre integralwerte längs gewisser manningfaltigkeiten**, Berichte Sächsische Akademie der Wissenschaften, Leipzig, Math-Phys. Kl., 1917, 69, pp 262-267.

J.G.Rogers, R.Harrop, G.H.Coombes, N.A.Wilkinson, M.S.Atkins, B.D.Pate, K.S.Morrison, M.Stazyk, C.J.Dykstra, J.S.Barney, P.W.Doherty and D.P.Saylor, **Design of a volume imaging positron emission tomograph**, IEEE Trans. on Nuclear Science, 1989, 36(1), pp 993-997.

L.A.Shepp, **Computerized tomography and nuclear magnetic resonance**, J. of Computer Assisted Tomography, 1980, 4(1), pp 94-107.

APPENDIX A

Inverse Fourier transform of the Hamming filter

From page 37, h is given by

$$h(p) = F^{-1}[H(f)] = F^{-1}\left[-2\pi^2 f^2 \left[1 + \cos\left(\frac{pf}{f_n}\right)\right] \text{rect}\left(\frac{f}{2f_n}\right)\right]$$

$$= -2\pi^2 \left(F^{-1}\left[f^2 \text{rect}\left(\frac{f}{2f_n}\right)\right] * F^{-1}\left[1 + \cos\left(\frac{pf}{f_n}\right)\right]\right) .$$

Now

$$F^{-1}\left[1 + \cos\left(\frac{pf}{f_n}\right)\right] = \delta(p) + \frac{1}{2}\left[\delta\left(p - \frac{1}{2f_n}\right) + \delta\left(p + \frac{1}{2f_n}\right)\right]$$

and

$$F^{-1}\left[f^2 \text{rect}\left(\frac{f}{2f_n}\right)\right] = \int_{-f_n}^{f_n} f^2 e^{2\pi i f p}\, df \triangleq k(p) .$$

Then, using $e^{i\theta} = \cos\theta + i \sin\theta$

$$k(p) = \int_{-f_n}^{f_n} f^2 (\cos(2\pi fp) + i \sin(2\pi fp)) \, df = 2 \int_0^{f_n} f^2 \cos(2\pi fp) \, df$$

(the $i \sin(2\pi fp)$ term integrates to 0)

$$= 2 \left[\frac{2f \cos(2\pi fp)}{(2\pi p)^2} + \frac{(2\pi fp)^2 - 2}{(2\pi p)^3} \sin(2\pi fp) \right]_0^{f_n}$$

that is,

$$k(p) = \frac{f_n}{\pi^2 p^2} \cos(2\pi f_n p) + \left[\frac{2\pi^2 f_n{}^2 p^2 - 1}{2\pi^3 p^3} \right] \sin(2\pi f_n p)$$

and h is given by

$$h(p) = -2\pi^2 \left(k(p) + \tfrac{1}{2} k\left(p - \tfrac{1}{2f_n}\right) + \tfrac{1}{2} k\left(p + \tfrac{1}{2f_n}\right) \right) .$$

APPENDIX B

Calculation of k(0)

From Appendix A, k is given by

$$k(p) = \frac{f_n}{\pi^2 p^2} \cos(2\pi f_n p) + \left[\frac{2\pi^2 f_n^{\,2} p^2 - 1}{2\pi^3 p^3}\right] \sin(2\pi f_n p) \ .$$

Using the expansion formula for the sine and cosine functions, that is,

$$\sin x = x - x^3/3! + x^5/5! - x^7/7! - \ldots$$

$$\cos x = 1 - x^2/2! + x^4/4! - x^6/6! + \ldots \quad ,$$

this gives

$$k(p) = \frac{f_n}{\pi^2 p^2}\left[1 - \frac{4\pi^2 f_n^2 p^2}{2!} + \frac{16\pi^4 f_n^4 p^4}{4!} - \ldots\right] + \frac{2\pi^2 f_n^2 p^2 - 1}{2\pi^3 p^3}\left[2\pi f_n p - \frac{8\pi^3 f_n^3 p^3}{3!} + \ldots\right]$$

$$= \frac{f_n}{\pi^2 p^2} - \frac{4\pi^2 f_n^3 p^2}{\pi^2 p^2 2!} + \frac{16\pi^4 f_n^5 p^4}{\pi^2 p^2 4!} - \ldots + \frac{4\pi^3 f_n^3 p^3}{2\pi^3 p^3} - \frac{2\pi f_n p}{2\pi^3 p^3} - \frac{16\pi^5 f_n^5 p^5}{2\pi^3 p^3 3!} + \frac{8\pi^3 f_n^3 p^3}{2\pi^3 p^3 3!} + \ldots$$

Ignoring terms with p^2 or higher powers of p

$$k(p) = \frac{f_n}{\pi^2 p^2} - 2f_n^3 + \ldots + 2f_n^3 - \frac{f_n}{\pi^2 p^2} + \frac{2}{3} f_n^3 + \ldots$$

$$= 2/3\, f_n^3 + (\text{terms with } p^2 \text{ and higher order})$$

and thus $k(0) = 2/3\, f_n^3$.

It is also possible to calculate k(0) using L'Hôpital's rule.

