INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Image reconstruction, recognition, using image processing, pattern recognition and the Hough transform

Seshadri, M. D., Ph.D.
The Ohio State University, 1992
IMAGE RECONSTRUCTION, RECOGNITION, USING IMAGE PROCESSING, PATTERN RECOGNITION AND THE HOUGH TRANSFORM

DISTRIBUTION

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

M. D. SESHADRI, B.E. M.Tech, M.S.

* * * * *

The Ohio State University
1992

Dissertation Committee
Dr. Don W. Miller
Dr. Tunc Aldemir
Dr. Yann Guzennec

Approved by

Advisor
Nuclear Engineering Program
To My Parents and my baby
ACKNOWLEDGEMENTS

I wish to gratefully acknowledge the guidance and the encouragement, moral and financial support given to me by my advisor, Professor Don. W. Miller.

A dissertation is not an individual effort but a cooperative effort with a group. I would like to thank all of my friends and colleagues for their support. I thank Nuclear Engineering Program, Department of Mechanical Engineering of the Ohio State University, Scantech Corporation, Ohio Technology Transfer Organization and Grimes Aerospace for the technical and financial support. I thank Prof. Tunc Aldemir for his encouragement and moral support and Prof. Yann Guzennec for being on my reading committee. I wish to thank Dr. Sumathi Shivaratri for her expert advice and Dr. Vishweshariah and my dad for the medical CT scans and radiologic diagnosis (findings).

Finally I thank my wife Suman for her understanding and also for typing and other editorial help.
VITA

November 28, 1959  Born - Bangalore, India
1982  B. E., Electrical Engineering, Bangalore University, India
1987-1991  Research and Teaching Associate, Nuclear Engineering Program, The Ohio State University, Columbus, Ohio

PUBLICATIONS


M.D. Seshadri and T. Aldemir, "Neutronic Scoping Calculations for OSURR Core Design With Standardized Fuel Plates", International Meeting on Reduced Enrichment for Test Reactors (RERTR), Gatlinburg, Tennessee, Nov. 05 1986.


FIELDS OF STUDY

Major Field: Nuclear Engineering
Studies in Medical Radiation Imaging Prof. Don W. Miller
Studies in AI Prof. Don W. Miller
Prof. B. Chandrasekaran
Studies in Nuclear Physics: Prof. Tunc Aldemir

