

Cranfield Institute of Technology

Department of Applied Computing and Mathematics

Ph.D. THESIS

Academic Year 1992-1993

Karam Ali Fayed

Computational Methods for Image Rotation and Computer Aided Tomography

Volume I: Theoretical Basis

Supervisor:

Dr. J. M. Blackledge

January 1993



IMAGING SERVICES NORTH

Boston Spa, Wetherby
West Yorkshire, LS23 7BQ
www.bl.uk

**TEXT CUT OFF IN THE
ORIGINAL**

ABSTRACT

This thesis, describes work whose principal goal has been to research and identify an optimum algorithm for computing the Radon transform - optimum in the sense of minimum CPU time for maximum image fidelity. This transform is the basis for a wide range of applications which are either directly or indirectly associated with Computer Aided Tomography. It has therefore been important to try and identify an optimum computational technique for the evaluation of this transform. A central theme to the computational techniques reported in this thesis has been the digital image rotation for which a new algorithm has been designed based entirely on integer arithmetic. This algorithm has been used for the evaluation of the forward and inverse Radon transforms (in particular, forward and back-projection). The relationship between the interpolation methods used for image rotation and the filtering techniques required to compute the inverse Radon transform has been studied in detail. The result of this study has been to identify a specific data processing path which provides an algorithm whose 'computational energy' is approximately half that of previously published algorithms.

With an optimum algorithm identified, research has been undertaken into the use of the Radon transform for two-dimensional array processing. It is shown that the Radon transform can be used to reduce the dimensionality of a problem from two-dimensions to a one-dimensional problem. By applying signal processors to the set of projections obtained by taking the Radon transform of an image, the inverse Radon transform can be used to reconstruct the processed image. The value of this approach lies in the fact that some useful one-dimensional signal processing algorithms do not have a two-dimensional extension or else the implementation of the two-dimensional algorithm is computationally intensive. An example of the latter case is reported.

ACKNOWLEDGMENTS

My sincere gratitude to Allah (God), Almighty, for giving me the ability, health, and knowledge to complete the requirements of my academic career.

With the number of people who have assisted me in this work, there is no way to have a short list of acknowledgments. This is my chance to publicly thank everyone for contributions towards this final goal. I am grateful to you all.

First, I would like to thank my supervisor Dr. J. M. Blackledge who has taken a great interest and guided me in my work. He was always enthused about my work and took special interest in it. Dr. Blackledge is an excellent teacher, and a great friend. It was a happy day for me when he confirmed the transfer of my registration from the Electronic System Design Department to the Applied Computing and Mathematics Department. It is a pleasure to be around an exuberant person with many great ideas, and with the ability to quickly sort out the not-so-great ideas. Working with such a person makes education very rewarding - from development of a theory, to implementation and experimentation, to modifications, and the excitement at the results. Dr. Blackledge has been this kind of mentor to me. I very much appreciate the time I spent learning with him. I am grateful also to the rest of the Applied Computing and Mathematics Department staff. Thank you for all of your help.

I would like also to express my gratitude to the Egyptian government who provided me with the financial support to carry out this research work.

My gratitude also go to many other consultants at the Computer Centre who helped me in my work. These include J. Haynes, the software support consultant,

and K. Caines, the system programmer.

Thanks to the members of the Cranfield Library who have been very supportive of my work, especially J. Harrington, the Information Services Manager and Miss Michelle Morris of the Inter-Library Loan Section.

My sincere thanks to the administrative group. Both professor P. Hutchinson, Head of the school, and Mr. S. Allison, the Assistant Registrar, have provided great motivational and technical support. Their confidence in me has helped me through this great hurdle in my life, and I appreciate all their efforts.

I was extremely fortunate to have a supportive, intelligent, interesting, and fun group of colleagues at the Cranfield Institute of Technology (CIT) with whom I had a great help. For this, I must thank, Dr. Hilal Al-Hinai of the Applied Energy Department, Khamis Al-Balushi of the Turbo-Machinery and Engineering Mechanics Department, Mahmoud Tahat of the Applied Energy Department, A. Abd Al-Maksoud of the School of Management, F. Al-Toney of the School of Management, Slahadin Ashair of the Thermal Power Group, and Saleh Mohamed of the Thermal Power Group.

Absolutely none of this work would have been possible without the encouragement, help, and love of my family and friends who have been extremely understanding throughout these last three years. A special thanks goes to my father and my mother, in-laws, for their support to finish this work. Also thanks to my brothers, sisters, in-laws, cousins – all of whom have issued words of comfort and have understood my absence for this work.

Most importantly, extreme thanks, praise, love, and adoration go to my wife and our children: to my son Mohammed, who has patiently waited for his dad to play ball, and my daughter Eman, who has always found the right time to adorn my office with the perfect inspirational smile. You are more wonderful children than any father's hope, and you make this all worthwhile. You have a share in this work,

and can all take pride in it, for it was your sacrifice as much as mine ... and I thank you with all my heart.

Finally, I dedicated this manuscript to my father and my mother whose interest in my education have shone a light, long beyond their death, who would have been fortunate enough to witness the culmination of their inspiration, enthusiasm, and love.

NOTATIONS

CT	Computed Tomography
1D	One Dimensional
2D	Two Dimensional
CPU	Central Processing Unit
CIT	Cranfield Institute of Technology
FRT	Fast Radon Transform
IFRT	Inverse Fast Radon Transform
ECT	Emission Computed Tomography
TCT	Transmission Computed Tomography
SPECT	Single Photon Emission Computed Tomography
PET	Positron Emission Tomography
ToF	Time of Flight
λ	The wave length
NMR	Nuclear Magnetic Resonance
RF	Radio Frequency
FM	Frequency Modulation
SAR	Synthetic Aperture Radar
θ	Angle of rotation
$P_{\theta}(z)$	Parallel projection or Radon transform
NDT	Non-Destructive Testing
R	Rotation matrix
x,y	are the input pixel coordinates
u,v	are the new output pixel in the rotated coordinates
m_N	is the multiplication operations
ad_N	is the number of addition operations
s_N	is the sbtraction operations
dv_N	is the number of division operations

sqN	is the number of square root operations
$N \times N$	Image size
MAF	Moving Average Filter
FIR	Finite Impulse Resposse filter
MF	Median Filter
3D	Three Dimension
$\hat{F}_1$	Fourier operator
$\hat{F}_1^{-1}$	Inverse Fourier operator
R^T	The Transposition of Rotation Matrix
R^{-1}	The Inverse of Rotation Matrix
$\hat{R}$	The Radon transform operator
$\hat{R}^{-1}$	The inverse Radon transform
$\hat{H}$	The Hilbert transform operator
$\hat{B}$	The back-projection operator
CE	The Computation Energy
T	The Total CPU reconstruction time
CE_c	The Computation Energy for the current technique
CE_o	The Computation Energy for the optimum technique
δCE	The relative Computation Energy saved by the optimum technique
MPE	Mean of the Projection square Error
SDPE	Standard Deviation of the Projection square Error
DSP	Digital Signal Processing
VLSI	Very Large Scale Integration

CONTENTS

Chapter 1: Major Fields of Applications

1.1	Introduction	1
1.2	Rotation and projection	3
1.3	Tomography	5
1.4	Industrial computer vision	5
1.5	Computerized Tomography (CT)	7
1.6	Industrial CT system design	7
1.7	Medicine; X-Ray Transmission Computed Tomography	8
1.8	Medicine; Emission Computed Tomography (ECT)	10
1.9	Medicine; Ultrasound Computed Tomography (UCT)	12
	i) Amplitude data	12
	ii) Propagation time data	12
1.10	Diffraction tomography	14
1.11	Astronomy	16
	1.11.1 Historical overview	16
	1.11.2 Brightness distribution	17
1.12	Nuclear magnetic resonance	18
1.13	Optics	19
1.14	Geophysics	21
1.15	Military application	21
1.16	Material science and others	22
1.17	Dissertation overview	23
1.18	References	25

Chapter 2: Theory of Digital Image Rotation

2.1	Introduction	41
-----	--------------------	----

2.2	Some basic relationships between pixels	41
2.2.1	Four-neighbours of a pixel	41
2.2.2	Eight-neighbours of a pixel	42
2.3	Basic problem	42
2.3.1	Rotation of axes	42
2.3.2	Basic mathematical expression	43
2.3.3	Problem	44
a)	Forward pixel mapping	44
b)	Reverse pixel mapping	44
2.4	Techniques for mapping Polar coordinate onto a discrete Cartesian grid ...	45
2.4.1	Nearest neighbour techniques	45
2.4.2	Interpolation Methods	45
1)	Bilinear or Rectangular interpolation	46
2)	Triangular interpolation	47
3)	2D Linear interpolation	48
2.4.3	Multi-neighbour interpolation	49
a)	Twelve-neighbours interpolation	49
1.	Rectangular-12 neighbours interpolation	49
2.	Triangular-12 neighbours interpolation	50
3.	Linear interpolation over 12 neighbour pixels	52
b)	Sixteen-neighbours interpolation	53
1.	Rectangular-16 neighbours interpolation	53
2.	Triangular-16 neighbours interpolation	55
3.	Linear interpolation over 16 neighbour pixels	56
2.5	Factorizations of the basic rotation matrix	58
2.5.1	Two-pass rotation	58
2.5.2	Three-pass rotation	59
2.6	The Trigonometric computing techniques	59

2.7 Holes in image rotation	60
2.8 References	60

Chapter 3: Optimum Digital Rotation

3.1 Introduction	76
3.2 Mathematical formulation	76
3.3 Algorithm for calculating the optimum rotation	83
3.4 Comparison of rotation techniques	83

Chapter 4: Image Rotation Using Filtering Techniques

4.1 Introduction	91
4.2 The Moving Average Filter (MAF)	91
4.3 Finite Impulse Response Filter (FIR)	92
4.4 Median Filter (MF)	92
4.5 Comparison of rotation techniques	93
4.6 References	94

Chapter 5: Relation of Rotation to Other Transform

5.1 Introduction	105
5.2 Relation to the Radon transform	105
5.3 Relation to the Fourier transform	105
5.3.1 Effect of rotation on Fourier transform	106
5.3.2 Connection to Fourier transform	108
5.4 Relation to the Hough transform	110
5.5 Relation to the Hotelling transform	111
5.6 Relation to the Abel transform	115
5.7 References	112

Chapter 6: The Radon Transform

6.1 Introduction	118
------------------------	-----

6.2	The Radon transform	119
6.3	The inverse Radon transform	121
6.4	Summary	121
6.5	References	122

Chapter 7: Application For Implementing 1D Processes

7.1	Introduction	129
7.2	Computing projections of digital images	129
7.3	Test example	133
7.4	Error discussion	133
7.5	The computational CPU time	133
7.6	The mathematical model	134
7.7	References	134

Chapter 8: Reconstruction Using Image Rotation

8.1	Introduction	143
8.2	Fast Radon transform and back projection	143
8.3	Reconstruction methods	145
8.4	Reconstruction by filtered back-projection	145
8.4.1	Reconstruction using Hilbert transform	146
8.4.2	Reconstruction using 1-D inverse filter	149
8.5	Reconstruction by back projection and deconvolution	150
8.6	Reconstruction using central-slice theorem	152
8.7	Noise analysis	153
8.8	Discussion	154
8.9	References	154

Chapter 9: Reconstruction Error and Statistical Analysis

9.1	Introduction	168
9.2	Errors in reconstruction	168

9.3	Statistical description of the mean square error	169
9.4	Projection square error	169
9.5	Statistical description of projection square error	170
9.6	Test example	170
9.7	Quantative study of the reconstruction techniques	170
9.7	Summary	171

Chapter 10: Conclusion and Future Work

10.1	Introduction	190
10.2	Conclusion	190
10.3	Future work	192
10.3.1	Fast reconstruction	192
10.3.2	Parallel implementation for Radon transform and its inverse	193
10.3.3	Multidimensional signal processing	193

APPENDIX A

A1	Introduction	195
A2	Software used	195

List of Figures

Figure 1.1	Application fields of image rotation.	30
Figure 1.2	The coordinate system in both (x, y) space and frequency space are rotated by angle θ	31
Figure 1.3	Geometry for tomography.	31
Figure 1.4	The fast Radon Transform (FRT) based computer vision system.	32
Figure 1.5	Mapping of a straight line in Cartesian space to a point in Radon space. ..	32
Figure 1.6	Common first-generation CT system design.	33
Figure 1.7	3D object composed of a stack of slices, the distribution of material in a single slice can be obtained by measuring x-ray intensity profile produced at different angles.	34
Figure 1.8	Definition of a nondiffracting tomographic projection.	35
Figure 1.9	The source and detector move together(z varies) so that the beam covers the entire region leading to the profile $P(z, \theta)$ for fixed angle θ	35
Figure 1.10	Methods of defining the ray direction for three different types of tomography.	36
Figure 1.11	Illustration of the diffraction tomography using pulsed fields.	36
Figure 1.12	Coordinate system associated with lunar disk and the transverse velocity V along the P axis.	37
Figure 1.13	Basic schematic of the important hardware on an NMR imaging system. ..	37
Figure 1.14	A 2D NMR image is reconstructed from a series of 1D projections obtained by recording NMR signals in gradients directed at different orientations relative to the sample.	38
Figure 1.15	Achromatic grating interferometer.	39
Figure 1.16	Filling the frequency domain by taking measurements at different angles. ..	39

Figure 1.17	Radar geometry as function of range and ground angle.	40
Figure 1.18	Typical geometry for a tomographic system.	40
Figure 2.1	Basic relationships between pixels.	61
Figure 2.2	Rotation of axes.	61
Figure 2.3	Cartesian and Polar coordinates representation.	62
Figure 2.4	Original image of 128x128 resolution and 16 grey level.	62
Figure 2.5	Normal digital rotation using nearest neighbour technique at 70° rotation.	63
Figure 2.6	Normal digital rotation using four neighbours technique at 70° rotation. ...	63
Figure 2.7	Normal digital rotation using eight neighbours technique at 70° rotation. .	64
Figure 2.8	Area interpolation scheme.	64
Figure 2.9	Digital rotation using bilinear interpolation at 70° rotation.	65
Figure 2.10	Digital rotation using triangular interpolation at 70° rotation.	66
Figure 2.11	Digital rotation using 2D-linear interpolation at 70° rotation.	67
Figure 2.12	Multi-neighbour interpolation.	68
Figure 2.13	Digital rotation using rectangular 12-neighbours interpolation at 70° rotation. 69	
Figure 2.14	Digital rotation using triangular 12-neighbours interpolation at 70° rotation. 70	
Figure 2.15	Digital rotation using linear interpolation over 12-neighbour at 70° rotation. 71	
Figure 2.16	Digital rotation using rectangular 16-neighbours interpolation at 70° rotation. 72	
Figure 2.17	Digital rotation using triangular 16-neighbours interpolation at 70° rotation. 73	
Figure 2.18	Digital rotation using linear interpolation over 16-neighbour at 70° rotation. 74	
Figure 2.19	The two-pass rotation (Top) and the augmented two-pass rotation (bottom). 75	

Figure 2.20	The point u, v is loaded with the value at x, y in two different ways for the two-and three-pass methods.	75
Figure 3.1	Representation of Polar coordinates on a Cartesian grid.	85
Figure 3.2	Optimum image rotation flowchart.	86
Figure 3.3	Original image of 128x128 resolution and 16 grey level.	87
figure 3.4	Optimum digital rotation using nearest neighbour technique at 70° rotation.	87
Figure 3.5	Optimum digital rotation using four neighbours technique at 70° rotation.	88
Figure 3.6	Optimum digital rotation using eight neighbours technique at 70° rotation.	88
Figure 4.1	Test image of 128 x 1128 resolution.	95
Figure 4.2	Test image after applying 70° rotation using nearest neighbour technique.	95
Figure 4.3	Test image after applying the moving average filter.	96
Figure 4.4	Test image after applying the finite impulse response filter.	96
Figure 4.5	Test image after applying the median filter.	97
Figure 4.6	3D-view after applying 70° rotation using nearest neighbour.	98
Figure 4.7	3D-view after applying the moving average filter.	99
Figure 4.8	3D-view after applying the FIR filter.	100
Figure 4.9	3D-view after applying the median filter.	101
Figure 4.10	Practical example of 128x128 resolution.	102
Figure 4.11	Practical example after applying 70° rotation using nearest neighbour technique.	103
Figure 4.12	Practical example after applying the median filter.	104
Figure 5.1	Relationship between rotation and Fourier transform.	117
Figure 5.2	Rotation of coordinate system.	117
Figure 6.1	Coordinate systems of computed tomography.	125
Figure 6.2	Spatial and Radon transform domains.	126
Figure 6.3	Images of parallel-beam projection CT for one cross-sectional plane through three cylinders.	127

Figure 6.4	Parallel Radon space of the object over 180 degrees.	128
Figure 6.5	Estimated reconstruction of the object from the Radon space.	128
Figure 7.1a	Head-Phantom image.	136
Figure 7.1b	Head-Phantom image with two spikes.	137
Figure 7.2	Result after applying 2D median filter.	138
Figure 7.3	Result after applying Radon transform followed by 1D median filter and then, taking the inverse Radon transform.	139
Figure 7.4	CPU time versus window size for implementing 1D MF using FRT and 2D MF. 140	
Figure 7.5	Polynomial least-squares fit.	141
Figure 8.1	Back-projection of $p_1(z)$ and $p_2(z)$ at (r, ϕ)	156
Figure 8.2	Schematic diagram of the reconstruction process using differntial filter and Hilbert transform method.	156
Figure 8.3	The amplitud responses of the differntial filters.	157
Figure 8.4	The phase responses of the differential filters.	157
Figure 8.5	Head-phantom image.	158
Figure 8.6	Estimated reconstruction of the head-phantom using filtered back-projection (Hilbert transform).	158
Figure 8.7	Schematic diagram of the reconstruction process using filtered back projection (1D inverse filter).	159
Figure 8.8	The amplitud response of the 1D inverse filter.	160
Figure 8.9	The phase response of the 1D inverse filter.	160
Figure 8.10	Radon space of the head-phantom image.	161
Figure 8.11	Estimated reconstruction of the head-phantom using filtered back projection (1D inverse filter).	161
Figure 8.12	Schematic diagram of the reconstruction process using back projection and deconvolution.	162
Figure 8.13	The amplitud response of the 2D inverse filter.	163

Figure 8.14	The phase response of the 2D inverse filter.	164
Figure 8.15	Radon space of the head-phantom image.	165
Figure 8.16	Estimated reconstruction of the head-phantom using back projection and deconvolution.	165
Figure 8.17	Schematic diagram of the reconstruction process using central-silce theorem.	166
Figure 8.18	Radon space of the head-phantom image.	167
figure 8.19	Estimated reconstruction of the head-phantom using central-slice theorem.	167
Figure 9.1	The projection error of a test image associated with nearest neighbour using filtered back projection (Hilbert transform).	173
Figure 9.2	The projection error of a test image associated with interpolation technique using filtered back projection (Hilbert transform).	174
Figure 9.3	The projection error of test image associated with nearest neighbour technique using filtered back projection (1D inverse filter)	175
Figure 9.4	The projection error of test image associated with interpolation technique using filtered back projection (1D inverse filter).	176
Figure 9.5	The projection error of a test image associated with nearest neighbour using back projection and deconvolution.	177
Figure 9.6	The projection error of a test image associated with the interpolation technique using back projection and deconvolut.	178
Figure 9.7	The projection error of a test image associated with the nearest neighbour using central-slice theorem.	179
Figure 9.8	The projection error of a test image associated with the interpolation technique using central-slice theorem.	180
Figure 9.9	The computation energy associated with the reconstruction techniques using different rotation algorithms.	181

List of Tables

Table 3.1	Comparison between the number of operations in the individual rotation algorithm	89
Table 3.2	Comparison of CPU rotation time of the nearest neighbour techniques at 70° rotation	89
Table 3.3	Comparison of CPU rotation time of the interpolation and multi-interpolation techniques at 70° rotation	90
Table 4.1	The CPU time for each filtering technique	97
Table 7.1	The CPU time versus window size for implementing 1D MF using FRT and 2D Median Filter	142
Table 9.1	Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using filtered-back projection technique (Hilbert transform)	181
Table 9.2	Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using filtered-back projection technique (Hilbert transform)	181
Table 9.3	Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using filtered-back projection technique (1D inverse filter)	182
Table 9.4	Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using filtered-back projection technique (1D inverse filter)	182
Table 9.5	Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using back projection and deconvolution technique	183
Table 9.6	Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using back projection and deconvolution technique	183

Table 9.7	Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using central slice theorem	184
Table 9.8	Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using central slice theorem	184
Table 9.9	The computation energy associated with the nearest neighbour techniques using filtered-back projection (Hilbert transform)	185
Table 9.10	The computation energy associated with the interpolation techniques using filtered-back projection (Hilbert transform)	185
Table 9.11	The computation energy associated with the nearest neighbour techniques using filtered-back projection (1D inverse filter)	186
Table 9.12	The computation energy associated with the interpolation techniques using filtered-back projection (1D inverse filter)	186
Table 9.13	The computation energy associated with the nearest neighbour techniques using back-projection and deconvolution	187
Table 9.14	The computation energy associated with the interpolation techniques using back-projection and deconvolution	187
Table 9.15	The computation energy associated with the nearest neighbour techniques using central slice theorem	188
Table 9.16	The computation energy associated with the interpolation techniques using central slice theorem	188

CHAPTER 1

Major Fields of Applications

1.1 Introduction

In recent years there has been a rapid advance in the science and technology of information processing and analysis. Previously, the large majority of research and development in this subject was almost exclusively stimulated by the need for military intelligence. Now, it is important in all physical and biological sciences.

Over the past twenty years in particular, many important developments have been occurred in information science. This has been due to a massive increase in the speed, power and availability of digital computers. One area of information technology that has grown rapidly because of this has been imaging science. This subject has become increasingly important because of the growing demand to obtain information about the structure, composition, and a behaviour of objects without having to inspect them visually.

On this basis, the rotation of the image is one of the fundamental functions in the image processing. Developments in the field of image processing such as pattern analysis for x-ray photographs and remote sensing images and character discrimination are very remarkable. Up to now, yet, only a part of those results has been applied to a real time system, since many of their processing times are rather long, even if run on large computer systems. Therefore, visual sensors in a

production line, industrial robots, and so on, have been developed from a different highest priority that the processing is performed in a very short time by a small-scale computer.

Image rotation has been considered as one of the most important basic processors in an image system with the object of processing matching, we can use more parameters for image features, and a configuration of complexity will be able to be recognized more easily. However, an expensive high-speed computer or image processor is necessary when image rotation is applied to this system. Therefore, the developments of a new method for digital image rotation are introduced.

The rotation problem is the fundamental part of the Radon transform. The approach to computing the Radon transform in this thesis is based on rotating the object about its centre and taking projections at regular steps in the angle of rotation. Hence, part of the research undertaking for this thesis has concentrated on generating accurate and efficient method of image rotation. This work forms a basis of Chapter 3.

The mathematical basis of projection tomography was derived in 1917 by the German mathematician Johannes Radon. Radon proved that the complete set of one-dimensional projections obtained from a continuous two-dimensional function contains all the information required to reconstruct same function. A projection is obtained by integrating a function over a set of parallel lines and is characteristic of its angle of rotation in the cartesian plan [1-2]. The application of image rotation for a large class of reconstruction problems is shown in Figure 1.1. The reconstruction problem is the problem of determining the internal structure or, more precisely, some property of the internal structure of an object without having to cut, crack, or otherwise macroscopically damage the object. The success of the new technology associated with medical imaging by use of computerized tomography is largely responsible for much of current interest in a variety of reconstruction meth-

ods. Not only do these methods and this technology hold much promise in medicine, but recent success there suggests interesting possibilities for expanded use of similar techniques in other fields. The image reconstructions are easily described by discrete mathematics. This process is performed by digital computer. The computation is time consuming and causes reconstruction times to be greater than a few seconds. Therefore, an optimum rotation algorithm for image reconstruction has been derived that can rapidly reduce the reconstruction time.

1.2 Rotations and Projections

The result that forms the basis of tomography will now derive. Let us consider the two-dimensional Fourier transform and inquire what happens if the coordinates in both (x, y) space and frequency space are rotated by an angle θ , as shown in Figure 1.2. It is clear that the value of s at the point Q in Figure 1.2a and the value of S at point P in Figure 1.2b must be independent of the coordinate system in which we express s or S . That is, $S(P)$ cannot depend on the coordinate system chosen. Therefore, upon referring to Figure 1.2b it is clear that because P has coordinates (K_x, K_y) in the reference system, denoted by O , and coordinates $(\acute{K}_x, \acute{K}_y)$ in the rotated system, denoted by θ , that :

$$S_o(K_x, K_y) = S_\theta(\acute{K}_x, \acute{K}_y) \quad (1.1)$$

where

S_o denotes S as measured in the O coordinate system.

S_θ signifies the value of S as measured in the θ coordinate system.

From Figure 1.2b one sees that (see Chapter 2) (k_x, k_y) are related to $(\acute{k}_x, \acute{k}_y)$ through

$$k_x = \acute{k}_x \cos \theta - \acute{k}_y \sin \theta \quad (1.2)$$

$$k_y = \acute{k}_x \sin \theta + \acute{k}_y \cos \theta$$

or conversely

$$\dot{k}_x = k_x \cos \theta + k_y \sin \theta \quad (1.3)$$

$$\dot{k}_y = -k_x \sin \theta + k_y \cos \theta$$

If Eqn.(1.2) is substituted into Eqn.(1.1), it can express all measurements of S in systems rotated by θ in terms of the Fourier transform measured in the reference system as

$$S_\theta(\dot{k}_x, \dot{k}_y) = S_o(\dot{k}_x \cos \theta - \dot{k}_y \sin \theta, \dot{k}_x \sin \theta + \dot{k}_y \cos \theta) \quad (1.4)$$

The result in Eqn.(1.4) can be used to derive the projection-slice theorem. In order to do this suppose that neither $s_\theta(u, v)$ nor $S_\theta(\dot{k}_x, \dot{k}_y)$ can be measured but it is possible to measure the projection of $s_\theta(u, v)$ as defined by

$$p_\theta(u) = \int_{-\infty}^{\infty} s_\theta(u, v) dv \quad (1.5)$$

The problem is then to deduce $s_o(x, y)$, or equivalently $S_o(k_x, k_y)$, from a knowledge of $p_\theta(u)$. That is, we are willing to continuously rotate the coordinate axes in Figure 1.2a and measure $p_\theta(u)$ at each rotation angle θ , and from these values $s_o(x, y)$ or $S_o(k_x, k_y)$ can be calculated.

In order to express $p_\theta(u)$ in terms of $S_o(k_x, k_y)$, let us take the Fourier transform $P_\theta(\dot{k}_x)$ of $p_\theta(u)$ with respect to u . This is

$$\begin{aligned} P_\theta(\dot{k}_x) &= \int_{-\infty}^{\infty} du e^{-j\dot{k}_x u} \int_{-\infty}^{\infty} dv s_\theta(u, v) \\ &= \int_{-\infty}^{\infty} du \int_{-\infty}^{\infty} dv s_\theta(u, v) e^{-j(\dot{k}_x u + 0v)} \\ &= S_\theta(\dot{k}_x, 0) \end{aligned} \quad (1.6)$$

That is, the Fourier transform of the projection $p_\theta(u)$ can be expressed in the rotated θ coordinate system. The final step in the use of Eqn.(1.4) to express $S_\theta(\dot{k}_x, 0)$ in terms of S_o . Then, we get

$$P_\theta(\dot{k}_x) = S_o(\dot{k}_x \cos \theta, \dot{k}_y \sin \theta) \quad (1.7)$$

Equation (1.7) is the projection theorem and allows one to calculate S_o from the Fourier transform of the projections.

1.3 Tomography

'Tomography' is derived from the Greek word $\tau\omicron\mu\omicron\sigma$ =slice. It stands for several different techniques in diagnostic radiology permitting the imaging of cross sections of the human body.

Computer-aided tomography is a method for obtaining a two-dimensional view of an object by digitally processing many one-dimensional views taken at different rotation angles. This is done by passing sets of narrow x-ray beams through the object and detecting their intensities using an array of sensors, as shown in Figure 1.3. The quantity measured here is given by equation (1.5) and from these measurements we can able to deduce the object function in the reference coordinate system.

1.4 Industrial Computer Vision

Computer vision is concerned with the analysis and recognition of images. It is particularly important to manufacturing industry for automatic inspection and for military applications (e.g. guided weapons systems and automatic targeting). Since a function is required to rotate the image in the pattern matching of the object to assist in the recognition of the object. However, the image rotation in the digital image processing requires a long time, when the object images are increased. From such a view point , it is requested to speed up the processing to reduce the processing time [3-5].

Computer vision system for automatic inspection of machine parts requires computing Fast Radon Transform (FRT) that transfer the image into Radon space, then extracting the information to describe the contents of the image (peaks in Radon space) using a point detector and the cluster averaging technique. Finally,

the image is reconstructed from the extracted points using Inverse Fast Radon Transform (IFRT) (see Figure 1.4).

the problem of edge detection plays a key role in computer vision since the recognition schemes are essentially based on interpreting the edges of an object in a scene. The need for enabling machines to identify objects in images of more complex scenes stems from applications in industrial automation, biomedical engineering, military application for guided weapons and for guided vehicles, remote sensing and robotic.

The projection transform utilized in computer vision is the Hough transform. The Hough transform was derived independently from the Radon transform. However, it has been shown that the Hough transform is just a special case of Radon transform [2]. The basic idea involves the identification of lines in digital images. The Radon transform of a function concentrated at a point described by the 2D delta function.

$$\delta^2(x - x_0, y - y_0) = \delta(x - x_0)\delta(y - y_0)$$

yields a sinusoidal curve

$$P = x_0 \cos \phi + y_0 \sin \phi$$

in $p\phi$ -plan. All colinear points in the xy -plan along a line determined by fixed values ϕ_0 and P_0 , map to sinusoidal curves in the $P\Phi$ -plane and intersect in the same point. Thus, if we choose a suitable method for plotting the projections of a digital image as a function of Φ and P , it follows that the Radon transform may be regarded as a line to point transform, as indicated by Fig. 1.5 . By utilizing the line detected properties, the manufactured objects can be analysed against their known characteristics. From these characteristics, identification of faults can be spotted. Also, from Fig. 1.5, by utilizing the line to point transform, it may be applied to image data compression. This is because of the ever-increasing desire to

transmit and store images, it is common to compress the data into a form less space consuming, thereby allowing speedier transmission or denser storage.

1.5 Computerized Tomography (CT)

CT was first developed in the early 1970's as a tool for medical diagnostic imaging [6] and its inventors were awarded the Nobel prize in medicine in 1979. It is now a familiar technique for physicians to obtain high resolution (0.5-1 mm) pictures of internal organs without surgery.

CT is the reconstruction of a clean image of the density from digital computational operations on measurements of emanations that have passed through the body. The fundamental problem of these computations is the digital image rotation. It has been applied in many scientific fields. An important version of the problem in medicine is that of obtaining the internal structure of the human body from multiple projections obtained using X-rays, γ -rays, Ultra-sound, etc. This process is referred to as computerized tomography. Computerized tomography is classified into two categories: 1-Industrial and 2- Medical applications. The subject of computerized tomography in medicine is conveniently divided into three major areas : 1- Transmission CT, 2- Emission CT, and 3- Ultrasound CT.

1.6 Industrial CT System Design

A typical experimental set-up for computed tomography system is shown in Figure 1.6. This simple design represents the first generation for transmission CT system. An x-ray source and associated detector are placed on opposite sides of the object and controlled by computer. The source/detector pair are moved in concert, perpendicular to the beam, along a straight-line path so that they always stay in a chosen transverse plane of the object (called the "slice plane"). For each individual ray position, the transmitted x-ray intensity values are collected and stored in the computer. One complete scan of all rays across the object at a fixed

angle gives a projection of the slice-plane for that angle. We can then rotate the object (or, equivalently, rotate the source/detector pair as in medical systems) to measure a similar set of rays through the same plane but at a different angle. Projections covering a range of angles from zero to 180 degrees (for parallel-beam data) are considered a full set of CT data. This data is then processed on a computer to reconstruct the two-dimensional (2D) attenuation coefficient distribution in the cross-sectional slice-plane. Algorithms to perform this task are described in Chapter 6. This 2D cross-section is then treated as an image and further processed, displayed, and analyzed. Most modern CT systems employ more sophisticated geometries than the first-generation system of Figure 1.6. For example, arrays or full rings of fixed detectors can be placed around the object.

1.7 Medicine; X-Ray Transmission Computed Tomography

The most widely known of computerized tomography is transmission computed tomography in which an image is produced that displays the absorption coefficient for X-rays passing through the body from an external source. This technique is commonly known as CT scanning. The X-ray process involves recording X-rays on a photographic plate as they emerge from a three dimensional object after having been attenuated by an amount that is terminated by the path followed by a particular ray through the object. This gives an image known as a radiograph. Each grey level of this type of image is determined by the combined effect of all the absorbing elements that lie along the path of an individual ray. In X-ray imaging, computer tomography provides a quantitative image of absorption coefficient of X-rays with initial intensity I_o . If an X-ray passes through a homogeneous material with attenuation coefficient μ , the beam intensity is observed to decrease [7] in accord with the equation :

$$I = I_o \exp(-\mu x) \quad (1.8)$$

Where :

I is the observed intensity after the beam passes a distance x through the material.

If the X-ray beam passes through two different materials (inhomogeneous), distance x_1 through medium 1 characterized by μ_1 and distance x_2 through medium 2 characterized by μ_2 , the fractional decrease in intensity is given by :

$$\frac{I}{I_o} = \exp[-\mu_1 x_1 - \mu_2 x_2]$$

For several media the relation is

$$\frac{I}{I_o} = \exp[-\sum_j \mu_j x_j]$$

If $\mu = \mu(x)$ is a continuous (or piecewise continuous) function of x , the sum goes over to an integral along the beam path L .

$$\frac{I}{I_o} = \exp[-\int_L \mu(x) dx] \quad (1.9)$$

Now, consider a transverse section through some three-dimensional objects Fig. (1.7). If the transverse section plane is perpendicular to the z axis (of an orthogonal coordinate system, the linear attenuation coefficient in the plane may be written as a function of two variables x , and y , which coordinatize the xy plane,

$$\mu = \mu(x, y)$$

If the narrow beam traverses the xy plane as indicated in Fig. (1.8), the fractional decrease in intensity is given by :

$$\frac{I}{I_o} = \exp[-\int_L \mu(x, y) ds] \quad (1.10)$$

Where the line integral is along the beam path L .

The natural logarithm of Eq. (1.10) yields a single projection

$$P = -\log\left(\frac{I}{I_o}\right) = \int_L \mu(x, y) ds. \quad (1.11)$$

By moving the source and detector as indicated in Fig. 1.9, it is possible to obtain a set of projections or profile for the angle θ . The value of the intensity and therefore P depends upon the point where the ray passes through the object which shall be denoted by z . therefore, by adjusting the source of X-rays and the orientation of the attenuating object, a full sequence of projections can be obtained which are related to two dimensional attenuation $\mu(x, y)$ by the equation :

$$P(z, \theta) = \int_L \mu(x, y) ds \quad (1.12)$$

Then if P is known for all values of z and θ , then P is the Radon transform of μ and that μ can be reconstructed from P by employing the inverse Radon transform [8] - [15].

1.8 Medicine; Emission Computed Tomography (ECT) :

Another form of imaging based on the use of projections is emission tomography. The use of ECT in medical science provides a dramatic advancement in nuclear medicine just as X-ray TCT does in radiology. The goal is to figure out the distribution (location and concentration) of some radioactive isotope inside the human body by studying the emitted photons.

There are two basic types of ECT problems, depending on whether the isotope utilized is a single photon emitter (SPECT), such as iodine-131 or technetium-99m, or a positron e^+ (or β^+) emitter (PET) such as carbon-11 or oxygen-15.

In transmission tomography, a single ray is shown in Fig. 1.10a. The source X-ray is a well-focussed spot on the anode of an X-ray tube, and this point, therefore, defines one end of the ray. On the opposite side of the patient, an X-ray detector is provided that can in some way sense the position of the detected X-ray photon. The source and detection points thus uniquely define the single ray of interest.

In emission tomography, the situation is made much more complicated by the

fact that one does not a priori know the position of the source point. The two methods of emission tomography, PET and SPECT, differ in the methods used to define the ray direction. The scheme used for PET is illustrated in Fig. 1.10b. The technique is limited to those radioisotopes that decay by emitting a positron. This positron travels a short distance (a few millimetres at most) before stopping and annihilating with a normal negative electron from the tissue of the patient ($e^+ + e^-$). Annihilation radiation is formed at the site of this combination to carry away the lost mass energy. The unique property of this annihilation radiation is that it consists of two electromagnetic photons, each of energy 0.511 MeV, that are emitted in time coincidence and are oppositely directed along exactly the same line. Thus by placing two position-sensitive detectors on opposite sides of the patient, one can fully define the ray if the two detected events correspond to the photons from a single annihilation event.

The second general approach to emission tomography is the SPECT. It differs fundamentally from PET in that any radioisotope that emits decay gamma rays may be used as the basis for the imaging. In contrast with annihilation radiation, these gamma rays are emitted as single individual photons. Definition of the ray direction in SPECT requires spatial collimators that are opaque to the gamma rays as shown in Fig. 1.10c. These collimation methods, by their very nature, eliminate most of the gamma rays that would otherwise strike the detector, and allow through only those that are incident in the prescribed direction. Therefore, a detector records both the position and direction of each incident gamma-ray photon to fully utilize all the information.

In either case, single gamma or positron, the reconstruction problem can be related directly to Radon transform. Since the direction of photons is known, the data recorded by a detector aimed along the line defined by angle θ is just a measure of the total radionuclide concentration along that line. The use of ECT has provided a dramatic advancement in nuclear medicine including investigations into brain and

heart metabolisms. Other possibilities include new methods for cancer detection [16] - [19].

1.9 Medicine; Ultrasound Computed Tomography (UCT)

As in X-ray computed tomography, the ultrasound can be identified with the image rotation. The aim of ultrasonic computed tomography is to reconstruct transverse cross sectional images from projection data obtained when ultrasound passes through the object. Under appropriate conditions (straight path approximation), the ultrasound may be used to determine acoustic attenuation and acoustic velocity distributions in tissues. Thus, ultrasonic imaging based on processing of amplitude information and propagation velocity are given as follow :

i) Amplitude data

Under the most ideal conditions, the attenuation coefficient $\mu(x, y)$ for an ultrasonic pulse propagating along the s direction may be related to the pressure amplitude A_o at the transmitting transducer and the pressure amplitude $A(z, \theta)$ at the receiving transducer by [20] the same type of equation as used in Equ. (1.12).

$$A(z, \theta) = \int_L \mu(x, y) ds \quad (1.13)$$

Then if A is known for all values of z, θ , then A is the Radon transform of μ that μ can be reconstructed from A by applying the inverse Radon transform. This technique allows a tomogram of the acoustic attenuation of a material to be obtained. Images of this kind may be interpreted as maps of the viscosity of the material since it is the viscous nature of a material that is responsible for attenuating acoustic radiation.

ii) Propagation time data (Time-of-Flight) ToF

Another approach is the use of propagation speed (Time-of-Flight) data to

reconstruct acoustic velocity distributions. ToF is based on emitting short pulses of ultrasound and recording the time taken for each pulse to reach a detector.

If the material in which the pulse propagates is homogeneous, the ToF for the pulse to traverse the distance between source and detector along a line L, is given by :

$$t = \frac{L}{v} \quad (1.14)$$

Where:

v is the velocity at which the pulse propagates through the material.

If the material is inhomogeneous along L, then the time of flight becomes:

$$t(z, \theta) = \int_L \frac{ds}{v(x, y)} \quad (1.15)$$

here, a tomogram of the inhomogeneous velocity of the material can be found by inverting the above equation. This result is the basis for an imaging technique known as UCT [21] - [22].

In addition to performing ToF experiments, the relative index of refraction (n) can be measured. A normalized ToF projection is obtained as follows [23] :-

$$t(z, \theta) = \int_L \frac{ds}{v_o} - \int_L \frac{ds}{v(x, y)} \quad (1.16)$$

Where :

v_o is the propagation velocity in the water bath surrounding the specimen.

Since the relative index of reflection n is given by [24] :

$$n(x, y) = \frac{v_o}{v(x, y)} \quad (1.17)$$

It follows that :

$$v_o \cdot t(z, \theta) = \int_L [1 - n(x, y)] ds. \quad (1.18)$$

If $v_o.t(z, \theta)$ is measured for arbitrary θ and z , then it is clear how the image rotation emerges once again to obtain the relative index of refraction $n(x, y)$.

1.10 Diffraction Tomography

Diffraction tomography is a method of imaging which is in its early stages of development [25] - [28]. The basic idea is to reconstruct the internal structure of an inhomogeneous material by the way in which it diffracts radiation. In practice, there is one very important difference between projection and diffraction tomography: whereas projection tomography is based on an exact reconstruction, diffraction tomography relies heavily on approximate solution to the scattering problem [29]. In other words, the inversion algorithm that enables us to recover the structure of an object from a sequence of projections is exact whereas in diffraction tomography. The inversion algorithm is based on an approximate solution for diffracted field. Two methods have been researched to date using (i) continuous wave fields and (ii) pulsed fields. In the latter case, it has been shown [25] that the time history or the diffraction pattern set up by a short pulse of radiation is related to the internal structure of the diffracting object by the Radon transform Fig. 1.11.

When the source emits pulse $P(x)$, then its characteristic spectrum can be written as :

$$P(x) = \int_{-\infty}^{\infty} \exp(-ik\tau) p(\tau) d\tau \quad (1.19)$$

Where :

τ = time x a acoustic wave speed.

In this case, the scattered pressure field at r induced by a pulsed acoustic point source at r_i is determined by the wave equation :

$$(\Delta^2 + k^2 P_s(r, r_i, k)) = -k^2 P(k) \gamma_k(r) g(r|r_i, k) \quad (1.20)$$

Where :

$$\gamma_k = \frac{k - k_o}{k_o}$$

$$g(r|r_i, k) = g(r, r_i, x)$$

ρ_o density

k_o compressibility

γ_k the compressibility fluctuations.

Using the expression for the 2D Green's function, the solution to equation (1.20) at r_s is then :

$$P_s(r_s, r_i, k) = \frac{ik}{8\pi} \int_A \frac{\exp(ik|r - r_i|) \exp(ik|r - r_s|)}{\sqrt{(|r - r_i|)} \cdot \sqrt{(|r - r_s|)}} P(k) \cdot \gamma_k(r) d^2r \quad (1.21)$$

The solution of Equ. (1.21) is developed under the conditions:

$$\frac{r}{|r_i|} \ll 1$$

and

$$\frac{r}{|r_s|} \ll 1$$

Where $|r_i| \simeq |r_s| \simeq a$, being the distance of the source and receiver from the centre of the scatterer as shown in Fig. 1.11. These conditions allow to use a Fourier-based inversion scheme because the scattered pressure field can be written as :

$$P_s(\hat{N}, k) = \frac{ik}{8\pi a} P(k) \exp(2ika) \int_A \exp[-ik\hat{N} \cdot r] \gamma_k(r) d^2r \quad (1.22)$$

Where :

$$\hat{n}_i = -\frac{r_i}{|r_i|}$$

$$\hat{n}_s = \frac{r_s}{|r_s|}$$

and

$$\hat{N} = \hat{n}_s - \hat{n}_i$$

$\hat{n}_i$ and $\hat{n}_s$ represent the direction of the incident and scattered fields respectively.

The function P_s describes the spectrum of the scattered pressure field. In practice, the spectrum of this field is not measured directly. What is actually measured is a time varying voltage which is proportional to the time history of the scattered field. This is given by the inverse Fourier transform of P_s . Using the convolution theorem and remembering that ik in Fourier space yields a derivative in real space, we can write [25] :

$$\begin{aligned} P_s(\hat{N}, \tau) &= \frac{1}{(8\pi a)} \frac{\partial}{\partial \tau} (p(\tau) \otimes \delta(\tau + 2a) \otimes \int_A \gamma_k \delta(\tau - \hat{N} \cdot r) d^2 r) \\ &= \frac{1}{(8\pi a)} \frac{\partial}{\partial \tau} p(\tau + 2a) \otimes \int_A \gamma_k(r) \delta(\tau - \hat{N} \cdot r) d^2 r \end{aligned} \quad (1.23)$$

where:

$$p_s(\hat{N}, \tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P_s(\hat{N}, \tau) \exp(ik\tau) dk$$

Equation (1.23) is a rather nice result because the integral over γ_k is just the Radon transform. Therefore, in practice, the object can be reconstructed by employing image rotation algorithms.

1.11 Astronomy

1.11.1 Historical overview

Image reconstruction entered radio astronomy around 1950 in response to a need for high angular resolution in the mapping of radio emission from celestial objects. As a technique, reconstruction offered a way of economizing on the initial

financial cost of a radio telescope and paying instead in terms of observing time. Such a research strategy is standard in fundamental research where time and effort cleverly deployed by a scientist may yield basic discoveries sooner than a larger organized effort involving more finances, more engineering and more staff. Anyhow, such was the general pattern of discovery during the many years when the phenomena was being discovered on which radio astronomy is now based [30]. These introductory remarks are introduced to emphasize a distinction between what happen in radio astronomy and what happened in the development of instruments for medical application, even though the underlying mathematical theory is the same.

A second distinction should be made clear. In x-ray tomography the line integral is taken along the direction in which the x-radiation is travelling. In radio astronomy, on the other hand, the line of integration is perpendicular to the direction in which the radio waves travel.

1.11.2 Brightness distribution

Radar astronomy is used to obtain Radar maps of the moon and planetary surfaces in various ways. An especially interesting method that leads directly to an interpretation in terms of the Radon transform is described by Thomson and Ponsonby [31]. The basic idea is to measure separately enough components of the Fourier transform $\tilde{f}$ or the Radon transform $\hat{f}$ to be able to construct the desired brightness distribution f . To do this, consider the coordinate system is associated with the lunar disk, as in fig.1.12. Furthermore, let the earth-based radar be in motion relative to this coordinate frame with the transverse velocity V . The reflected signal received by the radar from points on the lunar surface along a line of constant p has a Doppler shifted frequency exceeding that from points along the line with $p=0$ by approximately

$$v = \frac{2Vp}{\lambda}$$

where:

λ is the wave length. let $f(x, y)$ be the radar brightness distribution normalized so that

$$\int \int_{moon} f(x, y) dx dy = 1$$

and let $\hat{f}(p, \phi)$ be the line integral (Radon transform) of $f(x, y)$ along the line defined by

$$x \cos \phi + y \sin \phi = p$$

The measured power spectrum of the received signals is normalized so that

$$\int_{all \ v} S(v, \phi) dv = 1$$

The important result obtained by Thomson and Ponsoby is that

$$\hat{f}(p, \phi) = \frac{2V}{\lambda} S\left(\frac{2V}{\lambda} P, \phi\right)$$

using the appropriate scaling, $\hat{f}$ can be computed from the measured power spectrum.

1.12 Nuclear Magnetic Resonance (NMR)

Another important situation where the image rotation problem arises is in magnetic resonance imaging (MRI). Being noninvasive, it is becoming increasingly attractive in medical imaging for measuring (most commonly) the density of protons (that is, hydrogen nuclei) in tissue. This imaging technique is based on the fundamental property that protons (and all other nuclei that have an odd number of protons or neutrons) possess a magnetic moment and spin. When placed in a magnetic field, the proton precesses about the magnetic field in a manner analogous to a top spinning about the earth's gravitational field. Initially the protons are aligned either parallel or antiparallel to the magnetic field. When An RF signal having an appropriate strength and frequency is applied to the object, The protons absorb energy, and more of them switch to the antiparallel state. When the

applied RF signal is removed, the absorbed energy is reemitted and is detected by an RF receiver. The proton density and environment can be determined from the characteristics of this detected signal. By controlling the applied RF signal and the surrounding magnetic field, these events can be made to occur along only one line within the object. The detected signal is then a function of the line integral of the MRI signal in the object. From such a view point, Radon inversion formula can be applied to reconstruct the object[32].

The basic hardware elements of an NMR imager are shown in Figure 1.13. These are as follow [33]:

1. A magnet which provides a strong, uniform, steady, magnetic field.
2. A Radio Frequency (RF) transmitter, which delivers RF magnetic fields to the sample.
3. A gradient system, which produces time-varying magnetic fields of controlled spatial nonuniformity.
4. A detection (receiver) system, which yields the output signal.
5. An imager system, including the computer, which reconstructs and displays the images.

The method of projection reconstruction imaging which illustrates the concept of spin density projection is of historical importance since Lauterbur [34] used it to produce the first NMR image (see Fig.1.14). In Figure 1.14, a one-dimensional projection can be obtained by recording the NMR spectrum in the presence of a linear magnetic field gradient as shown in Fig. 1.13. Multiple projections are obtained by changing the relative orientation of the gradient, and the data manipulated in a computer via a standard projection reconstruction algorithm to give an image resolved in two-dimensions.