APPENDIX C

Source code

```
      PROGRAM GENTAB

C     Generates a lookup table for a limited number of event lines.
C     The events put into the table are those that have PHI=0 and
C     pass through the origin.  The table is accessed by THETA of the
C     event line and contains the angle of the planes in which the
C     event line lies.  The planes are specified by the normal to
C     the plane.

      INTEGER*4 eTHETA, rTHE, rPHI, TABLE(0:179,0:220), I, COUNT
      REAL*8    eX, eZ, ROOT, ALPHA, mALPHA, TEMP, FPEFF, rTHETA, PSI
      LOGICAL*1 DONE
      CHARACTER*6 cALPHA

C     eTHETA  - theta of the event line
C     eX      - X dimension of the event line
C     eZ      - Z dimension of the event line
C     rTHETA  - theta of the Radon plane  REAL*4
C     rTHE    - theta of the Radon plane  INTEGER*4
C     rPHI    - phi of the Radon plane

      WRITE (*,5) ' alpha: '
      READ (*,5) cALPHA
5     FORMAT (A$)
      READ (cALPHA,*) ALPHA           ! change to real number
      WRITE (*,6) ALPHA
6     FORMAT (1X,F6.3)

      mALPHA = COSD(90 - ALPHA)
      DO eTHETA=0,179                 ! loop through event lines
        eX = SIND(FLOAT(eTHETA))
        eZ = COSD(FLOAT(eTHETA))
        I = 0
        DONE = .FALSE.
        DO rPHI=0,179                 ! loop through all Radon PHI's
          ROOT = SQRT((eX*COSD(FLOAT(rPHI)))**2 + eZ**2)
          IF (ROOT .EQ. 0) THEN                                  ! math from p.77
            rTHETA = 90                                          ! of lab book.
          ELSE                                                   ! simplified using
            rTHETA = ACOSD((eX*COSD(FLOAT(rPHI)))/ROOT)          ! ePHI=0 and unit
            IF (eTHETA .LT. 90) rTHETA = 180 - rTHETA            ! vectors.
          END IF
          PSI = -(MOD(rTHETA, DBLE(180)))
          rTHE = MOD(NINT(rTHETA) + 1, 180)
          TEMP = ROOT * SIND(rTHE + PSI)
          COUNT = 0
C       Test that (1) - the event line is within alpha of the plane
C                 (2) - no more than 180 planes have been found for any 1 event
C                 (3) - only 1 plane (1 phi) for the theta=0 plane is used
          DO WHILE (ABS(TEMP) .LE. mALPHA .AND.
     1              COUNT .LE. 179)                  ! (1) .AND. (2)
            IF ((.NOT. DONE) .AND. rTHE .EQ. 0) THEN  ! (3)
              DONE = .TRUE.
              I = I + 1
              TABLE(eTHETA,I) = rPHI*1000 + rTHE
            ELSE IF (rTHE .NE. 0) THEN
              I = I + 1
              TABLE(eTHETA,I) = rPHI*1000 + rTHE
            END IF
            rTHE = MOD(rTHE + 1, 180)
            TEMP = ROOT * SIND(rTHE + PSI)
            COUNT = COUNT + 1
          END DO
          IF (COUNT .LE. 179) THEN          ! only do 2nd loop if
```

```
            rTHE = MOD(NINT(-PSI), 180)        ! the 1st didn't go to 180
            TEMP = ROOT * SIND(rTHE + PSI)
            DO WHILE (ABS(TEMP) .LE. mALPHA .AND. COUNT .LE. 179) ! (1) .and. (2)
              IF ((.NOT. DONE) .AND. rTHE .EQ. 0) THEN            ! (3)
                DONE = .TRUE.
                I = I + 1
                TABLE(eTHETA,I) = rPHI*1000 + rTHE
              ELSE IF (rTHE .NE. 0) THEN
                I = I + 1
                TABLE(eTHETA,I) = rPHI*1000 + rTHE
              END IF
              rTHE = rTHE - 1
              IF (rTHE .EQ. -1) rTHE = 179
              TEMP = ROOT * SIND(rTHE + PSI)
              COUNT = COUNT + 1
            END DO
          END IF
        END DO                  ! Radon PHI values
        TABLE(eTHETA,0) = I     ! # planes for this event
      WRITE (*,7) 'eTHETA: ', eTHETA, '# planes = ', I
7     FORMAT (1X,A8,I3,4X,A11,I)
      END DO                    ! event lines

      OPEN(UNIT=1,FILE='LOOKUP.TBL',STATUS='NEW')
      WRITE (1,10) ((TABLE(eTHETA,I),I=0,220),eTHETA=0,179)
10    FORMAT (10(1X,I6.6)/)
      CLOSE (UNIT=1)

      END
```

```
      PROGRAM TBLE

C     Takes individual events in the format (X1, Y1, Z1, X2, Y2, Z2) finds
C     the planes in which the line lies and increments them.  The PHI and
C     THETA values of the Radon planes are found in a lookup table, the P
C     values are calculated for each plane.