V
TABLE OF CONTENTS

ACKNOWLEDGMENTS iii

VITA iv

LIST OF FIGURES viii

CHAPTER PAGE

I. INTRODUCTION

1.1 Overview 1

1.2 Objectives 2

1.3 Motivation 3

1.4 Methodology 5

1.5 Dissertation Organization 7

II. COMPUTER TOMOGRAPHY AND IMAGE PROCESSING

2.1 Introduction 9

2.2 Image Reconstruction and CT 10

2.2.1 Filtered Back Projection 11

2.2.2 Fast Fourier Transform 14

2.2.3 Filter Functions 17

2.3 Image Processing 22

2.3.1 Filtering 23

2.3.2 Edge Detection 24

2.3.3 Segmentation 38

2.3.4 Region Growing, Splitting and Merging 44

2.3.5 Region Growing Algorithm 46

2.3.6 Region Splitting 46

2.3.7 Low Level Description 47

2.3.8 Contour Based Descriptive Approaches 48

2.3.9 Boundary Description and Encoding 48

2.3.10 Chain Codes 50

2.3.11 Histogram Evaluation 53

2.4 Template Matching 54

2.4.1 Normalized Correlation 57

2.4.2 Matching with Generalized Geometric Distortions 57

vi
### 2.4.3 Thresholding

#### III. PATTERN RECOGNITION AND IMAGE ENHANCEMENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Introduction</td>
<td>60</td>
</tr>
<tr>
<td>3.2 Pattern Recognition</td>
<td>61</td>
</tr>
<tr>
<td>3.2.1 Hierarchical Clustering</td>
<td>64</td>
</tr>
<tr>
<td>3.2.2 Nearest Neighbor Algorithm</td>
<td>66</td>
</tr>
<tr>
<td>3.2.3 Furthest Neighbor Algorithm</td>
<td>67</td>
</tr>
<tr>
<td>3.3 Image Enhancement</td>
<td>68</td>
</tr>
<tr>
<td>3.4 Neural Network Theory</td>
<td>70</td>
</tr>
<tr>
<td>3.4.1 Massive Parallelism</td>
<td>73</td>
</tr>
<tr>
<td>3.4.2 Design of Neural Network Computers</td>
<td>74</td>
</tr>
<tr>
<td>3.4.3 Learning</td>
<td>75</td>
</tr>
<tr>
<td>3.4.4 Tasks</td>
<td>77</td>
</tr>
<tr>
<td>3.4.5 Delta Rule</td>
<td>77</td>
</tr>
<tr>
<td>3.4.6 Supervised Classifiers</td>
<td>79</td>
</tr>
<tr>
<td>3.4.6.1 Single Layered Perceptron</td>
<td>79</td>
</tr>
<tr>
<td>3.4.6.2 Multi-Layer Perceptron</td>
<td>81</td>
</tr>
<tr>
<td>3.4.7 Back-Propagation Network</td>
<td>82</td>
</tr>
<tr>
<td>3.4.7.1 Forward Activation Flow</td>
<td>82</td>
</tr>
<tr>
<td>3.4.7.2 Backward Error Flow</td>
<td>83</td>
</tr>
<tr>
<td>3.4.8 Generalized Delta Rule</td>
<td>84</td>
</tr>
<tr>
<td>3.4.9 Unsupervised Classifiers</td>
<td>85</td>
</tr>
<tr>
<td>3.4.9.1 ART</td>
<td>85</td>
</tr>
</tbody>
</table>

#### IV. AN INTEGRATED IMAGING SYSTEM DESIGN

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>89</td>
</tr>
<tr>
<td>4.2 Matching Invariant to Position, Scale and Orientation</td>
<td>91</td>
</tr>
<tr>
<td>4.3 An Imaging System</td>
<td>92</td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>111</td>
</tr>
</tbody>
</table>

#### V. EVALUATION OF THE IMAGE INTERPRETATION SYSTEM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>114</td>
</tr>
<tr>
<td>5.2 Digital Radiography and DADR system</td>
<td>115</td>
</tr>
<tr>
<td>5.2.1 Tomography with the DADR system</td>
<td>119</td>
</tr>
<tr>
<td>5.2.2 The experimental set-up</td>
<td>120</td>
</tr>
<tr>
<td>5.3 Test Plan</td>
<td>121</td>
</tr>
<tr>
<td>5.3.1 Brain scans</td>
<td>122</td>
</tr>
<tr>
<td>5.3.2 Abdominal scans</td>
<td>127</td>
</tr>
<tr>
<td>5.4 Results and Analysis</td>
<td>135</td>
</tr>
<tr>
<td>5.5 Summary</td>
<td>168</td>
</tr>
</tbody>
</table>

#### VI. Summary and Conclusion

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>170</td>
</tr>
<tr>
<td>6.2 Application</td>
<td>172</td>
</tr>
<tr>
<td>6.4 Future Work</td>
<td>172</td>
</tr>
</tbody>
</table>