1.13 Optics

In optics, as in several other fields a concern here is with the way in which the Radon transform emerges in the problem of determining the refractive index rather than the way in which the refractive index relates to various properties and media mentioned earlier. Optical interferometry provides a method for measuring optical path length differences in optically transparent objects or media. Holographic interferometry, which is the interferometric comparison of two or more waves, provides a convenient and very accurate means for measuring optical path length differences [35]. Such measurements are related to the refractive index of the media and used to obtain information about internal properties of the media. these properties and media include mass density of fluids and gases, chemical species concentrations in reading gases and state of stress of transparent solids.

Cheng [36] obtained the Fourier coefficients of a near object by analyzing the interference pattern in an achromatic grating interferometer. As a result, the Fourier plane is filled and the spatial irradiance distribution of a 2D incoherent object is reconstructed by using the projection slice theorem. Figure 1.15 shows the achromatic grating interferometer in which all grating are perpendicular to the z -axis with their line structure parallel to the y -axis. The illuminating source is situated in the $\xi - \eta$ plane at (ξ_o, η_o) . The $x - y$ coordinates are a version of $\xi - \eta$ coordinates rotated by an angle σ about the optical axis (z -axis). Thus, the Fourier transform of the intensity distribution of the illuminating source $s(\xi, \eta)$ on radial lines (Fig. 1.16) can be obtained by rotating the grating interferometer about the optical axis by angles $\sigma_1, \sigma_2, \sigma_3, \dots$ and by completing the measurements for various angle σ . Filling the frequency plane (Fig.1.16, the 2D object can be reconstructed using the well known technique in computed tomography (inverse Radon transform).

The important optical applications were primarily for radially symmetric media and thus involved inversion of Abel's transform. As we shall see in Chapter 5, the Radon transform reduces to the Abel transform under appropriate symmetry conditions.

1.14 Geophysics

Other area in which the Radon transform and reconstruction method has been found useful is in geophysics. Recent research in theoretical seismology has been resulted in effective techniques for computing synthetic seismograms. Chapman[37] has developed a new way to compute these synthetic seismograms ((some aspects of earth structure) that makes use of Radon transform standard methods are used for the computation up to the point of evaluating certain integrals over frequency and wave slowness (reciprocal velocity). Chapman's technique of evaluation the frequency integral first and keeping the final wave slowness integral real is new and gives rise to an interpretation of each seismogram as an integrated cross section of a density function at a given angle. The inversion of the Radon transform leads directly to the concept of the velocity stack (slant stack) of the seismograms.

1.15 Military Applications

The idea of combining the clinical procedure known as reconstruction tomography and the radar processing for linear FM pulse compression is explained by Bernfeld[38], in order to map radar backscatter energy. The idea is conceived of combining the clinical procedure and CHIRP pulse compression in a system called CHIRP Doppler radar. The role intended for the new system is to map radar backscatter energy as a function of range and Doppler. Bernfeld has explained how to recover the backscatter energy using Radon inversion formula.

There are many similarities between image reconstruction algorithms and Synthetic Aperture Radar (SAR), SAR is a known technique for creating radar imagery. Of special interest are the surface mapping radars, such as the Seasat and the Shuttle imaging radars, which utilize a technique known as SAR to create high resolution images (pictures).

the data collected by a radar system is the echo signal. The value of an echo

signal at a time delay τ represents the average (integrated) reflectance of all targets in the radar illumination at a range $r = c\tau/2$, where c is the speed of light and the division by 2 accounts for the round trip propagation. If the reflectance observed by a sensor at azimuthal position x is considered as a 2D function of range and ground angle $\sigma(r, \theta)$, then the received echo can be written

$$s_x(r) = \int_{beam} \sigma(r, \theta) d\theta \quad (1.24)$$

The path of integration is illustrated in Figure 1.17. Estimation of the original reflectivity distribution from the echo measurements is accomplished using synthetic aperture techniques. A projection, which is similar to a radar echo, represents the average (integrated) transmission through a sample. Illumination of the sample is usually performed with nondiffracting sources, such as x-rays, so that the projections will be along straight lines as in Figure 1.18. Typically a set of parallel projections are collected by physically scanning the transmitter/receiver pair in a plane or by using planar illumination together with a sensor array as in Figure 1.18. The equation describing the linear attenuation of the illumination as it propagates through the object is given by [39]

$$p_\theta(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sigma(x, y) \delta(x \cos \theta + y \sin \theta - t) dx dy \quad (1.25)$$

where

$\sigma(x, y)$ refer to as the transmission function in optical applications.

Here again, the function $p_\theta(t)$ is called the parallel projection or Radon transform. Recalling Radon inversion formula, we can reconstruct (estimate) the transmission attenuation function.

1.16 Material science and others

Willis [40] made extensive use of the Radon transform of the traction vector in developing a method for determining the stresses in a composite body consisting

of dissimilar isotropic elastic half spaces bonded over a circular region. The basic idea is formulating an integral equation for the Radon transform of the relative displacement of the faces of the crack.

Other work in the area of nondestructive testing (NDT) is discussed by Ellingson and Berger [41] and Azevedo [42].

Other areas in which the Radon transform and reconstruction methods have been found useful in including air pollutant studies, pattern recognition, communications and applications of a purely mathematical nature.

1.17 Dissertation overview

The research results described in this manuscript represent the most recent work in both computed tomography for image reconstruction and multidimensional signal processing. We emphasize the real-time computational problem of the computed tomography. In this manuscript, we will summarize the work presented that will render a greater understanding of the image reconstruction problem.

Computer algorithms and architectures for fast implementation of CT algorithms are an important continuing area of research. As the reconstruction algorithms for CT have become more sophisticated over the years, the demands on computer systems have become correspondingly greater. The speed of reconstruction becomes increasingly important as we move toward real-time tomography.

The first goal of our research is to assess the applications of the Radon transform for image processing. Since the appropriate unifying mathematical framework for a large class of reconstruction problems is the Radon transform. It has a broad spectrum of applications such as industrial computer vision, medical imaging, diffraction tomography, astronomy, optics, nuclear magnetic resonance, geophysics, material science, molecular biology and military applications.

The second goal is to find out the relationship between image rotation and the reconstruction process, since the approach adopted in this research to computing the Radon transform is based on rotating the object about its centre and taking projections at regular steps in the angle of rotation. As a result, a new method for digital image rotation has been derived that can rapidly reduce the reconstruction time.

The third goal of our research is to investigate how the quality of the reconstructed image is influenced by rotation that occurs in computing the Radon transform and its inverse, and how this rotation time can be minimized during the reconstruction process. A numerical study has been carried out to test the relationship between different methods of image rotation and their effect on the performance for computing the Radon transform. The results show that the optimum algorithm is based on the optimum nearest neighbour method of image rotation to recover the object by using the filtered back projection (1D inverse filter). As a result, the new technique for image reconstruction offered four advantages over the current technique:

- 1- It provides better resolution of the reconstructed image (by visual inspection).
- 2- It drastically reduces the CPU time for image reconstruction algorithm.
- 3- It gives minimum least square error compared to the current algorithm.
- 4- It gives minimum projection square error along the image axis compared to the current algorithm.

The forth goal of our research is to combine the aforementioned features in a mathematical formula to describe the system performance. This formula is called the computation energy. A quantitative study has been carried out to compute the computation energy for each reconstruction technique using different rotation algorithm. The results show that the new technique saved computation energy by 54.04 % compared to the current one.

Another goal of our research is to derive the mathematical formulation of the reconstruction using central-slice theorem. Since we have proved that the mathematical formula should include the 2D inverse filter as described in Section 5.3.2 and Section 8.6.

The final research has been undertaken in assessing the Radon transform that can be naturally applied in other areas of multidimensional signal processing. For example, the median filtering operations have used the dimension-reducing property of the Radon transform to perform the simpler 1D median filter on each projection independently, thereby allowing computational parallelism; i.e., rather than computing a 2D median filter of an image, we have computed a number of 1D median filter independently.

1.18 References

- [1] Radon, J., 1917 "Über die Bestimmung von Funktionen durch ihre Integralwerte Langs gewisser Mannigfaltigkeiten, Berichte Sachsische Akademie der wissenschaften", Leipzig, Math.-phys. Kl., 69, 262-267. (N.B. a translation of this paper is given in [2]).
- [2] Deans, S. R., The Radon Transform and some of its Applications, Wiley-Interscience, 1983.
- [3] John, F. Jarvis, " Research Directions in industrial machine vision : A workshop summary ", proc. IEEE, Dec. 1982, pp. 55-61.
- [4] Rafael, C. Ganzalez and Reza Safabakhsh, " Computer Vision Techniques for Industrial Applications and Robot Control ", proc. IEEE, Dec. 1982, pp. 17-22.
- [5] King-sun Fu, " Pattern recognition for automatic visual inspection ", proc. IEEE, Dec. 1982, pp. 34-40.

- [6] Hounsfield, G. N., "Computerized transverse axial scanning (tomography: part 1: Description of system", *Brit. J. Radiology*, 46:1016-1022, 1973.
- [7] Segre, E. (1977). *Nuclei and practices*, Benjamin, Reading, MA.
- [8] Jamon Inouye, " Image reconstruction with limited angle projection data ", *IEEE Trans. Nuclear Science*, Vol. NS-26, No. 2, April 1979.
- [9] R. A. Robb, E. A. Hoffman, L. J. Sinak, L. D. Harris, and E. L. Ritman, " High-Speed Three-Dimensional X-Ray Computed Tomography : The dynamic spatial reconstructor ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [10] L. Axel, P. H. Argee and R. A. Zimmerman, " Applications of computerized tomography to diagnostic radiology ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [11] Albert Macovskj, " Physical problems of computerized tomography ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [12] Jayaram K. Udupa, " Display of 3D information in discrete 3D scenes produced by computerized tomography ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [13] P. Bloch and J. K. Udupa, " Application of computerized tomography to radiation therapy and surgical planning ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [14] K. C. Tam, J. W. Eberhard and K. W. Mitchell, " Incomplete-Data CT Image Reconstructions in Industrial Applications ", *IEEE Trans. Nuclear Science*, Vol. 37, No. 3, June 1990.
- [15] L. A. Shepp and J. B. Kruskal, " Computerized tomography: the new medical X-ray technology ", *American math. monthly*, 85, 420-439, June-July 1978.
- [16] P. E. Kinaham, and et al, " Three-dimensional image reconstruction in object space ", *IEEE Trans. Nuclear Science*, Vol. 35, No. 1, Feb. 1988.

- [17] G. F. Knoll, " Single-photon Emission computed tomography ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [18] P. E. Kinaham and J. G. Rogers, " Analytic 3D image reconstruction using all detected events ", *IEEE Trans. Nuclear Science*, Vol. 36, No. 1, Feb. 1989.
- [19] D. W. Townsend, and et al, " Three dimensional reconstruction of PET data from a multi-ring camera ", *IEEE Trans. Nuclear Science*, Vol. 36, No. 1, Feb. 1989.
- [20] Green J. F., S. A. Johnson, and A. H. Lent " Measurement of spatial distribution of refractive index in tissues by Ultrasonics computer assisted tomography ", *Ultrasound in med.& Biol.*, 3, pp. 327-339, 1978.
- [21] Gary H. G. and J. C. Sharp, " Reconstruction of Ultrasound propagation speed distribution in soft tissue: Time-of-Flight tomography ", *IEEE Trans. Sonics and Ultrasonics*, Vol. 5U-24, No. 4, July 1977.
- [22] James F. Green Leaf, " Computerized Tomography with ultrasound ", *proc. IEEE*, Vol. 71, No. 3, March 1983.
- [23] Kak A. C., C. V. Jakowatz, N. A. Baily, and R. A. Keller, " computerized tomography using video recorded fluoroscopic images. ", *IEEE Trans. Biomed. Eng.*, BME-24, pp. 157-169, 1977.
- [24] Kak A. C., " Computerized tomography with x-ray, emission, and ultrasound sources. ", *proc. IEEE*, 67, pp. 1245-1272, 1979.
- [25] Blackledge J. M., " Quantitative coherent imaging ", *Academic press*, 1989.
- [26] Blackledge J. M., R. E. Burge, K. I. Hopcraft and R. J. Wombell, " quantitative diffraction tomography: I. pulsed acoustic fields ", *J. phys. D: Appl. phys.*, Vol. 20, pp. 1-10, 1987.
- [27] Blackledge J. M., R. E. Burge, K. I. Hopcraft and R. J. Wombell, " quantitative

- diffraction tomography: II. pulsed elastic waves ", J. phys. D: Appl. phys., Vol. 20, pp. 11-17, 1987.
- [28] Devaney A. J., " Diffraction Tomography ", Inverse Methods in electromagnetic imaging, Reidel, 1985.
 - [29] Taylor J. G., Quantum Mechanics (An Introduction) George Allen and Unwin, 1970. Chapter 6 provides a good introduction to quantum Scattering theory using the Born approximation.
 - [30] Hey, J. S., " The evolution of radio astronomy", science history publications, New York, 1973.
 - [31] Thomson, J. H. and J. E. B. Ponsonby, "Two-dimensional aperture synthesis in lunar radar astronomy", Proc.R.Soc., London, A303, pp. 477-491, 1968.
 - [32] Shepp, L. A., "Computerized tomography and nuclear magnetic resonance", J.computer assist. tomog., No. 4, pp. 94-107, 1980.
 - [33] Willim P. Rothwell, "Nuclear magnetic resonance imaging", Applied optics, vol.24, No.23, pp.3958-3968, 1 Dec. 1985.
 - [34] Lauterbur, P. C., "Image formation by induced local interactions: Examples employing nuclear magnetic resonance", Nature London 242, 190, 1973.
 - [35] Vest, C. M., "Holographic interferometry", 1979, Wiley, New York.
 - [36] Cheng, Yih-shyang, "Two-dimensional grating interferometric imaging of a near object br computed tomography", Applied optics, vol.30, No.11, pp. 1332-1334 , 10 April 1991.
 - [37] Chapman, C. H., "On impulsive wave propagation in a spherically symmetrical model", Geophys. J. R. Astron. Soc., 58 , pp. 229-234, 1979.
 - [38] Bernfeld, M. "CHIRP Doppler Radar", Proc. IEEE, vol. 72, No. 4, April 1984.

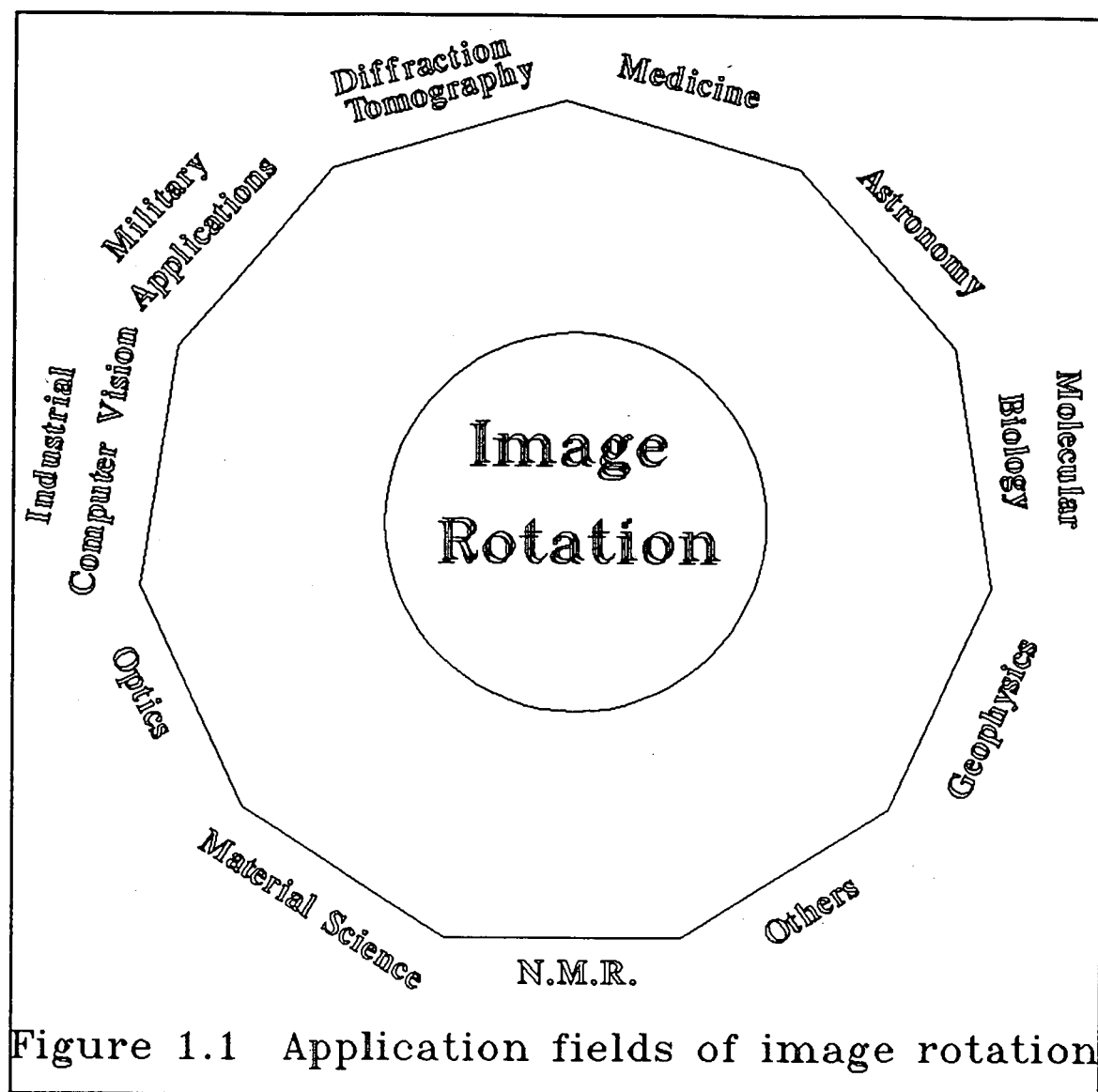


Figure 1.1 Application fields of image rotation

- [39] Fitch, J. P., "Synthetic Aperture Radar", Springer-verlag, New York, 1988.
- [40] Willis, J. R., "Interfacial stresses induced by arbitrary loading of dissimilar elastic half-spaces joined over a circular region", J. Int. Maths. Applics., 7, pp. 179-197, 1971.
- [41] Ellingson, W. A. and H. Berger, "Three-dimensional radiographic imaging" , in vol. 4 of research techniques in nondestructive testing, R. S. Sharpe, Ed., Academic, New York, pp. 1-38, 1980.
- [42] Azevedo, S. G., "Model-based computed tomography for nondestructive evaluation", Ph.D. thesis, University of California, Livermore, California, March 1991.

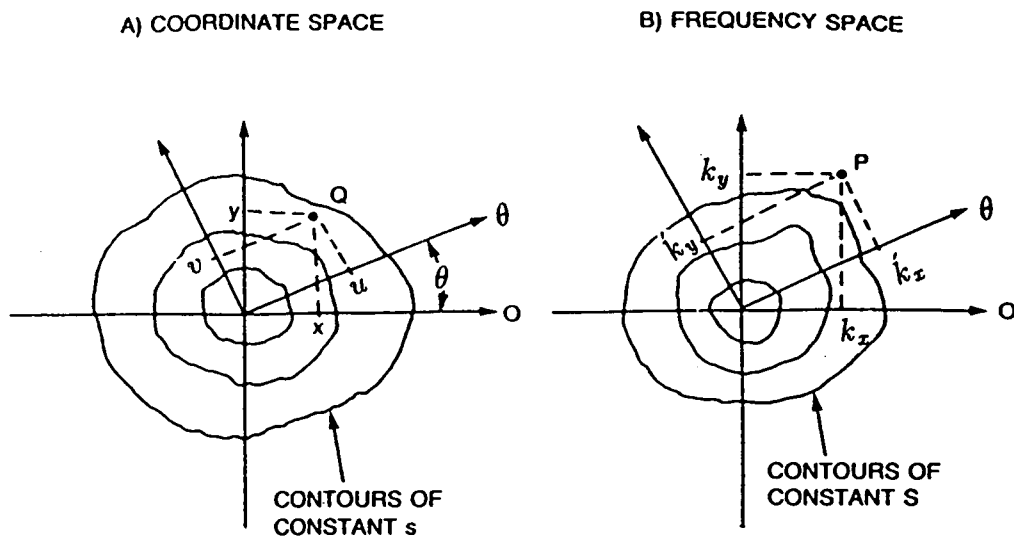


Figure 1.2 The coordinate system in both (x, y) space and frequency space are rotated by angle θ .

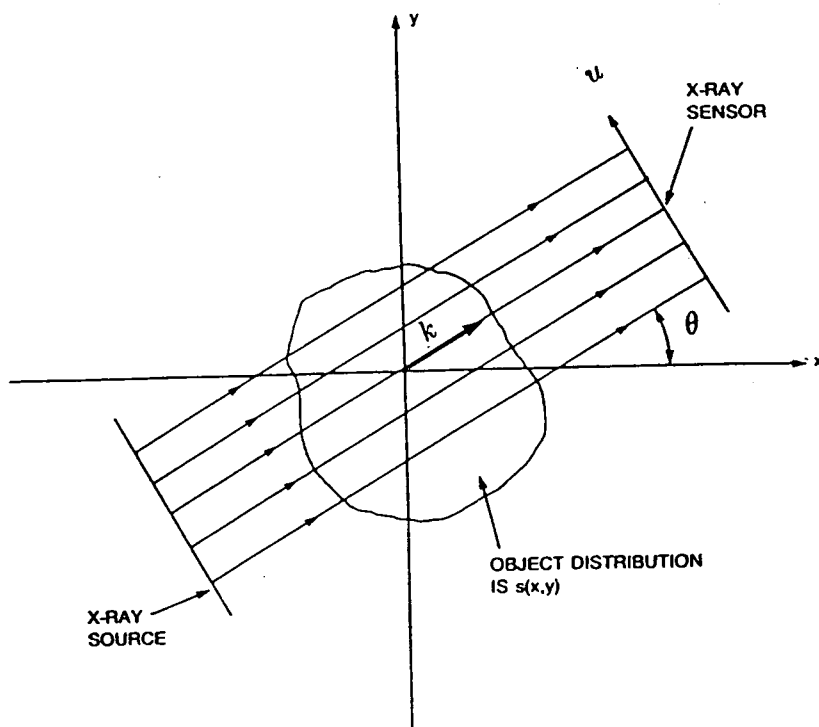


Figure 1.3 Geometry for tomography.

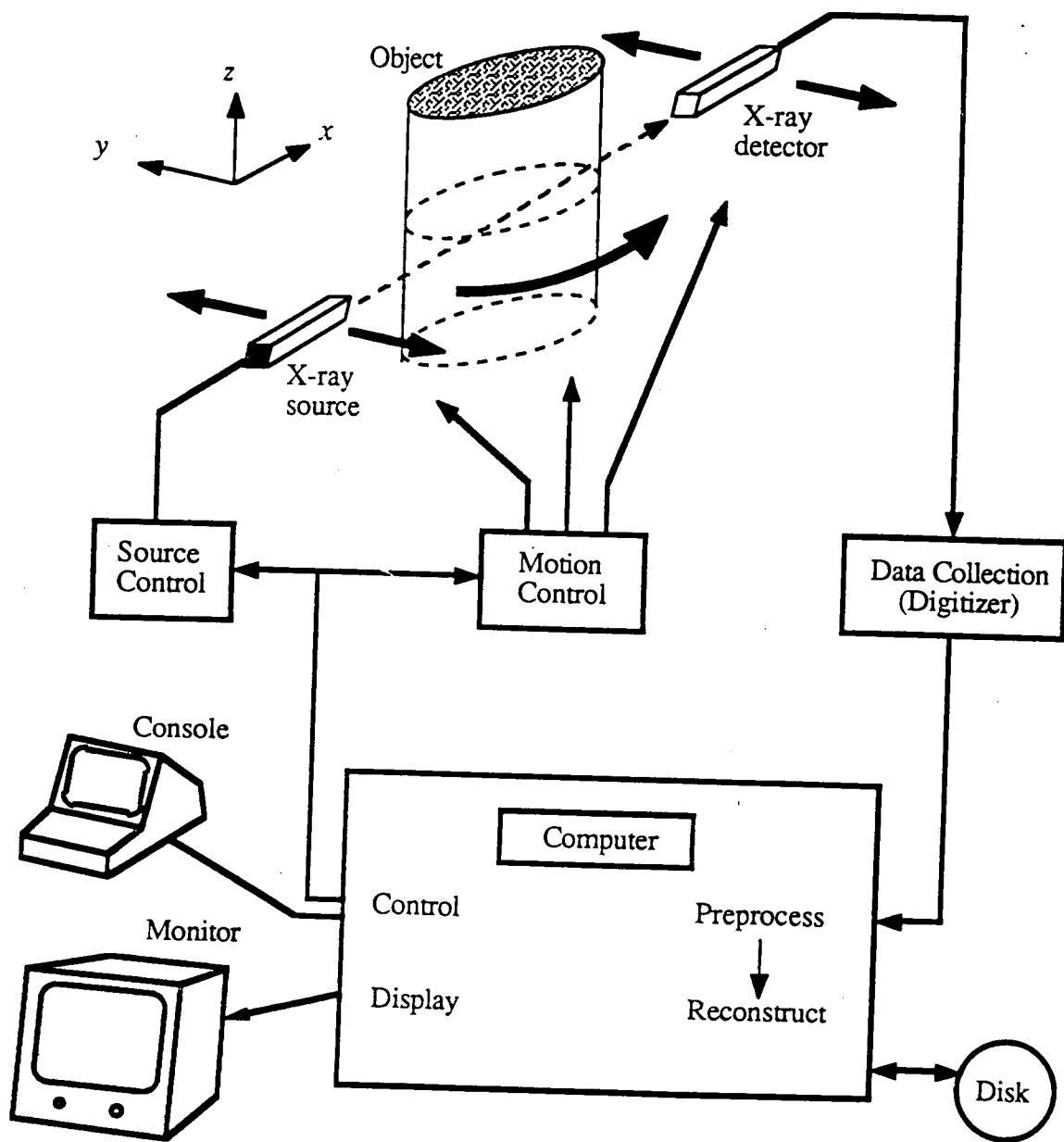


Figure 1.6 Common first-generation CT system design.

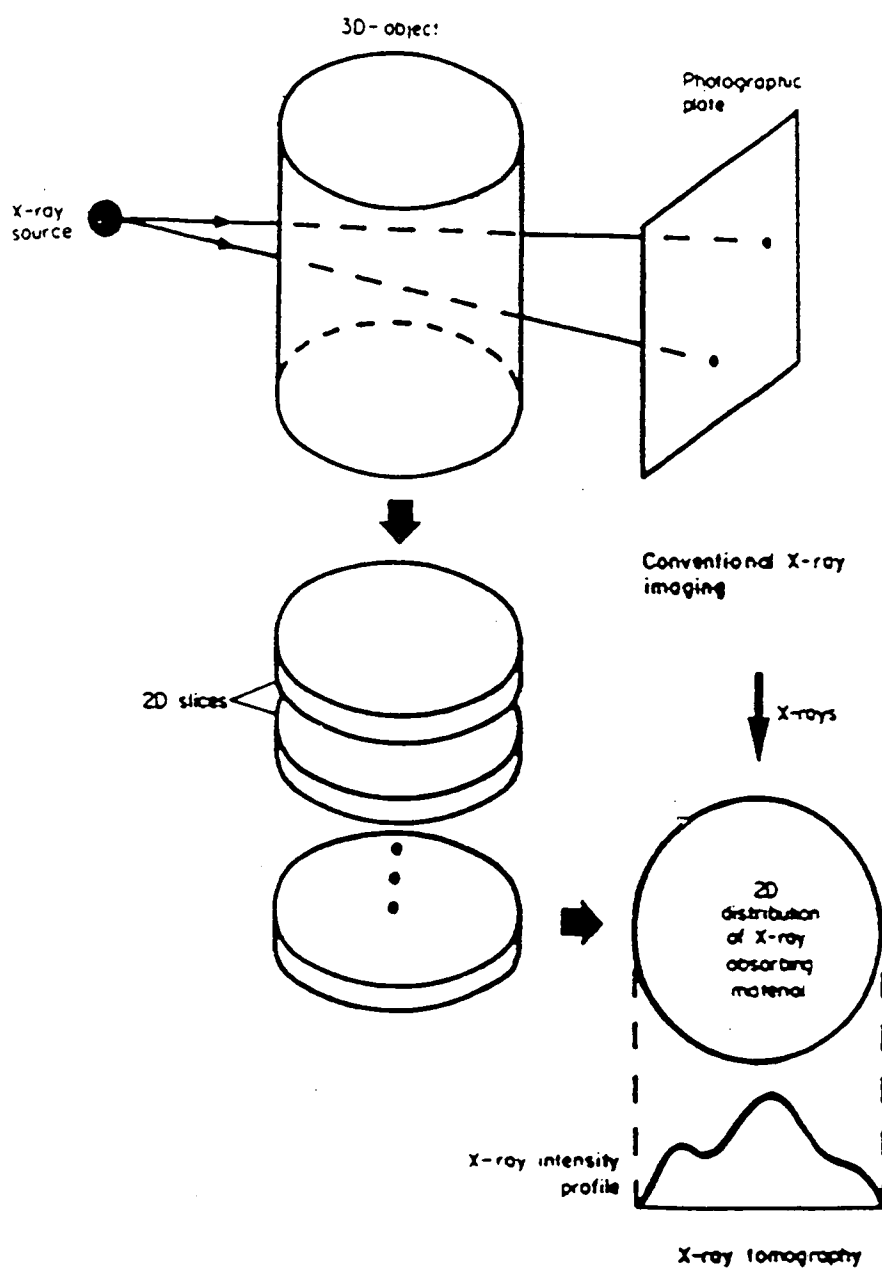


Figure 1.7 3D object composed of a stack of slices, the distribution of material in a single slice can be obtained by measuring x-ray intensity profile produced at different angles.

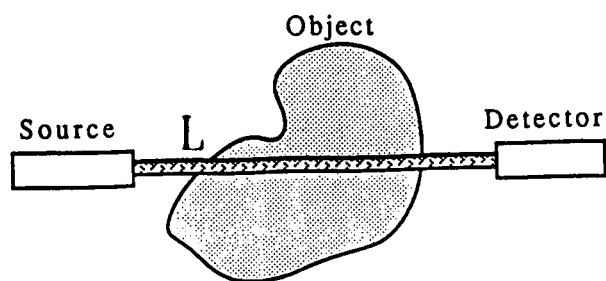


Figure 1.8 Definition of a nondiffracting tomographic projection.

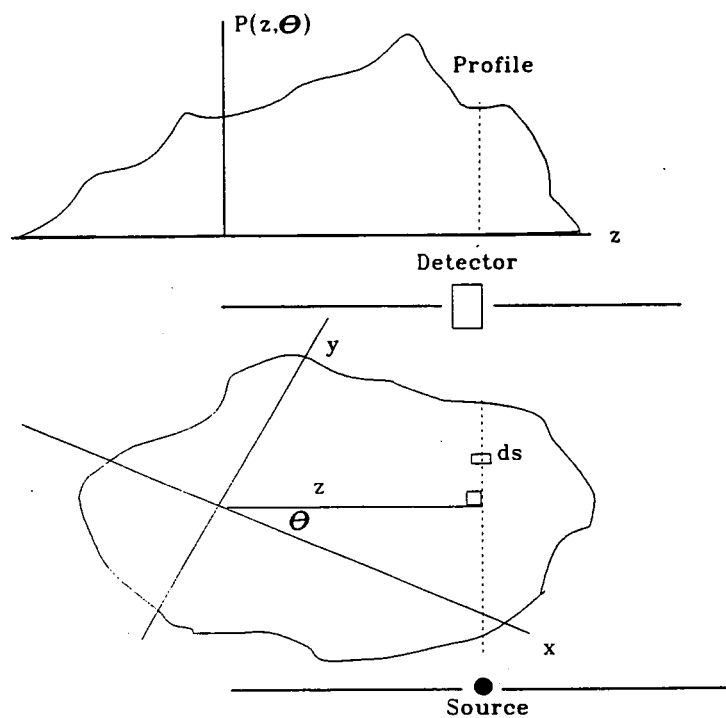


Figure 1.9 The source and detector move together (z varies) so that the beam covers the entire region leading to the profile $P(z, \theta)$ for fixed angle θ .

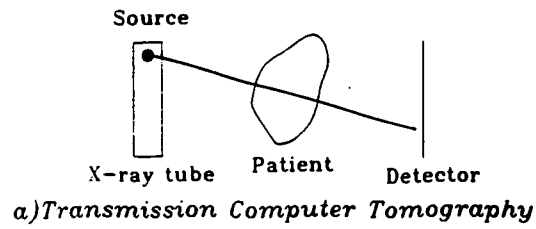
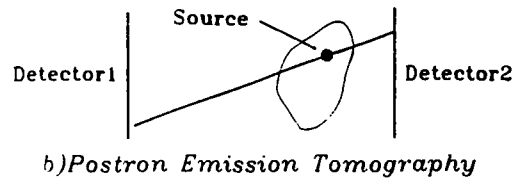
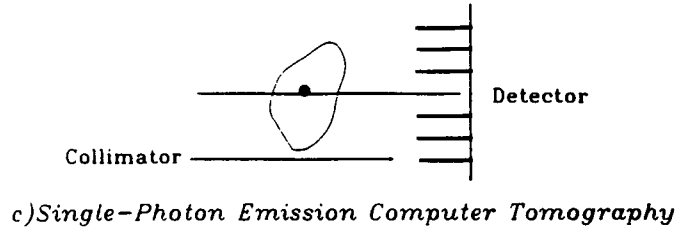


Figure 1.10 Methods of defining the ray direction for three different types of tomography.

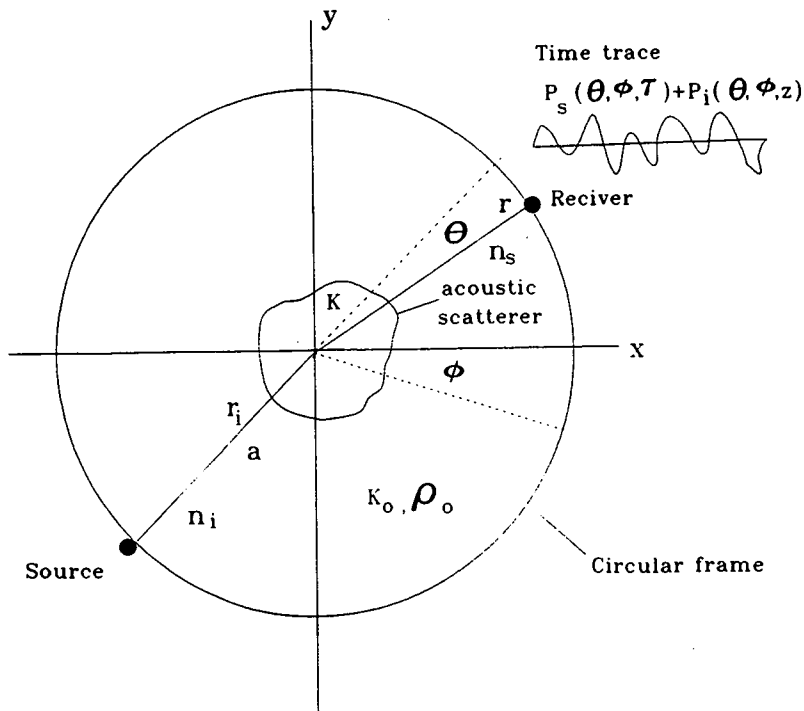


Figure 1.11 Illustration of the diffraction tomography using pulsed fields.

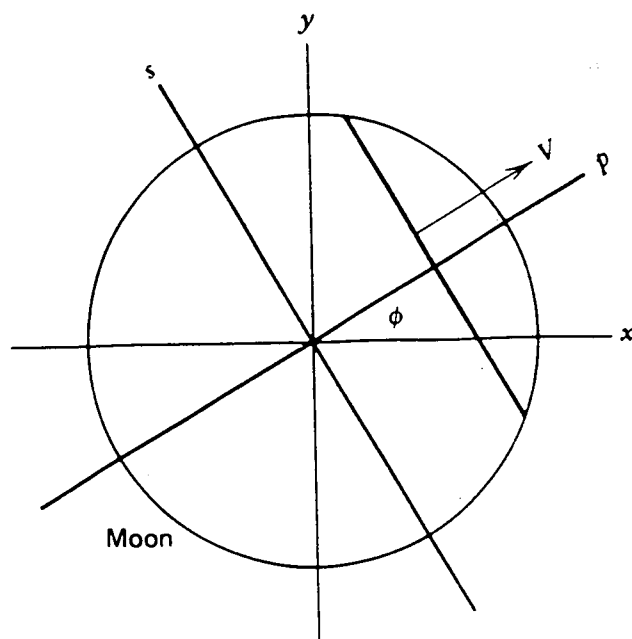


Figure 1.12 Coordinate system associated with lunar disk and the transverse velocity V along the p axis.

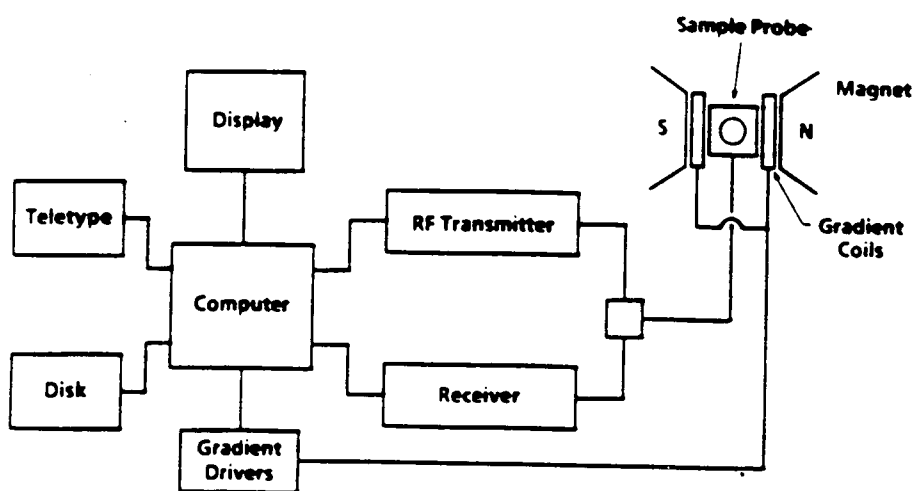


Figure 1.13 Basic schematic of the important hardware on an NMR imaging system.

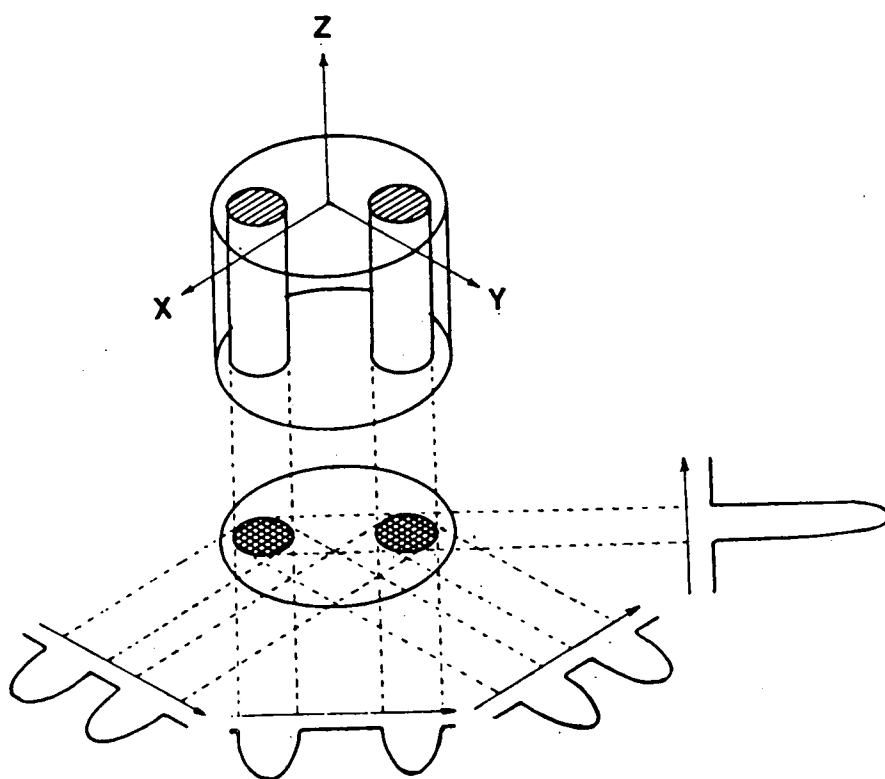


Figure 1.14 A 2D NMR image is reconstructed from a series of 1D projections obtained by recording NMR signals in gradients directed at different orientations relative to the sample.

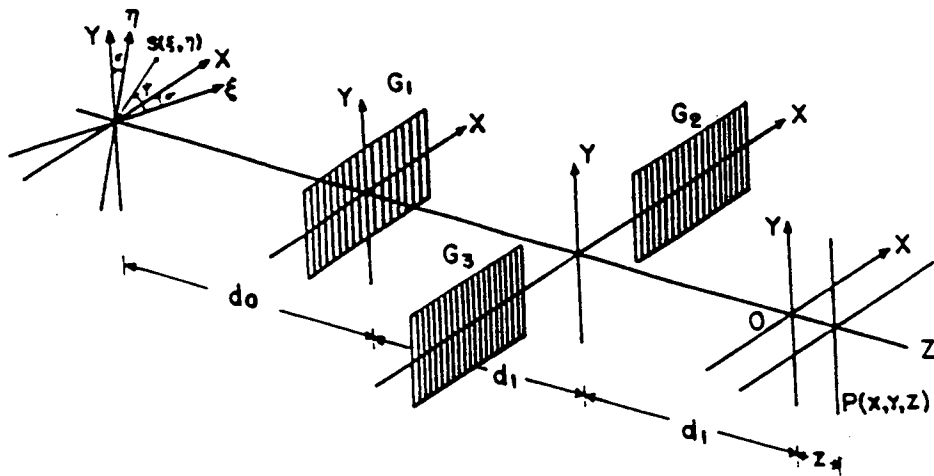


Figure 1.15 Achromatic grating interferometer.

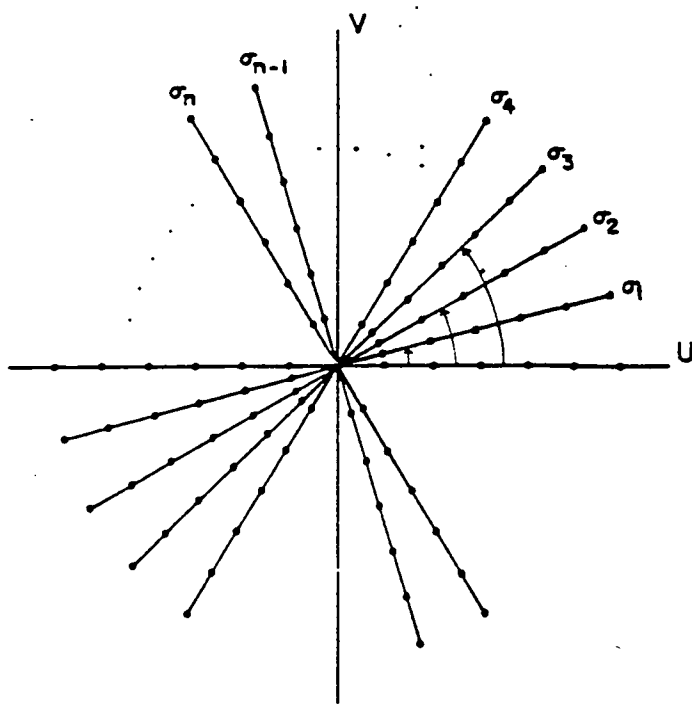


Figure 1.16 Filling the frequency domain by taking measurements at different angles.

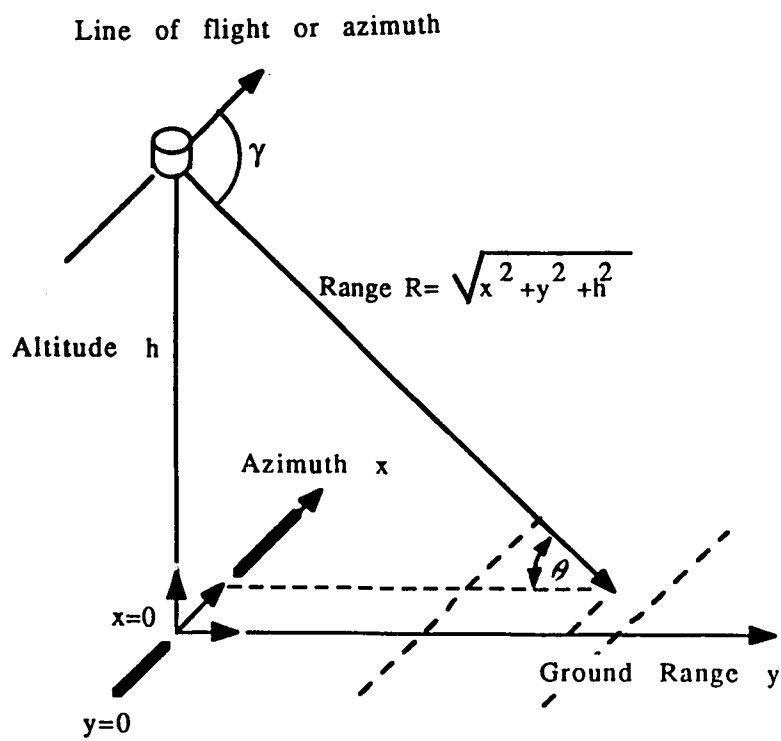


Figure 1.17 Radar geometry as function of range and ground angle.

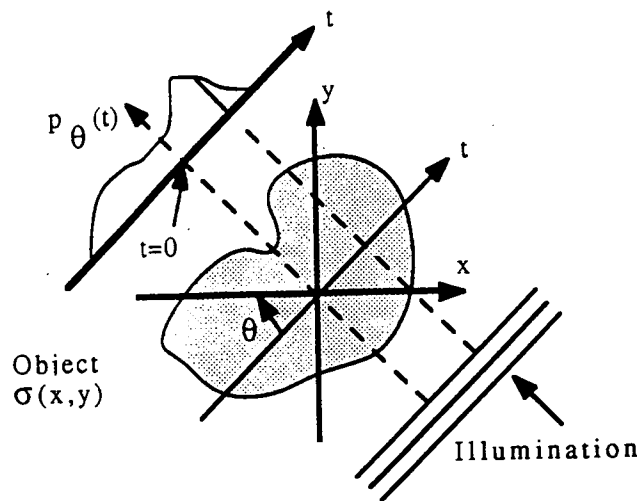


Figure 1.18 Typical geometry for a tomographic system.

CHAPTER 2

Theory of Digital Image Rotation

2.1 Introduction

Digital image rotation is one of the fundamental functions in the image processing and is applied widely to various problems such as medical and industrial computer vision. The image rotation in the digital image processing requires a long time when the object images are increased. From such a view point, it is requested to speed up the processing to reduce the processing time.

A concern in image rotation comes up when rotating an image by an angle that is not a multiple of 90° . The 90° rotation maps pixels one for one from the input to output image. This is also holds true for 180° and 270° rotation. When rotating through an angle that is not a multiple of 90° , several ways are discussed to handle the problem. Before going in details, let us discuss some basic relationship between pixels.

2.2 Some Basic Relation Ships Between Pixels

In this section we consider several primitives, but important relationships between pixels in a digital image. Consider an image is denoted by $a(k_y, k_x)$.

2.2.1 Four-Neighbours of a pixel

A pixel P at coordinates (k_y, k_x) has four-neighbours whose coordinates are given by :

$$(k_y + 1, k_x), (k_y - 1, k_x), (k_y, k_x + 1), (k_y, k_x - 1)$$

It is noted that each of these pixels is a unit distance from (k_y, k_x) and also that some neighbours of P will be outside the digital image if (k_y, k_x) is on the border of the image as shown in Figure 2.1.

2.2.2 Eight-Neighbours of a pixel

The four diagonal neighbours of a pixel P have coordinates $(k_y + 1, k_x + 1), (k_y + 1, k_x - 1), (k_y - 1, k_x + 1), (k_y - 1, k_x - 1)$.

These points, together with the four-neighbours defined above, are called 8-neighbours of P . As before, some of points will be outside the image if (k_y, k_x) is on the border of the image, as shown in Figure 2.1.

2.3 Basic Problem

Digital images are represented using a Cartesian mesh or grid where each node of the grid contains an individual pixel value. Image rotation ideally requires a Polar grid coordinate system. The problem is how to represent accurately a point with polar coordinates using a **discrete** Cartesian system.