      REAL*4    ePHI, eTHETA, x1, y1, z1, x2, y2, z2,
     1          midX, midY, midZ, eX, eY, eZ, EVENT(20)
      INTEGER*2 CHAN, IOSB(4)
      INTEGER*4 I, J, K, TABLE(0:180,0:220), eTHE, TEMP, nEVENT,
     1          RADON(-100:100,0:89,0:89), NUMPL, rTHE, rPHI, P,
     2          SYS$ASSIGN, STATUS, C
      LOGICAL   FIRSTCALL
      CHARACTER OUTFILE*15

C     EVENT           - holds the event parameters
C     CHAN            - I/O channel for the tape drive
C     IOSB(*)         - I/O status block (called INIOSB(,) in EVENT)
C     FIRSTCALL       - if the tape has been read yet
C     rTHE            - theta of the Radon plane
C     rPHI            - phi of the Radon plane
C     P               - distance of the plane from the origin
C     x,y,z(1);x,y,z(2) - endpoints of the event line
C     e(x,y,z)        - directionof the event line
C     eTHETA          - theta of the event line  REAL*4
C     eTHE            - theta of the event line  INT*4
C     ePHI            - phi of the event line
C     midX, midY, midZ - midpoint of the event line
C
C     do all arithmetic operations in REAL to avoid truncation problems

      OPEN (UNIT=1,FILE='LOOKUP.TBL',STATUS='OLD',READONLY)
      READ (1,10) ((TABLE(eTHE,I), I=0,220), eTHE=0,179)
10    FORMAT (10(1X,I6)/)
      CLOSE (UNIT=1)

      WRITE (*,15) ' number of events: '
15    FORMAT (A$)
      READ (*,*) nEVENT

      WRITE (*,15) ' output file name: '
      READ (*,15) OUTFILE
      C = INDEX(OUTFILE,'.')          ! if the file does not have an extension
      IF (C .EQ. 0) THEN              ! then default to .PLN.
        C = INDEX(OUTFILE,' ')
        OUTFILE(C:C+3) = '.PLN'
      END IF

      STATUS = SYS$ASSIGN('MUA0:',CHAN,,)     ! set up tape drive
      IF (.NOT. STATUS) THEN                  ! check for error
        WRITE (*,*) ' problem with tape drive '
        CALL LIB$STOP(%VAL(STATUS))
      END IF
      FIRSTCALL = .TRUE.              ! tape hasn't yet been read

      C = 0
100   CALL GET_EVENT(EVENT,FIRSTCALL,CHAN,IOSB)       ! read 1 event from tape
        IF (.NOT. IOSB(1)) THEN       ! test for EOT
          WRITE (*,*) ' end of tape, # events used = ', C
          GOTO 200
        END IF
        IF ( MOD(C,100) .EQ. 0) WRITE (*,*) ' event # ', C
        IF (C .EQ. nEVENT) GOTO 200           ! cut the data set short
```

```
          C = C+1
          x1 = EVENT(1)
          y1 = EVENT(2)
          z1 = EVENT(3)
          x2 = EVENT(4)
          y2 = EVENT(5)
          z2 = EVENT(6)
          midX = ((x1+x2) /2) *10                ! find the midpoint of the line
          midY = ((y1+y2) /2) *10                ! and convert to mm.
          midZ = ((z1+z2) /2) *10
          eX = x1*10 - midX                      ! find the direction of the line
          eY = y1*10 - midY
          eZ = z1*10 - midZ
C        convert to cartesian co-ordinates
          eTHETA = ACOSD(eZ/SQRT(eX**2 + eY**2 + eZ**2))
          IF (eTHETA .EQ. 0) THEN
            ePHI = 0
          ELSE
            ePHI = ACOSD(eX/SQRT(eX**2 + eY**2))
          END IF
          IF (eZ .LT. 0) THEN
            eTHETA = 180 - eTHETA
            ePHI = 180 - ePHI
          END IF
          eTHE = MOD(NINT(eTHETA), 180) ! 0 <= THETA <= 179
          ePHI = MOD(ePHI,180.0)        ! 0 <=  PHI  <= 179
          NUMPL = TABLE(eTHE,0)         ! get the number of planes for this THETA.
          DO I=1,NUMPL
            TEMP = TABLE(eTHE,I)        ! get PHI and THETA of each plane.
            rPHI = TEMP/1000
            rTHE = TEMP - rPHI*1000
            rPHI = rPHI + NINT(ePHI)    ! rotate for correct phi value, note that
                                        ! PHI is binned to 1 degree here.
            IF (rPHI .GE. 180) THEN     ! use the part of the line in the
              rPHI = rPHI - 180         ! +ive Y hemishere.
              rTHE = 180 - rTHE
            END IF
            P = NINT(midX*SIND(REAL(rTHE))*COSD(REAL(rPHI)) +
     1             midY*SIND(REAL(rTHE))*SIND(REAL(rPHI)) +
     2             midZ*COSD(REAL(rTHE)))
            IF (MOD(P,2) .NE. 0) P = P + 1      ! round to nearest 2mm
            RADON(P/2,MOD(NINT(REAL(rTHE)/2), 90),