**LIST OF REFERENCES**

vii
<table>
<thead>
<tr>
<th>FIGURES</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Conversion from Fan beam geometry to Parallel beam geometry</td>
<td>13</td>
</tr>
<tr>
<td>2. The four filter functions</td>
<td>19</td>
</tr>
<tr>
<td>3. Typical edges encountered in an image</td>
<td>25</td>
</tr>
<tr>
<td>4. Brice\Fenneman Edge enumeration approach for contour determination</td>
<td>40</td>
</tr>
<tr>
<td>5. Edge tracking and growing</td>
<td>42</td>
</tr>
<tr>
<td>6. Connectedness and neighbors</td>
<td>44</td>
</tr>
<tr>
<td>7. Typical extracted contour for parameterization</td>
<td>48</td>
</tr>
<tr>
<td>8. Sample chain code contour representation</td>
<td>51</td>
</tr>
<tr>
<td>9. Neural net classifiers</td>
<td>73</td>
</tr>
<tr>
<td>10. Operation of a backpropogation network</td>
<td>77</td>
</tr>
<tr>
<td>11. The image interpretation system</td>
<td>95</td>
</tr>
<tr>
<td>12. Locus of parameters with no directional information</td>
<td>99</td>
</tr>
<tr>
<td>13. Neural network module input extraction</td>
<td>105</td>
</tr>
<tr>
<td>14. The DADR system for the tomographic configuration</td>
<td>118</td>
</tr>
<tr>
<td>15. Semantic network representing a part of the abdomen</td>
<td>128</td>
</tr>
<tr>
<td>16. Images for a normal Brain Scan 1</td>
<td>138</td>
</tr>
<tr>
<td>17. Additional images for a normal Brain Scan 1</td>
<td>139</td>
</tr>
<tr>
<td>18. Images for a normal Brain Scan 2</td>
<td>140</td>
</tr>
<tr>
<td>19. Images for a normal Brain Scan 3</td>
<td>140</td>
</tr>
<tr>
<td></td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>20</td>
<td>Images for a normal Brain Scan 4</td>
</tr>
<tr>
<td>21</td>
<td>Backprop learning for a normal brain</td>
</tr>
<tr>
<td>22</td>
<td>Images for a Haemmoraged Brain Scan 1</td>
</tr>
<tr>
<td>23</td>
<td>Images for a Haemmoraged Brain Scan 2</td>
</tr>
<tr>
<td>24</td>
<td>Images for a Lacunar Infarcted Brain Scan 3</td>
</tr>
<tr>
<td>25</td>
<td>Images for a Lacunar Infarcted Brain Scan 4</td>
</tr>
<tr>
<td>26</td>
<td>Backprop learning for a haemorraged brain</td>
</tr>
<tr>
<td>27</td>
<td>Backprop learning for a lacunar infarcted brain</td>
</tr>
<tr>
<td>28</td>
<td>Images for Abdominal region scan 1</td>
</tr>
<tr>
<td>29</td>
<td>Additional images for Abdominal region scan 1</td>
</tr>
<tr>
<td>30</td>
<td>Images for Abdominal region scan 2</td>
</tr>
<tr>
<td>31</td>
<td>Images for Abdominal region scan 3</td>
</tr>
<tr>
<td>32</td>
<td>Images for Abdominal region scan 4</td>
</tr>
<tr>
<td>33</td>
<td>Images for Abdominal region scan 5</td>
</tr>
<tr>
<td>34</td>
<td>Images for Abdominal region scan 6</td>
</tr>
<tr>
<td>35</td>
<td>Images for Abdominal region scan 7</td>
</tr>
<tr>
<td>36</td>
<td>Images for Abdominal region scan 8</td>
</tr>
<tr>
<td>37</td>
<td>Tomographic 2-D slice of an industrial phantom</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

1.1 Overview

The most familiar images are those formed directly by optical instruments using visible light reflected or transmitted by an object. However, in many applications in which an image is required, indirect measurements can be made by probing the object with penetrating radiation such as x-rays. Often the measured data are not in a form suitable for interpretation, but are related to the image in a known way. The general aim of all image reconstruction procedures or algorithms [1] is to process data to form an image and thus to facilitate the interpretation of measurements done on the object.

Further image processing is necessary to delineate and recognize the features present in an image. Pattern recognition is required to distinguish an abnormal object from a normal object. Statistical pattern recognition, is sometimes implemented using neural networks [2]. In some situation where the object conforms to a standard geometrical form, an image enhancement procedure called the Hough transform [3], or Hough procedure is applicable.
The use of X-rays for medical diagnosis and industrial non-destructive evaluation has grown tremendously since the discovery of X-rays by Roentgen. Computerized Tomography (CT) has improved upon the use of penetrating radiation, by devising a method through which three-dimensional information about the organ or object can be achieved by rotating the object or the x-ray source. The concept of CT has been applied to both in medical diagnosis, and has been adapted to industrial applications in chemical engineering, thermal hydraulics, and welding engineering. Even though the data obtained during CT are digital in nature, some of the images are reproduced on a CT film thus making them unsuitable for further processing. If the images were preserved in the digital format, then post processing techniques can easily be implemented on these images.

1.2 Objectives

The main objective of this dissertation is to integrate image reconstruction, pattern recognition, image processing and expert system methodologies into an imaging system capable of human like performance in object recognition. In the process utilize the higher level primitives of segmentation, shape and size, with the lower level primitives of grey value and edge images information to form a practical image interpretation system.
Specifically the objectives are to develop a radiation image reconstruction and image processing system for medical and industrial non-destructive evaluation protocol, that

(1) Performs image reconstruction to obtain a 2-D image of the object under evaluation.

(2) Filters the 2-D image to reduce noise in the images.

(3) Identifies objects in industrial scans, and organs in medical CT scans.

(4) Enhances the image to depict the object boundaries.

(5) Recognizes and compares a given 2-D objects or organ from a normal (organ) or image.