2.3.1 Rotation of axes

The original system is shown in Figure 2.2 with solid lines, and the new system with equi-spaced dashed lines; the systems have common origin. The new axes are derived by rotating the old ones through an angle θ radians anti-clockwise about the origin (This is the usual mathematical way of measuring angles). If the point P in Figure 2.2 has coordinates (x, y) relative to the old system and (u, v) relative

to the new, then we have the relationships

$$\begin{aligned}
 u &= OU \\
 &= O\acute{B} + \acute{B}U \\
 &= A\acute{A} + \acute{P}P \\
 &= OA \sin \theta + AP \cos \theta \\
 &= OA \sin \theta + OB \cos \theta \\
 &= y \sin \theta + x \cos \theta
 \end{aligned}$$

$$\begin{aligned}
 v &= OV \\
 &= \acute{A}O - \acute{A}V \\
 &= A\acute{B} - A\acute{P} \\
 &= OA \cos \theta - AP \sin \theta \\
 &= OA \cos \theta - OB \sin \theta \\
 &= y \cos \theta - x \sin \theta
 \end{aligned}$$

and the rotation matrix R is

$$R = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

2.3.2 Basic Mathematical Expressions

The mathematical expression for rotation can be classified into two categories according to their pixel mapping:

1- The coordinate transformation equations using forward pixel mapping are:

$$u = x \cos \theta + y \sin \theta \quad (2.1)$$

$$v = y \cos \theta - x \sin \theta$$

where,

x, y are the input pixel coordinates (**integer coordinates**)

u, v are the new output pixel locations (**floating point coordinates**)

The output pixel coordinates are calculated for each possible input coordinate.

2- The coordinate transformation equations using reverse pixel mapping :

For every transformation there is an inverse transformation which will restore the coordinates of a point to their original value, therefore,

$$x = u \cos \theta - v \sin \theta \quad (2.2)$$

$$y = u \sin \theta + v \cos \theta$$

where:

x, y are the input pixel coordinates (**floating point coordinates**)

u, v are the new output pixel locations (**integer coordinates**)

The input pixel coordinates are calculated for each possible output coordinate.

2.3.3 The problem

How to represent the non-integer (u, v) coordinates on a discrete integer coordinate grid (see Figure 2.3). All solutions to this problem are based on some type of interpolation scheme.

a)-Forward Pixel Mapping Problem

Given (x, y) represent (u, v) .

b)-Reverse Pixel mapping Problem

Given (u, v) represent (x, y) .

2.4 Techniques for mapping (u, v) onto a discrete Cartesian grid

The rotation techniques can be classified as follows:

2.4.1 Nearest Neighbour Techniques

Nearest neighbour techniques are the simplest techniques for image rotation. Simply, these methods are based on the distribution of the approximated pixel location over a wider section of the grid.

1. Nearest Neighbour

Take coordinates of rotated image to be $nint(u), nint(v)$.

2. Four Nearest Neighbours

Assign value of pixel at coordinate $nint(u), nint(v)$ to the four nearest pixels (nearest neighbours).

3. Eight Nearest Neighbours

Assign value of pixel at coordinate $nint(u), nint(v)$ to the eight nearest pixels.

Examples of applying these techniques for digital image given in Figure 2.4 are shown in Figure 2.5 through Figure 2.7 at 70° rotation respectively.

Although nearest neighbour techniques are really simple to implement and of lower CPU time, these methods often have the drawback of producing undesirable artifacts, such as distortion of straight edges in images of fine resolution. Smoother results can be obtained by applying more sophisticated techniques.

2.4.2 Interpolation methods

These techniques form a weighted average of the intensities associated with areas or distances formed between the rotated output pixel and its four nearest neighbours. From such a view point, these techniques can be classified as follows:

1. Bilinear or Rectangular Interpolation

Forms a weighted average of the intensities associated with areas of the rectangles formed between the rotated output pixel and its four nearest neighbours (Figure 2.8a).

Let:

$$k_y = \text{int}(v)$$

$$k_x = \text{int}(u)$$

The fractional part of the new point are:

$$f_x = u - k_x$$

$$f_y = v - k_y$$

The rectangle areas for each data points are:

$$k(0,0) = (1 - f_x)(1 - f_y)$$

$$k(0,1) = f_x(1 - f_y) \tag{2.3}$$

$$k(1,1) = f_x f_y$$

$$k(1,0) = (1 - f_x)f_y$$

The intensity of the output pixel can be defined as:

$$b(v, u) = \sum_{L=0}^1 \sum_{n=0}^1 a(k_y + L, k_x + n)k(L, n) \tag{2.4}$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$\begin{aligned} m_N &= 8N^2 \\ ad_N &= 3N^2 \\ s_N &= 4N^2 \end{aligned} \tag{2.5}$$

where:

m_N is the multiplication operations

s_N is the subtraction operations

ad_N is the number of addition operations.

An example of applying this technique for digital image given in Figure 2.4 is shown in Figure 2.9 at 70° rotation.

2. Triangular Interpolation

Forms a weighted average of the intensities associated with areas of the triangles formed between the rotated output pixel and its four nearest neighbours as shown in Figure 2.8b. The triangle areas for each the data points are:

$$\begin{aligned} k(0,0) &= (1. - f_y) \\ k(0,1) &= f_x \\ k(1,1) &= f_y \\ k(1,0) &= (1. - f_x) \end{aligned} \tag{2.6}$$

The intensity of the output pixel can be defined as:

$$b(v,u) = \frac{1}{2} \sum_{L=0}^1 \sum_{n=0}^1 a(k_y + L, k_x + n)k(L,n) \tag{2.7}$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$\begin{aligned} m_N &= 5N^2 \\ ad_N &= 3N^2 \\ s_N &= 2N^2 \end{aligned} \tag{2.8}$$

Example of applying this technique is shown in Figure 2.10 at 70° rotation.

3 Two-Dimensional Linear Interpolation

This method computes the intensities associated with the linear distances formed between the rotated output pixel and its four nearest neighbours (Figure 2.8b).

The linear distances for each data points are:

$$\begin{aligned} k(0,0) &= \sqrt{((u - (1 + k_x))^2 + (v - (1 + k_y))^2)} \\ k(0,1) &= \sqrt{((u - k_x)^2 + (v - (1 + k_y))^2)} \\ k(1,1) &= \sqrt{((u - k_x)^2 + (v - k_y)^2)} \\ k(1,0) &= \sqrt{((u - (1 + k_x))^2 + (v - k_y)^2)} \end{aligned} \tag{2.9}$$

The intensity of the output pixel can be defined as:

$$b(v,u) = \sum_{L=0}^1 \sum_{n=0}^1 a(k_y + L, k_x + n)k(L,n) \tag{2.10}$$

This method requires a long computation time.

$$m_N = 12N^2$$

$$s_N = 8N^2$$

$$ad_N = 11N^2 \quad (2.11)$$

$$sq_N = 4N^2$$

Where:

Sq is the number of the square roots operations.

An example of applying this technique is shown in Figure 2.11 at 70° rotation.

2.4.3 Multi-Neighbour Interpolation

Based on the same principals as bilinear interpolation but considers more than just the four nearest neighbours. They can be classified according to the number of pixels (e.g. twelve or sixteen nearest neighbours) as shown in Figure 2.12 as follows:

a) Twelve-neighbours interpolation

The twelve-neighbours pixel can be classified as:

1.Rectangular-12 neighbours interpolation

It proceeds by forming a weighted average of the intensities of the twelve nearest neighbours.

The rectangle areas for each data points are:

$$k(1,0) = (1. - f_x)(1. + f_y)$$

$$k(0,0) = (2 - f_x)(1. - f_y)$$

$$k(0,1) = f_x(2 - f_y)$$

$$k(1,1) = (1. + f_x)f_y$$

$$k(-1,0) = (1. - f_x)(1. - f_y)$$

$$k(0,2) = f_x(1. - f_y)$$

$$k(2, 1) = f_x f_y \quad (2.12)$$

$$k(1, -1) = (1 - f_x) f_y$$

$$k(2, 0) = (2 - f_x) f_y$$

$$k(0, -1) = (2 - f_y)(1 - f_x)$$

$$k(-1, 1) = (1 + f_x)(1 - f_y)$$

$$k(1, 2) = (1 + f_y) f_x$$

$$k(-1, -1) = k(-1, 2) = k(2, -1) = k(2, 2) = 0$$

The total area A_r is:

$$A_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L, n) = 7$$

The intensity of the output pixel can be defined as:

$$b(v, u) = \frac{1}{7} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L, n) \quad (2.13)$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$m_N = 24N^2$$

$$dv_N = N^2$$

$$s_N = 12N^2 \quad (2.14)$$

$$ad_N = 15N^2$$

where:

dv_N is the number of division operations.

An example of applying this technique is shown in Figure 2.13 at 70° rotation.

2. Triangular-12 Neighbours Interpolation

This method proceeds by forming a weighted average of the intensities associated with the triangles of the twelve nearest neighbours shown in Figure 2.12.

The triangle areas for each data points are:

$$\begin{aligned}
 k(0,0) &= 2. - f_y \\
 k(0,1) &= 1 + f_x \\
 k(1,1) &= 1. + f_y \\
 k(1,0) &= 2. - f_x \\
 k(2,0) &= 1. - f_x \\
 k(0,-1) &= 1. - f_y \\
 k(-1,1) &= f_x \\
 k(1,2) &= f_y \\
 k(0,2) &= k(0,-1) \\
 k(2,1) &= k(-1,1) \\
 k(1,-1) &= k(1,2) \\
 k(-1,0) &= k(2,0) \\
 k(-1,-1) &= k(-1,2) = k(2,-1) = k(2,2) = 0
 \end{aligned} \tag{2.15}$$

The total area A_r is:

$$A_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L,n) = 10$$

The intensity of the output pixel can be defined as:

$$b(v,u) = \frac{1}{10} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L,n) \tag{2.16}$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$\begin{aligned} m_N &= 13N^2 \\ s_N &= 4N^2 \\ ad_N &= 13N^2 \end{aligned} \tag{2.17}$$

An example of applying this technique is shown in Figure 2.14 at 70° rotation.

3. Linear Interpolation over 12-Neighbour Pixels

This method computes the intensities associated with the linear distances formed between the rotated output pixel and its twelve nearest neighbours. The linear distances for each output data points are:

$$\begin{aligned} k(1,0) &= \sqrt{([v - (k_y - 1)]^2 + [u - (k_x + 1)]^2)} \\ k(0,0) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x + 2)]^2)} \\ k(0,1) &= \sqrt{([v - (k_y + 2)]^2 + [u - k_x]^2)} \\ k(1,1) &= \sqrt{([v - k_y]^2 + [u - (k_x - 1)]^2)} \\ k(-1,0) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x + 1)]^2)} \\ k(0,2) &= \sqrt{([v - (k_y + 1)]^2 + [u - k_x]^2)} \\ k(2,1) &= \sqrt{([v - k_y]^2 + [u - k_x]^2)} \\ k(1,-1) &= \sqrt{([v - k_y]^2 + [u - (k_x + 1)]^2)} \\ k(2,0) &= \sqrt{([v - k_y]^2 + [u - (k_x + 2)]^2)} \\ k(0,-1) &= \sqrt{([v - (k_y + 2)]^2 + [u - (k_x + 1)]^2)} \\ k(-1,1) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x - 1)]^2)} \end{aligned} \tag{2.18}$$

$$k(1, 2) = \sqrt{([v - (k_y - 1)]^2 + [u - k_x]^2)}$$

$$k(-1, -1) = k(-1, 2) = k(2, -1) = k(2, 2) = 0$$

The total distance d_r is:

$$d_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L, n) = \sqrt{5}$$

The intensity of the output pixel can be written as :

$$b(v, u) = \frac{1}{\sqrt{5}} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L, n) \quad 2.19$$

The no.of operations of $N \times N$ image required to compute the distance coefficients are given by :

$$\begin{aligned} m_N &= 36N^2 \\ dv_N &= N^2 \\ s_N &= 28N^2 \\ ad_N &= 35N^2 \\ sq_N &= 13N^2 \end{aligned} \quad (2.20)$$

An example of applying this technique is shown in Figure 2.15 at 70° rotation.

b) Sixteen-Neighbours Interpolation

These methods compute the intensities associated with the areas and linear distances formed between the rotated output pixel and its sixteen-nearest neighbours as follows:

1. Rectangular-16 Neighbours Interpolation

It proceeds by forming a weighted average of the intensities of the sixteen nearest neighbours. The rectangle areas for each data points are:

$$k(1,0) = (2. - f_x)(1. + f_y)$$

$$k(0,0) = (2 - f_x)(2. - f_y)$$

$$k(0,1) = (1. + f_x)(2 - f_y)$$

$$k(1,1) = (1. + f_x)(1. + f_y)$$

$$k(-1,-1) = (1. - f_x)(1. - f_y)$$

$$k(-1,2) = f_x(1. - f_y)$$

$$k(2,2) = f_x f_y$$

$$k(2,-1) = (1. - f_x)f_y$$

$$k(1,-1) = (2. - f_x)f_y \tag{2.21}$$

$$k(-1,0) = (2 - f_y)(1. - f_x)$$

$$k(0,2) = (1. + f_x)(1 - f_y)$$

$$k(2,1) = (1. + f_y)f_x$$

$$k(2,0) = (1. - f_x)(1. + f_y)$$

$$k(0,-1) = (1 - f_y)(2. - f_x)$$

$$k(-1,1) = f_x(2. - f_y)$$

$$k(1,2) = (1. + f_x)f_y$$

The total area A_r is:

$$A_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L,n) = 16$$

The intensity of the output pixel can be defined as:

$$b(v, u) = \frac{1}{16} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L, n) \quad (2.22)$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$\begin{aligned} m_N &= 32N^2 \\ dv_N &= N^2 \\ s_N &= 16N^2 \\ ad_N &= 23N^2 \end{aligned} \quad (2.23)$$

An example of applying this technique is shown in Figure 2.16 at 70° rotation.

2 Triangular-16 Neighbours Interpolation

Similarly, the triangle areas for each data points are:

$$\begin{aligned} k(0, 0) &= 2. - f_y \\ k(0, 1) &= 1 + f_x \\ k(1, 1) &= 1. + f_y \\ k(1, 0) &= 2. - f_x \\ k(-1, 0) &= k(-1, 1) = k(0, 0) \\ k(0, 2) &= k(1, 2) = k(0, 1) \\ k(2, 1) &= k(2, 0) = k(1, 1) \\ k(1, -1) &= k(0, -1) = k(1, 0) \\ k(2, 2) &= f_y \end{aligned} \quad (2.24)$$

$$k(2, -1) = 1 - f_x$$

$$k(-1, -1) = 1 - f_y$$

$$k(-1, 2) = f_x$$

The total area A_r is:

$$A_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L, n) = 20$$

The intensity of the output pixel can be defined as:

$$b(v, u) = \frac{1}{20} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L, n) \quad (2.25)$$

The no. of operation of $N \times N$ image required to compute the area coefficients are given by:

$$m_N = 16N^2$$

$$dv_N = N^2$$

$$s_N = 4N^2 \quad (2.26)$$

$$ad_N = 17N^2$$

An example of applying this technique is shown in Figure 2.17 at 70° rotation.

3. Linear Interpolation over 16-Neighbour Pixels

Similarly, it computes the intensities associated with the linear distances associated with the sixteen neighbour pixels. The linear distances for each output data points are:

$$k(0, 0) = \sqrt{([v - (k_y + 2)]^2 + [u - (k_x + 2)]^2)}$$

$$k(0, 1) = \sqrt{([v - (k_y + 2)]^2 + [u - (k_x - 1)]^2)}$$

$$k(1, 1) = \sqrt{([v - (k_y - 1)]^2 + [u - (k_x - 1)]^2)}$$

$$\begin{aligned}
k(1,0) &= \sqrt{([v - (k_y - 1)]^2 + [u - (k_x + 2)]^2)} \\
k(-1,0) &= \sqrt{([v - (k_y + 2)]^2 + [u - (k_x + 1)]^2)} \\
k(-1,1) &= \sqrt{([v - (k_y + 2)]^2 + [u - k_x]^2)} \\
k(2,1) &= \sqrt{([v - (k_y - 1)]^2 + [u - k_x]^2)} \\
k(2,0) &= \sqrt{([v - (k_y - 1)]^2 + [u - (k_x + 1)]^2)} \\
k(0,-1) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x + 2)]^2)} \\
k(0,2) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x - 1)]^2)} \\
k(1,2) &= \sqrt{([v - k_y]^2 + [u - (k_x - 1)]^2)} \\
k(1,-1) &= \sqrt{([v - k_y]^2 + [u - (k_x + 2)]^2)} \\
k(-1,-1) &= \sqrt{([v - (k_y + 1)]^2 + [u - (k_x + 1)]^2)} \\
k(-1,2) &= \sqrt{([v - (k_y + 1)]^2 + [u - k_x]^2)} \\
k(2,2) &= \sqrt{([v - k_y]^2 + [u - k_x]^2)} \\
k(2,-1) &= \sqrt{([v - k_y]^2 + [u - (k_x + 1)]^2)}
\end{aligned} \tag{2.27}$$

The total distance d_r is:

$$d_r = \sum_{L=-1}^2 \sum_{n=-1}^2 k(L, n) = 3$$

The intensity of the output pixel can be written as :

$$b(v, u) = \frac{1}{3} \sum_{L=-1}^2 \sum_{n=-1}^2 a(k_y + L, k_x + n) k(L, n) \tag{2.28}$$

The no. of operations of $N \times N$ image required to compute the distance coefficients are given by :

$$\begin{aligned}
m_N &= 48N^2 \\
dv_N &= N^2 \\
s_N &= 40N^2 \\
ad_N &= 47N^2 \\
sq_N &= 16N^2
\end{aligned} \tag{2.29}$$

An example of applying this technique is shown in Figure 2.18 at 70° rotation.

2.5 Factorizations of the Basic Rotation Matrix

Factorized versions of the rotation matrix corresponds to execution in more than one pass. The straight forward one-pass rotation is given by equation (2.2). The advantage of factorization is the fact that all operations are one-dimensional interpolations and data moves only one row or column at a time.

2.5.1 Two-pass rotation

It has been observed by several authors [1,2] that the transformation matrix such as that in Eqn.(2.2) can be decomposed. Hence we can write equation (2.2) as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan \theta \\ 0 & \frac{1}{\cos \theta} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.30}$$

Friedman [2] observed a serious flaw when using this decomposition. the intermediate image, from the first transformation, is unable to retain all the high-frequency content of the original image. A rotation angle θ causes a compression with $\cos \theta$ in the vertical dimension. This is illustrated in Fig.2.19, top row, where the input image $i(x, y)$ is first converted to $g(x, y)$ using the last rotation matrix of (2.30). The data compression is obvious since the area of the g -image is $\sqrt{2}$ less than that of the input image.

To avoid loss of information, Friedman resamples and enlarges the original image $i(x, y)$ in the horizontal direction prior to the vertical skewing. The enlargement factor should be $\frac{1}{\cos \theta}$, which means that we augment the previous factorization (2.30) and get

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sin \theta \\ 0 & \frac{1}{\cos \theta} \end{bmatrix} \begin{bmatrix} \cos \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.31)$$

This decomposition corresponds to the image sequence in the bottom of Fig.2.19, where $h(x, y)$ is extended in the x-direction so that $g(x, y)$ is now able to hold the data without loss of information. The computation is then really a three-pass procedure.

2.5.2 Three-pass rotation

As was shown by Paeth [3] that the rotation matrix in (2.2) can be decomposed in a different way than that proposed by Friedman *et al.* This is shown by [4]:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -\tan \frac{\theta}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin \theta & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan \frac{\theta}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.32)$$

The virtue of this decomposition, compared to the two-pass algorithm, is that it consists of three skewing matrices (movements in one coordinate only) and that all these skewings can be performed as a kind of parallel shifts row by row and column by column in the digital image as shown in Fig. 2.20.

2.6 The Trigonometric Computing Techniques

The CPU rotation time can be computed in terms of the number of the basic arithmetic operations by the corresponding rotation algorithms. The analytic expressions for the number of the basic operations needed for the given rotation algorithms are presented in Table 3.1. Also, the overall CPU rotation time for each algorithm is shown in Table 3.2 and Table 3.3 respectively.

For the triangular interpolation, the analytic expressions for the number of basic operations are lower compared with the bilinear interpolation. However, for the linear interpolation, the computing requirements are larger compared with the other algorithms.

2.7 Holes in image rotation

The nearest neighbour technique has the drawback of producing undesirable artifacts such as 'holes' (zeros) due to missing pixels in the rotated image (See Fig.2.5 and Fig.3.4). The number of holes become large when approaching 45° and 135° rotation respectively. They can be cured by applying four neighbour or some type of interpolation techniques.

2.8 References

- [1] E. Catmull and A.R. Smith, "3d-transformations of images in scan-line order", in SIGGRAPH Proceedings, vol.14, pp. 279-285, 1980.
- [2] D.E. Friedman, "Two-dimensional resampling of line scan imagery by one-dimensional processing", Photogrammetric Engrg. Remote Sensing 47, pp. 1459-1467, Oct. 1981.
- [3] A.W. Paeth, "A fast algorithm for general raster rotation", In Proceedings, Graphics Interface '86-Vision Interface'86, pp. 77-81, 1986.
- [4] P. Danielsson and M. Hammerin, " High-accuracy rotation of images", CVGIP: Graphic Models and Image Processing, vol.54, No.4, pp. 340-344, July, 1992.

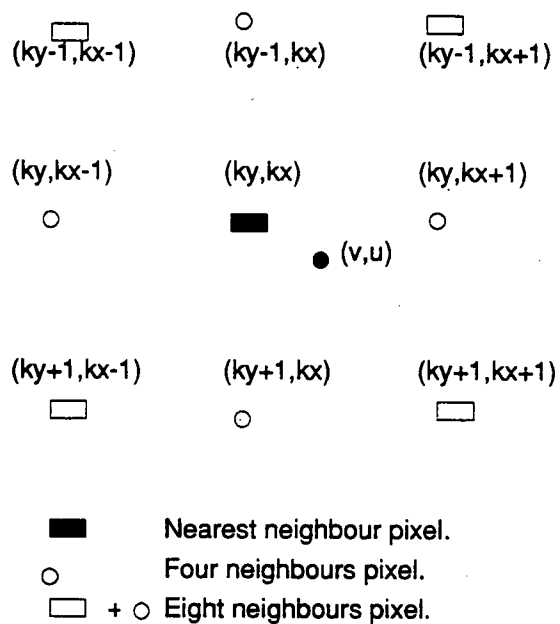


Figure 2.1 Basic relationships between pixels.

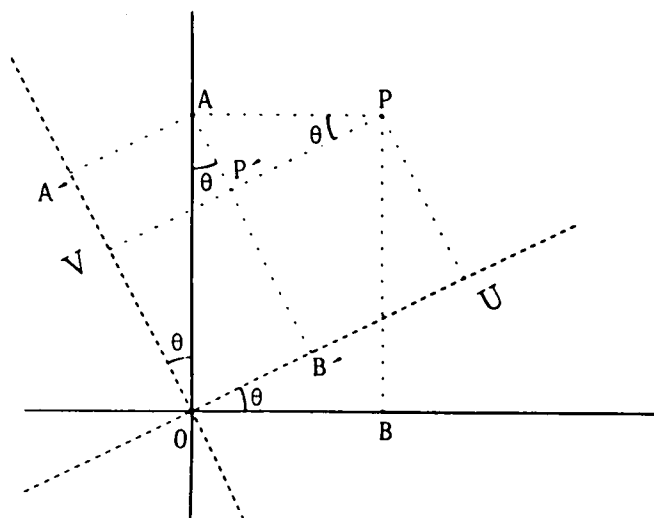


Figure 2.2 Rotation of axes.

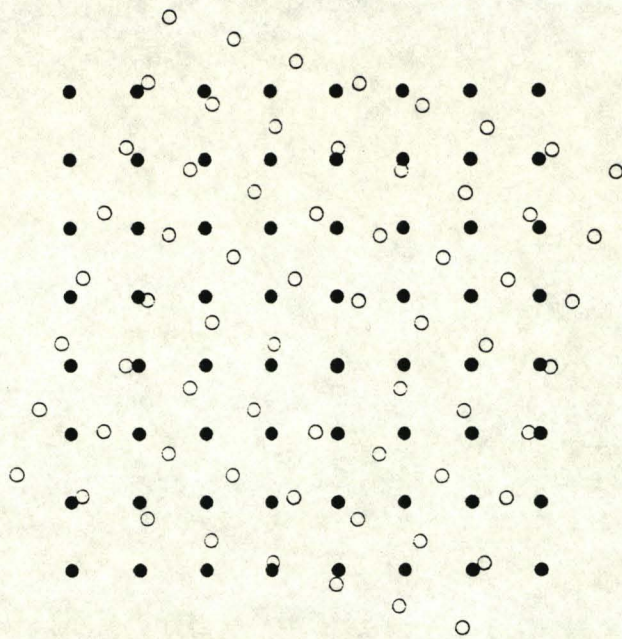


Figure 2.3 Cartesian and Polar coordinates representation.

- Polar coordinates of 8 x 8 pixels.
- Cartesian mesh of 8 x 8 pixels.

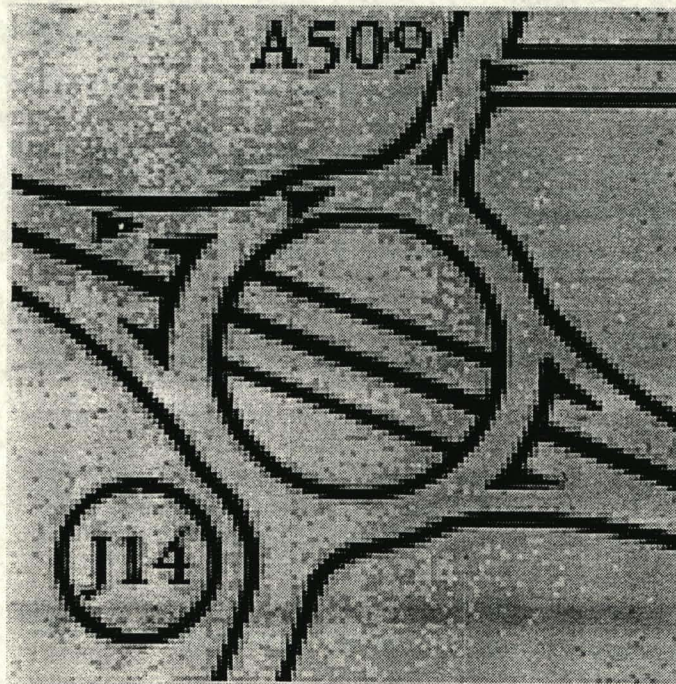


Figure 2.4 Original image of 128 x 128 resolution and 16 grey level.

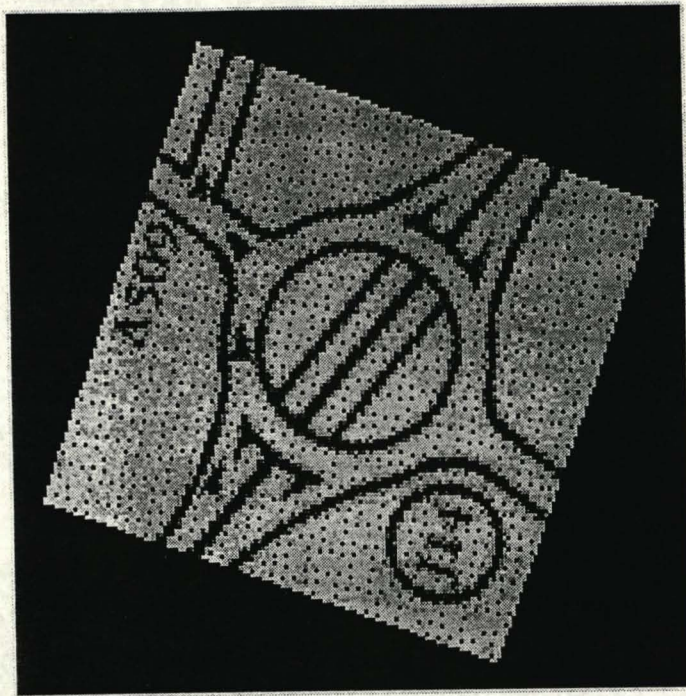


Figure 2.5 Normal digital rotation using nearest neighbour technique at 70° rotation.

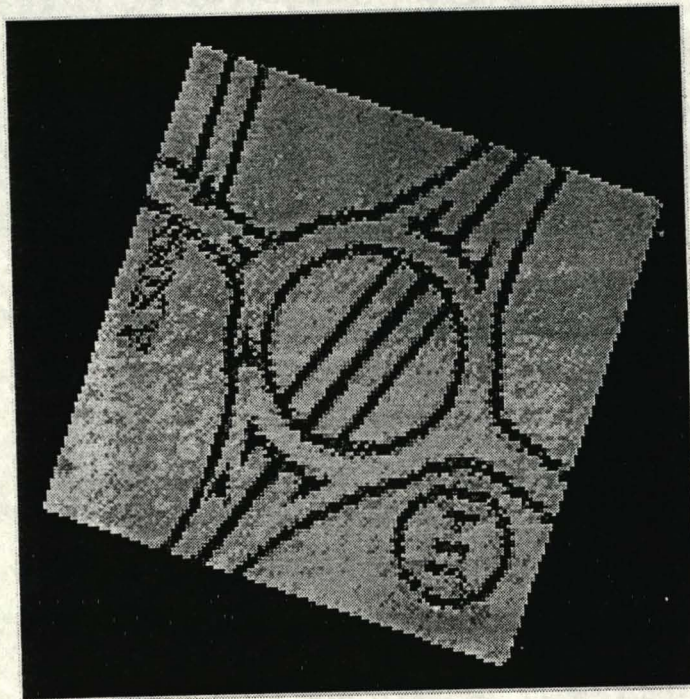


Figure 2.6 Normal digital rotation using four neighbours technique at 70° rotation.



Figure 2.7 Normal digital rotation using eight neighbours technique at 70° rotation.

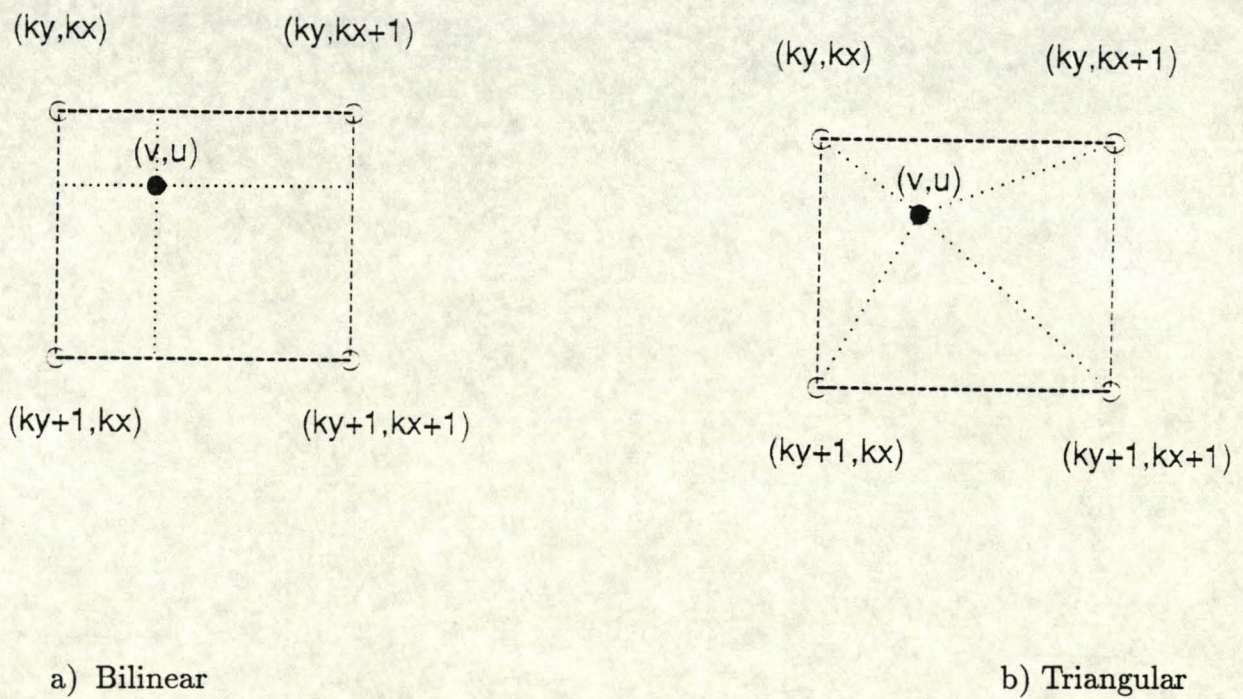
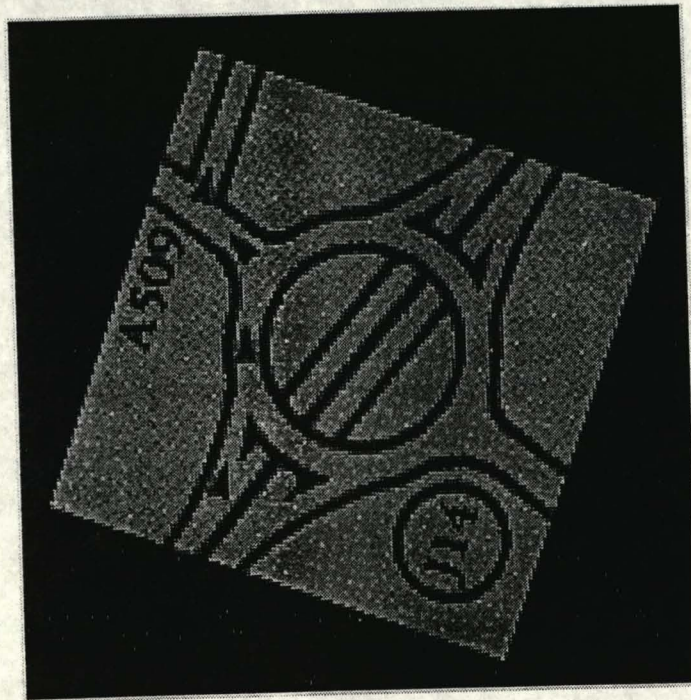


Figure 2.8 Area interpolation scheme.

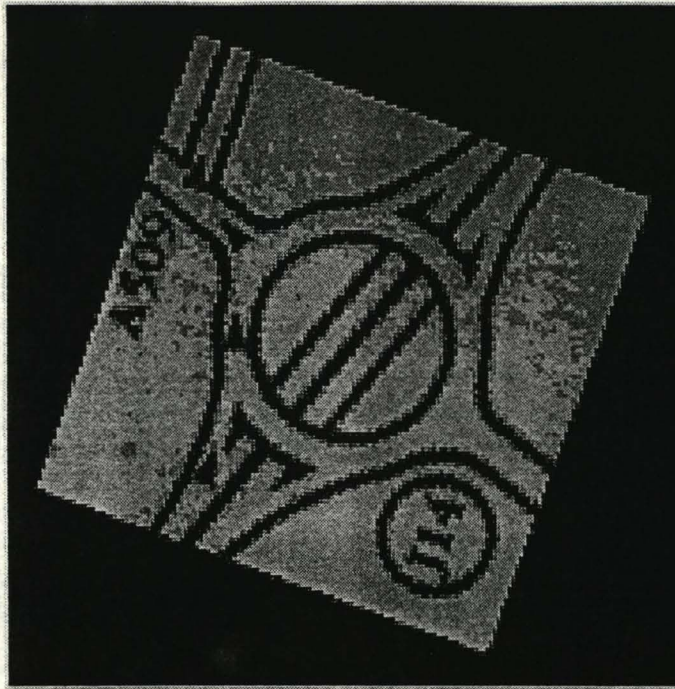


a) Forward.

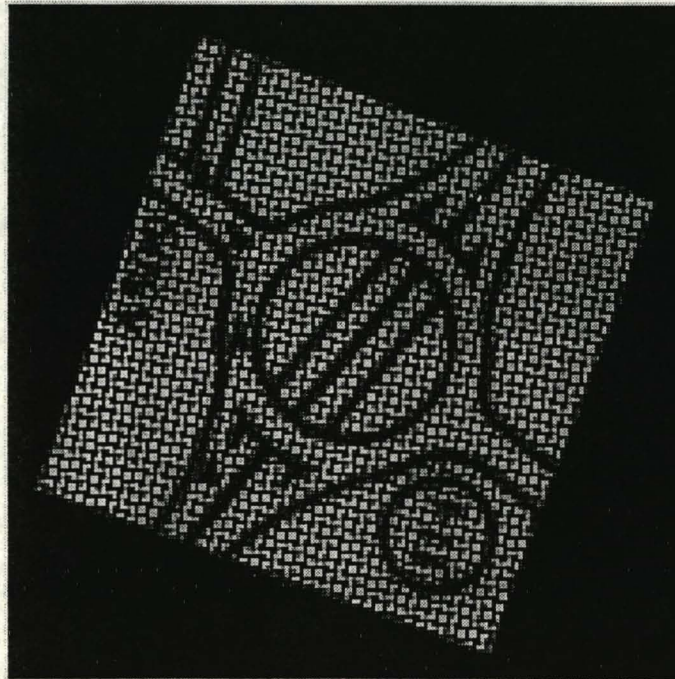


b) Reverse.

Figure 2.9 Digital rotation using bilinear interpolation at 70° rotation.

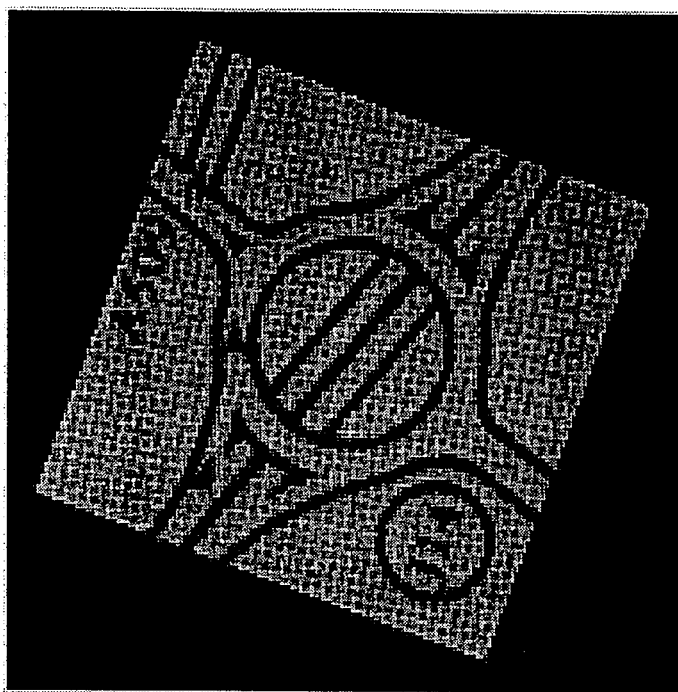


a) Forward.

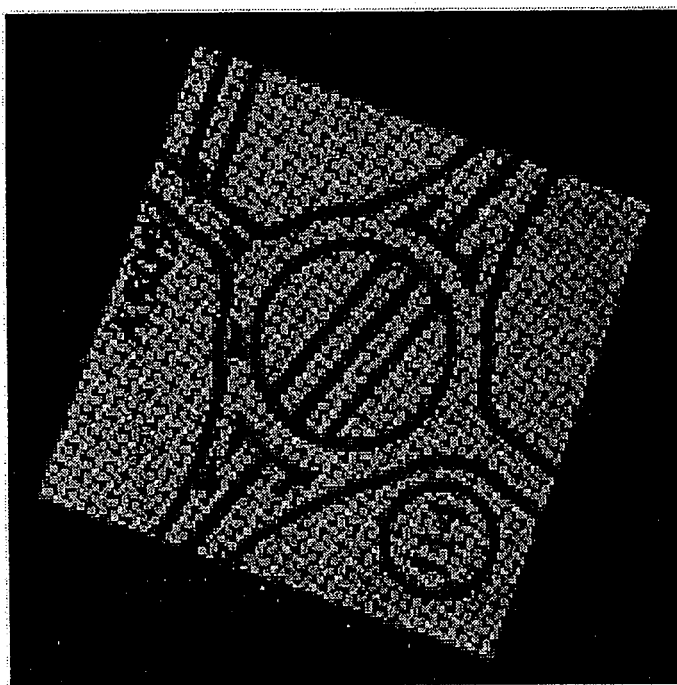


b) Reverse.

Figure 2.10 Digital rotation using triangular interpolation at 70° rotation.



a) Forward.



b) Reverse.

Figure 2.11 Digital rotation using 2D-linear interpolation at 70° rotation.

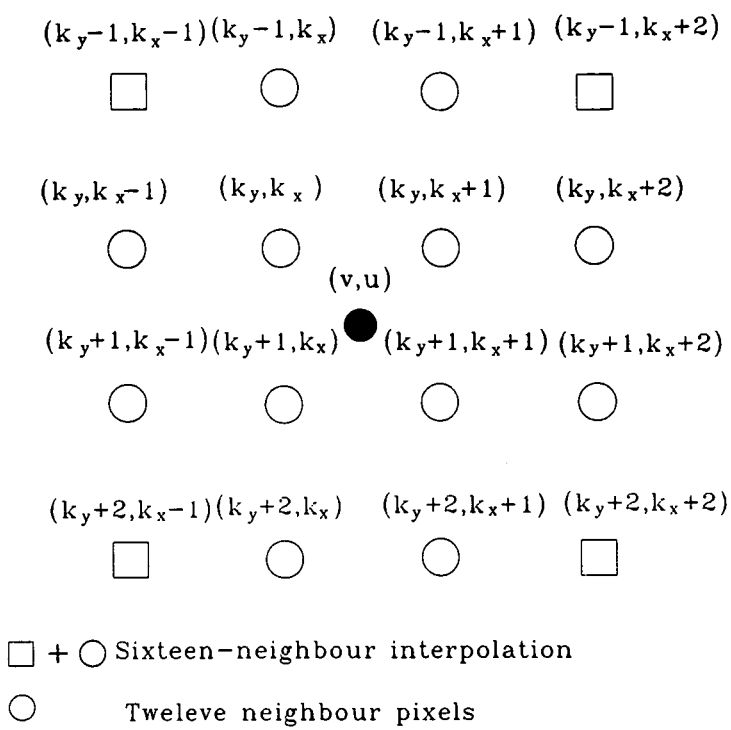
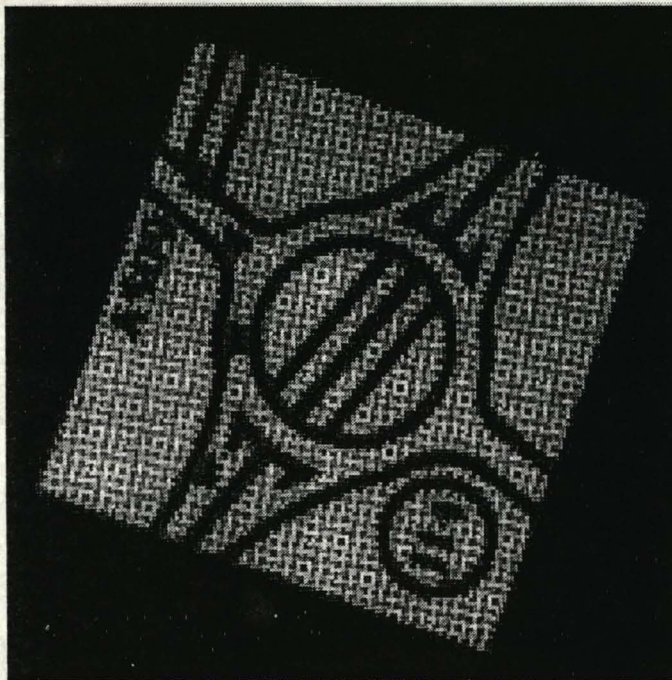


Figure 2.12 Multi-neighbour interpolation.



a) Forward.

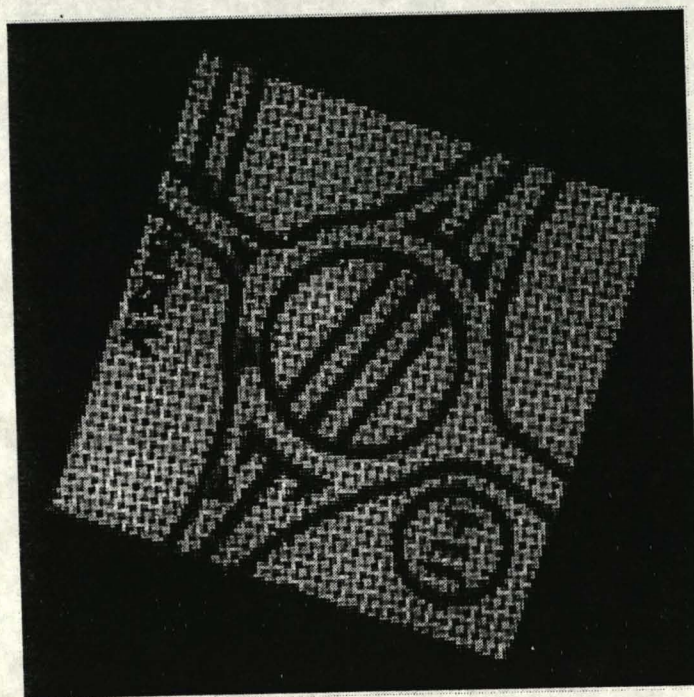


b) Reverse.

Figure 2.13 Digital rotation using rectangular 12-neighbours interpolation at 70° rotation.



a) Forward.

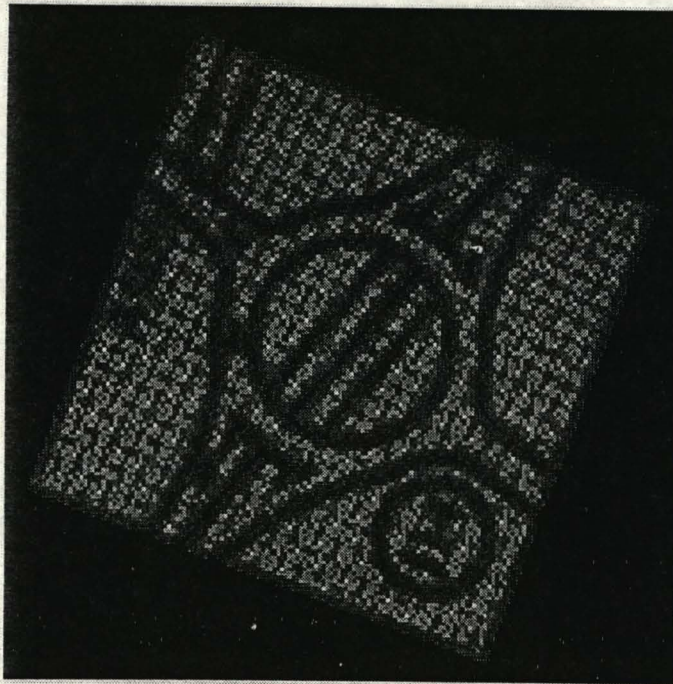


b) Reverse.

Figure 2.14 Digital rotation using triangular 12-neighbours interpolation at 70° rotation.

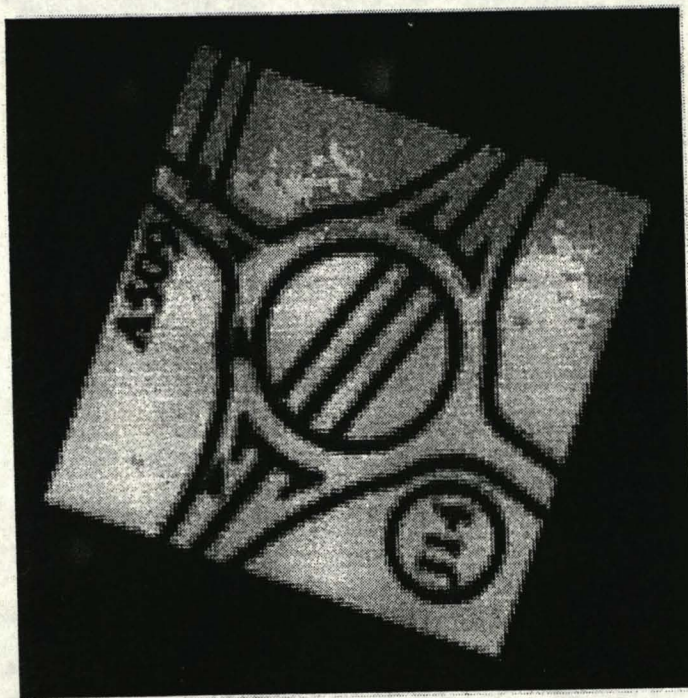


a) Forward.