     1              MOD(NINT(REAL(rPHI)/2), 90)) =
     2        RADON(P/2,MOD(NINT(REAL(rTHE)/2), 90),
     3                  MOD(NINT(REAL(rPHI)/2), 90)) + 1
          END DO
          GOTO 100                  ! get next event

200       CLOSE (UNIT=1)  ! used all events

          OPEN (UNIT=1,FILE=OUTFILE,STATUS='NEW',RECL=1206)
          WRITE (1,210) (((RADON(I,J,K), I=-100,100), J=0,89), K=0,89)
210       FORMAT (101(1X,I5))
          CLOSE (UNIT=1)  ! written all planes

          END
```

```
      PROGRAM DIRECT

C     Takes individual events from tape in the format
C     (X1, Y1, Z1, X2, Y2, Z2), finds the planes in which
C     the line lies (within some angle alpha) and increments them.
C     Prints out the full set of Radon planes, ready for filtering and
C     backprojection.

      REAL*8    ALPHA, mALPHA,
     1          x1, y1, z1, x2, y2, z2, LENGTH,
     2          eX, eY, eZ, ROOT, TEMP, rTHETA, PSI,
     3          midX, midY, midZ, sinP, cosP
      REAL*4    EVENT(20)
      INTEGER*4 I, J, K, COUNT, C, MAX, P, SYS$ASSIGN, STATUS,
     1          RADON(-40:40,0:35,0:35), rTHE, rPHI
      INTEGER*2 CHAN, IOSB(4)
      LOGICAL   FIRSTCALL
      CHARACTER INFILE*15, cALPHA*6, OUTFILE*15

C     EVENT   - holds the event parameters
C     x,y,z(1) x,y,z(2) - 2 endpoints of the event line
C     LENGTH  - length of the event line
C     STATUS  - return status of a system call
C     CHAN    - I/O channel for the tape drive
C     IOSB    - I/O status block (called INIOSB(,) in EVENT)
C     FIRSTCALL - if the tape has been read yet
C     eX      - X dimension of the event line
C     eY      - Y dimension of the event line
C     eZ      - Z dimension of the event line
C     rTHETA  - theta of the Radon plane  REAL*4
C     rTHE    - theta of the Radon plane  INTEGER*4
C     rPHI    - phi of the Radon plane
C     midX, midY, midZ - midpoint of the event line
C     MAX     - maximum number of events to process
C     P       - distance of the Radon plane from the origin along the normal

      WRITE (*,5) ' alpha: '
      READ (*,5) cALPHA
5     FORMAT (A$)
      READ (cALPHA,*) ALPHA           ! change to real number
      mALPHA = COSD(90 - ALPHA)

      WRITE (*,5) ' number of events: '
      READ (*,*) MAX

      WRITE (*,5) ' output file name: '
      READ (*,5) OUTFILE
      C = INDEX(OUTFILE,'.')          ! if the file does not have an extension
      IF (C .EQ. 0) THEN              ! then default to .PLN.
        C = INDEX(OUTFILE,' ')
        OUTFILE(C:C+3) = '.PLN'
      END IF

      STATUS = SYS$ASSIGN('MUA0:',CHAN,,)     ! set up tape drive
      IF (.NOT. STATUS) THEN                  ! check for error
        WRITE (*,*) ' problem with tape drive '
        CALL LIB$STOP(%VAL(STATUS))
      END IF
      FIRSTCALL = .TRUE.              ! tape hasn't yet been read

      C = 0
100   CALL GET_EVENT(EVENT,FIRSTCALL,CHAN,IOSB)       ! read 1 event from tape
        IF (.NOT. IOSB(1)) THEN       ! test for EOT
          WRITE (*,*) ' end of tape, # events used = ', C
```

```
          GOTO 200
        END IF
        IF ( MOD(C,100) .EQ. 0) WRITE (*,*) ' event # ', C
        IF (C .EQ. MAX) GOTO 200                ! cut the data set short
        C = C+1
        x1 = EVENT(1)
        y1 = EVENT(2)
        z1 = EVENT(3)
        x2 = EVENT(4)
        y2 = EVENT(5)
        z2 = EVENT(6)
        midX = ((x1+x2) /2) *10                 ! find the midpoint of the line
        midY = ((y1+y2) /2) *10                 ! and convert to mm.
        midZ = ((z1+z2) /2) *10
        eX = x1*10 - midX                       ! find the direction of the line
        eY = y1*10 - midY
        eZ = z1*10 - midZ
        LENGTH = SQRT(eX**2 + eY**2 + eZ**2)    ! use unit vectors
        eX = eX/LENGTH
        eY = eY/LENGTH
        eZ = eZ/LENGTH
        I = 0
        DO rPHI=0,179                   ! loop through all Radon PHI's
          sinP = SIND(REAL(rPHI))
          cosP = COSD(REAL(rPHI))
          ROOT = SQRT((eX*cosP + eY*sinP)**2 + eZ**2)
          IF (ROOT .EQ. 0) THEN                                  ! math from p.77
            rTHETA = 90                                          ! of lab book.