(6) Displays a 2-D reconstructed image with flaws in the object highlighted.

1.3 Motivation

The basic idea for this research was conceived when the author was investigating a Computerized Tomography (CT) application for industrial weld defect detection. The result of that research was 2-D scans of an industrial object. It was realized that, if post processing techniques were applied to these 2-D scans, then there would be a dramatical improvement in the resulting image. The need for post processing has been addressed in this dissertation research using techniques adopted from image processing such as edge detection, contour following, segmentation, pattern
recognition, and image enhancement, which have been used in other fields of research as diverse as photogrammetry, radar engineering, and computer vision.

This research addresses the issue of image recognition using pattern recognition and image enhancement for an image reconstructed from projections. The main objective is that these techniques in a knowledge base environment will assist in identifying tumors in medical organs, and defects and cracks in industrial objects. The original contribution of this research is the combination of all these techniques to yield a tool for radiation image reconstruction, enhancement and object recognition.

One of the major problems in industrial radiography is that a defect such as a hairline crack that occur during manufacturing of the object is often too minute to be visually seen and cannot easily be perceived from methods currently in use. X-rays have been used by many researchers for detection of flaws in an industrial object. X-rays, as they penetrate an object have the capability to give density information. Using X-rays which are laterally incident on the object, one can get a 2-D view of the object, and the resulting reconstructed image will be a 2-D slice of the object. Using suitable image reconstruction methodologies many such slices around an object are formed, which will yield a 3-D image of the object.
Techniques such as pattern recognition have been evaluated for the past thirty years, and image enhancement has been studied for the past two decades. Although they are relatively old techniques, they have not been used in a protocol complementing one another. In this research, the Hough transform has been used for image enhancement in conjunction with pattern classification and object recognition routines, so that the organs and defects were enhanced. Expert system which has only been mentioned and not implemented in any previous research for organ and defect recognition has been identified as a useful tool in this research. A typical intelligent knowledge base has 2-D images of some common internal organs, along with a description of their shape and position (with respect to other organs) in a typical 2-D image (slice).

1.4 Methodology

CT images obtained following suitable reconstruction are fed into an image processing system which performs basic image processing, like removal of noise and finding the edges in the image. The edge image is then fed into a pattern recognition system which implements segmentation algorithms and a Hough transform unit to determine the basic shape of the object (organ) and defect (if any). Finally all the above information is fed in to an expert system which selects a suitable model for the test object that is being imaged.
The components of the model in the expert system knowledge base are in the form of verbal descriptions, straight lines, rectangles, circles and ellipses. This model is compared with the normal image model and then the statistical pattern recognition system implemented using neural nets, decides and displays the deviation of the experimental data from data base knowledge. Comparison of selective features in the two images is a pattern recognition problem, which is implemented in this research by the neural nets which will infer any abnormal tumors or defects in the test image within a pre-determined standard deviation.

Computerized Tomography (CT) has been widely used in medical imaging for brain scans and other invasive procedures. It has also been used for industrial tomography by Seshadri et al. [4], and other researchers. Pattern recognition is a well established field in audio and alphabet recognition as described in Duda Hart [5], and in visual recognition of objects and images. The Hough transform as invented by Hough [6], and elucidated by Rosenfield [7] is a misnomer, because it is not really a transform but a procedure identified by Hough for detecting objects conforming to standard geometrical shapes. Neural networks are another vibrant area and its application envisages pattern recognition.
This dissertation research assembles these powerful technologies of image reconstruction, together with image processing techniques to form a protocol that can be utilized to detect objects and defects in medical images (scans), and industrial images with minimal human feedback or intervention.

1.5 Dissertation Organization

In section 2.2 the theoretical basis for computer tomography is described, followed by a review in section 2.2.1 of the popular convolution backprojection algorithm and the theory behind image reconstruction from projections. Section 2.2.2 introduce the algorithmic implementation of the Fast Fourier Transform (FFT). Section 2.3 is concerned with the theoretical basis behind image processing and has details in section 2.3.1 about spatial filters in section 2.3.2 has about edge detection operators, and in section 2.3.3 describes the segmentation schemes, and closed contour generation using chain code and Fourier techniques. In section 2.4 template matching is addressed, along with normalized and unnormalized correlations.