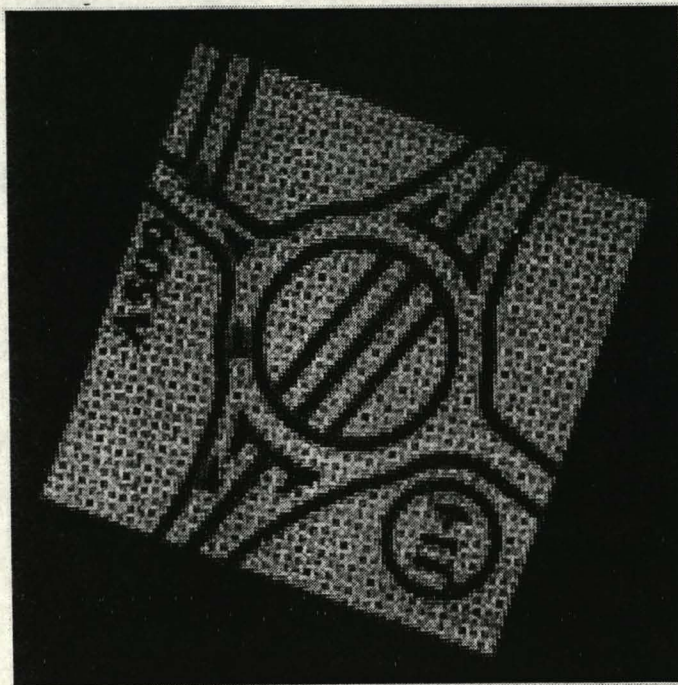


b) Reverse.

Figure 2.15 Digital rotation using linear interpolation over 12-neighbours at 70° rotation.



a) Forward.

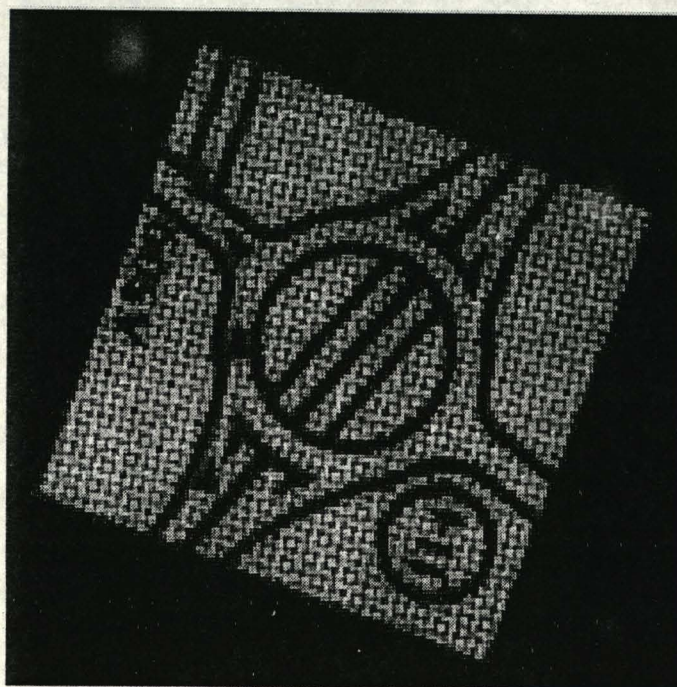


b) Reverse.

Figure 2.16 Digital rotation using rectangular 16-neighbours interpolation at 70° rotation.

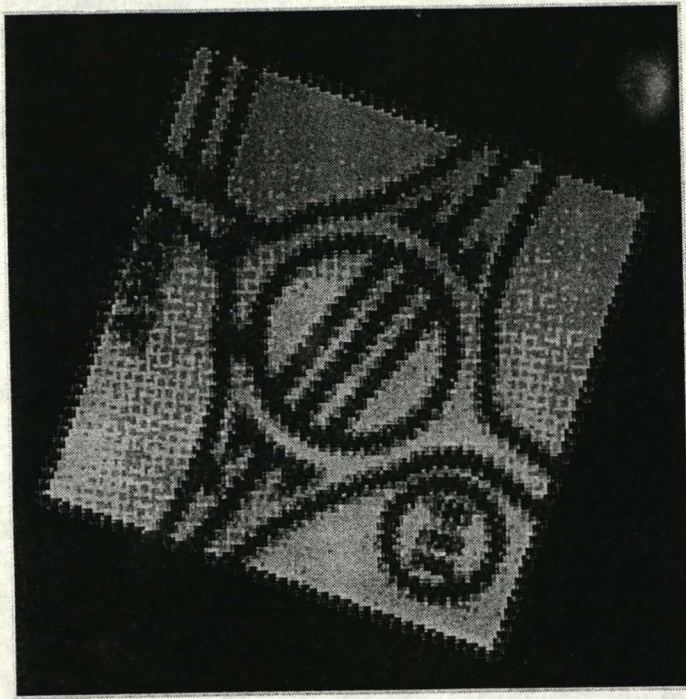


a) Forward.

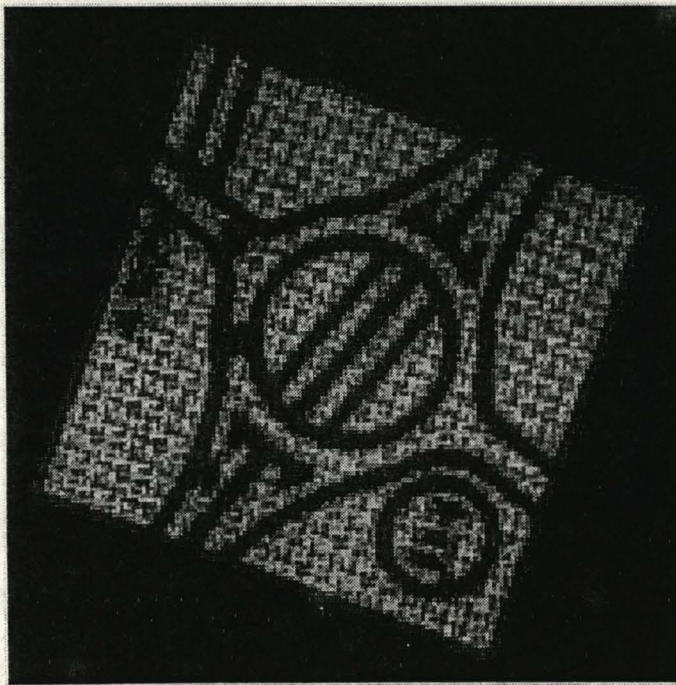


b) Reverse.

Figure 2.17 Digital rotation using triangular 16-neighbours interpolation at 70° rotation.



a) Forward.



b) Reverse.

Figure 2.18 Digital rotation using linear interpolation over 16-neighbours at 70° rotation.

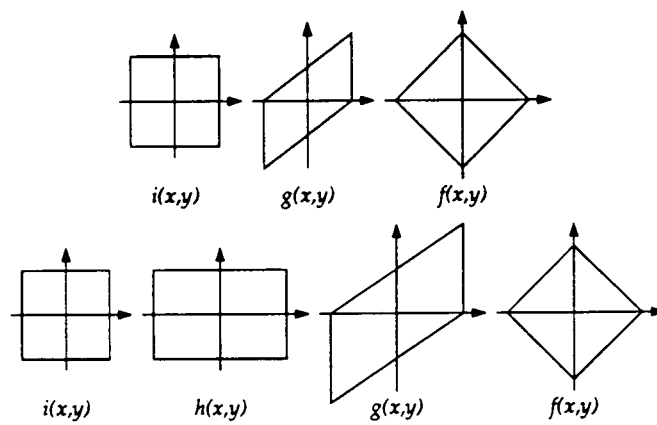


Figure 2.19 The two-pass rotation (Top) and the augmented two-pass rotation (Bottom).

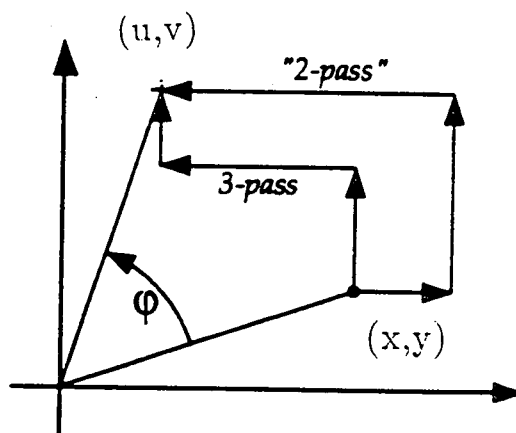


Figure 2.20 The point u,v is loaded with the value at x,y in two different ways for the two- and three-pass methods.

CHAPTER 3

Optimum Digital Rotation

3.1 Introduction

The coordinate transformation equations for image rotation are given by Eqns.2.1-2.2 and are recalled here:

$$u = x \cos \theta + y \sin \theta \quad (2.1)$$

$$v = y \cos \theta - x \sin \theta$$

or conversely

$$x = u \cos \theta - v \sin \theta \quad (2.2)$$

$$y = u \sin \theta + v \cos \theta$$

The above equations are a kind of transformation and must be calculated for every pixel in order to rotate an image. It is almost impossible to calculate Eqns.(2.1)-(2.2) for every pixel in real time by a small-scale computer. This is because it contains four multiplications for floating decimal fraction and two for its additions. The aforementioned techniques do not allow a quick real time rotation, therefore, a new technique is derived that allow a quick real time rotation.

3.2 Mathematical Formulation:

The idea behind this technique comes from the application of basic integer arithmetic operations to figure out the nearest pixel location.

Consider an $N \times M$ digital image centred at the pixel grid P_o . the appropriate rotated pixel position can be determined from the relative size of t_1 and $\Delta y_u(1)$, and $\Delta X_u(1)$ and S_1 respectively as shown in Figure 3.1. From the elementary trigonometry, the following parameters can be obtained as follows:

$$\frac{\Delta x_u}{2} = \frac{N}{2} \cos \theta \quad (3.1)$$

$$\frac{\Delta y_u}{2} = \frac{N}{2} \sin \theta \quad (3.2)$$

$$\frac{\Delta x_v}{2} = \frac{M}{2} \sin \theta \quad (3.3)$$

$$\frac{\Delta y_v}{2} = \frac{M}{2} \cos \theta \quad (3.4)$$

Where :

θ the angle of rotation.

N image width in pixels.

M image height in pixels.

Δx_u is the horizontal displacement along the X-axis which corresponds to the image width-pixels movement along the U-axis.

Δx_v is the horizontal displacement along the X-axis which corresponds to the image height-pixels movement along the V-axis.

Δy_u is the vertical displacement along the Y-axis which corresponds to the image width-pixels movement along the U-axis.

Δy_v is the vertical displacement along the Y-axis which corresponds to the image height-pixels movement along the V-axis.

To represent a non integer coordinate on a Cartesian coordinate, consider the following cases:

- Case 1:

$$\Delta y_u(1) > t_1:$$

a- $\Delta x_u(1) > s_1$ then T_1 is Nearest to P_1

b- $\Delta x_u(1) < s_1$ then T_1 is Nearest to Q_2

• Case 2:

$\Delta y_u(1) < t_1$:

a- $\Delta x_u(1) > s_1$ then T_1 is Nearest to Q_1

b- $\Delta x_u(1) < s_1$ then T_1 is Nearest to P_o

where:

$$\begin{aligned}\Delta y_u(1) &= \sin \theta = \frac{\Delta y_u}{N} \\ \Delta x_u(1) &= \cos \theta = \frac{\Delta x_u}{N}\end{aligned}\tag{3.5}$$

From the geometry of Fig. 3.1, distances t_1 and s_1 are given by :

$$t_1 = 1 - \Delta y_u(1)\tag{3.6}$$

$$s_1 = 1 - \Delta x_u(1)\tag{3.7}$$

With measurements made in pixel-space unit (i.e. by considering the situation depicted in Figure 3.1 in which pixels are of unit length in the x and y directions as shown).

• For one pixel movement along the U-axis :

$$\begin{aligned}\Delta y_u(1) - t_1 &= 2\Delta y_u(1) - 1 \\ &= 2(\Delta y_u/N) - 1\end{aligned}\tag{3.8}$$

Equation (3.8) would generally have to be processed using fractional arithmetic. As, however, simpler integer arithmetic is preferred. Equation (3.8) is recast the following alternative form :

$$V_u(1) = N(\Delta y_u(1) - t_1) = 2\Delta y_u - N\tag{3.9}$$

Where

$V_u(1)$ is an integer quantity having the same sign as $(\Delta y_u(1) - t_1)$.

ii) The horizontal displacement $\Delta x_u(1)$:

Similarity :

$$\begin{aligned}\Delta x_u(1) - s_1 &= 2\Delta x_u(1) - 1 \\ &= 2(\Delta x_u/N) - 1 \\ H_u(1) &= N(\Delta x_u(1) - s_1) = 2\Delta x_u - N\end{aligned}\tag{3.10}$$

Where:

$H_u(1)$ is an integer quantity having the same sign as $(\Delta x_u(1) - s_1)$

• For Two-pixel movements along the U axis :

i) The vertical displacement $\Delta y_u(2)$:

$$\Delta y_u(2) - t_2 = 2\Delta y_u(2) - 1\tag{3.11}$$

and from Fig. 3.1 we can find :

$$\Delta y_u(2) + \Delta y_p(1) = 2\frac{\Delta y_u}{N}\tag{3.12}$$

From Eqn.(3.11) and Eqn. (3.12) we have :

$$V_u(2) = N(\Delta y_u(2) - t_2) = 4\Delta y_u - 2\Delta y_p(1).N - N\tag{3.13}$$

ii) The horizontal displacement $\Delta x_u(2)$:

Similarly ;

$$H_u(2) = N(\Delta x_u(2) - s_2) = 4\Delta x_u - 2\Delta x_p(1).N - N\tag{3.14}$$

Where :

$\Delta y_p(1)$ and $\Delta x_p(1)$ are the numbers of pixel space unit of the $(i - 1)$ vertical and horizontal movements respectively.

• **For Three-pixel movements along the U axis :**

i) The vertical displacement $\Delta y_u(3)$:

Similarly,

$$V_u(3) = 6\Delta y_u - 2\Delta y_p(2).N - N \quad (3.15)$$

ii) The horizontal displacement $\Delta x_u(3)$:

$$H_u(3) = 6\Delta x_u - 2\Delta x_p(2).N - N \quad (3.16)$$

From Eqs.(3.13) - (3.16), the functions $V_u(2)$, $V_u(3)$, $H_u(2)$, and $H_u(3)$ depend on the state of the previous pixel coordinate. Clearly, this point toward an iterative solution for V_u and H_u .

That is,

$$\begin{aligned} V_u(3) - V_u(2) &= 2\Delta y_u - 2N(\Delta y_p(2) - \Delta y_p(1)) \\ V_u(3) &= V_u(2) + 2\Delta y_u - 2N(\Delta y_p(2) - \Delta y_p(1)) \end{aligned} \quad (3.17)$$

Similarly,

$$H_u(3) = H_u(2) + 2\Delta x_u - 2N(\Delta x_p(2) - \Delta x_p(1)) \quad (3.18)$$

Generalizing Equations (3.17) and (3.18), we obtain an expression for $V_u(i + 1)$ and $H_u(i + 1)$ in terms of the present value of $V_u(i)$ and $H_u(i)$ respectively.

$$V_u(i + 1) = V_u(i) + 2\Delta y_u - 2N(\Delta y_p(i) - \Delta y_p(i - 1)) \quad (3.19)$$

$$H_u(i + 1) = H_u(i) + 2\Delta x_u - 2N(\Delta x_p(i) - \Delta x_p(i - 1)) \quad (3.20)$$

Equations (3.19) and (3.20) can be reduced to the more convenient form :

$$V_u(i + 1) = V_u(i) + 2\Delta y_u - 2N.K_u(i) \quad (3.21)$$

$$H_u(i + 1) = H_u(i) + 2\Delta x_u - 2N.G_u(i) \quad (3.22)$$

and

$$K_u(i) = \Delta y_p(i) - \Delta y_p(i - 1)$$

$$G_u(i) = \Delta x_p(i) - \Delta x_p(i - 1)$$

Where :

$K_u(i)$ and $G_u(i)$ can be either 0 or 1 depending on the state of the pixels number i -th and $(i-1)$ th.

Similarly, we can find the expression for $V_v(i + 1)$ and $H_v(i + 1)$, that is,

$$V_v(i + 1) = V_v(i) + 2\Delta y_v - 2MK_v(i) \quad (3.23)$$

$$H_v(i + 1) = H_v(i) + 2\Delta x_v - 2MG_v(i) \quad (3.24)$$

If the digitized images are of equal size (i.e. $N = M$) then, the above four equations (3.21) - (3.24) can be reduced to two equations only.

Analysis of equations (3.21) and (3.22) :

1- $V_u(i) < 0$:

$$K_u(i) = 0$$

$$V_u(i + 1) = V_u(i) + 2\Delta y_u$$

$$= V_u(i) + D_{u1}$$

2- $V_u(i) > 0$:

$$K_u(i) = 1$$

$$V_u(i + 1) = V_u(i) + 2(\Delta y_u - N)$$

$$= V_u(i) + D_{u2}$$

3- $H_u(i) < 0$:

$$G_u(i) = 0$$

$$H_u(i+1) = H_u(i) + 2\Delta x_u \quad (3.25)$$

$$= H_u(i) + E_{u1}$$

4- $H_u(i) > 0$:

$$G_u(i) = 1$$

$$H_u(i+1) = H_u(i) + 2(\Delta x_u - N)$$

$$= H_u(i) + E_{u2}$$

where:

$$D_{u1} = 2\Delta y_u$$

$$D_{u2} = 2(\Delta y_u - N)$$

$$E_{u1} = 2\Delta x_u$$

$$E_{u2} = 2(\Delta x_u - N)$$

Following the concept of Eqn. (3.25), the values of k_u and G_u at the i -th pixel can be obtained. By rotating the digital image about its centre, the computations of K_u and G_u are reduced to half value of its size. Since,

$$K_u(i) = K_u\left(\frac{N}{2} + i\right), i = 1, 2, \dots, \frac{N}{2}$$

$$G_u(i) = G_u\left(\frac{N}{2} + i\right), i = 1, 2, \dots, \frac{N}{2} \quad (3.26)$$

Finally, testing the signs of Δx_u and Δy_u are required to generate 4-quadrant Cartesian plane. Applying the coordinate transformation for digital rotation, we get :

$$\begin{aligned} V_{ij} &= m_1 G_u(i) - n_1 K_u(j) \\ U_{ij} &= m_1 G_u(j) + n_1 K_u(i) \end{aligned} \quad (3.27)$$

Where :

m_1 and n_1 are the signs of Δx_u and Δy_u respectively and can be either -1 or 1.

i and j are the input pixel location.

U_{ij} and V_{ij} represent the integer value of rotated output pixel.

Note, that the method operates on each line of rotation (u and v axis) once the location of the pixels along the line have been obtained and not on a pixel-pixel basis

3.3 Algorithm for Calculating the Optimum Rotation :

Applying the nearest neighbour techniques discussed in Chapter 2 to eqn. (3.27), result in a quick real time image rotation and drastically reduces the computation time as compared with using the elementary trigonometry image rotation algorithm (Chapter 2). A good agreement is found between the results of both algorithms. A flow chart illustrating the building structure and the possible sequences of computations of the optimum digital rotation is shown in Figure 3.2. A test image of 128 x 128 resolution is shown in Figure 3.3. Examples of applying this technique are shown in Figure 3.4 through Figure 3.6 at 70° rotation.

3.4 Comparison of Rotation Techniques

The aforementioned rotation techniques discussed in Chapter 2 and in this Chapter verse the CPU time are illustrated in Table 3.2 through Table 3.3 at 70° rotation.

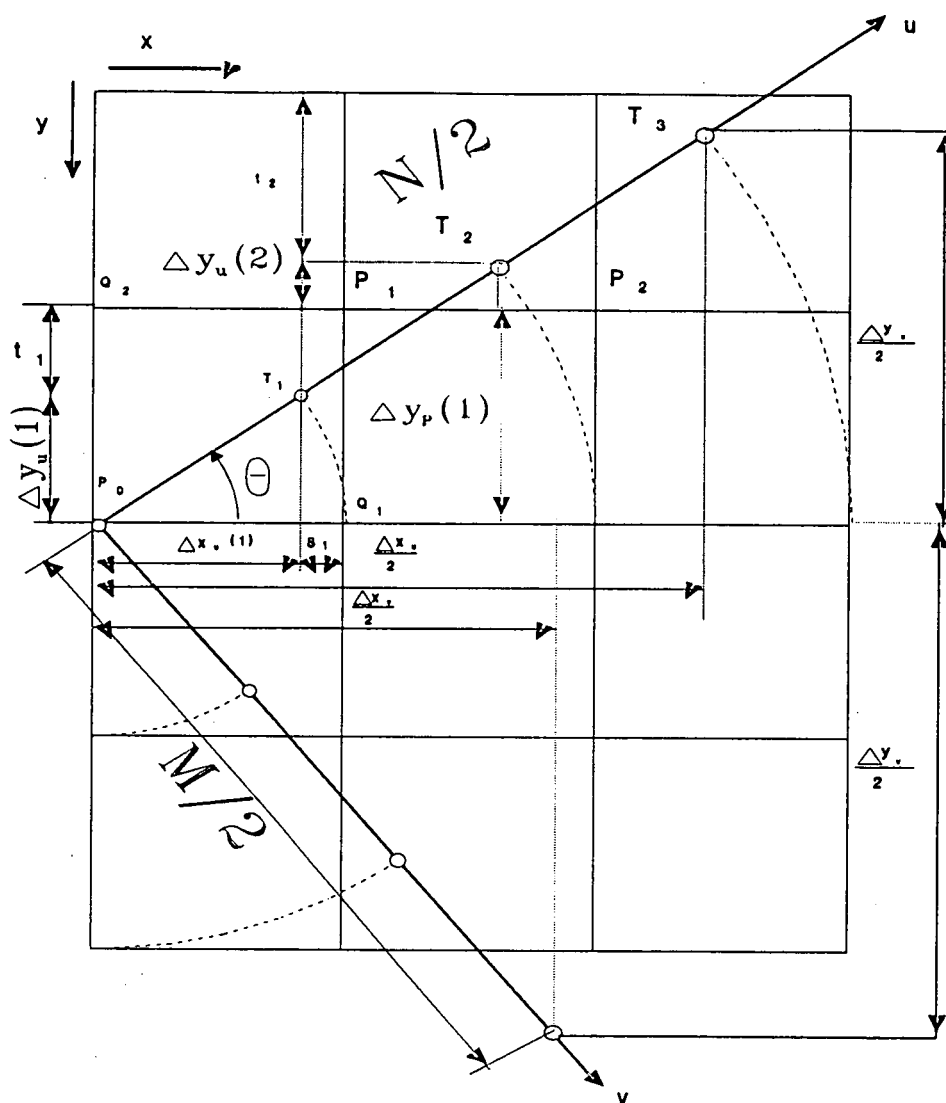
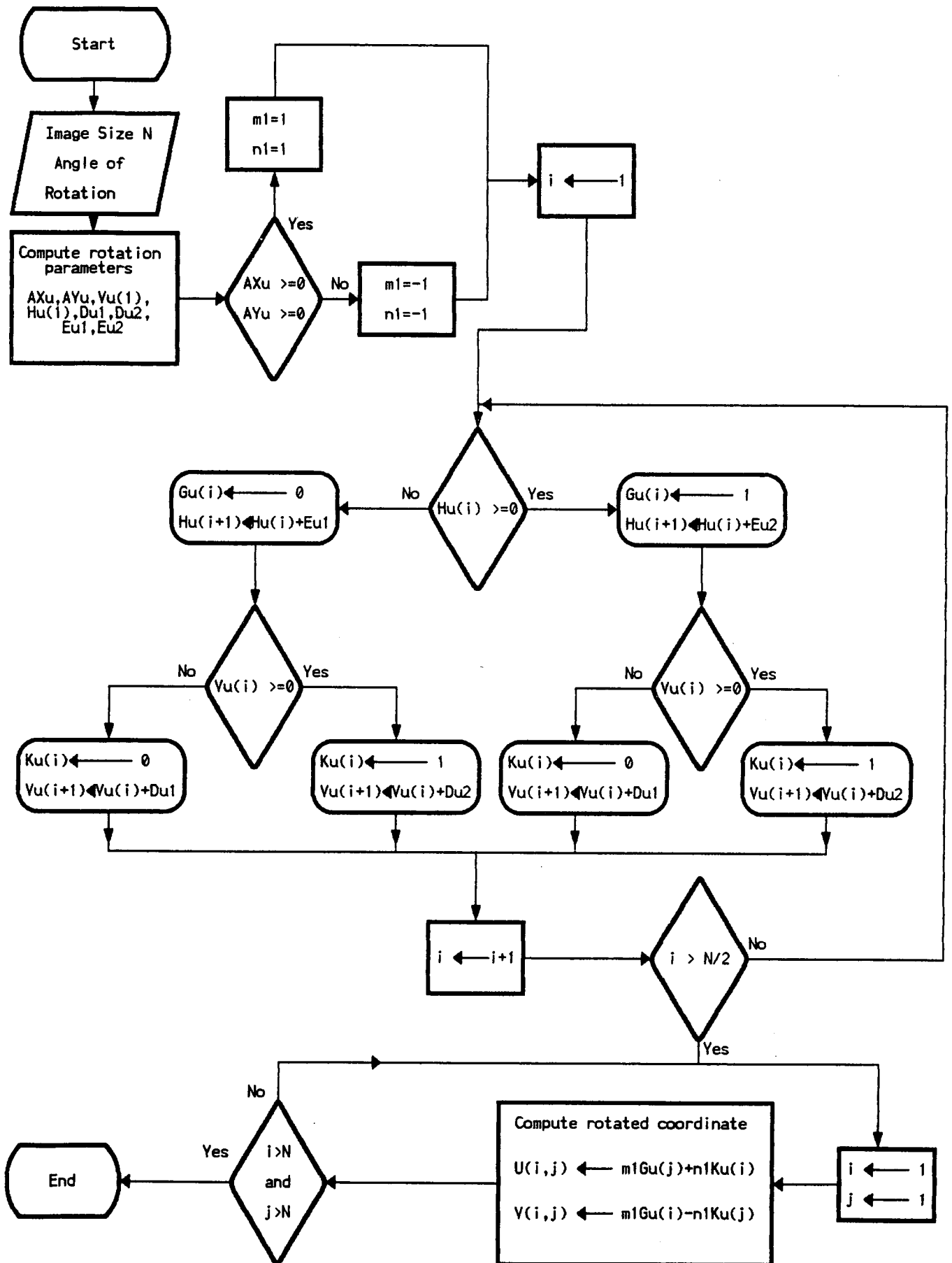


Figure 3.1 Representation of Polar coordinates on a Cartesian grid.

Figure 3.2 Optimum image rotation flowchart.



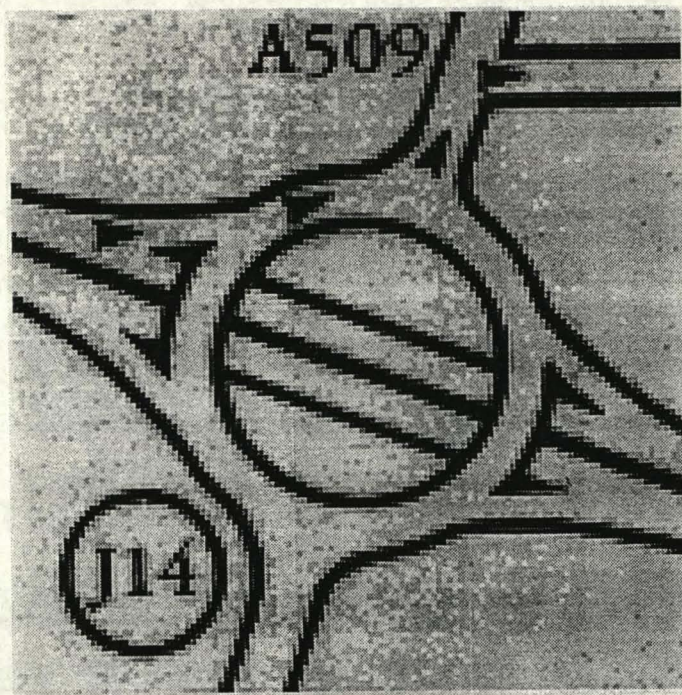


Figure 3.3 Original image of 128 x128 resolution and 16 grey level.

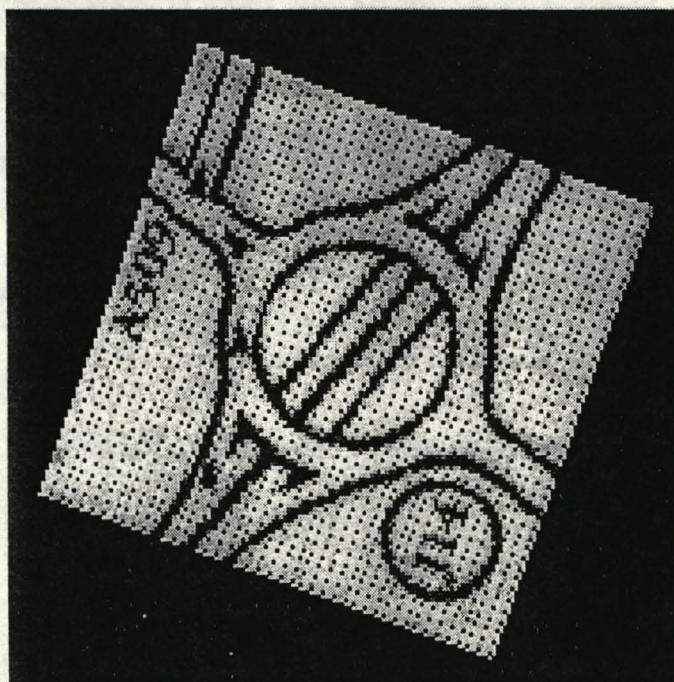


Figure 3.4 Optimum digital rotation using nearest neighbour technique at 70° rotation.

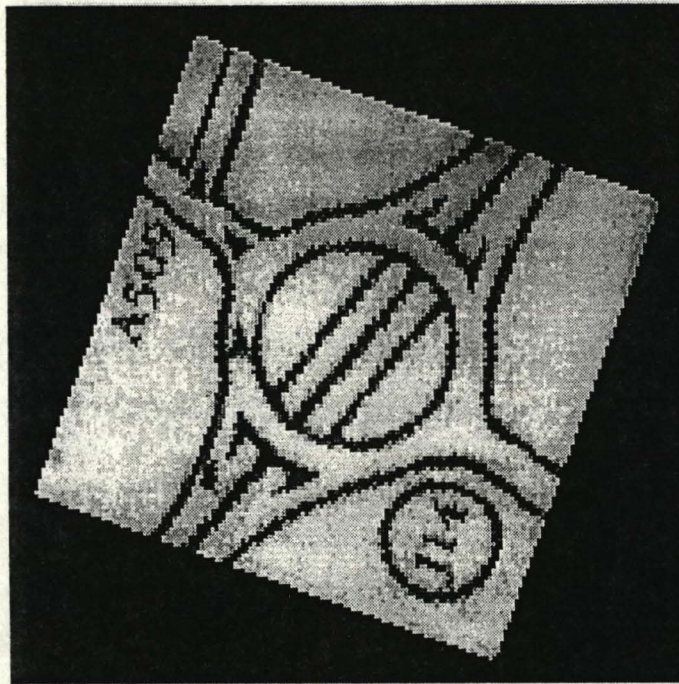


Figure 3.5 Optimum digital rotation using four neighbours technique at 70° rotation.

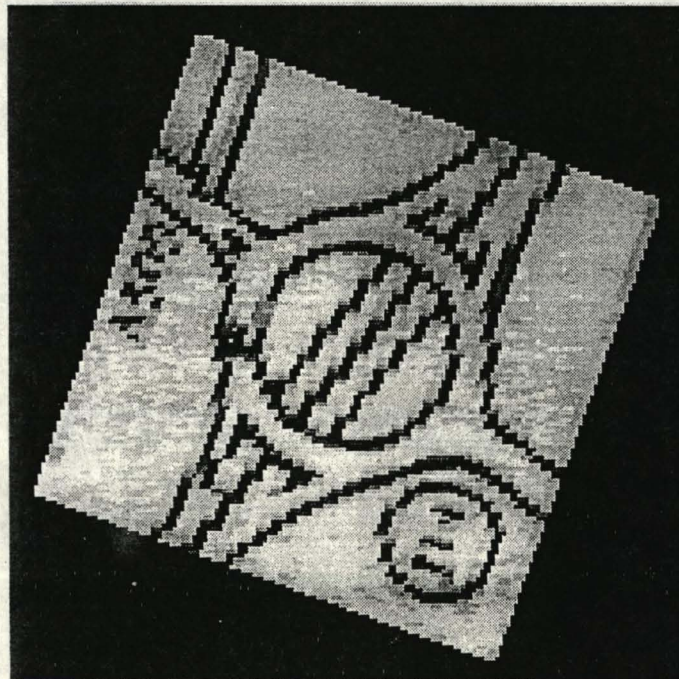


Figure 3.6 Optimum digital rotation using eight neighbours technique at 70° rotation.

Table 3.1 Comparison between the number of operations in the individual rotation algorithm.

Rotation techniques	No.of operations in the interpolation process				
	*	/	+	-	√
Nearest neighbours	-	-	-	-	-
Bilinear interpolation	$8N^2$	-	$3N^2$	$4N^2$	
Triangular interpolation	$5N^2$	-	$3N^2$	$2N^2$	
2D linear interpolation	$12N^2$	-	$11N^2$	$8N^2$	$4N^2$
Rectangular over 12 neighbours	$24N^2$	N^2	$15N^2$	$12N^2$	-
Triangluar over 12 neighbours	$13N^2$	-	$13N^2$	$4N^2$	-
Linear interpo. over 12 neig.	$36N^2$	N^2	$35N^2$	$28N^2$	$13N^2$
Rectangular over 16 neighbours	$32N^2$	N^2	$23N^2$	$16N^2$	-
Triangular over 16 neighbours	$16N^2$	N^2	$17N^2$	$4N^2$	-
Linear interpo. over 16 neig.	$48N^2$	N^2	$47N^2$	$40N^2$	$16N^2$

Where:

* Multiplication operator

/ Division operator

+ Addition operator

- Subtraction operator

√ Square root operator

Table 3.2 Comparison of CPU rotation time of the nearest neighbour techniques at 70° rotation.

Techniques		Methods	Rotation CPU time(sec)
Nearest neighbour techniques	Optimum techniques	Nearest	00:00.06
		Four	00:00.08
		Eight	00:00.11
	Normal techniques	Nearest	00:00.13
		Four	00:00.12
		Eight	00:00.15

Table 3.3 Comparison of CPU rotation time of the interpolation and multi-interpolation techniques at 70° rotation.

Techniques		Methods	Rotation CPU time(sec)
Interpolation techniques	Forward	Bilinear	00:00.65
		Triangular	00:00.62
		2D linear	00:01.75
	Reverse	Bilinear	00:00.62
		Triangular	00:00.58
		2D linear	00:01.64
Multi Interpolation techniques	Forward	Rectangular 12	00:01.64
		Triangular 12	00:00.93
		Linear 12	00:04.62
		Rectangular 16	00:01.32
		Triangular 16	00:01.07
		Linear 16	00:05.98
	Reverse	Rectangular 12	00:01.03
		Triangular 12	00:00.90
		Linear 12	00:04.66
		Rectangular 16	00:01.51
		Triangular 16	00:01.12
		Linear 16	00:05.73

CHAPTER 4

Image Rotation Using Filtering Techniques

4.1 Introduction

The principle objective of filtering techniques is to process the rotated image so that the result is suitable like the original image. Because processed images are usually stored as quantized pixels in computer memory, image processing tends to rely on digital filter as a discrete array of possibly complex numbers whose effect is to alter the spatial frequency content of an image during some processing operation (for example, rotation effect). The filtering operation can be in either image space or Fourier space. Three filter approach are applied to the rotated image. These filters are the moving average , finite impulse response (convolution) filter and the median filter.

4.2 The Moving Average Filter (MAF)

The moving average filter is a spatial domain technique. A window is chosen which encloses a predetermined neighbourhood of pixels. The average value of the pixels enclosed by this window is then computed and assigned to the pixel at the centre of the neighbourhood. By moving the window and repeating this process a neighbourhood averaged image is obtained. The size of the neighbourhood is defined by the user. Although different-shaped windows can be employed, a square

window is easier to use in practice. The size of the neighbourhoods is typically 3x3. The effect of computing the average of a neighbourhood of pixels is to eliminate any sudden jumps in the grey level which caused by the rotation process.

In mathematical term the neighbourhood averaging is given by [1]:

$$g(j, i) = \frac{1}{M} \sum_{(n,m) \in S} f(n, m) \quad (4.1)$$

Where :

S is the window enclosing $n \times m$ neighbours whose centre is located at (j, i)

M is the total number of pixels enclosed by S .

A test image of 128x128 resolution is shown in Figure 4.1. Figure 4.2 shows the test image after 70° rotation using the nearest neighbour technique. The result of applying the MA filter is shown in Figure 4.3.

4.3 Finite Impulse Response Filter (FIR)

The FIR (convolution) filters are the most commonly used in image processing due to their stability, zero phase, and ease of implementation. An FIR filter can be described mathematically as a convolution of the form [2]:

$$g(j, i) = \sum_{(n,m) \in S} p(n, m) f(j - n, i - m) \quad (4.2)$$

where:

$p(n, m)$ are the filter weights.

The result of applying this filter is shown in Figure 4.4.

4.4 Median Filter (MF)

A median filter is a nonlinear process which replaces the input sequence with the running median over a window of some specified length. The window always spans an odd number of points so that the median will always be one of the points. Let the window width be $2N + 1$, and let $med[x(m), x(n)]$ represent the median of the x values from $x(m)$ through $x(n)$. Then for an input sequence $\{x(i)\}$ [3],

$$y(i) = med(x(i - N), x(i + N)) \quad (4.3)$$

A brute-force implementation (programming) of a median filter simply copies the values $x(i - N), \dots, x(i + N)$, sorts the copied values, and outputs the central point in the sorted sequence (see Figure 4.5).

Note that if the central point in the window is already the median, it will be unchanged by the filter. Also, the output of a median filter is always one of the input points. If the central point is not the median, one of the neighbouring values, moved to the center by the sorting process, will be selected instead.

The begining and end of a sequence always present problems for the median filter. These problems are most commonly solved by appending N points at each end of the given sequence. The points appended to each end have values equal to the respective end points.

4.5 Comparison of Rotation Techniques.

Figure 4.2 is filtered using the MA, FIR, and median filter respectively. The result of these filtering process are shown in Figure 4.3 through Figure 4.5. A 3D-view of the test image after applying 70° rotation is shown in Figure 4.6. The missing dat represents the amount of noise in the image. The significance of these kind of filters can be seen by exammining the 3D-view of Figure 4.7 through Figure 4.9. As a result, the median filter completely removes the noise. A practical example of applying the median filter is shown in Figure 4.12. Table 4.1 illustrates the compar-

ison of CPU time of the different methods at 70^0 rotation using the aforementioned filtering techniques.

4.6 References

- [1] J.M. Blackledge,"Quantative coherent imaging", Academic press, 1989.
- [2] Anil k. Jain,"Fundamental of digital image processing", Prentice Hall, Englewood Cliffs, 1989.
- [3] R.A. Haddad and T.W. Parsons,"Digital signal processing: Theory, Application, and Hardware", computer science press, 1991.

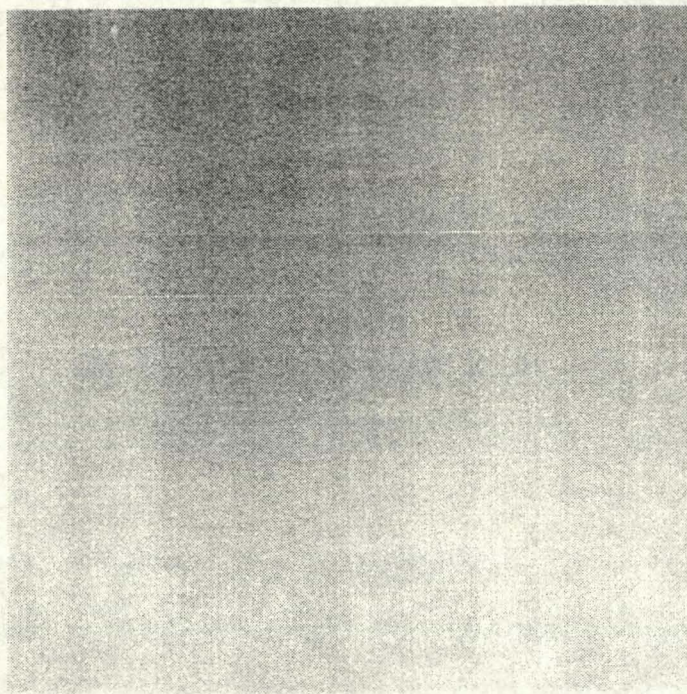


Figure 4.1 Test image of 128 x 128 resolution.

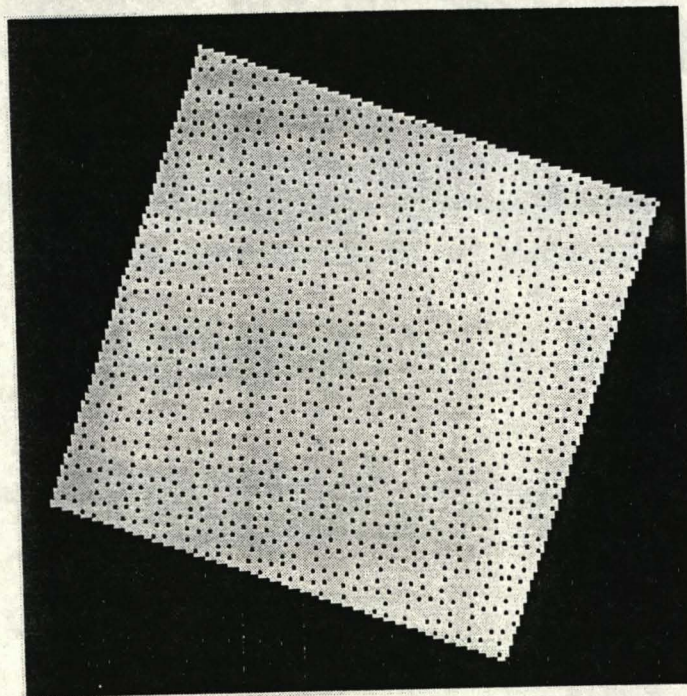


Figure 4.2 The test image after applying 70° rotation using nearest neighbour technique.

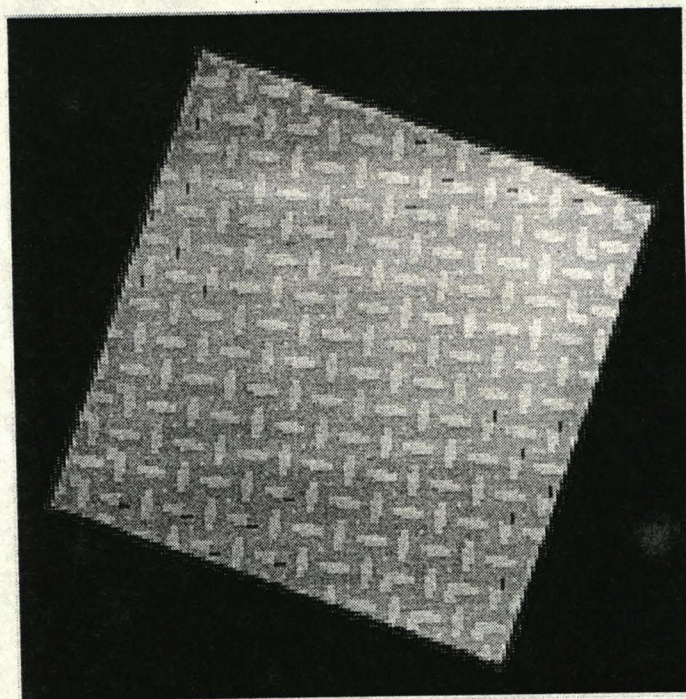


Figure 4.3 The test image after applying the moving average filter.

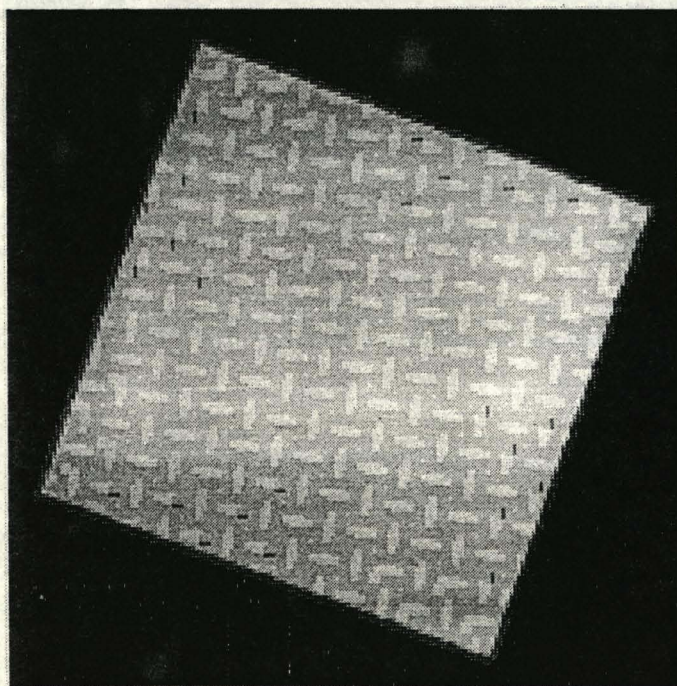


Figure 4.4 The test image after applying the finite impulse response filter.

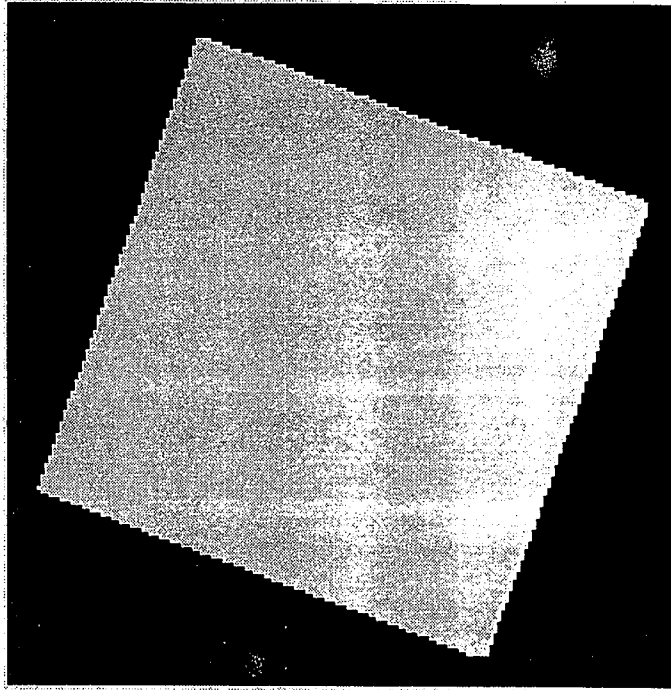


Figure 4.5 The test image after applying the median filter.

Filter type	CPU time (Sec.)
Moving Average	01.50
Finite Impulse Response	02.11
Median Filter	02.86

Table 4.1 The CPU time for each filtering technique.

Figure 4.6 3D-view after applying 70 rotation using nearest neighbour.