          ELSE                                                   ! simplified using
            rTHETA = ACOSD((eX*cosP + eY*sinP)/ROOT)             ! unit vectors.
            IF (eZ .GE. 0) rTHETA = 180 - rTHETA
          END IF
          PSI = -(MOD(rTHETA, DBLE(180)))
          rTHE = MOD(INT(rTHETA) + 1, 180)
          TEMP = ROOT * SIND(rTHE + PSI)
          COUNT = 0

C       Test that (1) - the event line is within angle alpha of the plane.
C                 (2) - no more than 180 planes have been found for any 1 event.

          DO WHILE (ABS(TEMP) .LE. mALPHA .AND.
     1              COUNT .LE. 179)                  ! (1) .AND. (2)
            I = I + 1

C       calculate P
            P = NINT(midX*SIND(REAL(rTHE))*cosP +
     1               midY*SIND(REAL(rTHE))*sinP +
     2               midZ*COSD(REAL(rTHE))) ! -200:200mm

C       increment the Radon plane
            RADON(NINT(REAL(P)/5),
     1            MOD(NINT(REAL(rTHE)/5), 36),
     2            MOD(NINT(REAL(rPHI)/5), 36)) =
     3        RADON(NINT(REAL(P)/5),
     4              MOD(NINT(REAL(rTHE)/5), 36),
     5              MOD(NINT(REAL(rPHI)/5), 36)) + 1
            rTHE = MOD(rTHE + 1, 180)
            TEMP = ROOT * SIND(rTHE + PSI)
            COUNT = COUNT + 1
          END DO
          IF (COUNT .LE. 179) THEN              ! only do 2nd loop if
            rTHE = MOD(INT(-PSI), 180)          ! the 1st didn't go to 180
            TEMP = ROOT * SIND(rTHE + PSI)
            DO WHILE (ABS(TEMP) .LE. mALPHA .AND. COUNT .LE. 179)
```

```
                I = I + 1

C       calculate P
                P = NINT(midX*SIND(REAL(rTHE))*cosP +
     1                   midY*SIND(REAL(rTHE))*sinP +
     2                   midZ*COSD(REAL(rTHE)))       ! -200:200mm

C       increment the Radon plane
              RADON(NINT(REAL(P)/5),
     1              MOD(NINT(REAL(rTHE)/5), 36),
     2              MOD(NINT(REAL(rPHI)/5), 36)) =
     3          RADON(NINT(REAL(P)/5),
     4                MOD(NINT(REAL(rTHE)/5), 36),
     5                MOD(NINT(REAL(rPHI)/5), 36)) + 1
              rTHE = rTHE - 1
              IF (rTHE .EQ. -1) rTHE = 179
              TEMP = ROOT * SIND(rTHE + PSI)
              COUNT = COUNT + 1
            END DO
          END IF
        END DO                  ! Radon PHI values
      GOTO 100                  ! get next event

200   CLOSE (UNIT=1)  ! used all events

      OPEN (UNIT=1,FILE=OUTFILE,STATUS='NEW',RECL=486)
      WRITE (1,210) (((RADON(I,J,K), I=-40,40), J=0,35), K=0,35)
210   FORMAT (81(1X,I5))
      CLOSE (UNIT=1)  ! written all planes to file

      END
```

```
      SUBROUTINE GET_EVENT(EVENT,FIRSTCALL,CHAN,IOSB)

C     Returns the parameters for one event, read from tape and
C     stored in the EVENT(20) array.  Parameters are retrieved
C     in the dame # and order as they were stored by PHANTOM.
C     Split events are ignored (1/block). Program assumes that
C     the tape drive has already been set up and an I/O channel
C     has been assigned.

      REAL*4  EVENT(20)        ! parameters of one event
      LOGICAL FIRSTCALL        ! TRUE the 1st time this routine is called
      INTEGER*2 CHAN,          ! tape drive I/O channel (called INCHAN in EVENT)
     1          IOSB(4),       ! I/O status block (called INIOSB(,) in EVENT)
     3          BUFFER(1:16383),      ! holds block of events from tape (INBUF)
     4          BLKSZw,        ! blocksize in words (INSUL)
     5          BLKSZb,        !    "      "  bytes (INBYTES)
     6          thisEV,        ! pointer to start of current event (INDEX0)
     7          nextEV         !    "     "   "   "  next event (INDEX1)
      INTEGER*4 STATUS,        ! return status of the system call
     1          SYS$QIOW       ! read from CHAN with wait
      EXTERNAL IO$_READVBLK

      BLKSZw = 16383
      BLKSZb = 16383*2

      IF (FIRSTCALL) THEN              ! if this is the first time the routine
        FIRSTCALL = .FALSE.            ! is called, need to get a block of
        nextEV = BLKSZw + 1            ! data from the tape.
      END IF

      IF (nextEV .GT. BLKSZw) THEN     ! need to get a new block