In section 3.2 the theoretical basis behind pattern recognition is described. Section 3.2.2 describes Neural net implementation of pattern recognition. Section 3.3.2 describes image enhancement implemented using the Hough transform.
Design of the integrated image interpretation system design is discussed in chapter 4. The theoretical basis for each technique is described and the integration of the various techniques is addressed. Finally the modelling scheme employed in this dissertation, which uses descriptive and geometrical information is described.

Chapter 5 discusses the results obtained using the modelling scheme described in chapter 4. Section 5.1 describes the experimental set-up for obtaining industrial images. The experiment and testing plan is described in section 5.2. Section 5.4 describes the results and analysis of the results.

Finally chapter 6 gives the conclusions drawn from this research and the direction for further research in this exciting area of image reconstruction, image processing and neural networks.
CHAPTER II

COMPUTER TOMOGRAPHY AND IMAGE PROCESSING

2.1 Introduction

In this chapter we will review selected publications related to image reconstruction, and image processing as well as the theoretical basis for these techniques. This literature review is related to these techniques with emphasis on the protocol and image system development related to the dissertation research goal.

An innovative and unique processing problem is the reconstruction of the image of an object from a set of transverse cross-sectional projections. In several applications, an image of an interior section of an object may be produced only in this manner without physically destroying the object. The significance of this technique is shown by its wide application in medical diagnosis, nuclear medicine, electron microscopy, radio astronomy, light microscopy, photogrammetry, and engineering. Rapid advances in the past two decades have created numerous algorithms for CT image reconstruction. These reconstruction algorithms are either iterative Herman [8], in which a large matrix of algebraic equations are solved by successive approximation
technique, or closed form. Closed form reconstruction algorithms which operate in frequency domain are frequently referred to as Fourier algorithms as described in the Donner manual [9], or more recently as Fast Fourier Transforms (FFT). Those that operate in the spatial domain are referred to as the Filtered Back Projection (FBP), or if convolution is used then they are referred to as Convolution Back Projection (CBP) algorithms, as described by Shepp and Logan [10]. There is ongoing research in CT directed to the development of more efficient CT machines and reconstruction algorithms, as described by Munshi [11]. The net result of their research has been improved image quality.

Image processing has been researched for well over three decades. The noise content in the images have been reduced by adopting various filtering techniques as given by Canny [12]. The edges in the images and thus the boundary have been identified by other researchers [23].[27] including Rosenfeld and A. C. Kak [13], as a very important part of image processing. Considerable research [14] effort has addressed this problem directly and have had some success in well defined domains. Most of the work has employed simulated images with little work performed with real images.

2.2 Image Reconstruction and CT

X-rays are the most widely used penetrating radiation to gather important information, such as the density of object.
The Fourier transform method reconstruction provides the easiest method of image reconstruction from multiple projections. This method is based on the fact that the Fourier transform of a two-dimensional projection of a three-dimensional object is exactly equal to the central section of the Fourier transform of the object. By rotating the projections and thus the Fourier transform section, the entire Fourier transform plane is first constructed and then the object reconstructed by simply taking the inverse Fourier transform.

The theory of image reconstruction from multiple projections using the Filtered Back Projection is described in the next section.

2.2.1 Filtered Back Projection

The reconstruction algorithm used in this study was suggested by Reed et al., [15] and has been used and implemented by Seshadri et al., [16] in their study of two-phase flow mixture. This algorithm is based on the principle of convolution. The algorithm development, is based on parallel x-ray beams, therefore in order to use the convolution technique in this study, a conversion from diverging beams to parallel beams geometry is required. The convolution method for image reconstruction is accurate and faster as compared to the other (FFT is faster but less accurate) methods of reconstruction. It eliminates an
integration (summation in the case of a computer program) and thus makes the program and hence the method more acceptable than the others where processing time and accuracy are important. This fact has been demonstrated by Seshadri et al., [1], in their work where different methods were taken and compared with respect to cpu time and reconstruction and the convolution method was overall superior to the other methods.