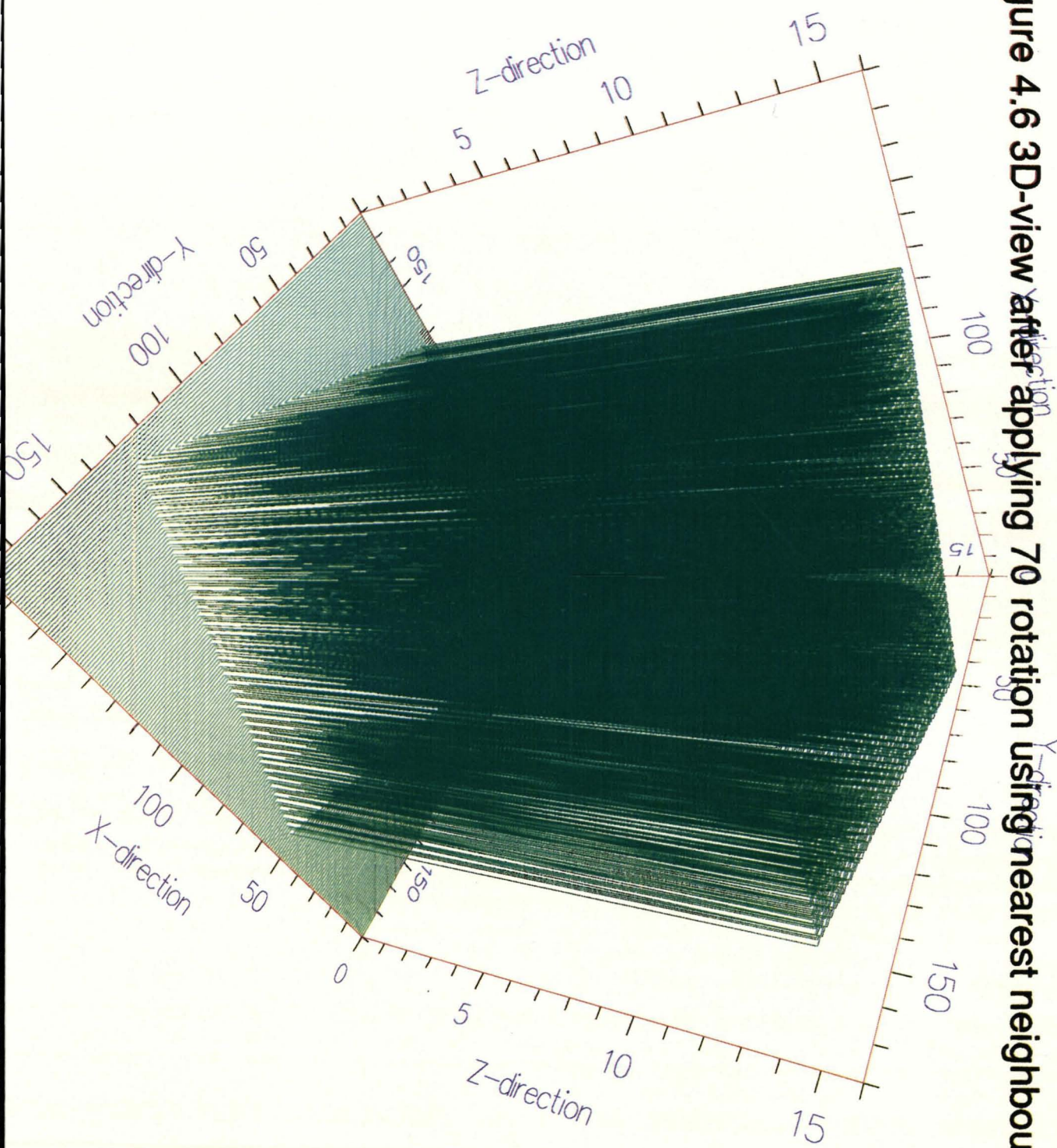


Figure 4.7 3D-view after applying the moving average filter.

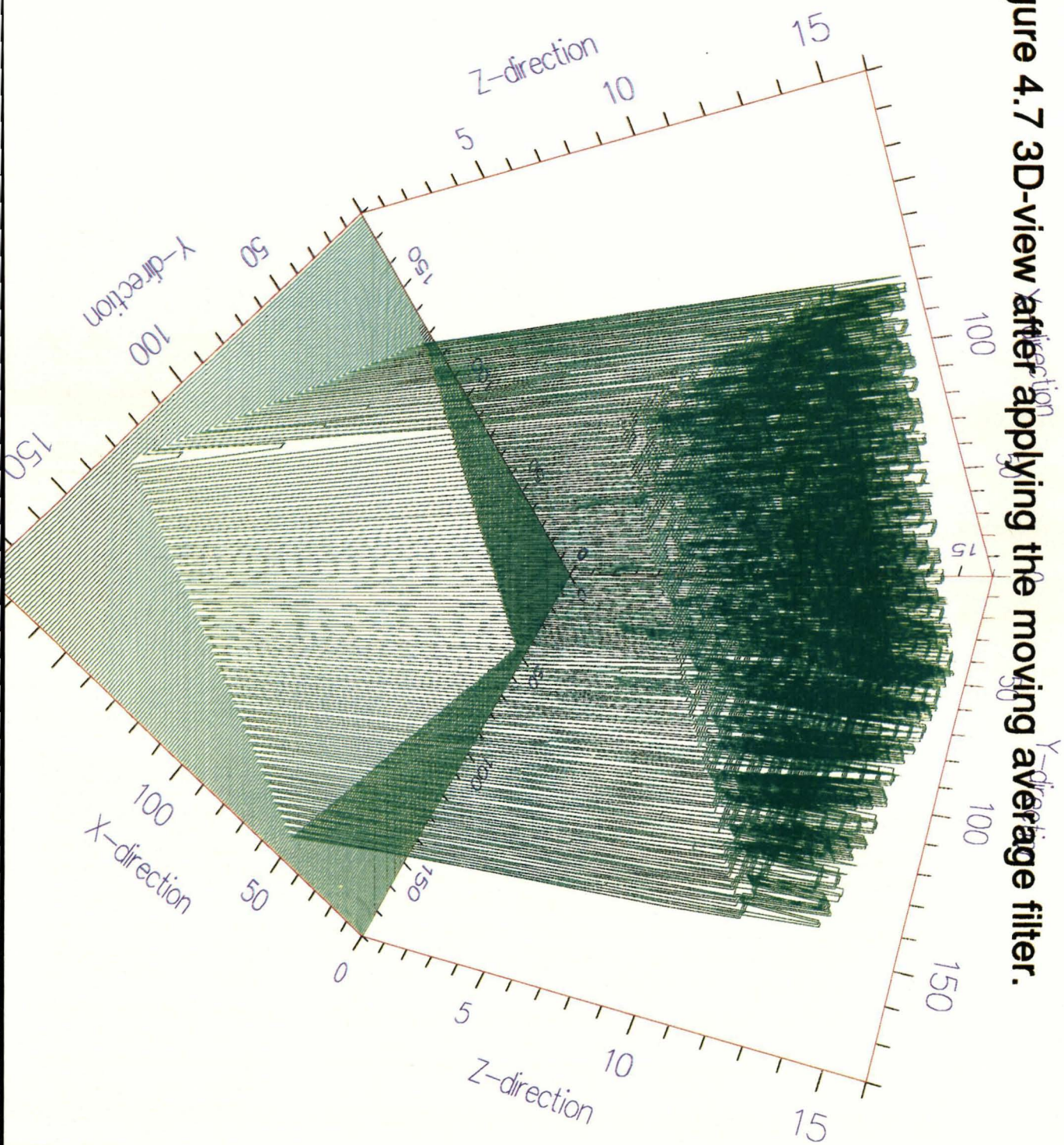


Figure 4.8 3D-view after applying the FIR filter.

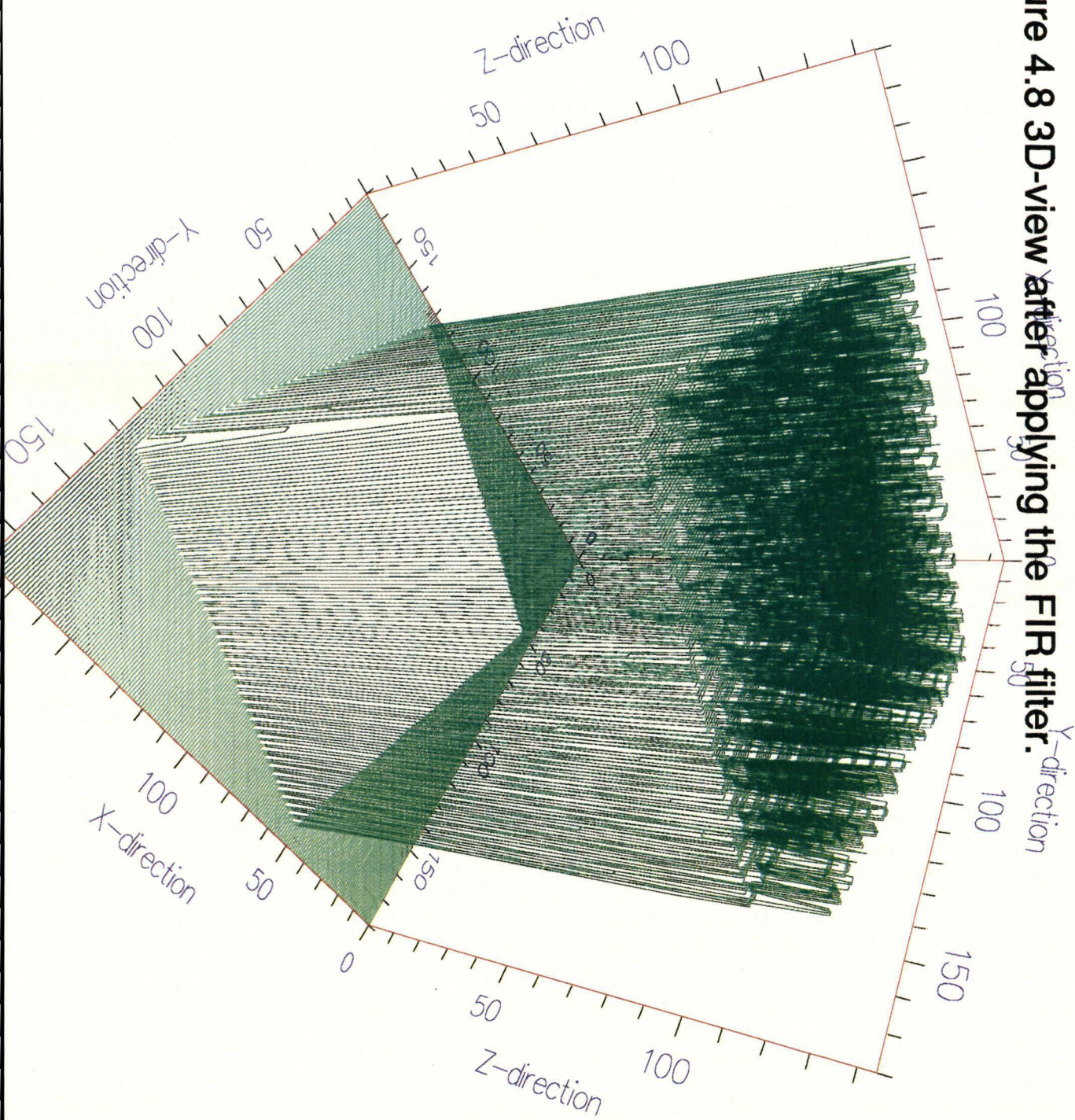


Figure 4.9 3D-view after applying the median filter:

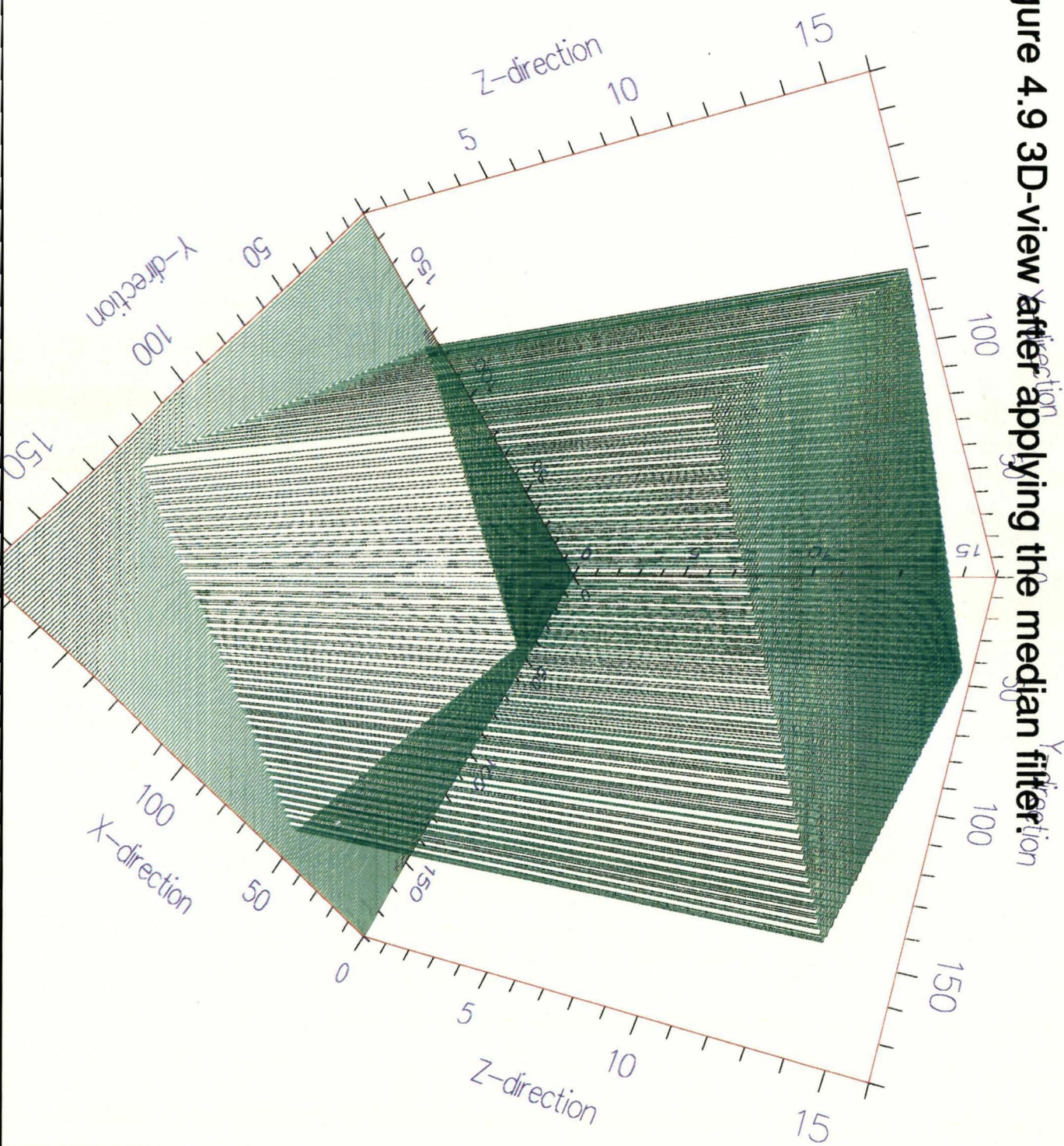


Figure 4.10 Practical example of 128 x 128 resolution.

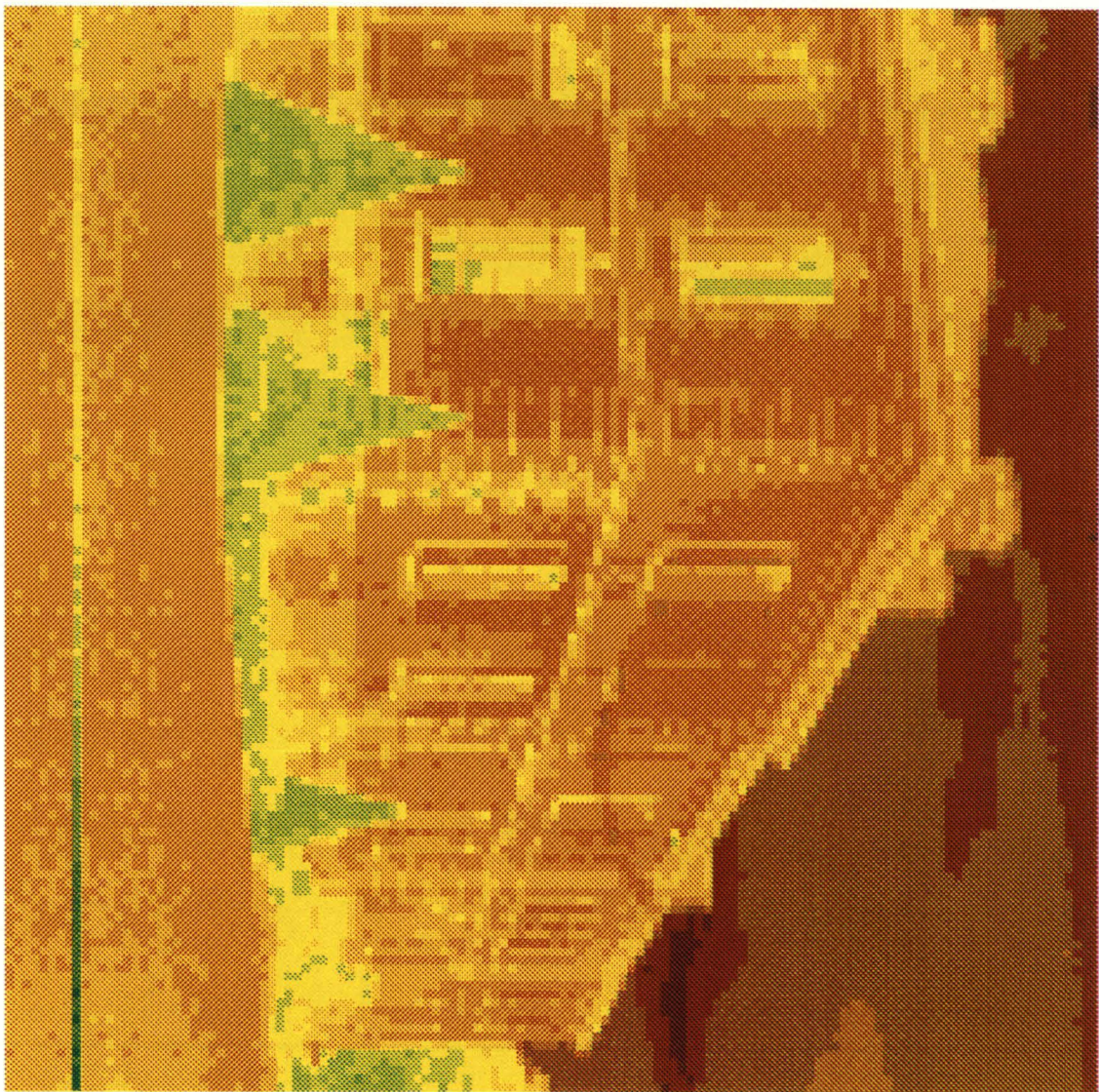


Figure 4.11 Practical example after applying 70 degree rotation using nearest neighbour technique.

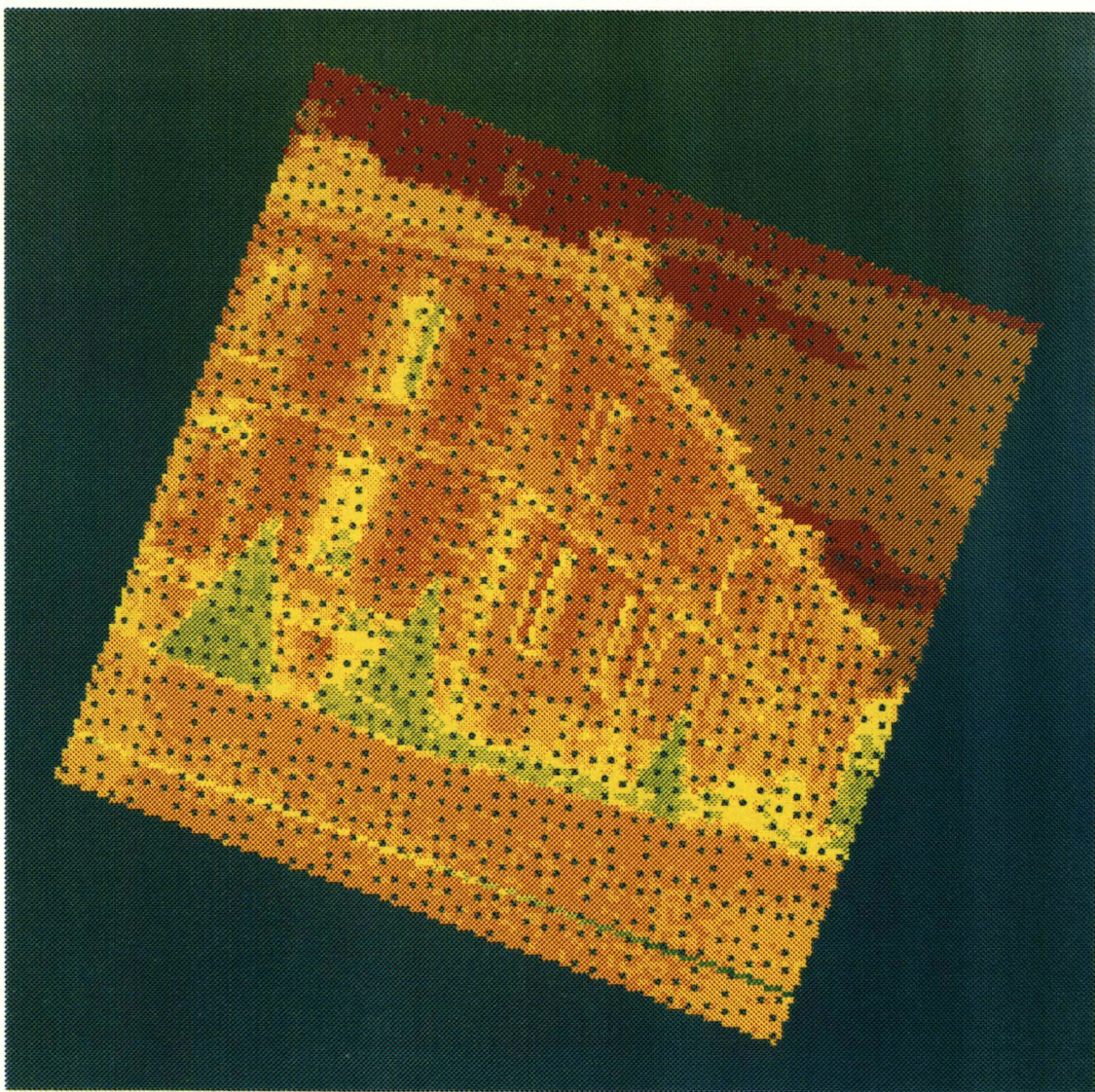
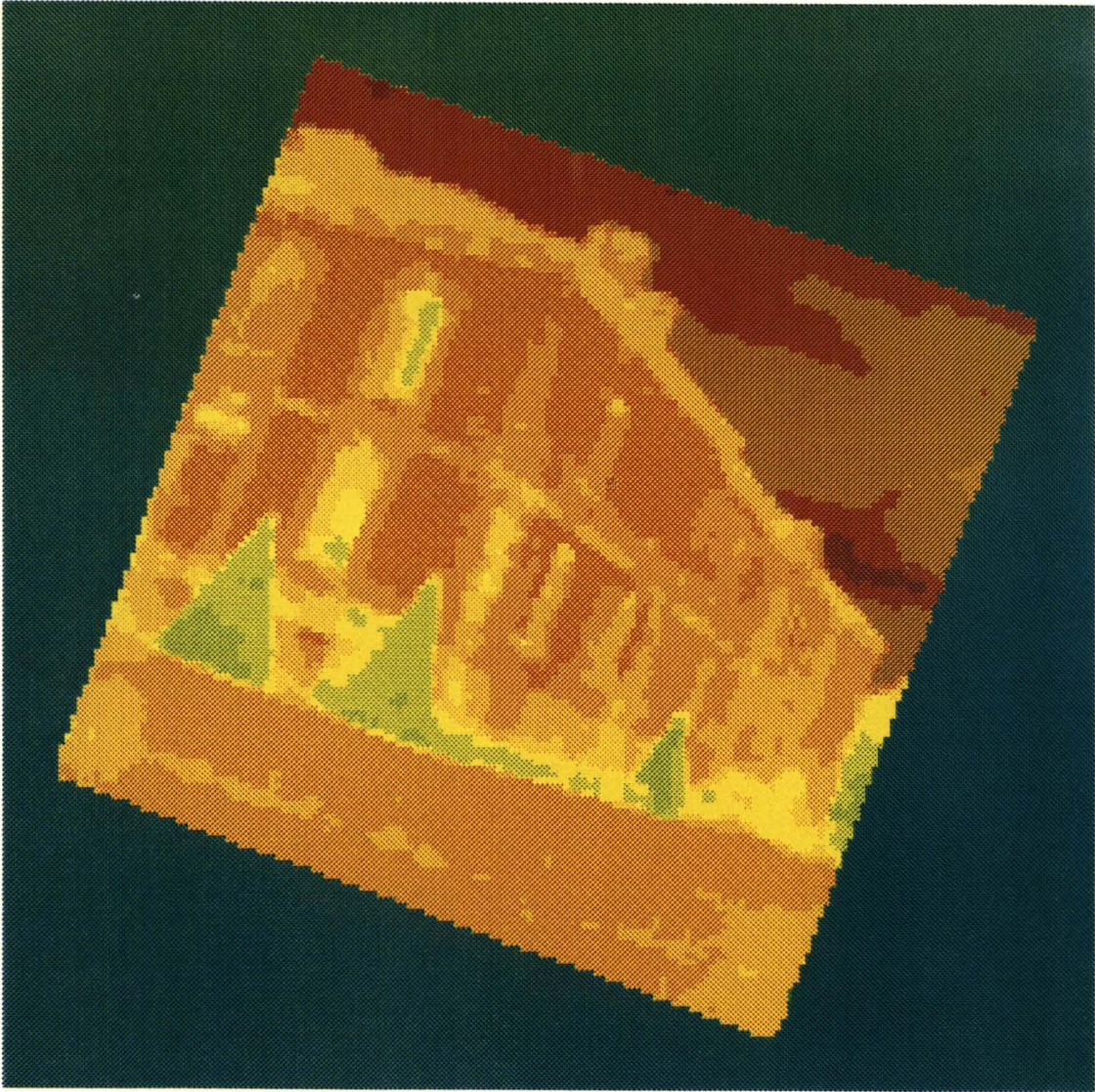
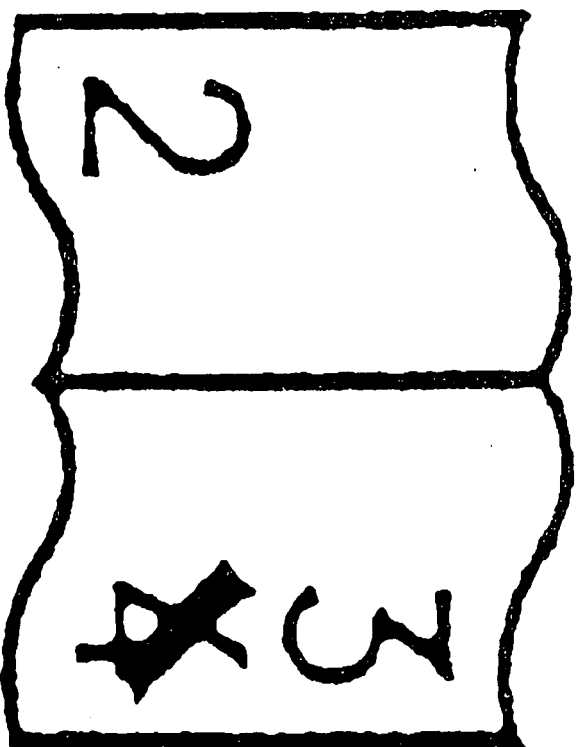


Figure 4.12 Practical example after applying the median filter.



PAGINATION ERROR



102-104

CHAPTER 5

Relation of Rotation to Other Transforms

5.1 Introduction

The material in this chapter deals primarily with the relation between the rotation process and Two-dimensional transform. Transform theory has played a key role in image processing for many years, and it continues to be a topic of interest in theoretical as well as applied work in this field.

5.2 Relation to Radon Transform

As in earlier discussions, the mathematical basis of projection tomography was derived in 1917 by the German mathematician Johannes Radon [1]. Radon proved that the complete set of one-dimensional projections obtained from a continuous two-dimensional function contains all the information required to reconstruct same function. A projection is obtained by integrating a function over a set of parallel lines and is characteristic of its angle of rotation in the Cartesian plane (see Chapter 6). Therefore, Radon transform is based on rotating the object about its centre and taking projections at regular steps in the angle of rotation.

5.3 Relation to the Fourier Transform

The Fourier transform of the function f can be written in the form [2]

$$F(k) = \hat{F}_1 f(x) = \int_{-\infty}^{\infty} f(x) \exp(-ikx) dx \quad (5.1)$$

The operator $\hat{F}_1$ is known as the Fourier operator. The hat denotes that $\hat{F}_1$ is a mathematical operator and the subscript 1 denotes the dimension of the operator.

Physically, the Fourier transform of a function provides a quantitative picture of the frequency content of the function. This is important in a wide range of physical problems and is fundamental to the processing and analysis of signals and images. The variable k has dimensions that are reciprocal to those of the variable x , there are two important cases that arise in imaging science:

- 1- x -time in seconds; k -temporal frequency in cycles per second (Hertz). Here, k is referred to as angular frequency which is given by $2\pi \times \text{frequency}$.
- 2- x -distance in meters, k -spatial frequency in cycles per meter. Here, k is known as the wave number and is given by:

$$k = \frac{2\pi}{\lambda}$$

Where:

λ is the wavelength.

The function $f(x)$ can be recovered from $F(k)$ by employing the inverse Fourier transform which is given by:

$$f(x) = \hat{F}_1^{-1} F(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k) \exp(ikx) dk \quad (5.2)$$

Where:

$\hat{F}_1^{-1}$ is the inverse Fourier operator, the superscript -1 is used to denote that this operator is an inverse operator.

5.3.1 Effect of rotation on Fourier transform

Recalling equation (2.1), the rotation of an image is represented as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.3)$$

Based on the aforementioned expression of matrix rotation, the transformation matrix is processed by using the vector notations as follows:

$$\vec{u} = R\vec{x}$$

where:

$$\vec{u} = \begin{pmatrix} u \\ v \end{pmatrix}$$

$$\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

R is the rotation matrix.

Consider the 2D object function $f(\vec{x})$, the Fourier transform is given by

$$F(\vec{k}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\vec{x}) e^{-j(\vec{k}^T \cdot \vec{x})} d\vec{x} \quad (5.4)$$

By rotating the function $f(\vec{x})$ by angle θ , then we have another object function is called $f(R\vec{x})$.

To figure out the effect of rotation on a function's Fourier transform, we evaluate:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(R\vec{x}) e^{-j(\vec{k}^T \cdot \vec{x})} d\vec{x} \quad (5.5)$$

By recasting the exponential term, we have:

$$\begin{aligned} \vec{k}^T \vec{x} &= \vec{k}^T R^{-1} R \vec{x} \\ &= (\vec{k}^T R^T) R \vec{x} \end{aligned} \quad (5.6)$$

note that $R^{-1} = R^T$

where the superscript T denotes the transposition.

Using the fact that:

$$(AB)^T = B^T A^T$$

then, equation (5.6) becomes:

$$\vec{k}^T \vec{x} = (R\vec{k})^T R\vec{x} \quad (5.7)$$

From equation (5.5) and equation (5.7) we have:

$$\begin{aligned} & \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(R\vec{x}) e^{-j((R\vec{k})^T R\vec{x})} d\vec{x} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\vec{\tau}) e^{-j((R\vec{k})^T \vec{\tau})} J d\vec{\tau} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\vec{\tau}) e^{-j((R\vec{k})^T \vec{\tau})} d\vec{\tau} = F(R\vec{k}) \end{aligned} \quad (5.8)$$

where

$$\vec{\tau} = R\vec{x}$$

$J = 1$, J is the Jacobian of the transformation.

Therefore, rotating a function rotates its transform. This is also true in higher dimensions.

5.3.2 Connection to Fourier transform

To connect the Fourier transform with the Radon transform, consider the two-dimensional object function $f(x, y)$, the Radon transform is given by (see Chapter 6) :

$$p(z, \theta) = \hat{R}f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(z - x \cos \theta - y \sin \theta) dx dy \quad (5.9)$$

Applying the Fourier transform to equation (5.9),

$$\begin{aligned} p(k, \theta) &= \hat{F}_1 p(z, \theta) \\ &= \hat{F}_1 \hat{R}f(x, y) = \int_{-\infty}^{\infty} dz e^{-ikz} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(z - x \cos \theta - y \sin \theta) dx dy \end{aligned}$$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-ik(x \cos \theta + y \sin \theta)} f(x, y) dx dy \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i(ux + vy)} dx dy \\
&= \hat{F}_2 f(x, y) = F(u, v)
\end{aligned} \tag{5.10}$$

Where:

$$\begin{aligned}
u &= k \cos \theta, & 0 \leq \theta \leq \pi \\
v &= k \sin \theta, & -\infty \leq k \leq \infty
\end{aligned}$$

$F(u, v)$ is the 2D Fourier transform in Polar coordinate.

Eqn. (5.10) illustrates the fundamental connection between the Fourier transform and the Radon transform [3]. See Fig. 5.1 for a diagrammatic interpretation. This relationship provides the theoretical basis for several image reconstruction algorithms.

The object function $f(x, y)$ can be obtained from the ordinary Fourier inversion formula by changing from polar (k, θ) to Cartesian coordinates (x, y) as follow:

The inverse Fourier transform of Eq.(5.10) is,

$$f(x, y) = \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^{\infty} F(u, v) e^{i(ux + vy)} du dv \tag{5.11}$$

Since, $F(u, v) = P(k, \theta)$ from Eq.(5.10) and by changing the variables in Eq.(5.11) in terms of (k, θ) , We get by definition,

$$f(x, y) = \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^{\infty} P(k, \theta) e^{ik(x \cos \theta + y \sin \theta)} J dk d\theta \tag{5.12}$$

where:

J is the Jacobian of the transformation.

By definition,

$$J = |k|$$

Equation (5.12) can be written as follow:

$$f(x, y) = \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^\infty P(k, \theta) e^{ik(x \cos \theta + y \sin \theta)} |k| dk d\theta \quad (5.13)$$

The factor $|k|$ that comes from the Jacobian is very important in image reconstruction. The result of computer simulation of Eq.(5.13) is shown in Chapter 8.

This leads to the projection theorem which states that the one-dimensional Fourier transform with respect to z of the projection $p(z, \theta)$ is equal to the central slice, at angle θ , of the two-dimensional Fourier transform of the object $f(x, y)$.

Thus given $p(z, \theta)$, it is possible to recover $f(x, y)$ by a radial Fourier transform followed by a 2-dimensional inverse Fourier transform.

5.4 Relation to the Hough Transform

An appropriate special case of a transform developed by J. Radon in 1917 [1] is discussed by Deans [4] to have the major properties of the Hough transform which is useful for finding line segments in digital pictures. The transform developed by Hough [5] has been discussed by several authors with interests in image analysis; for example, see Pratt [6], Duda and Hart [7], and Shapiro and Iannion [8] and references contained therein. The purpose of this communication is to point out that the Hough transform is deduced as a special case of a much more general transform now as the Radon transform which has been around for well over half a century.

The projection-space coordinates (z, θ) arise naturally because of the data gathering mechanics. This coordinate system plays an important role in many image processing applications. For example, the Hough transform, is useful for detection of straight-line segments of polygonal shapes, is a representation of a straight line in the projection space. As a result, image rotation is useful in image segmentation, whereas geometrical analysis of objects in processing applications requires transformation between polar and rectangular coordinates.

5.5 Relation to the Hotelling Transform

The principle uses of this transform are in data compression and rotation applications. The Hotelling transform is based on statistical properties of an image.

As an introduction to the problem, consider $N \times N$ image $f(x, y)$ is transmitted M times over some communication channel. Since any physical channel is subject to random disturbances, the set of received images, $\{f_1(x, y), \dots, f_M(x, y)\}$, represents a statistical ensemble whose properties are determined by the channel characteristics and the nature of the disturbance [9].

Many objects of interest in image processing applications can be easily standardized by performing a rotation that aligns the coordinate axes with the axes of maximum variance of the pixel in the object.

The original coordinates of each pixel in the object can be interpreted as two-dimensional random variables with mean :

$$m_x \cong \frac{1}{M} \sum_{i=1}^M x_i$$

and covariance matrix

$$c_x \cong \frac{1}{M} \left[\sum_{i=1}^M x_i x_i^T \right] - m_x m_x^T$$

Where:

M is the number of pixel in the object to be rotated.

x_i is the vector composed of the coordinates of i^{th} pixel.

Since the eigenvectors of c_x point in the direction of maximum variance, a logical choice is to select the new coordinate system so that it will be aligned with these eigenvectors (see fig. 5.2).

The rotation from the original coordinate system to this new system is given by the relation :

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (5.14)$$

which is seen to be in the familiar form $y = Ax$, where

$$\begin{aligned} A &= \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \end{aligned}$$

Equation 5.14 is analogous to a two-dimensional Hotelling transform where the mean has not been subtracted from the original vectors. Subtraction of the mean vector simply centralizes the object so that its centre of gravity is at the origin of the new coordinate system. In this case, the transformation has the form $y = A(x - m_x)$.

5.6 Relation to Abel Transform

Image rotations suggest that there are close relations between the Radon and the Abel transforms. To investigate these relations it is convenient to define the Radon transform through equation (6.1) given in Chapter 6 and recalled here.

$$\hat{R}f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(z - x \cos \theta - y \sin \theta) dx dy \quad (6.1)$$

This phenomena arises when circularly symmetrical distribution in two dimensions is projected in one dimension. A typical example is the electrical response of a television camera as it scans across a narrow line; another is the electrical response of a microdensitometer whose slit scans over a circularly symmetrical density distribution on a photographic plate.

Fractional-order derivatives are also closely connected with the Abel transform, which therefore also arises in fields, such as conduction of heat in solids or transmission of electrical signals through cables, where fractional-order derivatives are encountered.

In the case that the function $f(x, y)$ is rotationally symmetric, we can write :

$$f(x, y) \equiv f(r), \quad r = \sqrt{x^2 + y^2} \quad (5.15)$$

Where:

$$x = r \cos \theta$$

$$y = r \sin \theta$$

For the case of $\theta = 0$, eqn(6.1) can be written as:

$$\begin{aligned} \hat{R}f(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(z - x) dx dy \\ &= \int_{-\infty}^{\infty} f(y) dy \end{aligned} \quad (5.16)$$

From equation 5.15 we get: $y = \sqrt{r^2 - x^2}$ therefore,

$$dy = \frac{1}{\sqrt{r^2 - x^2}} \cdot r dr$$

Therefore, equation (5.16) is rewritten in the form

$$f_A(x) = \hat{R}f(x, y) = 2 \int_x^{\infty} \frac{f(r) r dr}{\sqrt{r^2 - x^2}} \quad (5.17)$$

Equation (5.17) is called Abel transform,

Where:

x is a rectangular coordinate (abscissa) and r is a radial coordinate.

Therefore, the Abel transform can be considered to be a special case of the more general Radon transform in the case when f is radially symmetric.

To get the inversion formula, equation (5.17) may be written in the following form:

$$f_A(x) = \int_0^{\infty} k(r, x) f(r) dr$$

where,

$$k(r, x) = \begin{cases} 2r(r^2 - x^2)^{-\frac{1}{2}}, & r > x ; \\ 0, & r < x. \end{cases}$$

A slight change of variable leads to a kernel $k(r, x)$ which simply shifts without change of form.

Thus putting $\xi = x^2$ and $\rho = r^2$, and letting $f_A(x) = F_A(x^2)$ and $f(r) = F(r^2)$, we have:

$$\begin{aligned} F_A(\xi) &= \int_0^\infty (\rho - \xi)^{-\frac{1}{2}} F(\rho) d\rho \\ &= \int_0^\infty K(\xi - \rho) F(\rho) d\rho \end{aligned} \quad (5.18)$$

where

$$K(\xi) = \begin{cases} (-\xi)^{-\frac{1}{2}}, & \xi < 0; \\ 0, & \xi \geq 0. \end{cases}$$

Eqn.(5.18) is the convolution integral, therefore, it can be written as

$$F_A = K \otimes F \quad (5.19)$$

Taking the Fourier transform of the above equation, we have

$$\tilde{F}_A = \tilde{K} \tilde{F}$$

Since

$$\tilde{K}(s) = \frac{1}{(-2is)^{\frac{1}{2}}}$$

It follows that:

$$\tilde{F} = (-2is)^{\frac{1}{2}} \tilde{F}_A$$

Multiplying the numerator and denominator of R.H.S. of the above equation by $\pi(-2is)^{\frac{1}{2}}$, we get

$$\begin{aligned}\tilde{F} &= -\frac{1}{\pi} \frac{1}{(-2is)^{\frac{1}{2}}} 2i\pi s \tilde{F}_A \\ &= -\frac{1}{\pi} K \otimes F'_A\end{aligned}$$

that is,

$$F(\rho) = -\frac{1}{\pi} \int_{\rho}^{\infty} \frac{d\xi}{(\xi - \rho)^{\frac{1}{2}}} \frac{d}{d\xi} F_A(\xi)$$

Reverting to f and f_A , we may write the solutions as

$$f(r) = \frac{-1}{\pi} \int_r^{\infty} \frac{1}{\sqrt{(x^2 - r^2)}} \frac{df_A(x)}{dx} dx \quad (5.20)$$

5.7 References

- [1] Radon, J., 1917 "Über die Bestimmung von Funktionen durch ihre Integralwerte Langs gewisser Mannigfaltigkeiten, Berichte Sachsische Akademie der wissenschaften", Leipzig, Math.-phys. Kl., 69, 262-267. (N.B. a translation of this paper is given in [2]).
- [2] Blackledge J. M., "Quantitative coherent imaging", Academic press, 1989.
- [3] Deans, S. R., The Radon Transform and some of its Applications, Wiley-Interscience, 1983.
- [4] S. R. Deans "Hough transform from Radon transform", IEEE Trans. pattern analysis and machine intelligence, vol. PAMI-3, No.2, March 1981.
- [5] P. V. C. Hough, "Method and means for recognizing complex patterns", U. S. Patent 3069654, 1962.
- [6] W. K. Pratt, "Digital image processing", New York; Wiley-Interscience, 1978.
- [7] R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures", Commun. Ass. Comput. Mach., vol. 15, pp. 11-15, Jan. 1972.

- [8] S. D. Shapiro and A. Iannino, "Geometric constructions for predicting Hough transform performance", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. PAMI-1, pp. 310-317, July 1979.
- [9] R. C. Gonzalez and P. Wintz, " Digital image processing", Second eddition, 1987.

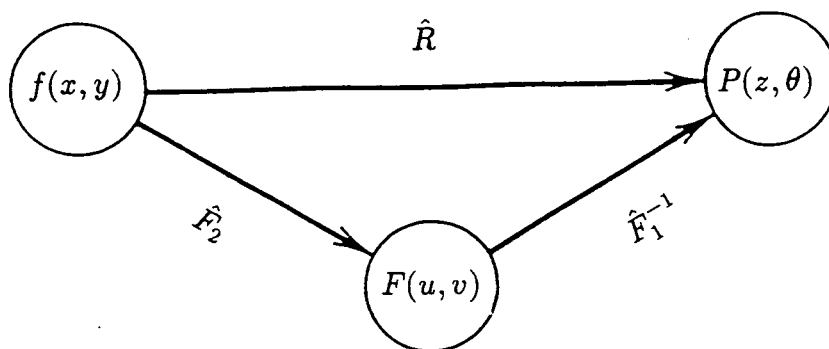


Figure 5.1 Relationship between rotation and Fourier transform.

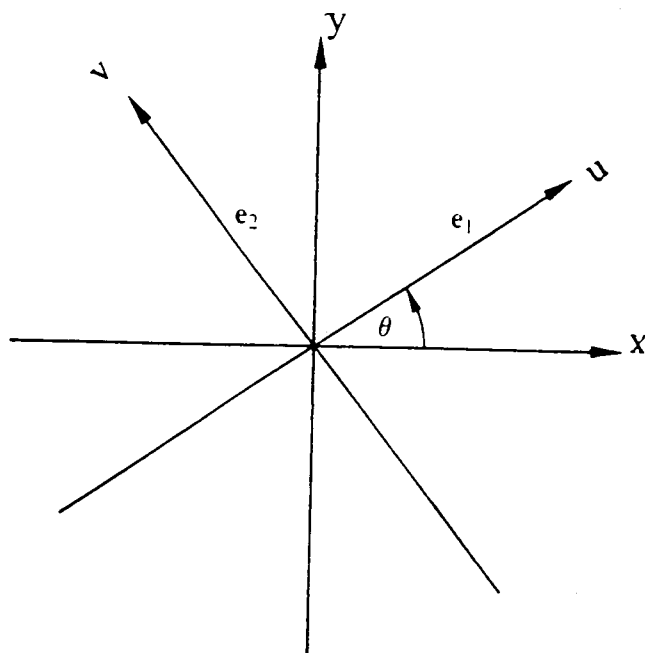


Figure 5.2 Rotation of coordinate system.

CHAPTER 6

The Radon Transform

6.1 Introduction

This chapter presents the theory of Radon transform-the central problem in C.T. Reconstructing object cross-sections from projections is a common problem in many scientific and medical areas. The original mathematics for reconstructing a 2D function from its projected line-integrals was developed by Radon in 1917 [1]. However, the first computed tomography researchers in the early 1960's were unaware of Radon's work, which was rediscovered later. Cormack [2] showed its importance in the field of radiology (for which he shared the 1979 Nobel prize). Bracewell [3] considered the same problem in regards to radio-astronomy applications. Other fields now making novel use of this technology include magnetic resonance imaging, ultrasound tomography, electron microscopy, optics, spotlight-mode synthetic aperture radar (SAR) [4]-[5], stress analysis, sonar array processing, and geophysics (see [6] for applications overview and bibliography).

Numerous books and articles have presented overviews of applying Radon's work to the particular problem of computed tomography. These include books by Herman [7], [8], [9], Barret and Swindell [10], Macovski [11], Stark [12], Bates and McDonnell [13], and Kak and Slaney [14]. General multidimensional signal processing problems, of which tomography is a sub-class, are described in [15], [16],

and [17]. Deeper mathematics of the Radon transform and image reconstruction can be found in books by Helgason [18], Deans [6], and Natterer [19]. Earlier work was summarized in articles by Mersereau and Oppenheim [20], Gordon and Herman [21], and Kak [22]. In additions, entire issues of technical journals have been devoted to this topic, including IEEE Transactions on Nuclear Science (April 1979), IEEE Transactions on Biomedical Engineering (February 1981), Proceedings of the IEEE (March 1983), Applied Optics (December 1, 1985), and IEEE Transactions on Medical Imaging (December 1986). Most relevant works of recent years have been published in the above journals as well as IEEE Transactions on Acoustics, Speech, and Signal Processing and The Journal of the Optical Society of America A: Optics and Image Science.

Obviously, it is requested to discuss the Radon transform, which provides the mathematical framework necessary for going back and forth between the spatial coordinates (x, y) and the projection coordinates (z, θ) .

6.2 The Radon Transform

The approach to computing the Radon transform is based on rotating the object about its centre and taking projections at regular steps in the angle of rotation. Therefore, to compute the Radon transform, first rotate a Cartesian array of the object about its centre and then compute the sum of the pixels which lie along each column or row of the rotated array (i.e., computes the family of discrete line integrals associated with a specific angle of rotation) [1].

The Radon transform of an object function $O(x, y)$, denoted as $P(z, \theta)$, is defined as its line integral along a line inclined at an angle θ from the X-axis and at a distance z from the origin (see Fig. 6.1). Mathematically, it is written as:

$$P(z, \theta) = \hat{R}O(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} O(x, y) \delta(x \cos \theta + y \sin \theta - z) dx dy \quad (6.1)$$

or, in polar form,

$$P(z, \theta) = \hat{R}O(x, y) = \int_0^\pi \int_{-\infty}^\infty O(r, \phi) \delta(r \cos(\theta - \phi) - z) |r| dr d\phi \quad (6.2)$$

Where:

the symbol $\hat{R}$, denoting the Radon transform operator, is also called the projection operator. The function $P(z, \theta)$, the Radon transform of $O(x, y)$, is the one-dimensional projection of $O(x, y)$ at an angle θ .

$\delta()$ is the Dirac delta function, and r and ϕ bear the standard polar relationship with x and y shown in Figure 6.2a:

$$\begin{aligned} x &= r \cos \phi & r &= \sqrt{x^2 + y^2} \\ y &= r \sin \phi & \phi &= \tan^{-1}(y/x) \end{aligned} \quad (6.3)$$

For a fixed point (r, ϕ) , this equation gives the locus of all points in (z, θ) , which is a sinusoid as shown in Fig. 6.2b.

The Radon transform maps the spatial domain (x, y) to the domain (z, θ) . Each point in the (z, θ) space corresponds to a line in the spatial domain (x, y) . Notice that the Dirac delta function of (6.1) and (6.2) are selecting only those points for integration that lie on line.

The alternative formulation can be found by rotating the (x, y) coordinate system by θ and labelling the new axes (u, v) as was shown in equation (2.1) (Chapter 2). With these, we can change the integration variables of (6.1) to u and v , which lets us separate and remove a $\int_{-\infty}^\infty \delta(u - z) du$ term to arrive at

$$P(z, \theta) = \int_{-\infty}^\infty O(z \cos \theta - v \sin \theta, z \sin \theta + v \cos \theta) dv \quad (6.4)$$

A value of $P(z, \theta)$ on a single line (i.e., for fixed z and θ) is often called a line integral or ray sum. The equivalent representations of $P(z, \theta)$ shown in (6.1), (6.2), and (6.4) are known as the (2D) Radon transform of $O(x, y)$.

6.3 The Inverse Radon Transform:

The image reconstruction problem is theoretically equivalent to finding the inverse Radon transform of $P(z, \theta)$. There are several ways to invert the Radon transform for the analytic case where all the projections in the interval $0 \leq \theta \leq \pi$ are known. The result is summarized by the following theorem.