        STATUS = SYS$QIOW(%VAL(1),             ! use event flag #1
     1                    %VAL(CHAN),          ! channel allocated for tape I/O
     2                    %VAL(%LOC(IO$_READVBLK)),   ! function to be performed
     3                    IOSB(1),,,           ! return status
     4                    BUFFER(1),           ! where to start writing at
     5                    %VAL(BLKSZb),,,,) ! # bytes to transfer
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        IF (.NOT. IOSB(1)) RETURN      ! end of tape
        thisEV = nextEV - BLKSZw       ! reset pointers in new buffer
        nextEV = thisEV + BUFFER(thisEV)
        IF (thisEV .LT. 0 .OR. thisEV .GE. 200) THEN  ! data is meaningless
          WRITE (*,*) ' invalid data on tape'
          CALL LIB$STOP(%VAL(1))
        END IF
      END IF

C     fetch one event from the buffer

      CALL LIB$MOVC3((BUFFER(thisEV)-2)*2,     ! # bytes to be moved (skip header)
     1               BUFFER(thisEV+2),         ! where from (skip header)
     2               EVENT)                    ! where to
      thisEV = nextEV                  ! reset pointers in current buffer
      nextEV = nextEV + BUFFER(nextEV)

      RETURN

      END
```

```
      PROGRAM PERFECT

C     Program generates a perfect Radon transform and
C     places it in a file for use with RECON.
C     UNITS ARE CURRENTLY 1mm

      INTEGER*4 PHI, THETA, P, ELIPS, numELIPS, I,
     1          Ainc, Sinc, REC_LEN
      REAL*4  RADON(-200:200), GREY, AREA
      REAL*8  Px, Py, Pz, D, a, b, c,
     1        ELLIPSOID(16,10), Ex, Ey, Ez, Va(3), Vb(3), Vc(3),
     2        ALPHAa, ALPHAb, ALPHAc, Ssqrd, PI
      CHARACTER OUTFILE*20

C     PHI, THETA, P   - define the Radon plane
C     Ainc, Sinc      - increments in the projection directions and
C                       along the projection line
C     Px, Py, Pz      - direction cosines of the normal to the Radon plane
C     ELLIPSOID(,)    - parameters of the ellipsoids
C     RADON()         - values of the Radon planes along 1 projection line
C     Ex, Ey, Ez      - co-ordinates for the center of the ellipsoid
C     Va, Vb, Vc      - direction cosines for the orthonormal axes of the ellipso
C     a, b, c         - lengths of the semiaxes
C     GREY            - grey scale of the ellipsoid
C     ALPHAa, ALPHAb, ALPHAc  - projection of the normal to the plane onto each
C                       axis of the ellipsoid
C     Ssqrd           - square of the distance from (Ex,Ey,Ez) to the tangent
C                       plane in the direction normal to the plane.
C     D               - distance from (Ex,Ey,Ez) to the plane,in the direction
C                       normal to the plane
C     AREA            - area of the intersection of the plane with the ellipsoid
C     REC_LEN         - size of the output record

      DATA PI /3.141592654/
      DATA numELIPS /6/
      DATA ELLIPSOID          ! ellipsoid data in mm
C     1 /0,0,0, 10,20,30, 1,0,0, 0,1,0, 0,0,1, 1,
C     1  0,0,0, 95,95,95, 1,0,0, 0,1,0, 0,0,1, 0,
C     1  0,0,0, 15,15,15, 1,0,0, 0,1,0, 0,0,1, 0,
C     1  112*0/
     2 /0,0,0, 72.3,96.4,127, 1,0,0, 0,1,0, 0,0,1, 1.02,        ! inner skull
     2 -8,-60.5,38.1, 4.6,2.3,2.3, 1,0,0, 0,1,0, 0,0,1, .01,    ! tumor 1
     2  0,-60.5,38.1, 2.3,2.3,4.6, 1,0,0, 0,1,0, 0,0,1, .01,    !  " 2
     2  6,-60.5,38.1, 2.3,4.6,2.3, 1,0,0, 0,1,0, 0,0,1, .01,    !  " 3
     2  22,0,38.1, 11,31,25.4,  .95,-.31,0,  .31,.95,0, 0,0,1, -.02,! r ventrical
     2 -22,0,38.1, 16,41,38.1, -.95,-.31,0, -.31,.95,0, 0,0,1, -.02,! l ventrical
     2  0,114,-19.6, 12.7,34,17, 1,0,0, 0,.54,-.84, 0,.84,.54, 1.5, ! nose
     2  48*0/

      WRITE (*,5) ' output file name: '
      READ (*,5) OUTFILE
5     FORMAT (A$)
      I = INDEX(OUTFILE,'.')          ! if the file lacks an extension
      IF (I .EQ. 0) THEN              ! then default to .PLN.
        I = INDEX(OUTFILE,' ')
        OUTFILE(I:I+3) = '.PLN'
      END IF

      WRITE (*,5) ' Angular increment (degrees): '
      READ (*,*) Ainc
      WRITE (*,5) ' Sampling increment (mm): '
      READ (*,*) Sinc

      REC_LEN = ((400/Sinc)+1)*6
```

```
      OPEN(UNIT=1,FILE=OUTFILE,STATUS='NEW',RECL=REC_LEN)