Referring to Figure 1 the following equations are applicable:

\[ \beta = i \alpha \quad 0 < i < N \quad \beta = \theta - \sin^{-1}(L/D) \] (2.1)

\[ \lambda = s \alpha \quad -M/2 < s < M/2 \quad \lambda = \frac{L}{\sqrt{1 - (L/D)^2}} \] (2.2)

\[ a = \lambda_{\text{max}} / M \quad M: \text{Number of diverging beams in the fan} \]

\[ a = \pi / N \quad N: \text{Number of Scans} \]

\[ P(L, \theta) = h(\lambda, \beta) = h(\frac{L}{\sqrt{1 - (L/D)^2}}, \theta - \sin^{-1}(L/D)) \] (2.3)

\[ C(La) = C(t_L, \theta_j) = \sum_{-M/2}^{M/2} p(t_k, \theta_j) W(t_L - t_k) \] (2.4)

\[ \theta_j \text{ is the } j^{th} \text{ angular position} = j \pi / N \]

\[ t_i = i \alpha \quad i = j, k, l \quad t = x \cos \theta_j + y \sin \theta_j \]

\[ b(t, \theta_j) = a \left[ \frac{t_{1+l-t}}{t_{1+l}-t_1} c(t_1, \theta_j) + \frac{t-t_1}{t_{1+l}-t_1} c(t_{1+l}, \theta_j) \right] \] (2.5)
\[
f(x,y) = \frac{1}{2N} \sum_{j=0}^{N-1} b(t, \theta_j)
\]
for $0 < j < N-1$

Figure 1 Conversion from Fan beam geometry to Parallel beam geometry

The equations (2.1) and (2.2), calculate the parameters so that the input data can be converted from diverging beam
to parallel beam geometry. Equation (2.3), represents the conversion of the input data of diverging beams to parallel beam data, for the application of the CBP algorithm. The convolution process is given by equation (2.4), wherein the converted data is convolved with the filter function. Back projection using first order interpolation is given by equation (2.5), and finally the reconstructed image is calculated by summing up the backprojected values and normalizing them as shown in equation (2.6).

The above equations were implemented in FORTRAN and C, optimized for speed to accomplish the objective of minimum processing time. The data after conversion from the diverging beams into a parallel beam are fed into a Shepp and Logan filter, which is a band pass filter with a cutoff frequency selected to minimize spatial oscillation in response to an edge or step function. The Shepp and Logan filter function is given by Equation (2.7),

$$W(t_L - t_k) = q(ka) = \frac{-4}{\pi(a^2(k^2 - 1))}. \quad \text{for } k=1,2,\ldots. \quad (2.7)$$

2.2.2 Fast Fourier Transform

The Fast Fourier Transform (FFT) is used to reduce the image reconstruction time with a tradeoff in accuracy. T. F. Budinger and G. T. Gullberg [17] suggested a method for doing image reconstruction using FFT techniques. The time taken for reconstruction using FBP is proportional to $n^3$, where $n$
is the matrix size. The time taken for reconstruction using FFT is proportional to \( n^2 \log(n) \). The FFT algorithm carries out the same steps as FBP, but uses filters in frequency space as compared to real space in the FBP, and uses the Fourier Transform instead of the Convolution.

The FFT algorithm performs the following sequence of operations: Fourier transforms the projection data vector; multiplies the complex values by one of the five optimal filters; inverse Fourier transforms these modified frequencies; and back projects the modified projection data. Equation (2.8) carries out this sequence of operation,

\[
X = \text{back-project} \left\{ \mathcal{F}^{-1}[c \ \mathcal{F}(p)] \right\}
\]  

(2.8)

where \( X \) is the transverse section, \( p \) are the projection data, \( c \) is the filter function and \( \mathcal{F} \) denotes the one-dimensional Fourier transformation. The filter function \( c \) is equal to the product of a window function \( w(R) \) and the absolute value of the frequency.

\[
c = R \ w(R)
\]  

(2.9)

If one considers the Fourier convolution theorem, this method of reconstruction is equivalent to the convolution method except that the convolution of the projection data is carried out in frequency space. The filter function \( c \) is the Fourier transform of the convolution function \( c \).

The digital implementation of the FFT algorithm was done by T. F. Budinger and G. T. Gullberg [18]. This performs the
discrete Fourier transform of the projection data as given by Equation (2.10),

\[ \tilde{p}_{km} = \frac{1}{K\text{DIMT}} \sum_{l=0}^{K\text{DIMT}-1} p_{lm} \exp\left(-i2\pi kl/K\text{DIMT}\right) \]  

(2.10)

where \( k \) is the projection bin index and \( m \) is the angle index. \( K\text{DIMT} \) is equal to \( 2^{\text{IPOW2}} \) where \( \text{IPOW2} = 2^x \) (the smallest power of two that is greater than or equal to number of projections). The factor of 2 is required so that the convolution result of one period does not overlap the convolution result of the succeeding period when using the discrete Fourier transform. After discrete Fourier transforming the projection data, Fourier transformed values \( \tilde{p}_{km} \) are multiplied by a filter function giving,

\[ \tilde{q}_{km} = \tilde{c}(k/K\text{DIMT}) \tilde{p}_{km} \]  