Given

$$P(z, \theta) = \hat{R}O(x, y), \quad -\infty < z < \infty, \quad 0 \leq \theta < \pi$$

then, the inverse Radon transform is:

$$O(r) = \hat{R}^{-1}P(\hat{n}, z) = -\frac{1}{2\pi^2} \int_0^\pi d\theta \int_{-\infty}^\infty dz \frac{1}{z - \hat{n} \cdot r} \frac{\partial}{\partial z} P(\hat{n}, z) \quad (6.5)$$

Where:

$\hat{R}^{-1}$ is the inverse Radon operator.

The variable P denotes the projection or line integral of O onto a line perpendicular to the direction of $\hat{n}$ defines the angle of rotation of the object and so,

$$P(\hat{n}, z) \equiv P(\theta, z)$$

The variable z may be interpreted as defining the location of $\hat{n}$ at which the rectilinear line integral through O is taken. Therefore, the object function O can be recovered or reconstructed from P using equation (6.5).

6.4 Summary

In this chapter, we have shown the mathematical groundwork of the CT reconstruction problem. The Radon transform and its inverse were described. Many conventional algorithms for inversion of the Radon transform will be pointed out in Chapter 8.

Figure 6.3 through Figure 6.5 show images of the object and Radon spaces. A simple simulated object function containing the cross-section of three cylinders can be seen schematically in Figure 6.3a and as an image in Figure 6.3b. The Radon transform of the object is usually displayed in the form shown in Figure 6.4, called a sinogram because a distinct (x, y) location in the object space traces a sinusoidal path through this space. This is the typical form of the raw data acquired from a CT scanner.

Our interest in the Radon transform is to find its inverse. That is, given the measurements of $P(z, \theta)$ for many projected lines around the object, find an estimate of the original cross-section function $O(x, y)$ in the slice. One possible estimate of the object described above from its Radon space is shown in Figure 6.5.

6.5 References

- [1] Radon, J., 1917 "Über die Bestimmung von Funktionen durch ihre Integralwerte Langs gewisser Mannigfaltigkeiten, Berichte Sachsische Akademie der wissenschaften," Leipzig, Math.-phys. Kl., 69, 262-267. (N.B. a translation of this paper is given in [6]).
- [2] A. M. Cormack, "Representation of a function by its line integrals with some radiological applications," J. Applied Phys., vol. 34, pp. 2722-2727, 1963.
- [3] R. N. Bracewell, "Strip integration in radio astronomy," Aust. J. Phys. , vol.9 pp. 198-217, 1956.
- [4] D. C. Munson, Jr., J. D. O'Brien, and W. K. Jenkins, "A Tomographic formulation of spotlight-mode synthetic aperture radar," Proceedings of IEEE, vol.71, No. 8, pp. 917-925, August 1983.
- [5] J. P. Fitch, "Synthetic aperture radar," Springer-Verlag, New York, 1988.
- [6] Deans, S. R., The Radon Transform and some of its Applications, Wiley-

Interscience, 1983.

- [7] G. T. Herman, "Image reconstruction from projections: The fundamentals of computerized tomography," Academic press, 1980.
- [8] G. T. Herman, editor, "Image reconstruction from projections: Implementation and application," Volume 32 of Topics in applied physics, Springer-Verlag, 1979.
- [9] G. T. Herman and H. K. Tuy, "Image reconstruction from projections: an approach from mathematical analysis. In P. C. Sabatier, editor, Basic methods of tomography and inverse problems, pages 3-126, Adam Hilger, 1987.
- [10] H. H. Barrett and W. Swindell, "Radiological imaging: Theory of image formation, detection and processing," Academic Press, New York, 1981.
- [11] A. Macovski, "Medical imaging systems," Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.
- [12] H. Stark, editor, "Image recovery: Theory and application," Academic Press, New York, 1987.
- [13] R. H. T. Bates and M. J. McDonnell, "Image restoration and reconstruction," Clarendon Press, Oxford, 1986.
- [14] A. C. Kak and M. Slaney, "Principles of computerized tomographic imaging," IEEE Press, New York, 1988.
- [15] S. Haykin, editor, "Array signal processing," Prentice-Hall, New Jersey, 1985.
- [16] D. E. Dudgeon and R. M. Mersereau, "Multidimensional digital signal processing," Prentice-Hall signal processing series, Prentice-Hall, 1984.
- [17] Anil K. Jain, "Fundamentals of digital image processing," Prentice Hall, Englewood Cliffs, 1989.
- [18] S. Helgason, "The Radon transform," Volume 5 of progress in mathematics,

Birkhauser, 1980.

- [19] F. Natterer, "The mathematics of computerized tomography," Wiley, Chichester, 1986.
- [20] R. M. Mersereau and A. V. Oppenheim, "Digital reconstruction of multidimensional signals from their projections," Proceedings of the IEEE, vol.62, No.10, pp. 1319-1338, October 1974.
- [21] R. Gordon and G. T. Herman, "Three-Dimensional reconstruction from projections: A review of algorithms," In G. H. Bourne and J. F. Danielli, editors, Int'l. Rev. of Cytology, Pages 111-151, Academic Press, New York, 1974.
- [22] Kak A. C., "Computerized tomography with x-ray, emission, and ultrasound sources. ", proc. IEEE, vol.67, No.9, pp. 1245-1272, September 1979.

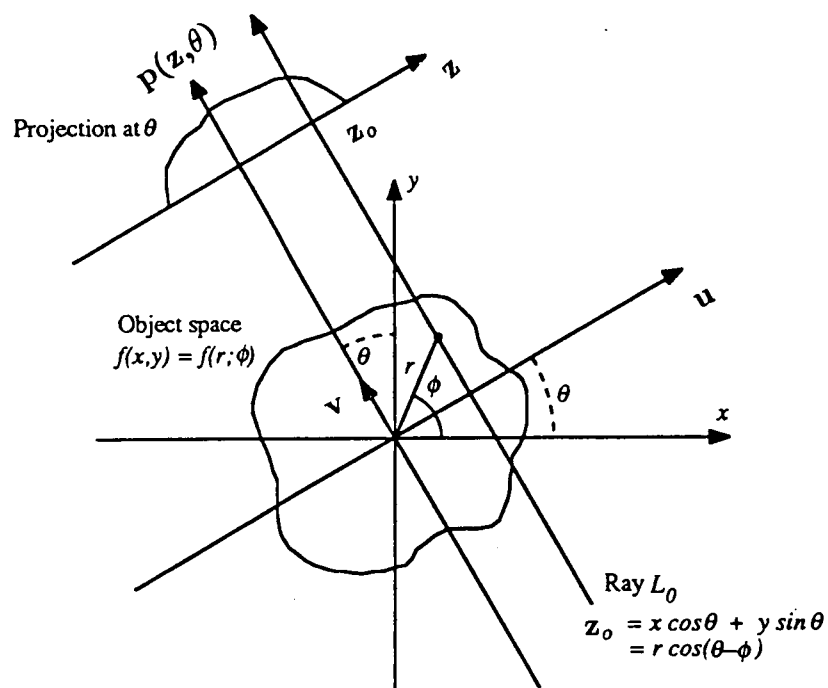
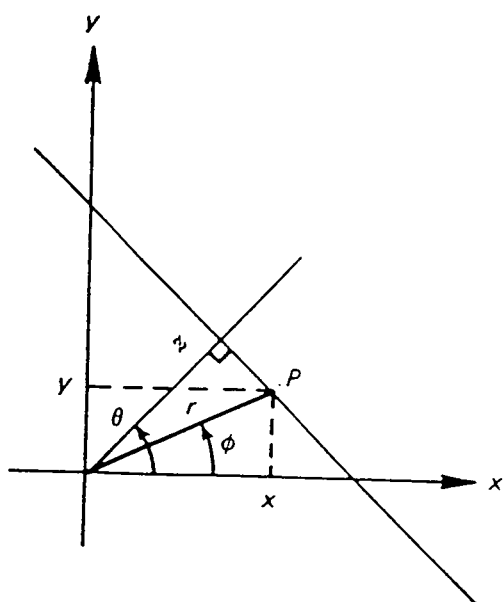
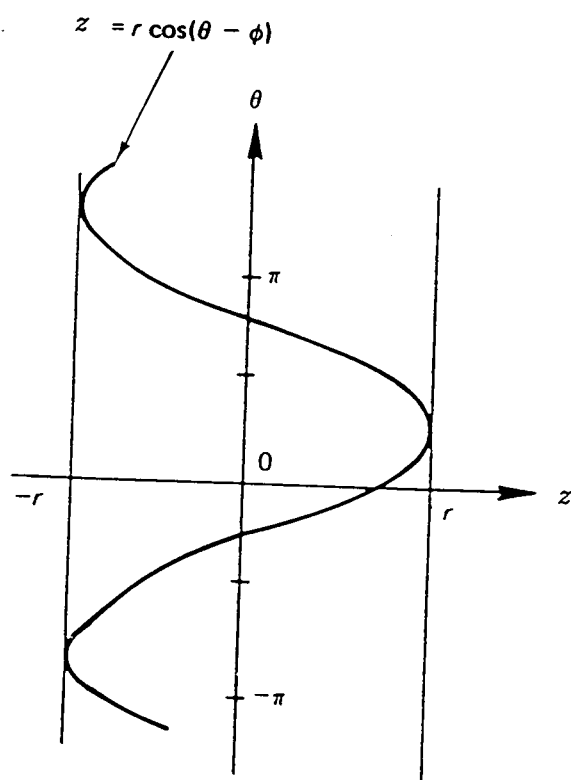


Figure 6.1 Coordinate systems of computed tomography.



a) Spatial domain (x, y) .



b) The point p maps into sinusoid in the (z, θ) plane.

Figure 6.2 Spatial and Radon transform domains.

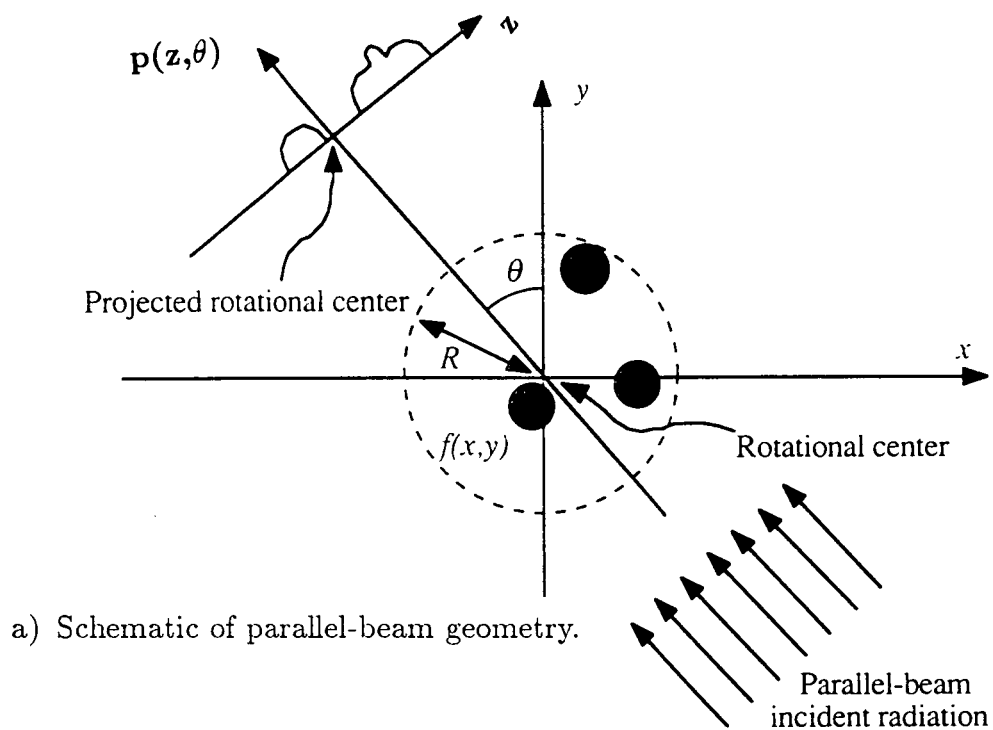
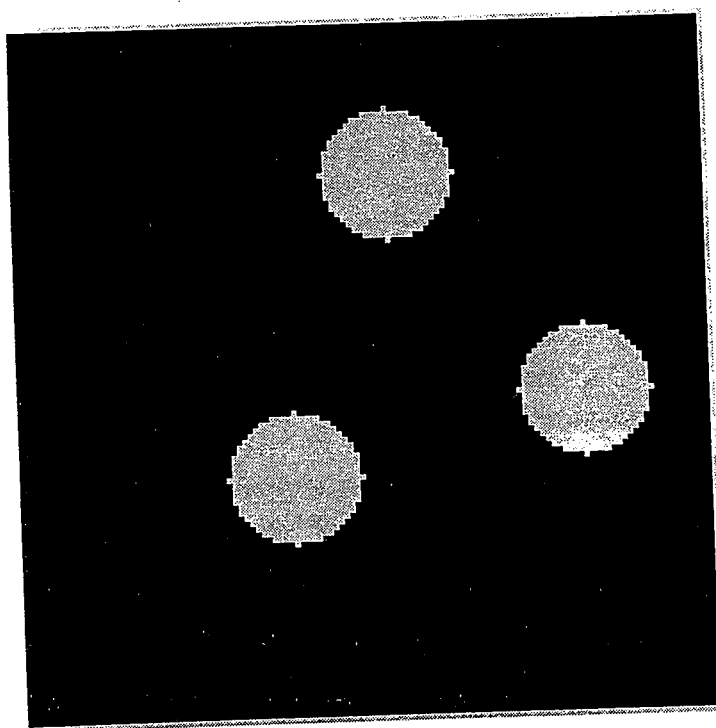


Figure 6.3 Images of parallel-beam projection CT for one cross-sectional plane through three cylinders.

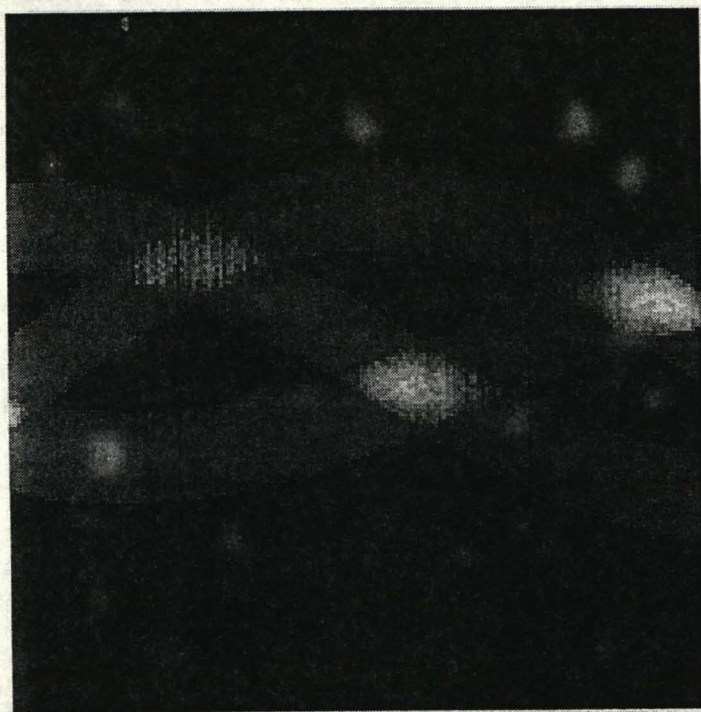


Figure 6.4 Parallel Radon space of the object over 180 degrees.

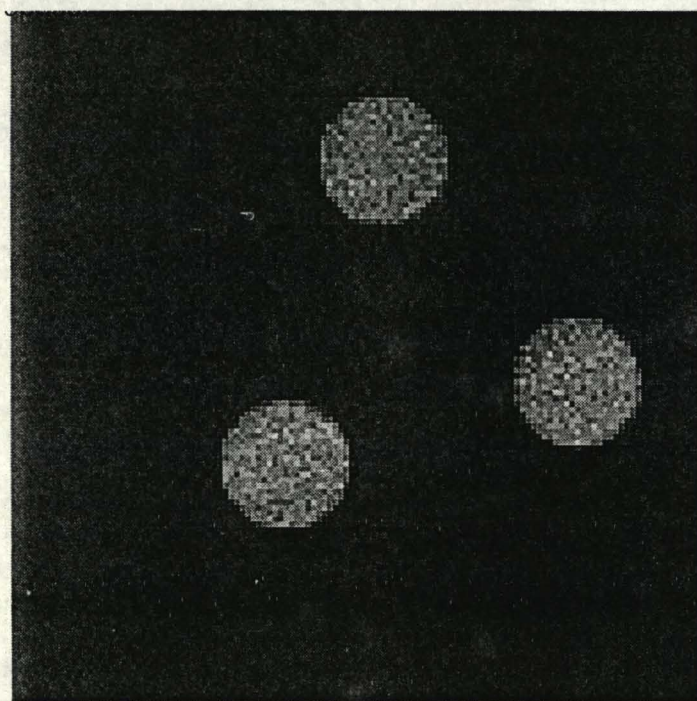


Figure 6.5 Estimated reconstruction of the object from the Radon space.

Chapter 7

Applications For Implementing 1D Processes

7.1 Introduction

There is a wide variety of signal and image processing techniques that are in constant use today. These methods have a wide range of applications. In this Chapter, we focus the attention on reducing the problem of implementing two dimensional signal processing algorithms that are computationally intensive and limited by the array size. The approach is considered that over-rides the computational difficulties of multi-dimensional processing by reducing the dimensionality of the problem to 1D. That is, instead of working (i.e., formulating and computing) in 2D or 3D, a transform is used in such a way that the computational aspects of a given procedure are reduced to 1D and the result then transformed back to 2D or 3D respectively. This means that any available 1D digital process can automatically be used without modification to process a 2D image or 3D data base. Here, only the 2D case (i.e., image processing) is considered and the transform used is the Radon transform, which has an exact analytical inverse.

7.2 Computing Projections of Digital Images

The Radon transform method for digital image processing can in principle be used for any linear processing scheme but it is most useful for those techniques which

in 2D involve the inversion of block matrix (i.e., matrix inversion for 2D arrays) [1]. Therefore, the Radon transform method offers a computational efficiency which in this case, cannot be matched by computing in 2D.

Using the central slice theorem discussed in Chapter 8 which states that 1D Fourier transform of a given projection $p_\theta(x)$ is equal to a slice at an angle θ through the Fourier plane of $I(x, y)$. With the convolution theorem, it is also the basis for the following theorem:

Theorem:

If

$$\hat{P}_{2D}I(x, y) = H(x, y) \otimes I(x, y) + A(x, y) \quad (7.1)$$

and

$$\hat{P}_{1D}f_\theta(x) = h_\theta(x) \otimes f_\theta(x) + a_\theta(x) \quad \forall \theta; 0 \leq \theta \leq \pi \quad (7.2)$$

where

$\hat{P}_{2D}$ is any linear 2D processing operator with a 1D version $\hat{P}_{1D}$ such that:

$$\hat{R}I(x, y) = f_\theta; \quad 0 \leq \theta \leq \pi \quad (7.3)$$

$$\hat{R}H(x, y) = h_\theta; \quad 0 \leq \theta \leq \pi \quad (7.4)$$

and

$$\hat{R}A(x, y) = a_\theta(x); \quad 0 \leq \theta \leq \pi \quad (7.5)$$

where

$\hat{R}$ is the Radon transform operator,

Then

$$\hat{P}_{2D}I(x, y) = \hat{R}^{-1} \hat{P}_{1D} \hat{R}I(x, y) \quad (7.6)$$

where

$\hat{R}^{-1}$ is the inverse Radon transform and $\otimes \otimes$ and $\otimes$ denote the 2D and 1D convolutions respectively.

Proof:

From the central slice theorem we can write

$$\hat{F}_{1D} \hat{R} I(x, y) = \hat{F}_{2D} I(x, y)|_{slice} \quad (7.7)$$

where

$\hat{F}_{1D}$ and $\hat{F}_{2D}$ are 1D and 2D Fourier transform operators. Thus, from equation (7.3) we have:

$$\hat{F}_{1D} f_{\theta}(x) = \hat{F}_{2D} I(x, y)|_{slice} \quad (7.8)$$

and similarly equations (7.4) and (7.5)

$$\hat{F}_{1D} h_{\theta}(x) = \hat{F}_{2D} H(x, y)|_{slice} \quad (7.9)$$

and

$$\hat{F}_{1D} a_{\theta}(x) = \hat{F}_{2D} A(x, y)|_{slice} \quad (7.10)$$

We are therefore able to construct the following equation

$$\begin{aligned} & \hat{F}_{2D} H(x, y)|_{slice} \hat{F}_{2D} I(x, y)|_{slice} + \hat{F}_{2D} A(x, y)|_{slice} \\ &= \hat{F}_{2D} H(x, y) \hat{F}_{2D} I(x, y)|_{slice} + \hat{F}_{2D} A(x, y)|_{slice} \\ &= \hat{F}_{1D} h_{\theta}(x) \hat{F}_{1D} f_{\theta}(x) + \hat{F}_{1D} a_{\theta}(x) \end{aligned} \quad (7.11)$$

Operating on both sides of this equation by the 1D inverse Fourier transform gives

$$\begin{aligned} & \hat{F}_{1D}^{-1} [\hat{F}_{2D} H(x, y) \hat{F}_{2D} I(x, y)]_{slice} + \hat{F}_{1D}^{-1} [\hat{F}_{2D} A(x, y)]_{slice} \\ &= h_{\theta}(x) \otimes f_{\theta}(x) + a_{\theta}(x) \end{aligned} \quad (7.12)$$

where we have used the convolution theorem

$$\hat{F}_{1D}^{-1}[\hat{F}_{1D}h_{\theta}(x)\hat{F}_{1D}f_{\theta}(x)] = h_{\theta}(x) \otimes f_{\theta}(x) \quad (7.13)$$

Finally, we note that for any 2D function $G(x, y)$

$$\hat{F}_{1D}^{-1}\hat{F}_{2D}G(x, y)|_{slice} = \hat{R}\hat{F}_{2D}^{-1}\hat{F}_{2D}G(x, y) \quad (7.14)$$

Thus, using the 2D convolution theorem

$$\hat{F}_{2D}H(x, y)\hat{F}_{2D}I(x, y) = \hat{F}_{2D}[H(x, y) \otimes \otimes I(x, y)] \quad (7.15)$$

equation (7.12) becomes

$$\hat{R}[H(x, y) \otimes \otimes I(x, y)] + \hat{R}A(x, y) = h_{\theta}(x) \otimes f_{\theta}(x) + a_{\theta}(x) \quad (7.16)$$

and from equation (7.1) and (7.2) we can write

$$\hat{R}\hat{P}_{2D}I(x, y) = \hat{P}_{1D}f_{\theta}(x) \quad (7.17)$$

Finally, since for any angle θ , $f_{\theta}(x)$ is given by equation (7.3), we have

$$\hat{R}\hat{P}_{2D}I(x, y) = \hat{P}_{1D}\hat{R}I(x, y) \quad (7.18)$$

and by operating on both sides of this equation by $\hat{R}^{-1}$ where $\hat{R}^{-1}\hat{R} = 1$, we obtain

$$\hat{P}_{2D}I(x, y) = \hat{R}^{-1}\hat{P}_{1D}\hat{R}I(x, y) \quad (7.19)$$

which complete the proof. This is a particularly important result. It states that any linear 2D filtering operation on a 2D function can be simulated exactly and uniquely by first taking the Radon transform of the function, operating on each resulting projection independently by the equivalent 1D filter and then using the inverse Radon transform to reconstruct. In terms of operators, we have the fundamental equivalence statements.

$$\hat{P}_{2D} = \hat{R}^{-1}\hat{P}_{1D}\hat{R} \quad (7.20)$$

The importance of this theorem becomes clear when we consider any 2D operator as a digital processing scheme which requires extensive computation in contrast to the 1D version of the scheme that may be computationally straight forward.

7.3 Test Example

Figure (7.1) represents the test image. By implementing equation (7.20), the left hand term is shown in Figure (7.2) by applying 2D median filter on the given test image. On the other hand, Radon transform is applied for the test image, then 1D median filter is processed to the projection and finally the inverse Radon transform is applied and the result is given in Figure (7.3) which should be similar as Figure (7.2).

7.4 Error Discussion

In any numerical treatment of the Radon transform and its inverse, a certain degree of corruption is inevitable due to the discretization of analytical formulae. It is evident that projection data is only required from 0° to 180° (i.e., the integral is evaluated from 0 to π). However, in practice, any reconstruction using this formula suffers from noise generated by numerical inaccuracies which are primarily due to the rotation of a discrete Cartesian array.

7.5 The Computational CPU Time

The computational CPU time versus the window size of the above example is shown in Table 7.1. This Table shows the CPU time versus the window size for implementing the 1D process (1D MF) using FRT and the 2D process (2D MF). The diagramatic representation of this Table is shown in Figure 7.4. It is evident that as the window size increases, the CPU time for the 2D process is drastically increased. On the other hand, the 1D process using FRT is approximately constant

along the hole window range. Therefore, for higher window size (approximately more than 17 pixels), the 1D process using FRT is more efficient compared to the 2D process.

7.6 The Mathematical Model

The mathematical model describing the above Figure using the polynomial least-squares method is given by

$$f(x) = \sum_{i=0}^N a_i x^i$$

where:

a_i the polynomial coefficients.

N the polynomial degree.

• For 1D process using FRT:

The mathematical model is given by:

$$\begin{aligned} f(x) &= a_0 + a_1 x \\ &= 111.901 + 0.63x \end{aligned}$$

which represents equation of straight line (See Fig.7.5a).

• For 2D process:

The mathematical model is given by:

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 \\ &= 31.53 - 16.82x + 2.93x^2 - 0.2x^3 + 0.006x^4 \end{aligned}$$

which represents a polynomial of degree 4 (See Fig.7.5b).

7.7 References

- [1] J.M. Blackledge, "Digital image processing in Radon space and the inversion of limited Fourier data," Optik, vol.73, No.2, pp. 74-82, 1986.

- [2] N. Srinivasa, and et all, "Two-Dimensional spectral estimation: A Radon transform approach," IEEE J.oceanic Eng., vol.OE-12, No.1, pp. 90-96, Jan. 1987.

Figure 7.1a Head-Phantom image.



Figure 7.1b Head-Phantom image with two spikes.

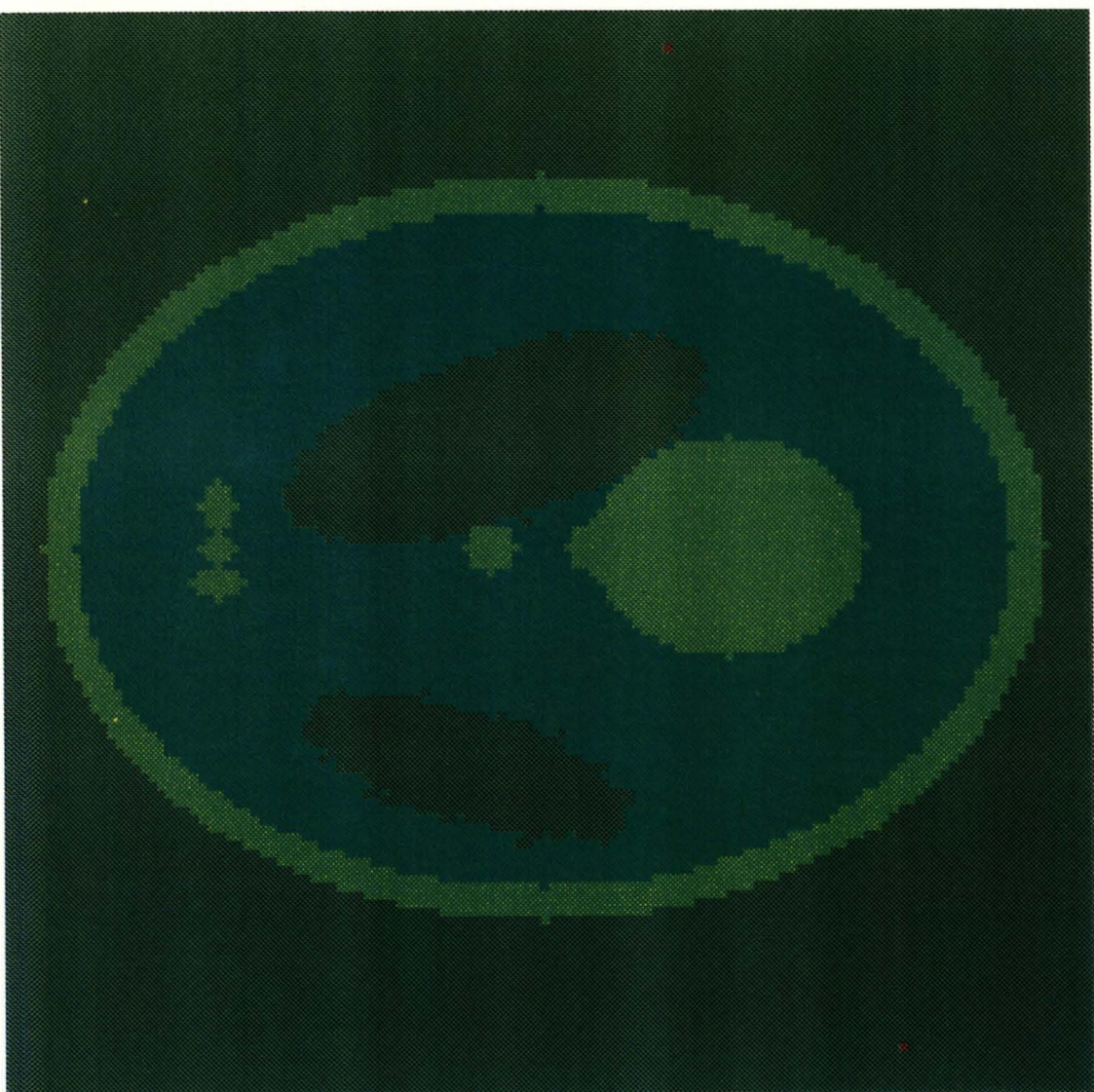


Figure 7.2 The result after applying 2D median filter.

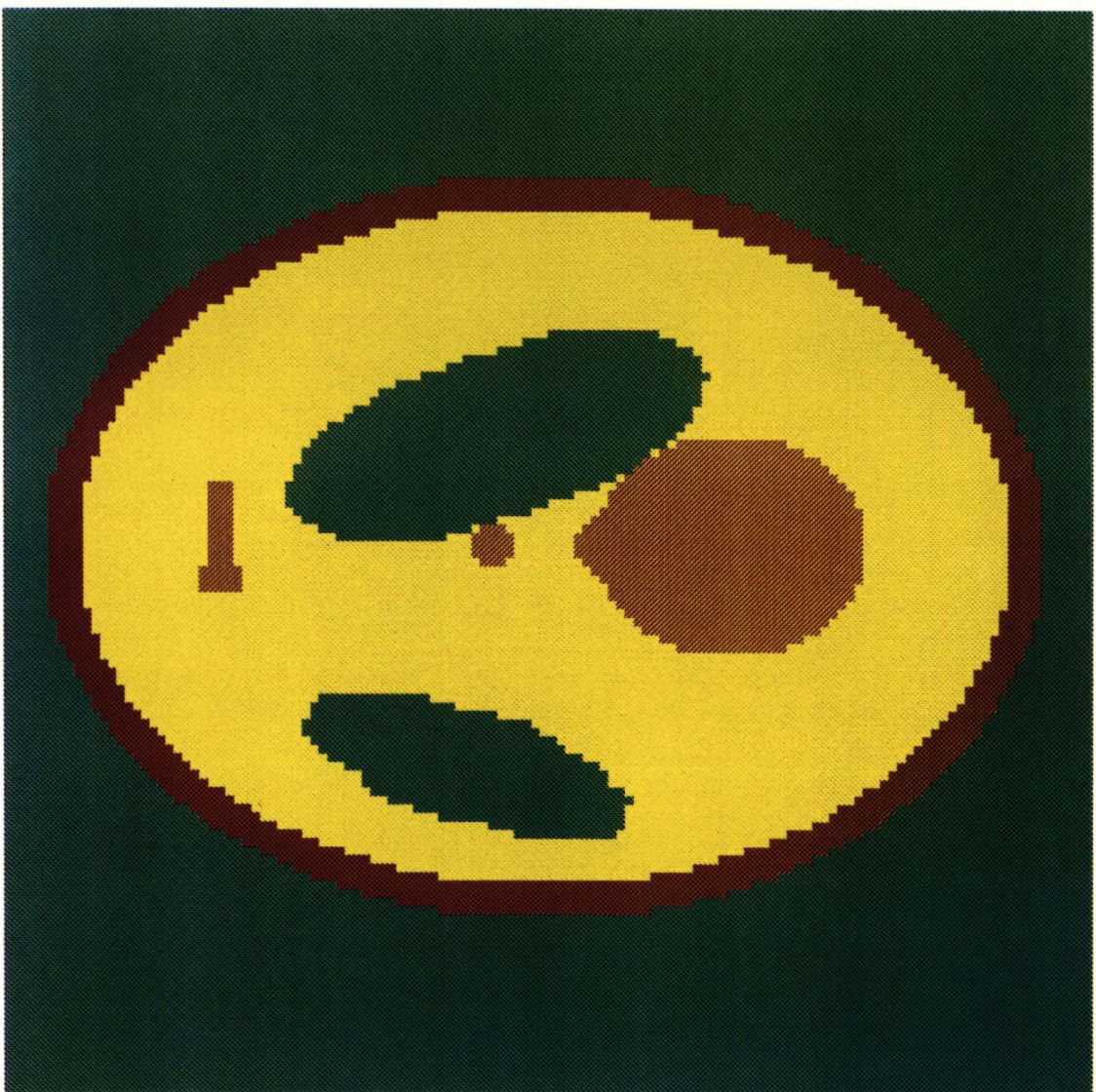
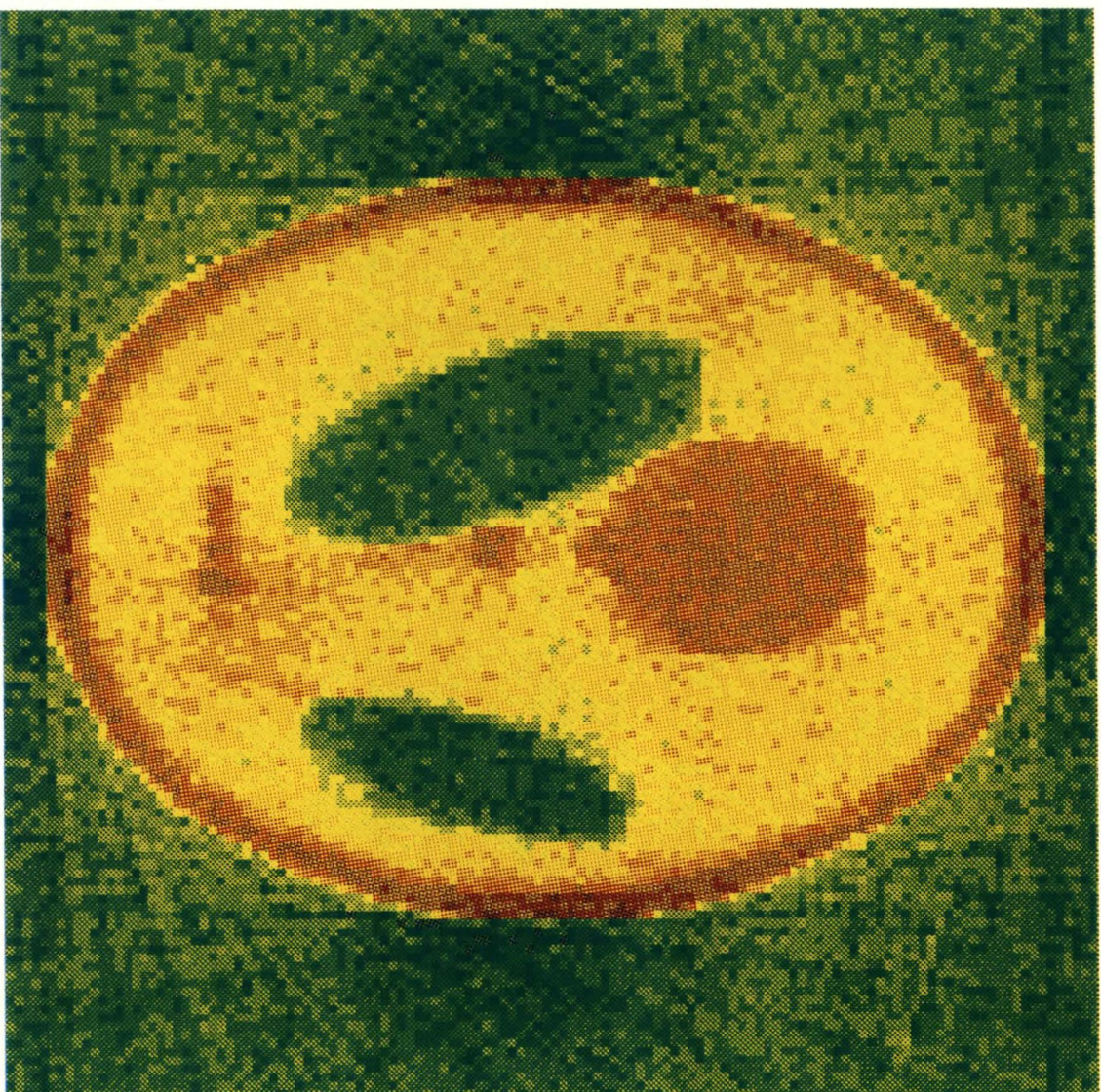


Figure 7.3 The result after applying RT followed by 1D MF and then, taking inverse RT.



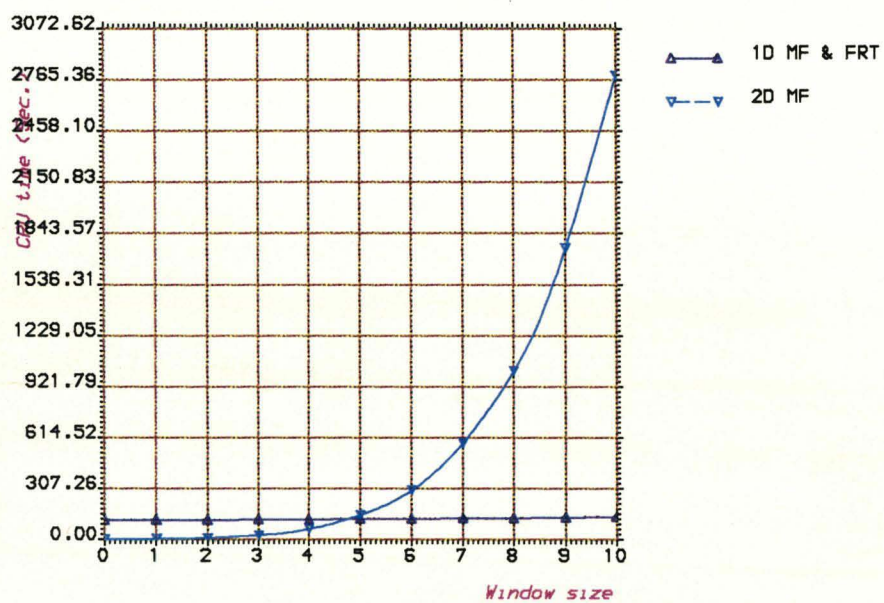
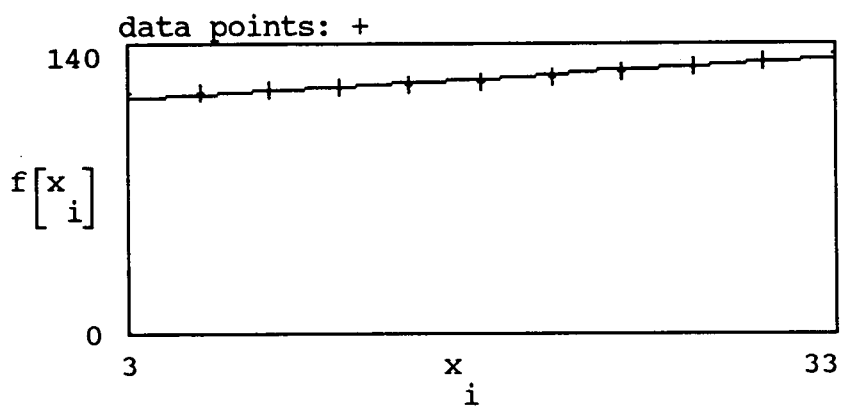
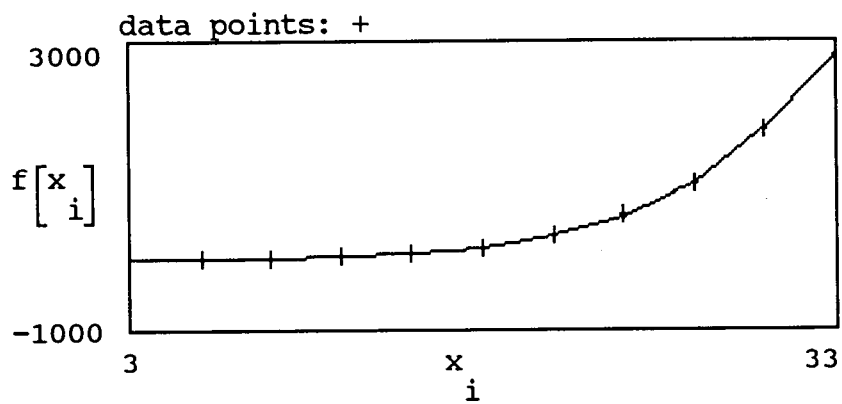


Figure 7.4 The CPU time versus window size For implementing 1D MF using FRT and 2D MF.



a) Data points representation of the 1D process using FRT.



b) Data points representation of 2D process.

Figure 7.5 Polynomial least-squares fit.

Table 7.1 The CPU time versus window size for implementing 1D MF using FRT and 2D Median Filter.

Window size (pixel)	1D MF and FRT (Sec.)	2D Median Filter (Sec.)
3	116.23	1.1
6	116.48	3.11
9	117.26	8.51
12	118.33	24.9
15	120.43	61.01
18	121.77	146.82
21	123.34	294.81
24	126.63	582.06
27	128.63	1015.34
30	131.23	1755.03
33	135.27	2793.29

CHAPTER 8

Reconstruction Using Image Rotation

8.1 Introduction

An important problem in image processing is to reconstruct a cross section of an object from several images of its transaxial projections [1-11]. The reconstruction problem is a particularly important subject area in image processing. In general, this problem is concerned with evaluating the structure of an object from observations on how it modifies certain properties of a probe. Mathematically, the data obtained are usually related to some object function via an integral transform. Here, the reconstruction problem is concerned with inverting certain classes of integral equations. This problem is compounded in the inversion of an integral transform called the Radon transform (see Chapter 6).

8.2 Fast Radon Transform and Back Projection

Computer algorithms and architectures for fast implementation of CT algorithms are important continuing areas of research. As the reconstruction algorithms for CT have become more sophisticated over the years, the demands on computer systems have become correspondingly greater. The speed of reconstruction becomes increasingly important as we move toward real-time tomographic reconstructions. The most time-consuming portions of the reconstruction algorithms lie in the com-

putation of back projection and the Radon transform.

Associated with the Radon transform is the back-projection operator $\hat{B}$ which is defined as:

$$B(x, y) = \hat{B}P(z, \theta) = \frac{1}{2\pi} \int_0^\pi P(x \cos \theta + y \sin \theta, \theta) d\theta$$

In polar coordinates (r, ϕ) , we have

$$B(r, \phi) = \frac{1}{2\pi} \int_0^\pi P(r \cos(\theta - \phi), \theta) d\theta \quad (8.1)$$

Back projection represents the accumulation of the ray-sums of all the rays that pass through the point (x, y) or (r, ϕ) .

The function $P(x \cos \theta + y \sin \theta, \theta)$ is the distribution of P along the family of lines L . For a fixed value of (r, ϕ) , $P(r \cos(\theta - \phi), \theta)$ is constructed by assigning the value of P at a point z to all points along the original line of projection L . By repeating the process for all values of z and for each value of θ , the function $P(r \cos(\theta - \phi), \theta)$ is obtained. Then, by summing all the functions P obtained for different values of θ between 0 and π , the back-projection function B is computed. This process is difficult to conceive from equation (8.1) alone. For this reason, If

$$P(z, \theta) = P_1(z)\delta(\theta - \theta_1) + P_2(z)\delta(\theta - \theta_2)$$

That is, if there are only two projections, then (see Fig. 8.1)

$$B(r, \phi) = P_1(z_1) + P_2(z_2)$$

Where

$$z_1 = r \cos(\theta_1 - \phi), z_2 = r \cos(\theta_2 - \phi)$$

In general, for a fixed point (x, y) or (r, ϕ) , the value of back projection $\hat{B}P(z, \theta)$ is evaluated by integrating $P(z, \theta)$ over θ for all lines that pass through that point. The Back-projection $B(x, y)$ at any pixel (x, y) requires projections from all directions.

This evident from equation (8.1). The back-projection is a 'blurred' representation of the true object function, this is because of the digital noise present in the image due to the process of rotation which inevitably requires some type of approximation to the coordinate transformation. This necessitates a filtering operation to amplify the high frequency content of $B(x, y)$.

The required filter is obtained by performing a Fourier analysis of the operation

$$\int_{-\infty}^{\infty} dz \frac{1}{z - \hat{n}.r} \frac{\partial}{\partial z} P(\hat{n}, z)$$

in equation (6.5) of Chapter 6 shows in the following section.

8.3 Reconstruction Methods

This section is concerned with methods of approximating the inverse Radon transform given by equation (6.5) of Chapter 6. The reconstruction methods presented are:

- 1- Reconstruction by Filtered Back-projection.
- 2- Reconstruction by Back-projection and deconvolution.
- 3- Reconstruction using central slice theorem.

Theoretically, the first two methods are completely equivalent and are essentially variation on equation (6.5) of Chapter 6. However, computationally, each method poses a different set of problems and requires an algorithm whose computational performance can vary significantly depending on the data type and its structure.

The reconstruction methods listed above using the back-projection process are classified according to whether filtering is applied before (1) or after (2) back-projection. These methods are discussed in the following sections.

8.4 Reconstruction by Filtered Back-Projection

This method can be classified according to the way of computational process into:

- 1- Reconstruction by using Hilbert transform.
- 2- Reconstruction by using 1-Dimensional inverse filter.

8.4.1 Reconstruction using Hilbert transform

To start with, let us rewrite equation (6.5) of Chapter 6 in the form

$$O(r) = -\frac{1}{2\pi} \int_0^\pi d\theta \frac{1}{\pi} \int_{-\infty}^{\infty} dz \frac{1}{z - \hat{n}.r} \frac{\partial}{\partial z} P(\hat{n}, z) \quad (8.2)$$

The integral over z is just the Hilbert transform of

$$\frac{\partial}{\partial z} P(\hat{n}, z)$$

Therefore,

$$\hat{H} \partial_z P(\hat{n}, z) = \frac{1}{\pi} \int \frac{\partial_z P(\hat{n}, z)}{z - \hat{n}.r} dz$$

Where:

$\hat{H}$ is the Hilbert transform operator.

$$\partial_z \equiv \frac{\partial}{\partial z}$$

The Hilbert transform is just a convolution in z . Using this operator, equation (8.2) can be written in the form

$$O(r) = \hat{R}^{-1} P(\hat{n}, z) = \hat{B} \hat{H} \partial_z P(\hat{n}, z) \quad (8.3)$$

Written in this form, clearly, the inverse Radon transform is actually composed of three separate operations; differentiation, Hilbert transform, and Back-projection.

• Numerical differentiation

Difference approximation is used to evaluate derivatives of a function using the functional values given at grid points. There are three different approaches for deriving difference approximations. These three approximations are called respectively forward, backward, and central difference approximation.

The foregoing results are useful for developing image reconstruction algorithms. In practice, the projections are available only on a finite grid. That is,

$$P_{\theta_n}(z_m) = P(z_m, \theta_n) = [\hat{R}O](z_m, \theta_n) \quad (8.4)$$

where,

$$0 \leq m \leq M-1, 0 \leq n \leq N-1$$

Thus, we have N projections taken at equally spaced angles. Equation (8.4) can be written in terms of discrete Fourier transform as:

$$P_{\theta_n}(z_m) = \frac{1}{N} \sum_{k_n=0}^{N-1} \tilde{P}_{\theta_n}(k_n) \exp\left(\frac{j2\pi k_n z_m}{N}\right) \quad (8.5)$$

Where:

$$\tilde{P}_{\theta_n}(k_n) = \sum_{z_m=0}^{N-1} P_{\theta_n}(z_m) \exp\left(\frac{-j2\pi k_n z_m}{N}\right)$$

is the discrete Fourier transform.