      DO PHI=0,179,Ainc
      DO THETA=0,179,Ainc
        Px = SIND(DBLE(THETA))*COSD(DBLE(PHI))        ! direction cosines
        Py = SIND(DBLE(THETA))*SIND(DBLE(PHI))        ! of the plane.
        Pz = COSD(DBLE(THETA))

C     get the ellipsoid parameters from the array
        DO ELIPS=1,numELIPS
          Ex = ELLIPSOID(1,ELIPS)
          Ey = ELLIPSOID(2,ELIPS)
          Ez = ELLIPSOID(3,ELIPS)
          a  = ELLIPSOID(4,ELIPS)
          b  = ELLIPSOID(5,ELIPS)
          c  = ELLIPSOID(6,ELIPS)
          DO I=1,3
            Va(I) = ELLIPSOID(6+I,ELIPS)      ! 7,8,9
            Vb(I) = ELLIPSOID(9+I,ELIPS)      ! 10,11,12
            Vc(I) = ELLIPSOID(12+I,ELIPS)     ! 13,14,15
          END DO
          GREY = ELLIPSOID(16,ELIPS)

C     find the distance from the origin of the ellipsoid to the
C     tangent of the plane with the ellipsoid.
          ALPHAa = Px*Va(1) + Py*Va(2) + Pz*Va(3)
          ALPHAb = Px*Vb(1) + Py*Vb(2) + Pz*Vb(3)
          ALPHAc = Px*Vc(1) + Py*Vc(2) + Pz*Vc(3)
          Ssqrd = ALPHAa**2*a**2 + ALPHAb**2*b**2 + ALPHAc**2*c**2
C     calculate the interception with each plane in this stack
          DO P=-200,200,Sinc
            D = P - (Px*Ex + Py*Ey + Pz*Ez)    ! distance of plane from ellipsoid
            IF (D**2 .LE. Ssqrd) THEN          ! does the plane intersect the elli
              AREA = (PI*a*b*c*GREY*(Ssqrd - D**2))/Ssqrd**(3.0/2.0)
              RADON(P/Sinc) = RADON(P/Sinc) + AREA
            END IF
          END DO
        END DO
C     write out values for this set of planes
        WRITE (1,10) (NINT(RADON(P)), P=-200/Sinc,200/Sinc)
        DO P=-200/Sinc,200/Sinc
          RADON(P) = 0
        END DO
      END DO
      WRITE (*,*) ' PHI:',PHI
      END DO

10    FORMAT (<REC_LEN/6>(1X,I5))
      CLOSE (UNIT=1)

      END
```

```
      PROGRAM RECON

C     does straight & filtered backprojection, taking the Radon transform
C     (set of planes) and backprojecting it onto a 3D array of
C     voxels.

      INTEGER*4 RADON(-200:200), I, PHI, THETA, P, Ainc, Sinc, numP
     1          X, Y, Z, CUTOFF, FILTER, TEMP, lo, hi, REC_LEN
     2          FILE_ORG, LEN, TYPE, WORD_SZ, SPE_ORG
      INTEGER*2 ARRAY(-50:50,-50:50)
      REAL*4    VOXEL(-50:50,-50:50,-50:50), sinT, cosT, sinP, cosP,
     1          J, Jm, Jp, k, PI, h(0:60), Fn, SUM, INTERP(-2000:2000),
     2          R(-100:100), Mx, SF
      CHARACTER INFILE*15, OUTFILE*132, DUMMY1*132, cFn*4

C     RADON(P) - Radon transform for constant phi, theta; varying p
C     Ainc, Sinc - increments in the prejection directions and along
C                  the projection line.
C     R(P)     - filtered Radon transform
C     PHI      - phi value of the Radon plane (0:179)
C     THETA    - theta value of the Radon plane (0:179)
C     P        - distance along the normal of the plane (-200:200 mm)
C     X,Y,Z    - cartesian co-ord of the voxel being reconstructed
C                (-115:115 mm)
C     CUTOFF   - sampling cutoff for Paul's filter
C     Fn       - Nyquist frequency
C     VOXEL    - 3D array of the reconstucted image
C     INTERP   - interpolated values along the normal of a stack of
C                planes (.1mm; -2000:2000)
C     Mx       - maximum absolute image value, used for scaling
C     SF       - scale factor
C     FILTER   - type of filter to be used in the reconsruction;
C                1 - 2nd difference filter
C                2 - Hamming filter
C     REC_LEN - length of the output record

      Mx = 0
      PI = 3.141592654

      WRITE (*,20) ' Radon plane file: '
      READ (*,20) INFILE
20    FORMAT (A$)
      TEMP = INDEX(INFILE,'.')          ! if the file does not have an extension
      IF (TEMP .EQ. 0) THEN             ! then default to .PLN.
        TEMP = INDEX(INFILE,' ')
        INFILE(TEMP:TEMP+3) = '.PLN'
      END IF

      WRITE (*,20) ' .SPE file: '
      READ (*,20) OUTFILE
      TEMP = INDEX(OUTFILE,'.')         ! if the file does not have an extension
      IF (TEMP .EQ. 0) THEN             ! then default to .SPE and add the
        TEMP = INDEX(OUTFILE,' ')       ! directory specifier
        OUTFILE(TEMP:TEMP+3) = '.SPE'
        DUMMY1 = 'EVENTDATA:'//OUTFILE
        OUTFILE = DUMMY1
      END IF

      WRITE (*,20) ' Angular increments (degrees): '
      READ (*,*) Ainc
      WRITE (*,20) ' Sampling increments (mm): '
      READ (*,*) Sinc
      REC_LEN = ((400/Sinc)+1)*6
      numP = 200/Sinc
```

```
      WRITE (*,20) ' filter ( 1-2nd diff. ; 2-Hamming ): '
      READ (*,*) FILTER
      IF (FILTER .EQ. 2) THEN
        WRITE (*,20) ' cutoff: '
        READ (*,*) CUTOFF
        WRITE (*,20) ' Fn: '
        READ (*,20) cFn
        READ (cFn,*) Fn        ! translate to a REAL number

        DO J=0,CUTOFF          !  generate filter
          Jm = J - (1/(2*Fn))
          Jp = J + (1/(2*Fn))
          h(J) = -2*PI**2*(k(J,Fn) + .5*k(Jm,Fn) + .5*k(Jp,Fn))
          IF (J .EQ. 0) THEN
            SUM = SUM + h(J)/2
          ELSE
            SUM = SUM + h(J)
          END IF
        END DO
        WRITE (*,*) ' SUM: ',SUM
      END IF

      CALL OPEN_SPECTRUM(OUTFILE,FILE_ORG,MAX_SN)      ! prepare output file
      OPEN (UNIT=2,FILE=INFILE,STATUS='OLD',READONLY,RECL=REC_LEN)
      DO PHI=0,179,Ainc                ! 0:179
        WRITE (*,15) 'PHI = ', PHI
15      FORMAT (1X,A6,I3)
        sinP = SIND(REAL(PHI))
        cosP = COSD(REAL(PHI))
      DO THETA=0,179,Ainc              ! 0:179
        sinT = SIND(REAL(THETA))
        cosT = COSD(REAL(THETA))

C     get the Radon transform data

        READ (2,10) (RADON(P), P=-numP,numP)
5       FORMAT (A)
10      FORMAT (<REC_LEN/6>(1X,I5))
        IF (FILTER .EQ. 2) THEN        ! Hamming filter.
          DO P=-numP,numP              ! for every element of the transform;
            R(P) = 0
            DO J=-CUTOFF,CUTOFF        ! convolve with the filter.
              R(P) = R(P) + RADON(P+J)*h(ABS(J))
            END DO
            R(P) = R(P) * (-1/(8*PI**2))
          END DO
        ELSE                           ! 2nd difference filter.
          DO P=-(numP-1),numP-1
            R(P) = -(RADON(P-1) + RADON(P+1) - 2*RADON(P))
          END DO
          R(-numP) = -(RADON(-numP))
          R(numP)  = -(RADON(numP))
        END IF

C     for each stack of planes (phi & theta constant; p varying), do an
C     interpolation along p, increasing the number of bins by a factor of 10.

        DO P=-numP,nump-1
          DO I=0,9
            INTERP(P*10+I) = REAL(10-I)/10*R(P) + REAL(I)/10*R(P+1)
          END DO
        END DO
        INTERP(numP*10) = R(numP)
```

```
C       Do reconstruction over the largest cube which entirely fits inside
C       the sphere where the Radon transform is defined.  Do it backwards
C       because of FORTRAN ordering of array elements.
C       scaling: 2.31mm/voxel.

          SF = 2.31/(0.1*Sinc)
          DO Z=-50,50           ! set x,y,z to reconstruct 100x100x100
          DO Y=-50,50
          DO X=-50,50

C       find which plane in this stack contains the point (x,y,z)
C       scale P to be .1(Sinc)mm
            P = NINT((X*sinT*cosP + Y*sinT*sinP + Z*cosT)*SF)
            VOXEL(X,Y,Z) = VOXEL(X,Y,Z) + INTERP(P)*sinT
            IF (ABS(VOXEL(X,Y,Z)) .GT. Mx) Mx = ABS(VOXEL(X,Y,Z))
          END DO
          END DO
          END DO        ! finished backprojecting one stack
        END DO
        END DO  ! finished backprojection
        CLOSE (UNIT=2)

        DO Z=-50,50     ! scale the image to 0:250 with 0 @ 125
          DO Y=-50,50
          DO X=-50,50
            ARRAY(X,Y) = NINT(((VOXEL(X,Y,Z)+Mx)*250.0) / (2*Mx))
          END DO
          END DO        ! output data in PERITEK displayable format
          CALL READ_SPE_DESCRIPTION(I,FILE_ORG,LEN,TYPE,WORD_SZ,SPE_ORG)
          CALL WRITE_SPECTRUM(ARRAY,LEN,SPE_ORG)
        END DO

        END

C*************************************************************************

        REAL FUNCTION k(J,Fn)

        REAL*4 J, Fn, PI, PIsqd, PIcbd

        PI = 3.141592654
        PIsqd = PI**2
        PIcbd = PI**3

        IF (J .NE. 0) THEN
          k = (Fn/(PIsqd*J**2)) * COS(2*PI*Fn*J) +
     1        ((2*PIsqd*Fn**2*J**2 - 1)/(2*PIcbd*J**3)) *
     2        (SIN(2*PI*Fn*J))
        ELSE
          k = (2.0/3.0)*Fn**3   ! k(0) term
        END IF

        RETURN
        END
```