To compute the first derivative or difference of $P_{\theta_n}(z_m)$ we have:

1- Forward difference approach:

$$\dot{P}_{\theta_n}(z_m) = \frac{P_{\theta_n}(z_m + h) - P_{\theta_n}(z_m)}{h} \quad (8.6)$$

Where:

h is an interval between two consecutive points.

Consider the case of $h = 1$ and using equation (8.5) we have:

$$\dot{P}_{\theta_n}(z_m) = \sum_{k_n=0}^{N-1} \tilde{P}_{\theta_n}(k_n) \exp\left(\frac{j2\pi k_n z_m}{N}\right) \cdot \left[\exp\left(\frac{j2\pi k_n}{N}\right) - 1\right] \quad (8.7)$$

2- Backward difference approximation

$$\dot{P}_{\theta_n}(z_m) = \frac{P_{\theta_n}(z_m) - P_{\theta_n}(z_m - h)}{h} \quad (8.8)$$

Using equation (8.5) and $h = 1$, we have:

$$\dot{P}_{\theta_n}(z_m) = \sum_{k_n=0}^{N-1} \tilde{P}_{\theta_n}(k_n) \exp\left(\frac{j2\pi k_n z_m}{N}\right) \cdot \left[1 - \exp\left(\frac{-j2\pi k_n}{N}\right)\right] \quad (8.9)$$

3- Central difference approximation

$$\dot{P}_{\theta_n}(z_m) = \frac{P_{\theta_n}(z_m + h) - P_{\theta_n}(z_m - h)}{2h} \quad (8.10)$$

Using equation (8.5) and $h = 1$, we have:

$$\begin{aligned} \dot{P}_{\theta_n}(z_m) &= \sum_{k_n=0}^{N-1} \tilde{P}_{\theta_n}(k_n) \exp\left(\frac{j2\pi k_n z_m}{N}\right) \cdot \left[\frac{\exp\left(\frac{j2\pi k_n}{N}\right) - \exp\left(\frac{-j2\pi k_n}{N}\right)}{2}\right] \\ &= \sum_{k_n=0}^{N-1} \tilde{P}_{\theta_n}(k_n) \exp\left(\frac{j2\pi k_n z_m}{N}\right) \cdot j \sin\left(\frac{2\pi k_n}{N}\right) \end{aligned} \quad (8.11)$$

• Filtering process

The difference approximations described by equation (8.7), (8.9), and equation (8.11) contain three types of filtering process. The transfer functions of these filters are given by:

$$H_f(k) = \exp(j2\pi k) - 1 \quad (8.12)$$

$$H_B(k) = 1 - \exp(-j2\pi k) \quad (8.13)$$

$$H_c(k) = j \sin(2\pi k) \quad (8.14)$$

Where:

$$k = \frac{k_n}{N}, k_n = 0, 1, 2, \dots, N-1$$

H_f is the filter transfer function of the forward difference approximation.

H_B is the filter transfer function of the backward difference approximation.

H_c is the filter transfer function of the central difference approximation.

Therefore, this analysis highlighted that the process of derivative introduces different types of filters which depend on the method used for derivative; forward, backward, or central difference method. The amplitude and phase response of these filtering approach are shown in figure (8.2) and figure (8.3) respectively.

A schematic diagram illustrating the possible sequence of computing the inverse Radon transform using Hilbert transform is shown in figure (8.4). An example of applying this technique is shown in figure 8.5 and figure 8.6 respectively.

8.4.2 Reconstruction using 1-Dimensional inverse filter

Since the Hilbert transform is a linear functional, therefore:

$$\hat{H}\partial_z P = \partial_z \hat{H}P$$

The exact form of the filter that is associated by this operation can be found by Fourier analysis. For a fixed value of $\hat{n}$ (i.e., fixed angle of rotation θ , we have:

$$\hat{H}\partial_z = \frac{1}{\pi z} \otimes \frac{\partial P}{\partial z}$$

Where:

P is the projection obtained for a given $\hat{n}$ and $\otimes$ is the convolution operation. Using the convolution theorem which states that the convolution of two functions in real space is the same as their product in Fourier space. Using this result:

$$\hat{F}_1\left(\frac{\partial P}{\partial z}\right) = ik\hat{F}_1 P$$

and

$$\hat{F}_1\left(\frac{1}{\pi z}\right) = -i \operatorname{sgn}(k)$$

Where $\hat{F}_1$ is the one-dimensional Fourier transform operator and k is the spatial frequency, therefore,

$$\hat{F}_1(\hat{H} \partial_z P) = -i \operatorname{sgn}(k) (ik \hat{F}_1 P)$$

Now,

$$-i \operatorname{sgn}(k) \cdot (ik) = \operatorname{sgn}(k)k = |k|$$

Hence, the operation $\hat{H} \partial_z P$ in real space is equivalent to applying the filter $|k|$ in Fourier space. Therefore, the reconstruction formula given by equation (8.3) can be written in the form:

$$O(r) = \hat{B} \hat{F}_1^{-1} [|k| \hat{F}_1 P(\hat{n}, z)] \quad (8.15)$$

This process is illustrated in figure (8.7). The amplitude and phase response of this filter are shown in Fig.(8.8) and Fig.(8.9) respectively. An example for applying this process is shown in figure 8.10 and figure 8.11 respectively.

8.5 Reconstruction by Back-Projection and Deconvolution

Another method of reconstructing O from P can be acquired by considering the effect of back-projecting without filtering. The result will be some blurred version of the object function. The blurring inherent in such a reconstruction can be represented mathematically by the convolution of the object function with a point spread function. By computing the functional form of the point spread function, the object reconstruction can be deconvolved.

• Calculation of the point spread function:

The point spread function can be computed by back-projecting the projections obtained from a single radially symmetric point located at $(0, 0)$ described analytically by a two-dimensional delta function. The projection of a two-dimensional delta function, is a one-dimensional delta function and so in this case,

$$P(x \cos \theta + y \sin \theta, \theta) = \delta(x \cos \theta + y \sin \theta); \forall \theta$$

Thus writing the above equation in polar coordinate system (r, ϕ) , we obtain:

$$B(r, \phi) = \frac{1}{2\pi} \int_0^\pi \delta[r \cos(\theta - \phi)] d\theta = \frac{1}{r} \quad (8.16)$$

Hence, the point spread function is given by:

$$P(x, y) = \frac{1}{\sqrt{(x^2 + y^2)}} \quad (8.17)$$

The back-projection function obtained from the sequence of projections taken through an object function O is therefore given by

$$B(x, y) = P(x, y) \otimes \otimes O(x, y)$$

Where $\otimes \otimes$ is the two-dimensional convolution operation. In order to reconstruct O from B we must deconvolve. This can be done by processing the equation above in Fourier space as follow:

$$\tilde{B}(k_x, k_y) = \tilde{P}(k_x, k_y) \tilde{O}(k_x, k_y)$$

Where:

$$\tilde{O}(k_x, k_y) = \hat{F}_2 O(x, y)$$

$$\tilde{P}(k_x, k_y) = \hat{F}_2 P(x, y)$$

and

$$\tilde{B}(k_x, k_y) = \hat{F}_2 B(x, y)$$

Rearranging

$$\tilde{O}(k_x, k_y) = \frac{\tilde{B}(k_x, k_y)}{\tilde{P}(k_x, k_y)}$$

The function $\frac{1}{\tilde{P}}$ is called the inverse filter and can fortunately be computed analytically. The result is

$$\tilde{P}(k_x, k_y) = \frac{1}{\sqrt{(k_x^2 + k_y^2)}}$$

Hence, the reconstruction formula for the object function is given by

$$O(x, y) = \hat{F}_2^{-1}[|k|\tilde{B}(k_x, k_y)] \quad (8.18)$$

Where:

k is the 2-dimensional spatial frequency vector.

A schematic diagram illustrating this reconstruction process is given in Figure 8.12. The amplitude and phase response of this filter are shown in figure 8.13 and Figure 8.14 respectively. An example of applying this technique is shown in Figure 8.15 and Figure 8.16 respectively.

8.6 Reconstruction Using Central Slice Theorem

We discussed the projection slice theorem in the section on the connection to Fourier transform (section 5.3.2). It is an easy transform-based technique to conceptualize as it represents a direct use of the projection-slice theorem. The Fourier transform of each projection is calculated, interpolated, placed at the proper slice angle onto a 2D Cartesian grid, and finally filtered using 2D inverse filter. The 2D inverse Fourier transform then yields the reconstructed image. Mathematically it can be written as

$$f(x, y) = \hat{F}_2^{-1}\{|K|[I_c(\hat{F}_1 P(\theta, z))]\} \quad (8.19)$$

where $I_c()$ represents an interpolation operator that acts on all the projections to convert from polar coordinates to a full Cartesian grid of Fourier domain values.

The 1D Fourier transform of the projections can be interpolated to more points in the frequency domain by the simple method of adding many zero data values in the spatial domain before performing the Fourier transform. Since the spatial sampling distance is fixed, adding more zeros has the effect of decreasing the Fourier domain sample spacing, essentially giving us interpolated frequency points. This technique is called zero-padding [12].

The 2D inverse filter $|K|$ in equation (5.13) (sec.5.3.2) is essential in image reconstruction. Since, the data values close to the origin is much larger than the radial one. As a result, the proper image reconstruction using central-slice theorem can be obtained. A schematic diagram illustrating this process is shown in figure 8.17. An example of applying this technique is shown in figure 8.18 and figure 8.19 respectively.

8.7 Noise analysis

Rotation problem is concerned with the reconstruction process by computing the Radon transform and its inverse. The most common technique is the optimum nearest neighbour which leads to holes in the rotated images. The effect of these holes lead to artifacts in computing the Radon transform specially at 45° and 135° rotation (See Fig. 8.10 and Fig. 8.18)

Rotation is used for both the Radon transform and back projection algorithms because they are very similar in structure. As a result, rotation noise arises in computing both of them. Therefore, the result of the reconstructed image using filtered back projection is better than using the back projection and deconvolution, since the filtering process in the first method is an intermediate stage between the Radon transform and the back projection. This is not the case with the back

projection and deconvolution which involved application of a 2D inverse filter after applying projection and back projection, both of which generate noise through the rotation process.

8.8 Discussion

The aim of this chapter has been discussed to show the computational methodology behind each technique for performing the inverse Radon transform. These techniques are computed using different rotation techniques to reconstruct the object function of a test image (see Chapter 9). The CPU time required to compute both Radon transform and its inverse are presented in Chapter 9. Also, error and statistical analysis using different rotation techniques are discussed in Chapter 9.

8.9 References

- [1] IEEE Trans. Nuci. Sci., 'Special issue on topics related to image reconstruction', NS-21, no.3 (1974); NS-26, no.2, (April 1979); NS-27, no.3, (June 1980).
- [2] IEEE Trans. Biomed. Engineering, 'Special issue on computerized medical imaging, 'BME-28, no.2 (February 1981).
- [3] Proc. IEEE, 'Special issue on computerized tomography, 'vol. 71, no.3, March 1983.
- [4] A. C. Kak, 'Image reconstruction from projections, 'in M.P. Ekstrom (editor), Digital image processing techniques, New York, Academic Press, 1984, pp.111-171.
- [5] G. T. Herman (editor), 'Image reconstruction from projections, 'Topics in Applied Physics, vol.32, New York, Springer-Verlag, 1979.
- [6] G. T. Herman, 'Image reconstruction from projections-The fundamental of computerized tomography, New York, Academic Press, 1980.
- [7] H. J. Scudder, 'Introduction to computer aided tomography, 'Proc. IEEE, vol.66, no.6, June 1978.

- [8] Z. H. Cho, H. S. Kim, H. B. Song, and J. Cumming, 'Fourier transform nuclear magnetic resonance tomographic imaging, 'Proc. IEEE, vol.70, no.10, pp.1152-1173, October 1982.
- [9] W. S. Hinshaw and A. H. Lent, 'An introduction to NMR imaging, 'in [3].
- [10] D. C. Munson, Jr., J. O'Brien, K. W. Jenkins, 'A tomographic formulation of spot-light mode synthetic aperture radar, 'Proc. IEEE, vol.71, pp.917-925, August 1983.
- [11] J. L. C. Sane, E. B. Hinkle, A. K. Jain, 'Radon and projection transform-based machine vision: Algorithms, a pipeline architecture, and industrial applications, 'Berlin: Springer-Verlag, 1988.
- [12] S. W. Rowland, 'Computer implementation of image reconstruction formula ', In G. T. Herman, editor, Image reconstruction from projections: Implementation and applications, Springer-verlag, Berlin, 1979.

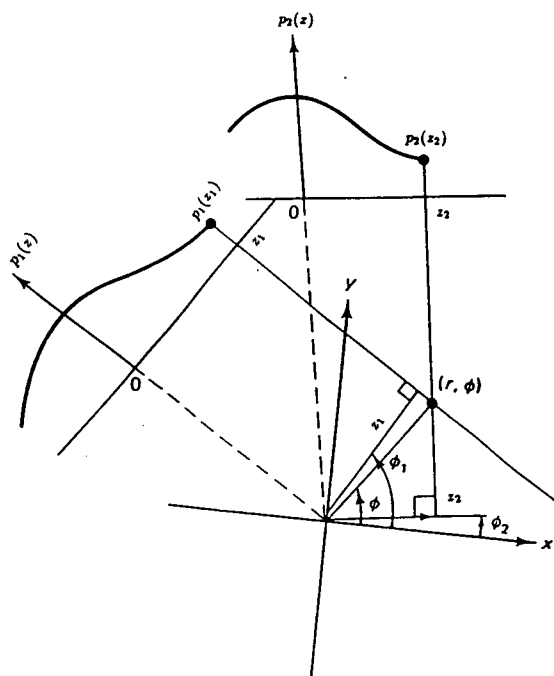


Figure 8.1 Back-projection of $p_1(z)$ and $p_2(z)$ at (r, ϕ) .

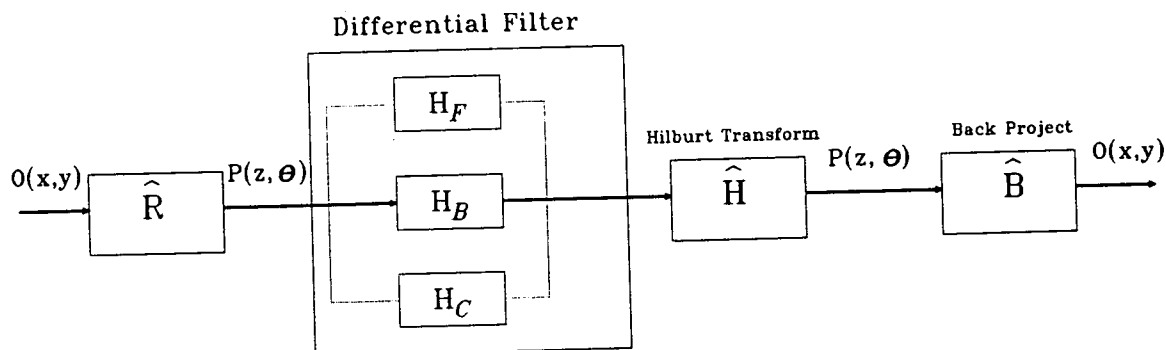


Figure 8.2 Schematic diagram of the reconstruction process using differential filter and Hilbert transform method.

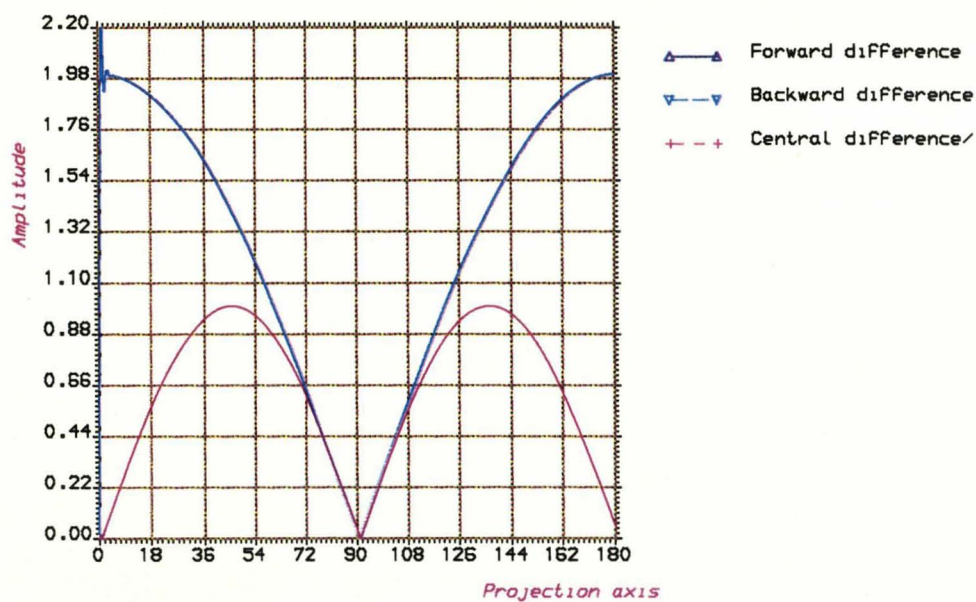


Figure 8.3 The amplitude responses of the differential Filters.

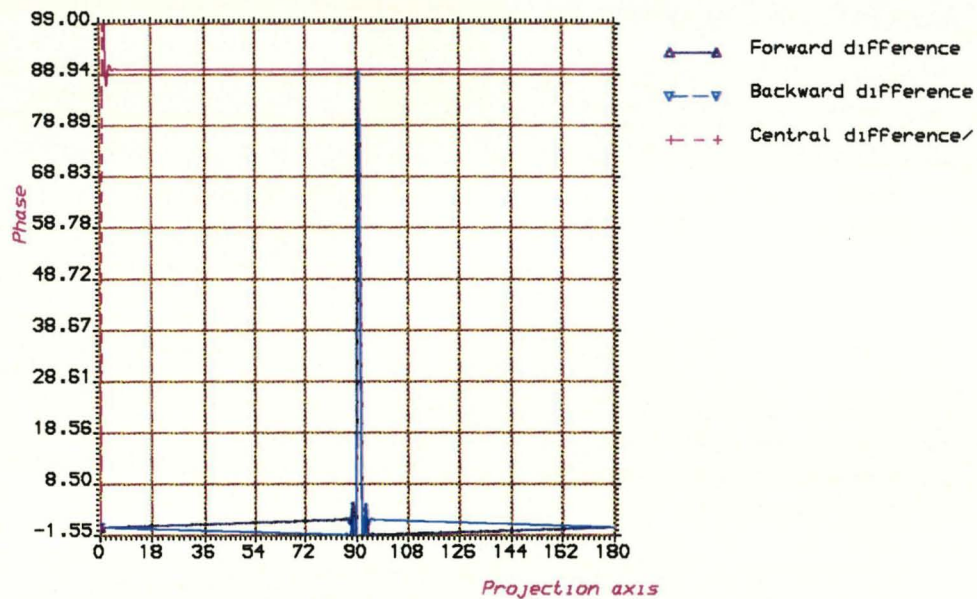
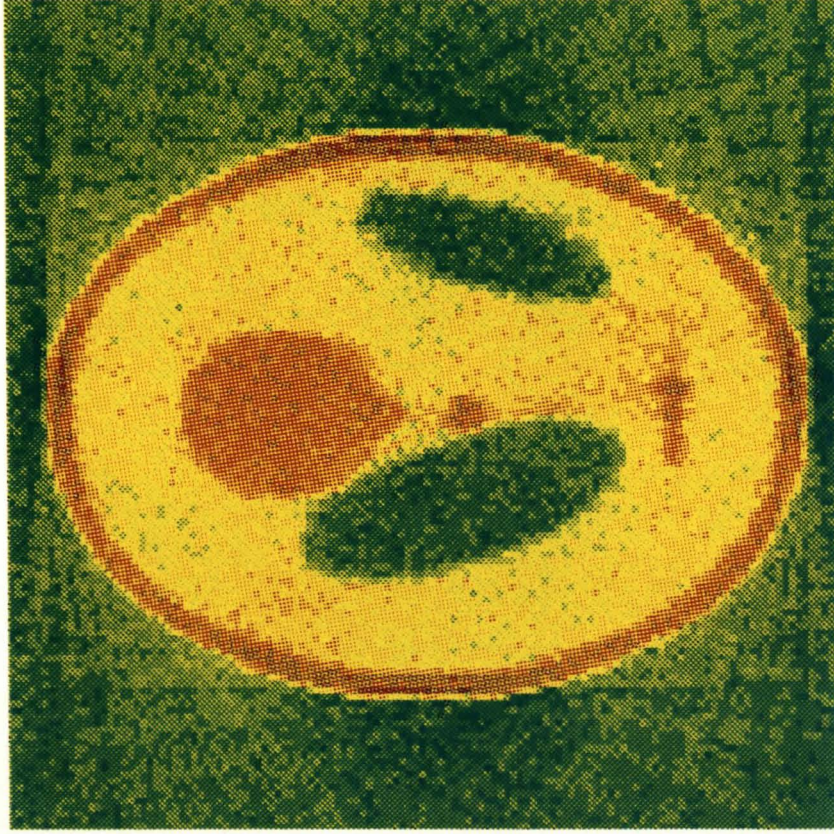


Figure 8.4 The phase responses of the differential Filters.

Figure 8.5 Head-phantom image.



Figure 8.6 Estimated reconstruction of the head-phantom using filtered back projection (Hilbert transform).



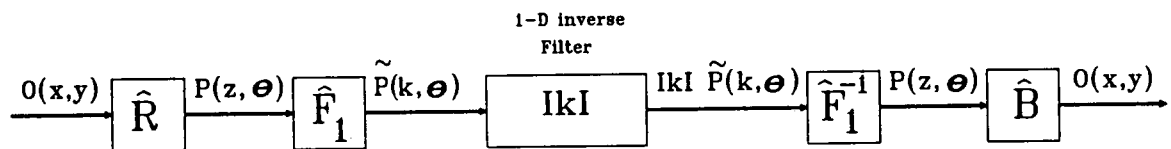


Figure 8.7 Schematic diagram of the reconstruction process using filtered back projection (1D inverse filter).

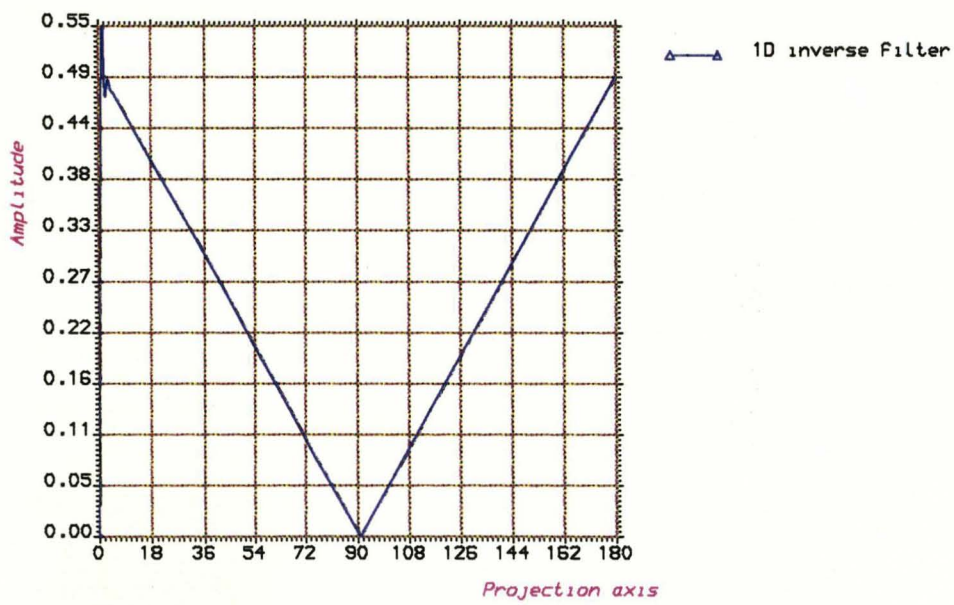


Figure 8.8 The amplitude response of the 1D inverse Filter.

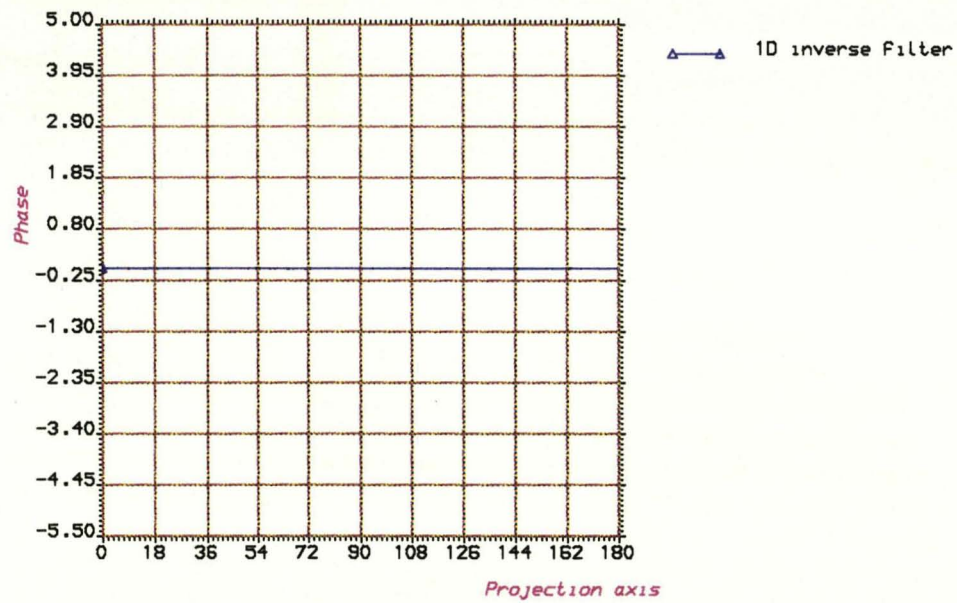


Figure 8.9 The phase response of the 1D inverse Filter.

Figure 8.10 Radon space of the head-phantom image.

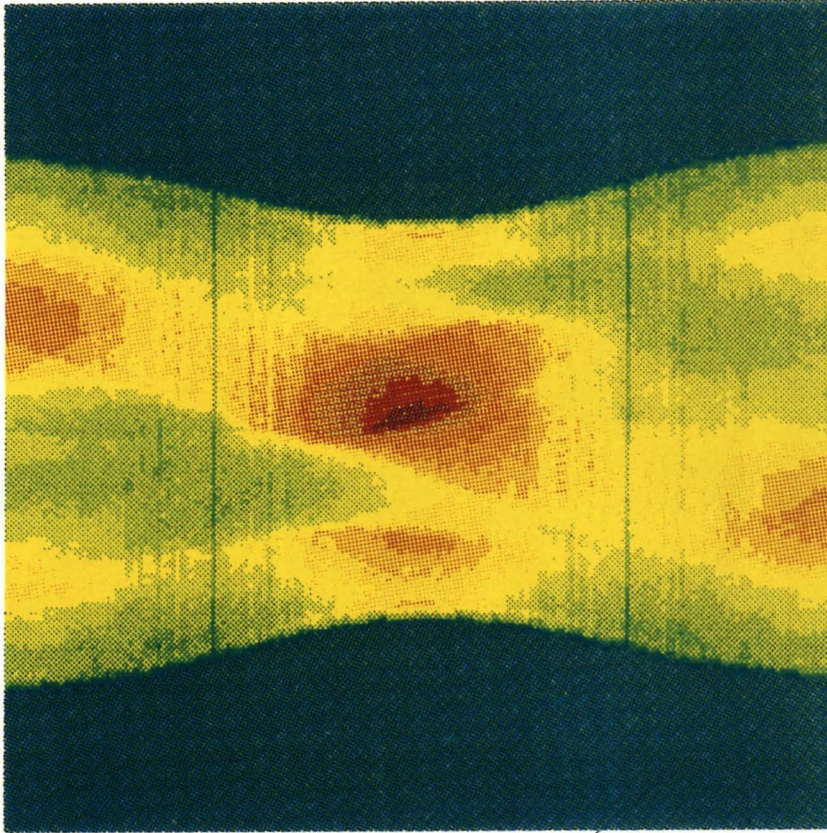
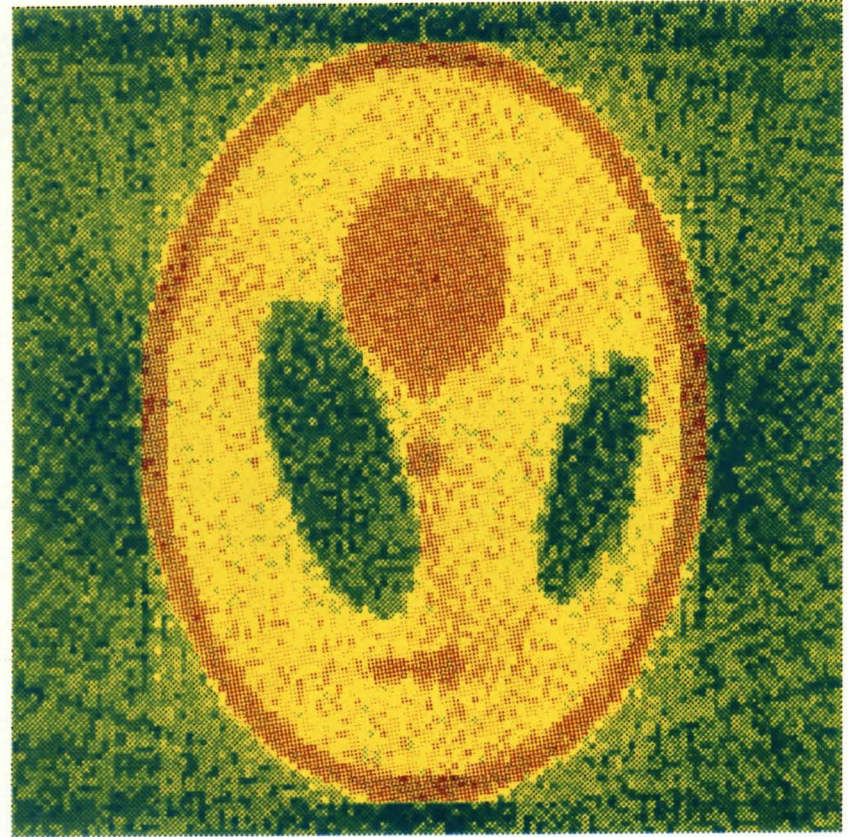


Figure 8.11 Estimated reconstruction of the head-phantom using filtered back projection (1D inverse filter).



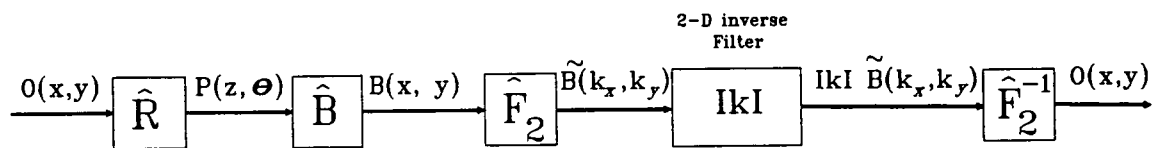


Figure 8.12 Schematic diagram of the reconstruction process using back projection and deconvolution.

Figure 8.13 The amplitude response of the 2D inverse filter.

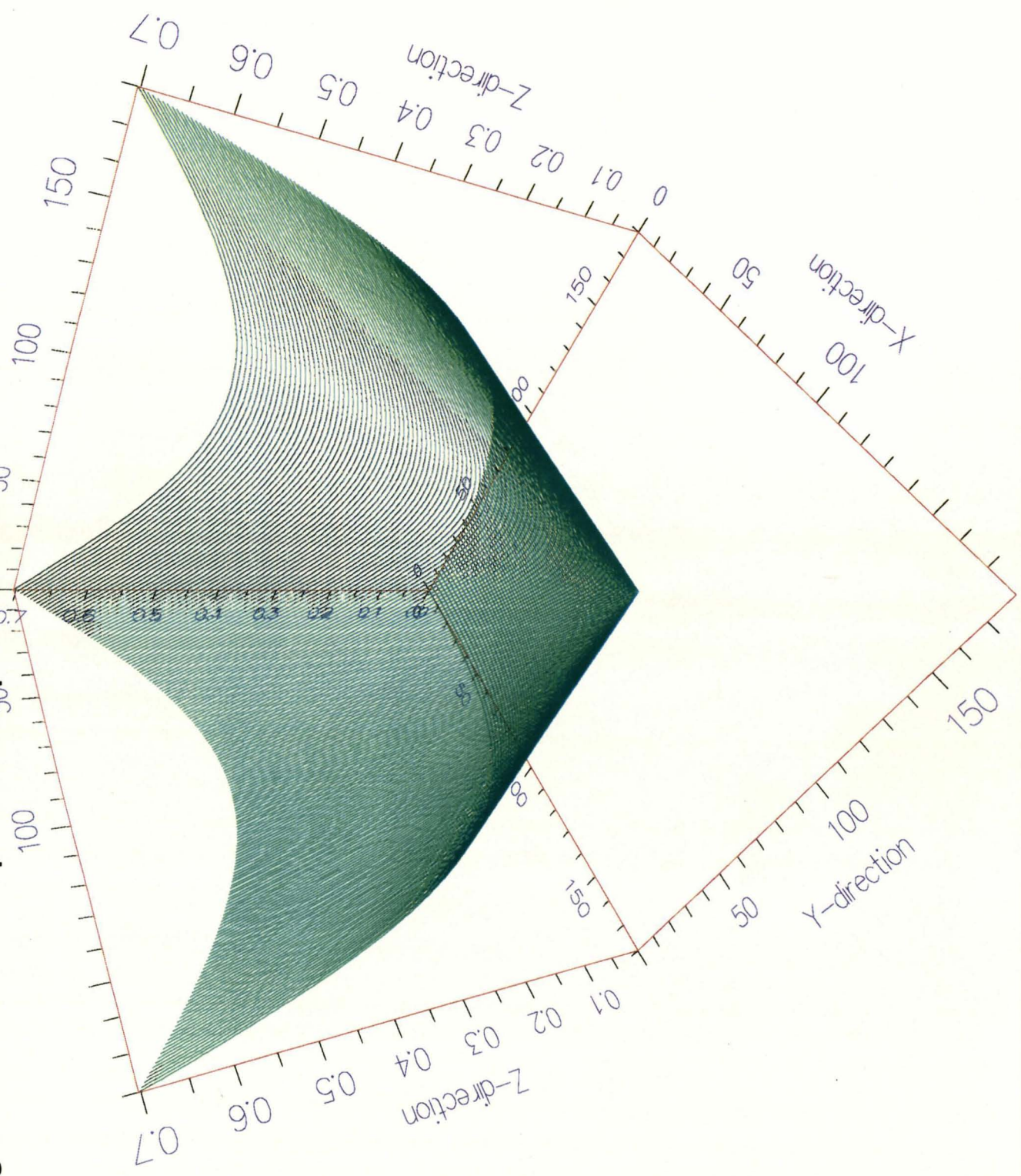


Figure 8.14 The phase response of the 2D inverse filter.

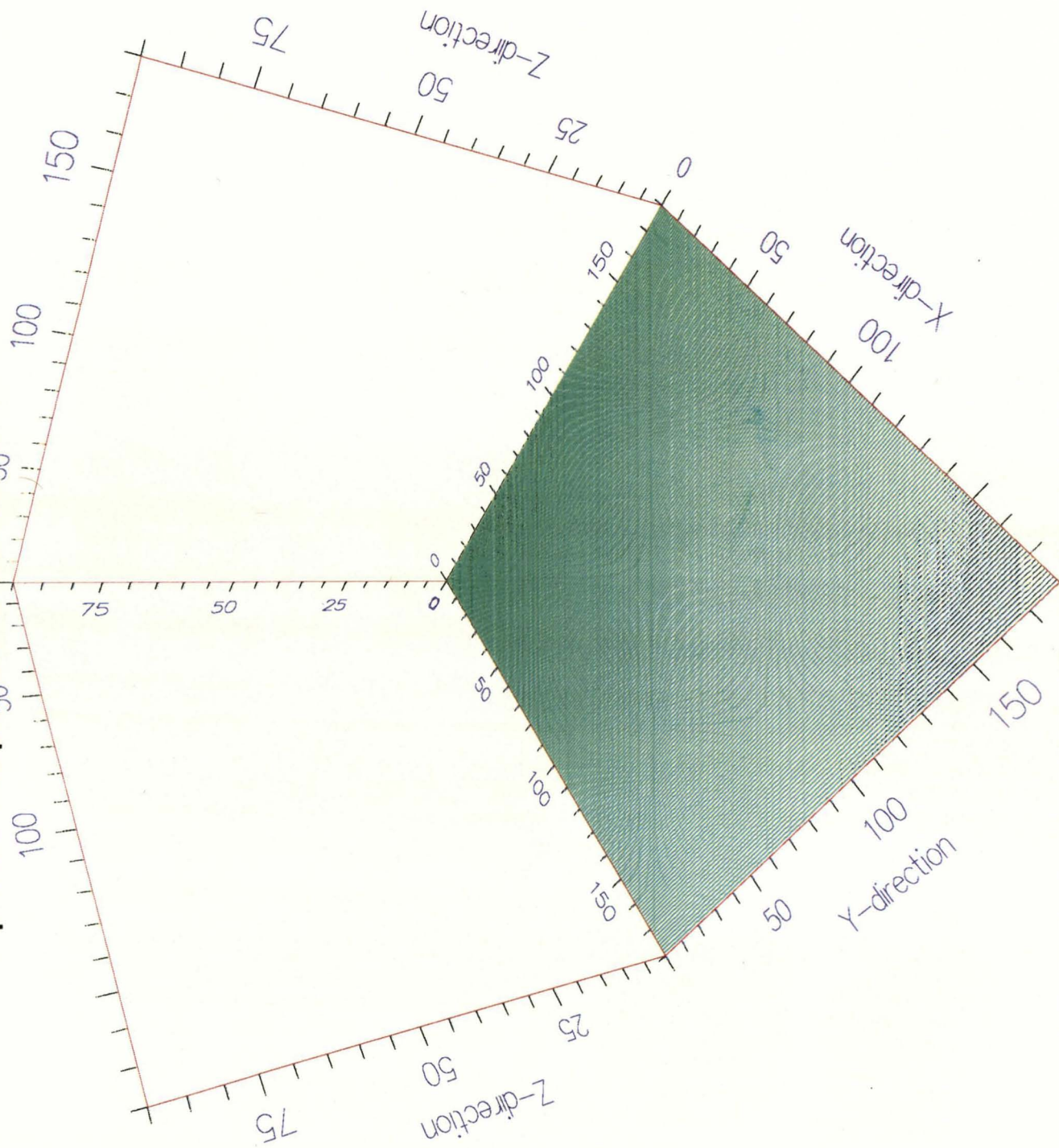


Figure 8.15 Radon space of the head-phantom image.

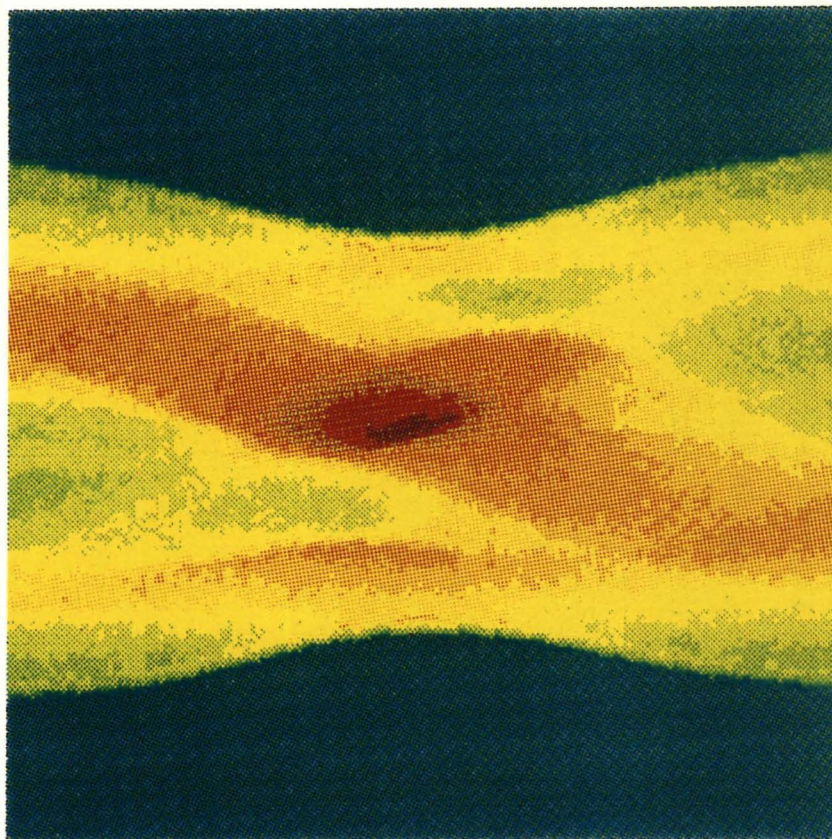
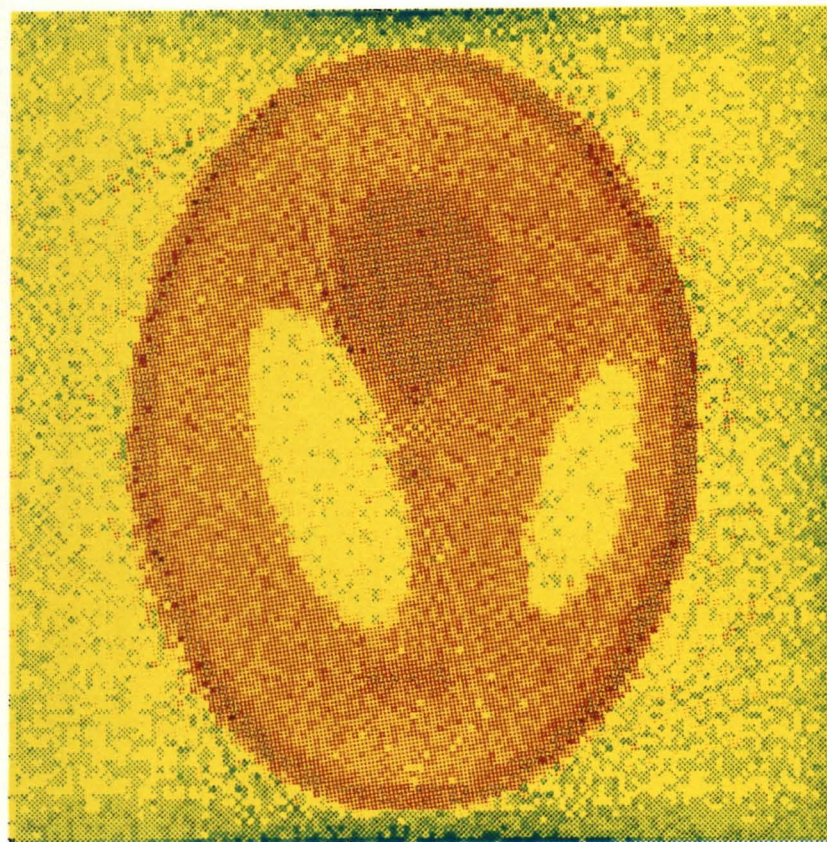


Figure 8.16 Estimated reconstruction of the head-phantom using back projection and deconvolution.



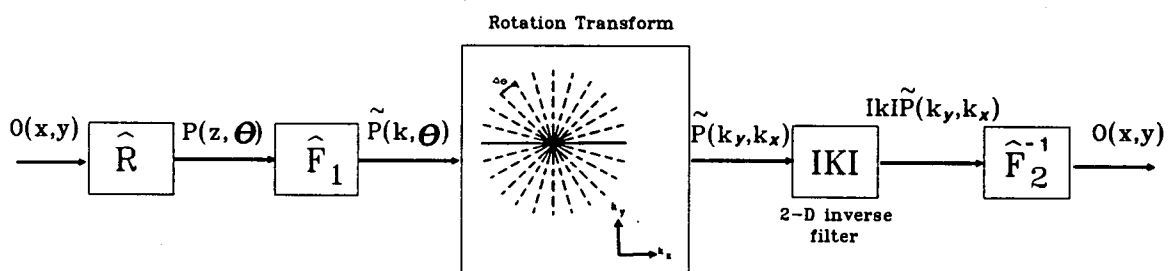


Figure 8.17 Schematic diagram of the reconstruction process using central-slice theorem.

Figure 8.18 Radon space of the head-phantom image.

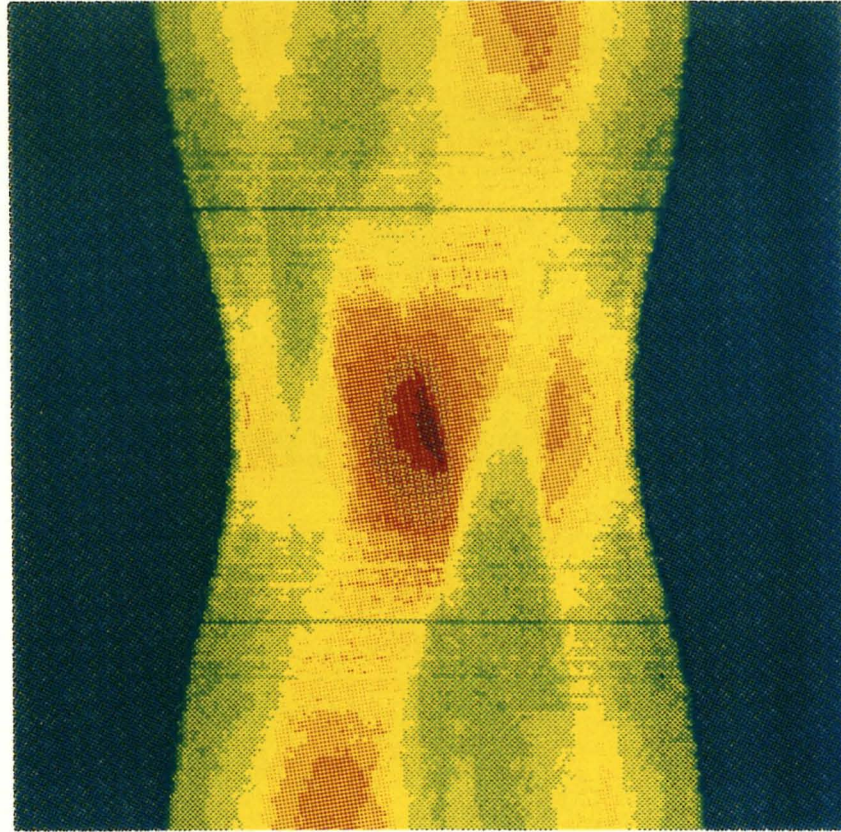
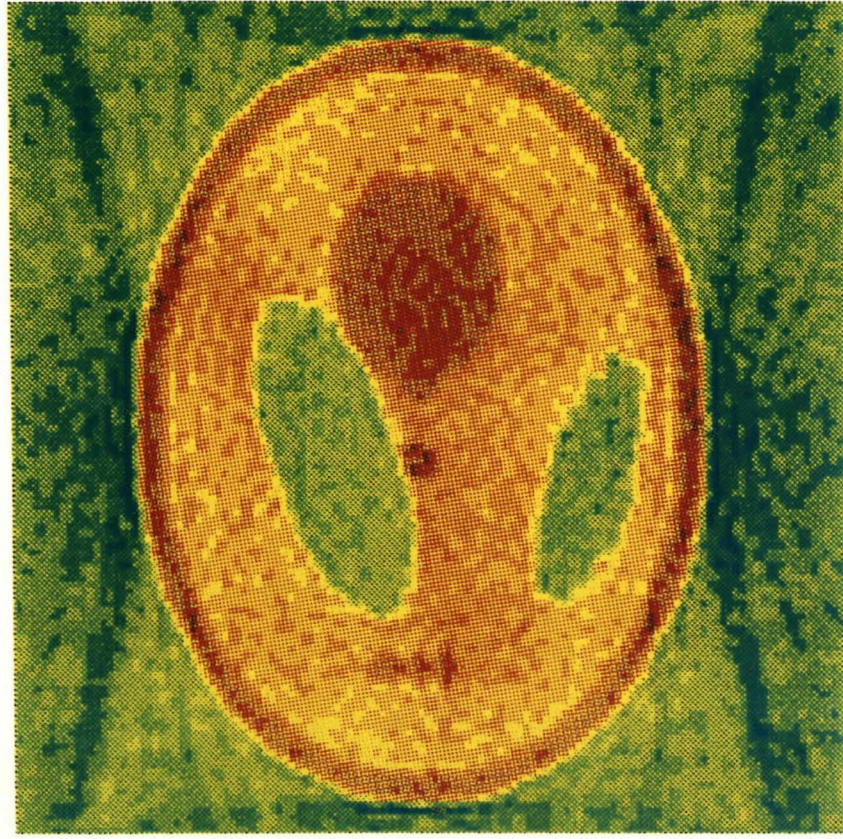


Figure 8.19 Estimated reconstruction of the head-phantom using central-slice theorem.



CHAPTER 9

Reconstruction Errors and Statistical Analysis

9.1 Introduction

In this chapter we are going to review and discuss the possible sources of errors associated with the reconstruction processes. This source of error mainly associated with digital image rotation. This error leads to the production of noise when computing the Radon transform. Therefore, a numerical study has been carried out to test the relationship between different methods of image rotation and their effect on the performance for computing the Radon transform, in particular, the evaluation of CPU time versus the mean squares error between a given test image and its reconstruction. The solution to this problem is based on utilizing the least squares principle and the statistical analysis.

9.2 Errors in Reconstruction

The first analysis concerns the error in the reconstruction process. In this analysis, the image to be reconstructed $f(x, y)$ contains no noise component. All the noise in the reconstruction process is caused by rotation in the projections and back-projections. The mathematical bases behind this error method is discribed as:

$$e = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \frac{[f(i, j) - \hat{f}(i, j)]^2}{\sum_{i=1}^N \sum_{j=1}^N [f(i, j)]^2} \quad (9.1)$$

Where:

e is the mean square error.

$f(i, j)$ is the original object function.

$\hat{f}(i, j)$ is the estimated function associated with the reconstruction process.

The absolute square error e_{ij} is defined by:

$$e_{ij} = \frac{[f(i, j) - \hat{f}(i, j)]^2}{\sum_{i=1}^N \sum_{j=1}^N [f(i, j)]^2} \quad (9.2)$$

9.3 Statistical Description of the Mean Square Error

The statistical description is defined by measuring the mean of the absolute square error and its standard deviation as follow:

$$\bar{e}_{2D} = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N e_{ij} \quad (9.3)$$

$$\sigma_{2D} = \sqrt{\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N [e_{ij} - \bar{e}_{2D}]^2} \quad (9.4)$$

where:

$\bar{e}_{2D}$ is the mean of the 2D absolute square error.

σ_{2D} is the standard deviation of the 2D absolute square error.

9.4 Projection Square Error

Projection square error is the mean square error projected along the image axis. This error is described mathematically by:

$$e_p(i) = \sum_{j=1}^N \frac{[f(i, j) - \hat{f}(i, j)]^2}{\sum_{i=1}^N \sum_{j=1}^N [f(i, j)]^2} \quad (9.5)$$

where:

$e_p(i)$ is the projection square error along the image axis.

Figure (9.1) through Figure (9.8) show the projection square error of a test image versus the image axis for each rotation algorithm and different reconstruction method. From Figure (9.3), the projection square error using the optimum nearest technique gives the lowest error along the image axis.

9.5 Statistical Description of Projection Square Error

The statistical description is defined by measuring the mean of the projection square error and its standard deviation as follow:

$$\bar{e}_p = \frac{1}{N} \sum_{i=1}^N e_p(i) \quad (9.6)$$

$$\sigma_p = \sqrt{\frac{1}{N} \sum_{i=1}^N [e_p(i) - \bar{e}_p]^2} \quad (9.7)$$

where:

$\bar{e}_p$ is the mean of the projection square error.

σ_p is the standard deviation of the projection square error.

9.6 Test Example

Table (9.1) through Table (9.8) show the CPU reconstruction time of a test image using different rotation algorithm and different reconstruction methods versus the mean square error (MSE). Table (9.3) shows that the optimum algorithm is based on the optimum nearest neighbour method of image rotation to recover the object by using the filtered back projection (1D inverse filter).

9.7 Quantative Study of the Reconstruction Techniques

Computing better reconstructions involve producing images that objectively show more information. The results showed that the optimum nearest neighbour

technique gives a better reconstruction than others. Computing the computation energy for each reconstruction process requires the determination of the total CPU reconstruction time and the mean of the projection square error. Therefore, consider the amplitude of the projection square error is measured in $[volt]^2$. Then, the computation energy per each pixel is given by:

$$CE = \frac{\bar{e}_p \cdot T}{N^2} \quad (9.8)$$

where:

CE is the computation energy.

T is the total CPU reconstruction time.

N^2 is the image size.

Table (9.9) through Table (9.16) show the CPU reconstruction time of a test image using different rotation algorithm and different reconstruction methods versus the mean of projection square error, its standard deviation and the computation energy. Table (9.11) shows that the minimum computation energy is based on the optimum nearest neighbour method of image rotation to recover the object by using the filtered back projection (1D inverse filter). Also, Figure (9.9) shows the computation energy for each reconstruction methods using different rotation algorithms. As a result, minimum computation energy gives better system performance.

9.8 Summary

The motivation for drawing the four methods of image reconstruction in a common figure (Fig.9.9) is to measure the stability of image recovery for each rotation algorithm. From Figure (9.9), it is evident that the filtered back projection (1D inverse filter) has the lowest computational energy and consistent over all the different rotation algorithms.

The computation energy saved by the new technique compared to the current

one is given by:

$$\delta CE = \frac{CE_c - CE_o}{CE_c} \cdot 100$$

$$\delta CE = 54.04\%$$

where:

CE_c is the computation energy for the current technique (normal eight).

CE_o is the computation energy for the optimum technique.

δCE is the relative computation energy saved by the optimum technique.

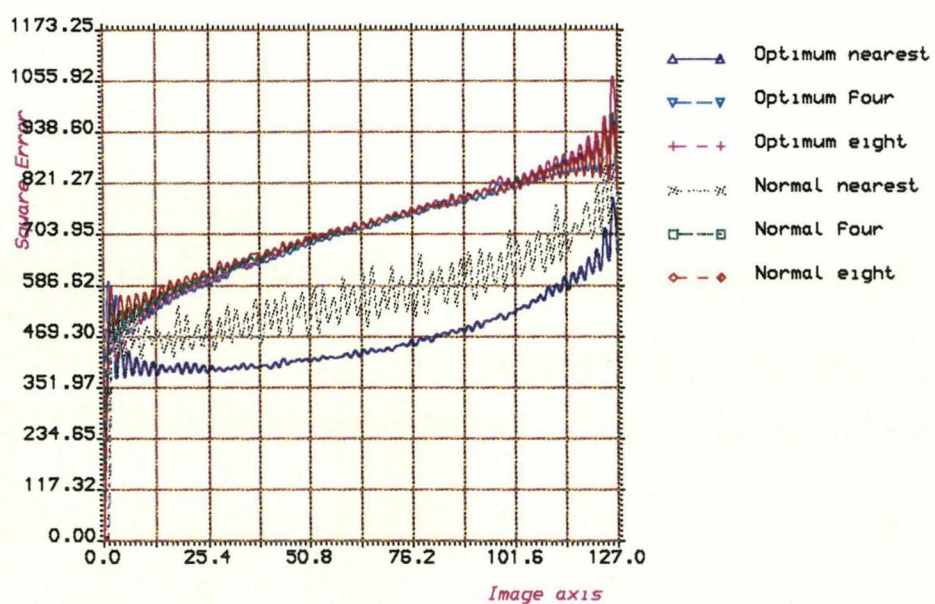


Figure 9.1 The projection error of a test image associated with nearest neighbour using Filtered back projection (Hilbert transform).

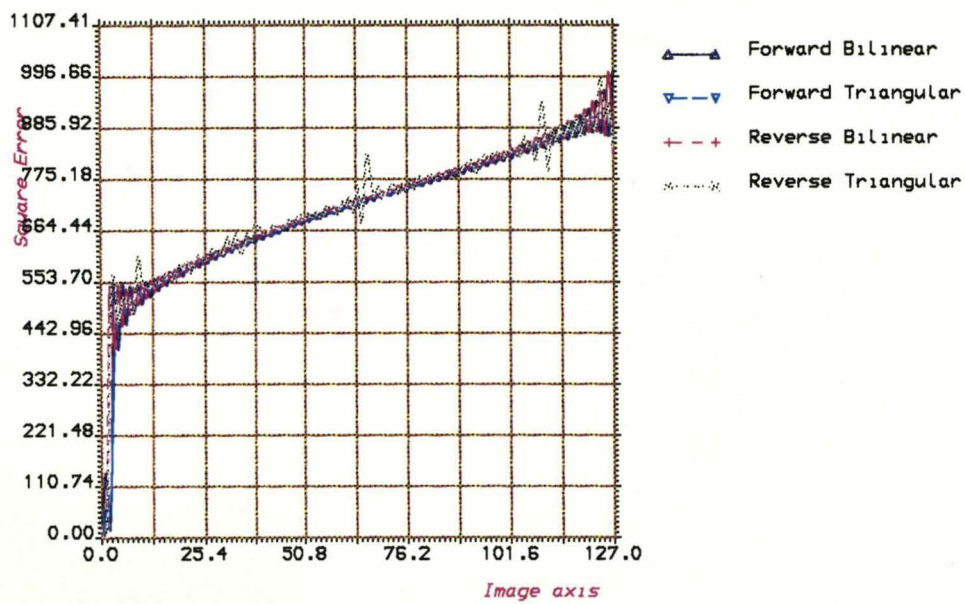


Figure 9.2a The projection error of a test image associated with interpo technique using Filtered back projection (Hilbert transform)

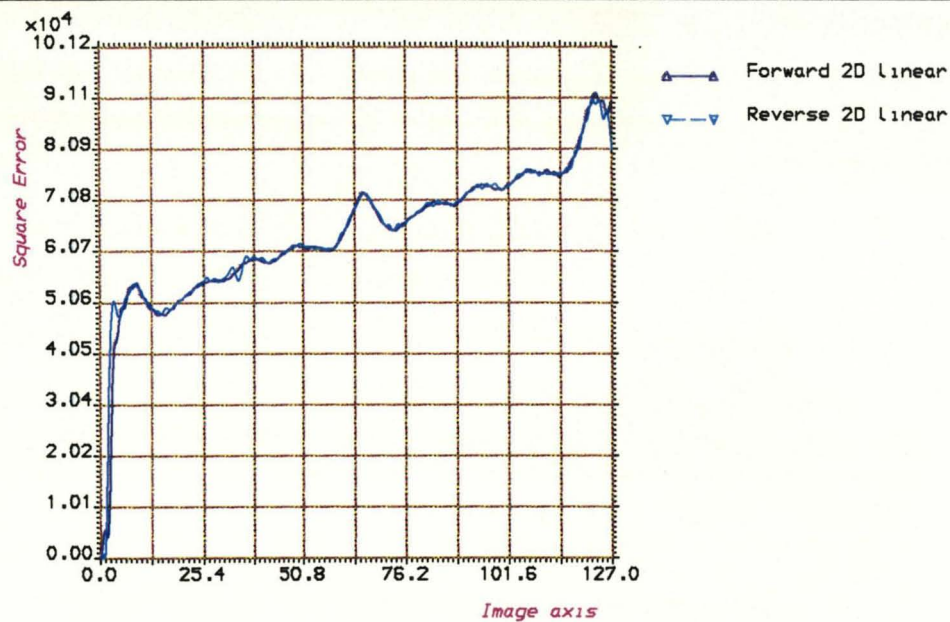


Figure 9.2b The projection error of a test image associated with interpo technique using Filtered back projection (Hilbert transform)

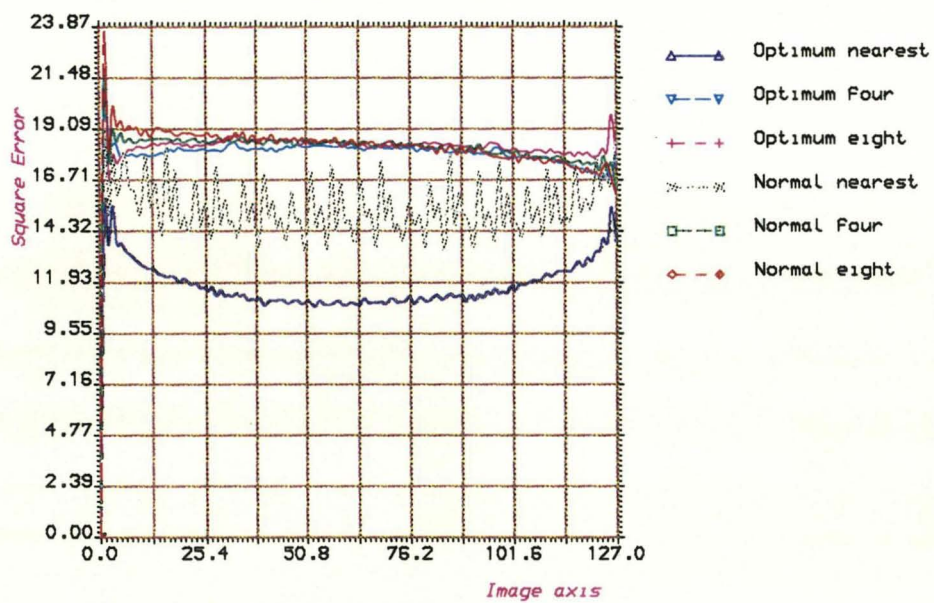


Figure 9.3 The projection error of a test image associated with nearest neighbour using Filtered back projection (1D inverse Filter).

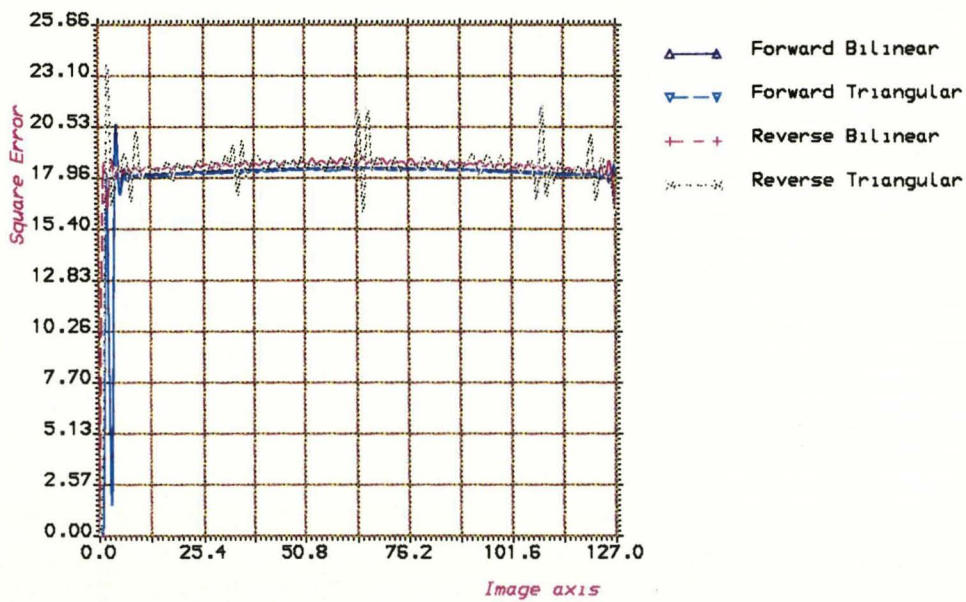


Figure 9.4a The projection error of test image associated with interpola technique using Filtered back projection (1D inverse Filter)

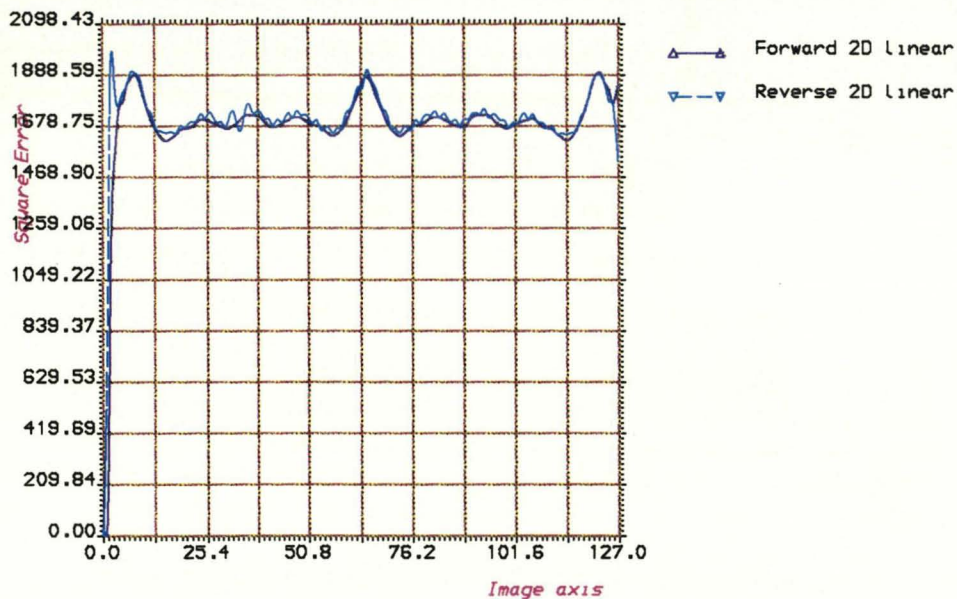


Figure 9.4b The projection error of test image associated with interpola technique using Filtered back projection (1D inverse Filter)

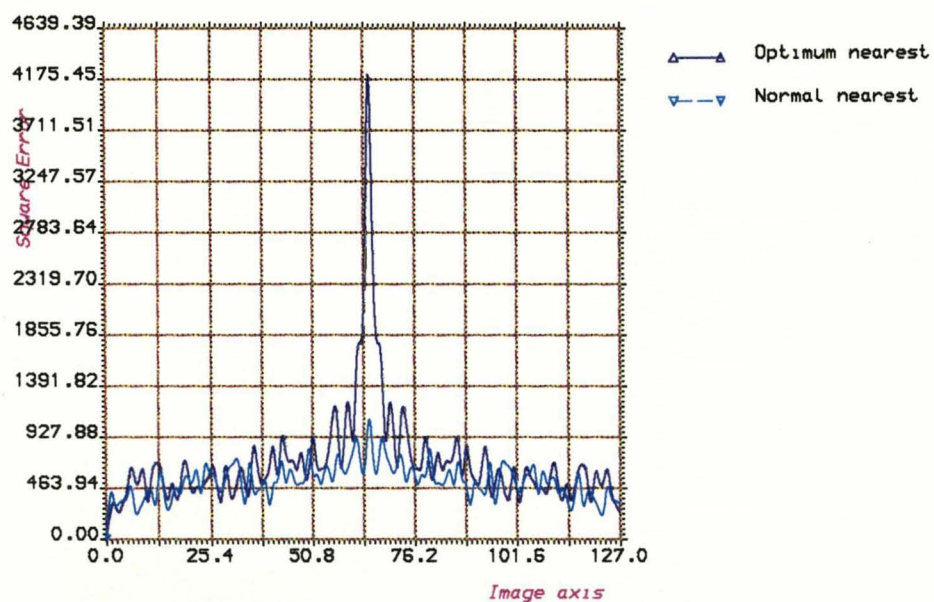


Figure 9.5a The projection error of a test image associated with nearest neighbour using back projection and deconvolution.

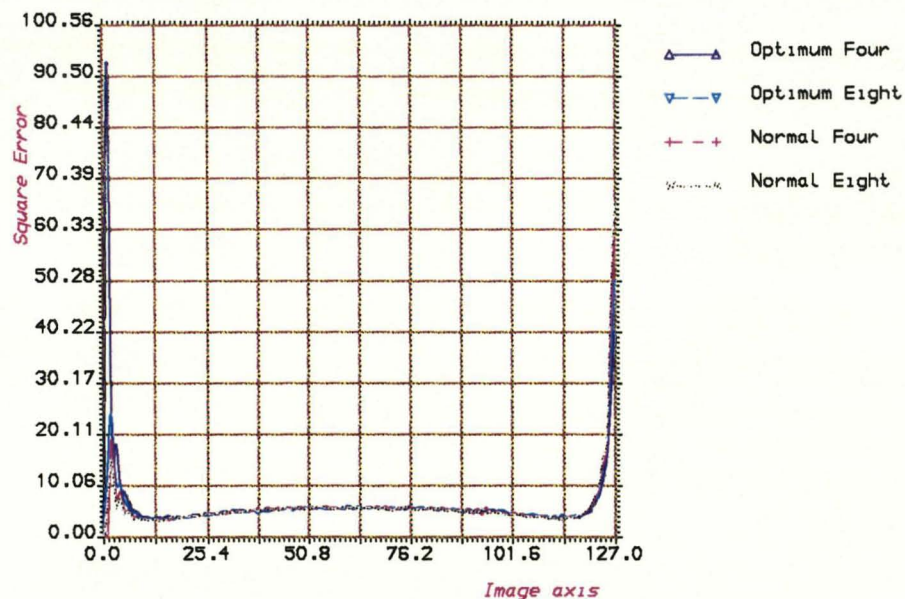


Figure 9.5b The projection error of a test image associated with nearest neighbour using back projection and deconvolution.

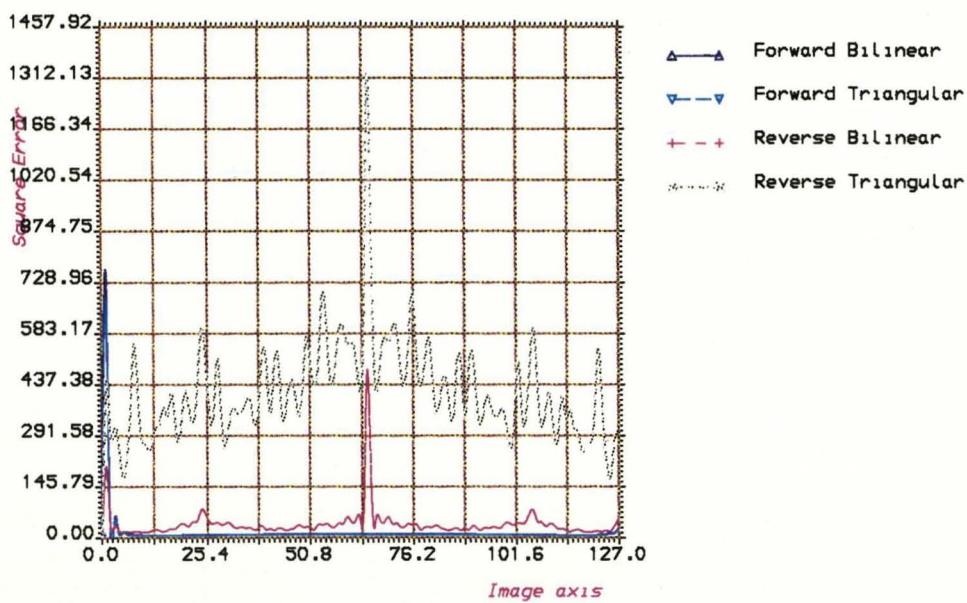


Figure 9.6a The projection error of a test image associated with the interpolation technique using back projection and deconvolut

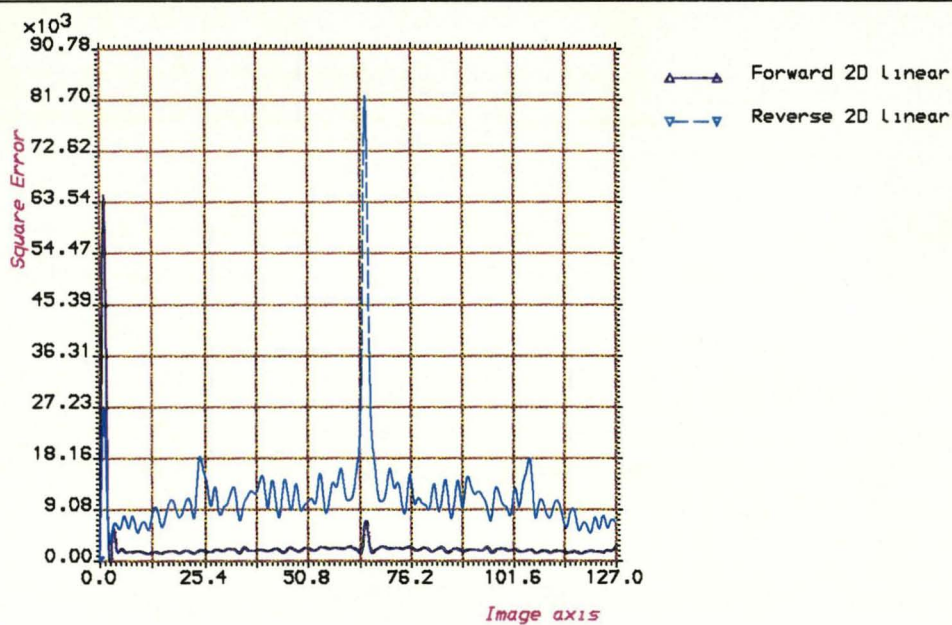


Figure 9.6b The projection error of a test image associated with the interpolation technique using back projection and deconvolut

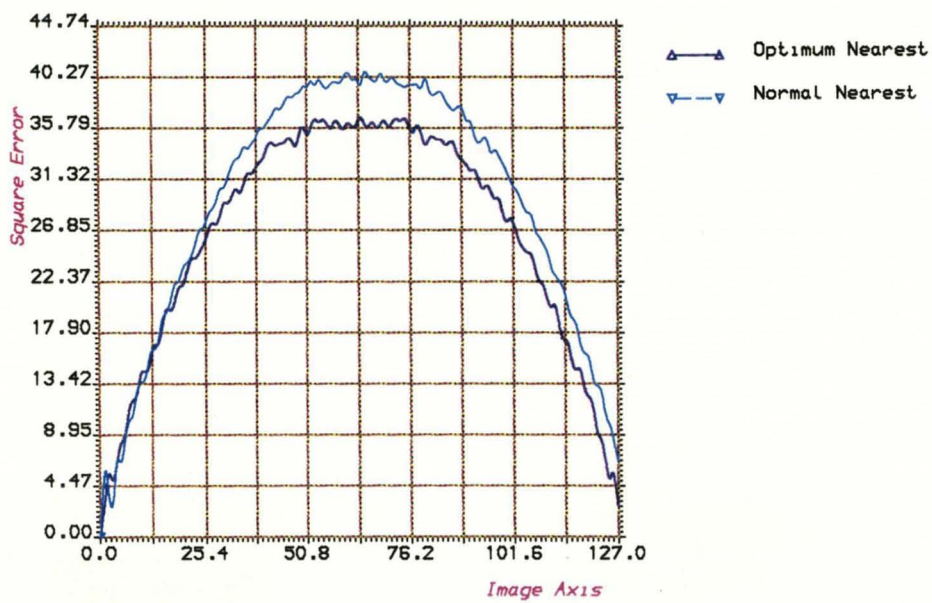


Figure 9.7a The projection error of a test image associated with the nearest neighbour using central-slice theorem.

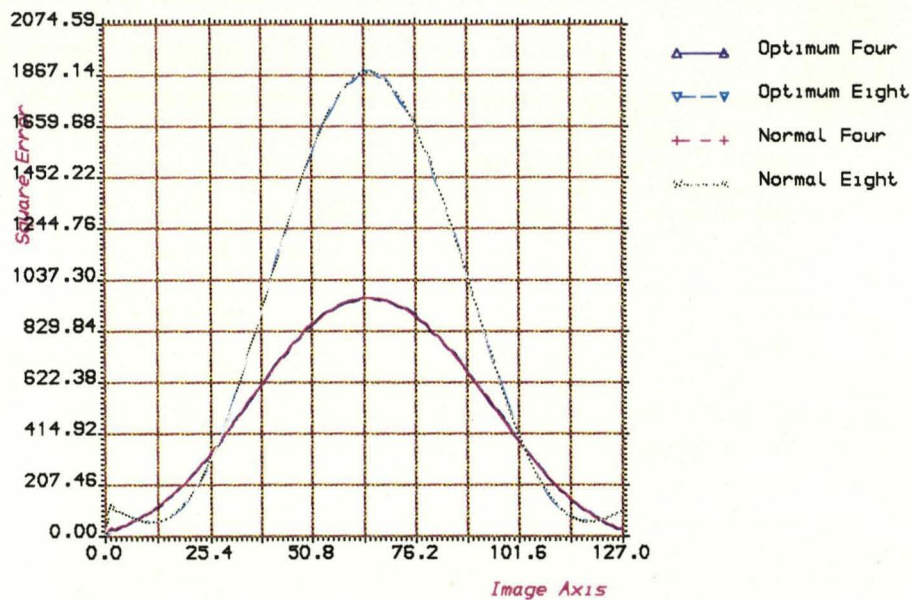


Figure 9.7b The projection error of a test image associated with the nearest neighbour using central-slice theorem.

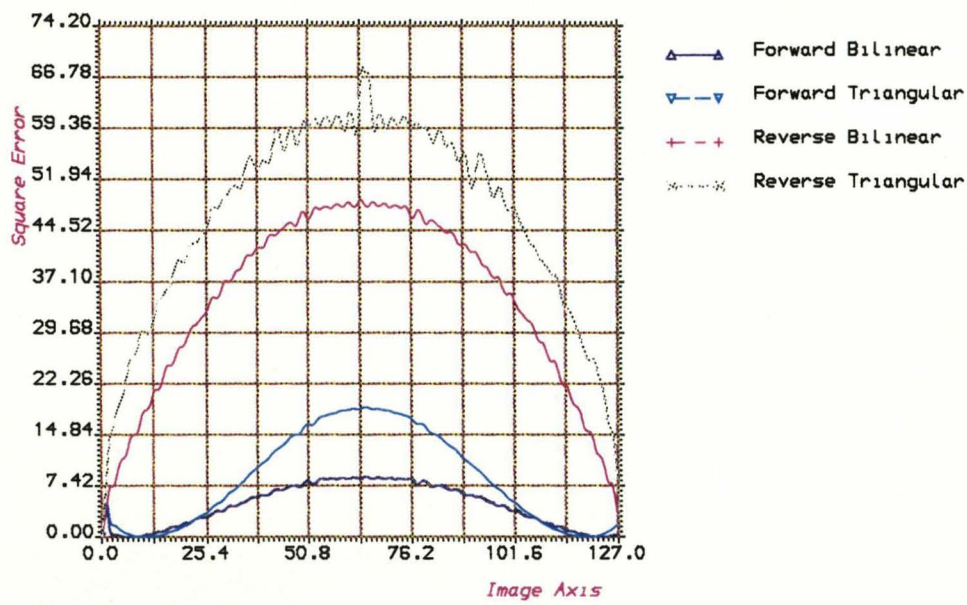


Figure 9.8a The projection error of a test image associated with the interpolation technique using central-slice theorem.

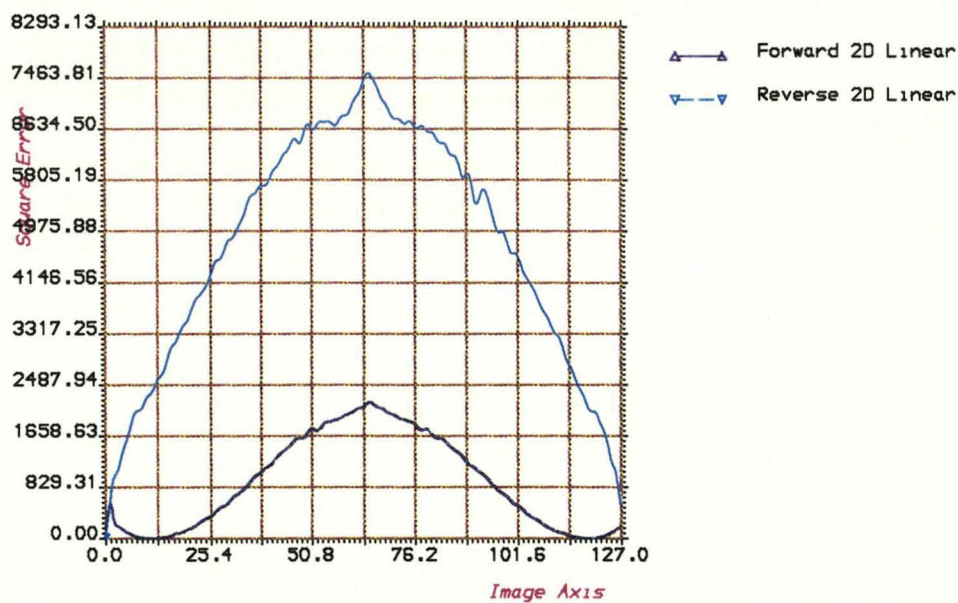


Figure 9.8b The projection error of a test image associated with the interpolation technique using central-slice theorem.

Table 9.1 Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using filtered-back projection technique(backward difference approximation and Hilbert transform).

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Nearest neighbour techniques	Optimum techniques	Nearest	00:46.35	01:11.67	3.6449
		Four	00:52.06	01:19.38	5.5505
		Eight	00:54.19	01:27.62	5.6162
	Normal techniques	Nearest	00:55.97	01:27.77	4.4636
		Four	01:00.97	01:36.23	5.6019
		Eight	01:00.28	01:49.40	5.6363

Table 9.2 Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using filtered-back projection technique(backward difference approximation and Hilbert transform).

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Interpolation techniques	Bilinear (rectangular)	Forward	02:26.66	04:29.07	5.5524
		Reverse	02:28.16	04:35.57	5.6164
	Triangular	Forward	02:18.37	04:19.65	5.5564
		Reverse	01:29.10	02:34.40	5.6293
	2-D linear interpolation	Forward	06:41.55	13:06.48	505.268
		Reverse	07:13.89	14:11.73	506.782

where:

RT is the Radon Transform.

IRT is the Inverse Radon Transform.

Table 9.3 Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using filtered-back projection technique(One-Dimensional inverse filter).

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Nearest neighbour techniques	Optimum techniques	Nearest	00:47.68	01:09.81	9.2181E-02
		Four	00:50.69	01:20.41	0.14055
		Eight	00:53.26	01:27.94	0.14310
	Normal techniques	Nearest	00:54.69	01:28.08	0.12065
		Four	00:58.30	01:37.13	0.14251
		Eight	00:59.74	01:45.02	0.14302

Table 9.4 Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using filtered-back projection technique(One-Dimensional inverse filter).

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Interpolation techniques	Bilinear (rectangular)	Forward	02:28.19	04:30.20	0.14178
		Reverse	02:27.99	04:35.19	0.14005
	Triangular	Forward	02:19.49	04:18.94	0.14104
		Reverse	02:23.90	04:28.62	0.14425
	2-D linear interpolation	Forward	05:40.17	11:10.18	13.2324
		Reverse	05:45.41	11:19.13	13.3824

where:

RT is the Radon Transform.

IRT is the Inverse Radon Transform.

Table 9.5 Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using back projection and deconvolution technique.

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Nearest neighbour techniques	Optimum techniques	Nearest	00:46.33	01:08.95	5.5168
		Four	00:48.69	01:17.21	6.3105E-02
		Eight	00:50.37	01:25.37	7.1370E-02
	Normal techniques	Nearest	00:55.27	01:27.95	4.14137
		Four	00:56.40	01:36.89	7.9810E-02
		Eight	00:57.94	01:43.52	8.8804E-02

Table 9.6 Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using back-projection and deconvolution.

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Interpolation techniques	Bilinear (rectangular)	Forward	02:28.04	04:30.72	0.13003
		Reverse	02:30.13	04:43.21	0.31016
	Triangular	Forward	02:22.50	04:21.34	0.11655
		Reverse	02:24.31	04:25.79	3.2702
	2-D linear interpolation	Forward	07:03.54	13:59.82	19.1871
		Reverse	06:07.54	11:44.56	89.8645

where:

RT is the Radon Transform.

IRT is the Inverse Radon Transform.

Table 9.7 Comparison of CPU reconstruction time and the mean square error associated with the nearest neighbour techniques using central slice theorem.

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Nearest neighbour techniques	Optimum techniques	Nearest	00:50.48	01:44.10	0.20616
		Four	00:54.06	01:45.03	3.8826
		Eight	00:53.49	01:44.52	6.20754
	Normal techniques	Nearest	00:55.29	01:39.76	0.22843
		Four	00:55.80	01:56.66	3.8904
		Eight	00:57.56	01:56.34	6.2129

Table 9.8 Comparison of CPU reconstruction time and the mean square error associated with the interpolation techniques using central slice theorem.

Techniques		Methods	RT CPU time(sec)	IRT CPU time(sec)	Mean Square Error
Interpolation techniques	Bilinear (rectangular)	Forward	01:40.79	02:23.13	3.532E-02
		Reverse	01:34.59	02:02.10	0.2671
	Triangular	Forward	01:37.93	02:23.15	6.5214E-02
		Reverse	01:30.73	02:00.23	0.36059
	2-D linear interpolation	Forward	03:02.99	02:29.67	7.0699
		Reverse	02:53.98	02:07.05	36.779

where:

RT is the Radon Transform.

IRT is the Inverse Radon Transform.

Table 9.9 The computation energy associated with the nearest neighbour techniques using filtered-back projection (Hilbert transform).

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Nearest neighbour techniques	Optimum techniques	Nearest	118.02	466.551	95.597	3.360
		Four	131.44	710.468	113.59	5.699
		Eight	141.81	718.880	131.42	6.222
	Normal techniques	Nearest	143.74	570.061	110.99	5.001
		Four	157.20	717.050	114.24	6.879
		Eight	169.68	721.451	106.47	7.471

Table 9.10 The computation energy associated with the interpolation techniques using filtered-back projection (Hilbert transform).

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Interpolation techniques	Bilinear (rectangular)	Forward	415.73	710.711	140.601	18.033
		Reverse	423.73	718.912	135.123	18.592
	Triangular	Forward	398.02	711.224	145.223	17.277
		Reverse	243.50	720.562	132.781	10.709
	2-D linear interpolation	Forward	1188.03	64674.13	13401.8	4689.62
		Reverse	1285.62	64868.19	12138.2	5090.07

where:

MPE is the mean of the projection square error.

SDPE is the standard deviation of the projection square error.

CE is the computation energy.

Table 9.11 The computation energy associated with the nearest neighbour techniques using filtered-back projection (1D inverse filter).

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Nearest neighbour techniques	Optimum techniques	Nearest	117.49	11.799	0.9838	0.0846
		Four	131.10	17.991	0.4151	0.1439
		Eight	141.20	18.317	0.2800	0.1578
	Normal techniques	Nearest	142.77	15.444	1.1813	0.13458
		Four	155.43	18.241	0.4753	0.1730
		Eight	164.76	18.307	0.7145	0.1841

Table 9.12 The computation energy associated with the interpolation techniques using filtered-back projection (1D inverse filter).

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Interpolation techniques	Bilinear (rectangular)	Forward	418.39	18.1487	1.5894	0.4634
		Reverse	423.18	18.5607	0.3857	0.4794
	Triangular	Forward	398.43	18.0541	1.5717	0.4390
		Reverse	412.52	18.4641	1.4733	0.4648
	2-D linear interpolation	Forward	1010.35	1693.747	167.764	104.44
		Reverse	1024.54	1712.953	120.177	107.116

where:

MPE is the mean of the projection square error.

SDPE is the standard deviation of the projection square error.

CE is the computation energy.

Table 9.13 The computation energy associated with the nearest neighbour techniques using back-projection and deconvolution.

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Nearest neighbour techniques	Optimum techniques	Nearest	115.28	706.148	481.485	4.968
		Four	125.90	8.0774	18.600	0.0620
		Eight	135.74	9.1361	35.277	0.0756
	Normal techniques	Nearest	143.22	530.097	150.972	4.6338
		Four	153.26	10.2156	47.739	0.0955
		Eight	161.46	11.367	59.755	0.1120

Table 9.14 The computation energy associated with the interpolation techniques using back-projection and deconvolution.

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Interpolation techniques	Bilinear (rectangular)	Forward	418.76	16.6450	66.286	0.4254
		Reverse	433.34	39.7011	45.8529	1.0500
	Triangular	Forward	403.84	14.9188	59.031	0.3677
		Reverse	410.10	418.5912	138.321	10.477
	2-D linear interpolation	Forward	1263.36	2455.946	5531.70	189.37
		Reverse	1132.10	1150.63	7781.96	794.807

where:

MPE is the mean of the projection square error.

SDPE is the standard deviation of the projection square error.

CE is the computation energy.

Table 9.15 The computation energy associated with the nearest neighbour techniques using central-slice theorem.

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Nearest neighbour techniques	Optimum techniques	Nearest	154.58	26.3889	10.0150	0.2489
		Four	159.09	496.980	326.148	4.8250
		Eight	158.01	794.568	669.991	7.662
	Normal techniques	Nearest	157.05	29.2396	10.9090	0.2802
		Four	172.46	497.975	325.970	5.2417
		Eight	173.90	795.261	671.143	8.440

Table 9.16 The computation energy associated with the interpolation techniques using central-slice theorem.

Techniques		Methods	Total CPU time(sec)	MPE	SDPE	CE
Interpolation techniques	Bilinear (rectangular)	Forward	243.92	4.522	3.0674	0.0673
		Reverse	216.69	34.193	13.2547	0.4522
	Triangular	Forward	241.08	8.3475	6.8409	0.1228
		Reverse	210.96	46.1562	14.747	0.5943
	2-D linear interpolation	Forward	332.66	904.963	754.524	18.374
		Reverse	301.03	4707.83	1958.44	86.499

where:

MPE is the mean of the projection square error.

SDPE is the standard deviation of the projection square error.

CE is the computation energy.

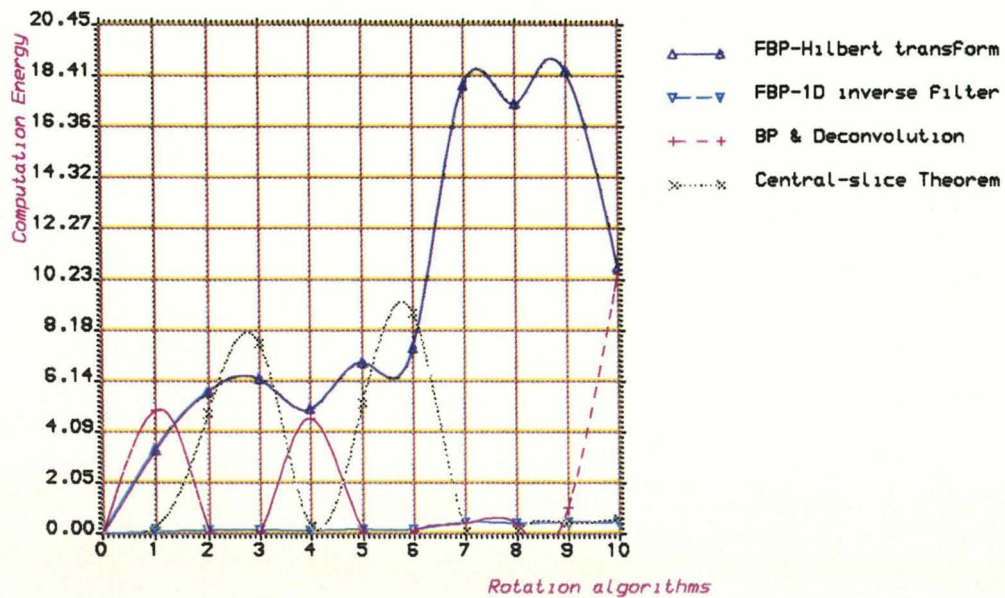


Figure 9.9 The computation energy associated with the reconstruction techniques using different rotation algorithms.

Where:

- | | |
|---------------------|------------------------|
| 1: Optimum Nearest | 2: Optimum Four |
| 3: Optimum Eight | 4: Normal Nearest |
| 5: Normal Four | 6: Normal Eight |
| 7: Forward Bilinear | 8: Forward Triangular |
| 9: Reverse Bilinear | 10: Reverse Triangular |

CHAPTER 10

Conclusion and Future Work

10.1 Introduction

The research results described in this manuscript represent the most recent work in both computed tomography for image reconstruction and multidimensional signal processing. We emphasize the real-time computational problem of the computed tomography. While a number of results have been shown and problems have been solved, there are still more avenues of research that must be addressed. In this final Chapter, we will summarize the work presented earlier and describe some new areas of study that will render a greater understanding of the image reconstruction problem.

10.2 Conclusions

Computer algorithms and architectures for fast implementation of CT algorithms are an important continuing area of research. As the reconstruction algorithms for CT have become more sophisticated over the years, the demands on computer systems have become correspondingly greater. The speed of reconstruction becomes increasingly important as we move toward real-time tomography.

The first goal of our research is to assess the applications of the Radon transform for image processing. Since the appropriate unifying mathematical frame

work for a large class of reconstruction problems is the Radon transform. It has a broad spectrum of applications such as industrial computer vision, medical imaging, diffraction tomography, astronomy, optics, nuclear magnetic resonance, geophysics, material science, molecular biology and military applications.

The second goal is to find out the relationship between image rotation and the reconstruction process, since the approach adopted in this research to computing the Radon transform is based on rotating the object about its centre and taking projections at regular steps in the angle of rotation. As a result, a new method for digital image rotation has been derived that can rapidly reduce the reconstruction time.

The third goal of our research is to investigate how the quality of the reconstructed image is influenced by rotation that occurs in computing the Radon transform and its inverse, and how this rotation time can be minimized during the reconstruction process. A numerical study has been carried out to test the relationship between different methods of image rotation and their effect on the performance for computing the Radon transform. The results show that the optimum algorithm is based on the optimum nearest neighbour method of image rotation to recover the object by using the filtered back projection (1D inverse filter). As a result, the new technique for image reconstruction offered four advantages over the current technique:

- 1- It provides better resolution of the reconstructed image (by visual inspection).
- 2- It drastically reduces the CPU time for image reconstruction algorithm.
- 3- It gives minimum least square error compared to the current algorithm.
- 4- It gives minimum projection square error along the image axis compared to the current algorithm.

The forth goal of our research is to combine the aforementioned features in a mathematical formula to describe the system performance. This formula is called

the computation energy. A quantitative study has been carried out to compute the computation energy for each reconstruction technique using different rotation algorithm. The results show that the new technique saved computation energy by 54.04 % compared to the current one.

Another goal of our research is to derive the mathematical formulation of the reconstruction using central-slice theorem. Since we have proved that the mathematical formula should include the 2D inverse filter as described in Section 5.3.2 and Section 8.6.

The final research has been undertaken in assessing the Radon transform that can be naturally applied in other areas of multidimensional signal processing. For example, the median filtering operations have used the dimension-reducing property of the Radon transform to perform the simpler 1D median filter on each projection independently, thereby allowing computational parallelism; i.e., rather than computing a 2D median filter of an image, we have computed a number of 1D median filter independently.

10.3 Future Work

In this final chapter, we will describe some new areas of study that will render a greater understanding of the image reconstruction problem. All of these topics loosely fit into one of three areas described below: Fast reconstruction, parallel computing for Radon transform and back projection, and multidimensional signal processing.

10.3.1 Fast reconstruction

As we move farther from the 2D to the 3D imaging problem, and as the reconstruction processes achieve greater usage, the speed of image reconstruction becomes a major bottle-neck to CT system throughput. A fast algorithm for computing the Radon transform and back-projection (the most time-consuming) has been shown.

Software simulations of this algorithm have been tested and a special-purpose hardware implementation of it will continue in a separate project. As a result, it is important to continue striving for faster computation of Radon transform and its inverse, in particular, for implementing the optimum algorithm.

10.3.2 Parallel implementation for Radon transform and its inverse

The Radon transform and its inverse (filtered back-projection) are receiving increasing attention for applications in image reconstruction. As the image reconstruction algorithms have become more sophisticated, the computational intensity of these problem has drastically increased. High-speed parallel hardware architectures will continue for the purpose of reducing image reconstruction times. An array of identical processors is organized to have both parallel and pipelined data paths will proposed for high-speed implementations of both the Radon transform and its inverse. Ideally, the array of processors is organized so that each processor handles the data for one projection angle. With this architecture it would be possible to handle images a pixel at a time in raster scan fashion at rates set by the speed of an individual processor.

10.3.3 Multidimensional signal processing

The advanced DSP architectures, VLSI circuit capabilities, and multiprocessor design techniques have made it possible and affordable to construct a computer for fast implementation of the forward and inverse Radon transform. Such a machine has many applications and, as with the fast Fourier transform, is likely to motivate new algorithms for general multidimensional signal processing using the Radon transform.

APPENDIX A

APPENDIX A

A1. Introduction

The software package in Volume II is divided into four appendices (Appendix A, B, C, D) which contains the FORTRAN source code for different application, i.e. rotation operation, filtering operations, fast Radon transform and 1D process, and reconstruction operations.

The aim of this software package is to use a set of routines as the basis for the development of image rotation and reconstruction package which is both user friendly and versatile in the sense that it can be expanded as more digital image processing and reconstruction routines become available.

Appendix A lists the rotation routines which rotate the image by an angle through 360° . It also lists a set of digital filter routines used to remove the noise from the rotated image and to enhance the quality of the image.

Appendix B lists the subroutines used for computing fast Radon transform and 1D process operation.

Appendix C lists the subroutines used for image reconstruction.

Appendix D lists the common subroutines used for the aforementioned appendices (A-C). These subroutines are the digital image processing routines. The routines designed for reading, writing and displaying images.

A2. Software Used

The software used was written in DEC FORTRAN using standard FORTRAN

77. Extensive use was made of UNIRAS-GIMAGE graphics library (Version 6.3) giving a 4-bit image.