(2.11)

Then the values \( q_{km} \) are then discrete inverse Fourier transformed giving the convolved projection

\[ q_{km} = \frac{1}{K\text{DIMT}} \sum_{l=0}^{K\text{DIMT}-1} \tilde{q}_{lm} \exp\left(-i2\pi kl/K\text{DIMT}\right) \]  

(2.12)

The convolved projection data are then back-projected as in the convolution method to give the reconstruction

\[ X_{ij} = \frac{P}{N\text{ANG}} \sum_{k \text{m} i j} F_{k \text{m}} q_{km} \]  

(2.13)

where \( F_{k \text{m}} \) are the weighting factors in the projection and back-projection routines. The factor \( \pi/N\text{ANG} \) is the step size in the numerical calculation of the back-projection integral.
A different approach to FFT is attributed to R. H. T. Bates, and T. M. Peters [19]. In this method the image data are back projected; Fourier transforming the two-dimensional back-projection image; multiplying the two-dimensionally distributed Fourier coefficients by one of the optional filter functions; and then performing the two-dimensional inverse Fourier transform.

There are different ways to approach the back projection part of image reconstruction and one of the ways is by linear interpolation. One has to be careful in doing this since there is a transformation from polar to rectangular geometry. So the proper sine and cosines must be taken which is the most time consuming part of the algorithm.

2.2.3 Filter Functions

The algorithms described above require a filter to be designated. These algorithms have been developed with various options for frequency space filters because frequency space manipulation lends itself to easy tradeoff between noise propagation vs. resolution properties of the convolution kernel. The user can improve resolution by changing the filter shape, but the noise amplification will increase. Alternatively the user can suppress noise; however, this noise suppression will come at the cost of resolution. A second reason for incorporation of various filters with the Fourier space algorithms is that the
computational method for reconstruction is more efficient using the FFT rather than convolution in real space.

An excellent review of filtering and filter comparison has been done by R. K. Otnes and L. Enochson [20], and R. W. Hamming [21]. The five windows used for filter function are Butterworth, Hann, Hamming, Parzen, and Rectangular. These window functions when multiplied by the ramp function yield the respective filters.

Real space convolution and frequency filtering are equivalent operations. Thus the Ramp filter used with the FFT algorithm achieves the same result as the Ramachandran & Lakshminarayanan convolution function used with FBP. The use of convolution instead of Fourier transform has been demonstrated earlier by Ramachandran and Lakshminarayanan [22]. In the FFT method the operation of filtering is done by multiplying the filter values by the Fourier transform of the projection data, then the inverse Fourier transforming the result. In the convolution method the projection data modified in the same fashion is obtained by convolving the projection data with the real space equivalent of the ramp function in FBP.

The shapes of the window function are shown in Figure 2. The width of the window is measured as the distance between the closest zeroes on each side of the center lobe of the inverse Fourier transform of the window function. Ideally,
for good resolution, the window function should have a central lobe that is high and narrow. The side lobes for the inverse Fourier transform of these window functions give rise to the Gibbs phenomenon, which is observed as artifacts that are contamination from adjacent parts of the reconstruction.

Figure 2 The Four filter functions
To measure the key performance parameters, the standard procedure is to employ a medical phantom for medical images. In this procedure, a calibration image is taken using water or air, the signal and noise components or quantified by this process. Next the standard phantom is imaged and another quantification of the signal and noise is obtained. Using the above data the noise component is separated from the signal and signal to noise ratio is computed. The value of the attenuated image is calculated using the standard attenuation equation. This theoretically calculated value is compared with the detector reading and thus the detector efficiency is calculated. By using the procedure outlined previously, the noise sources can be separated, if different calibration runs are taken and also standard phantoms are used, except in the case where there is nonlinear noise.

Using materials of known attenuation and thickness we can measure the target photons to non-target photons. Using the same strategy as above in an industrial setting, we employed an industrial phantom, which gave a measure of the key features that were to be recognized in the object.

There can be other sources of noise, like misalignment, under sampling, and interpolation errors. When the projection data are systematically distorted in such a way
that they are not linearly proportional to true ray-sums of the attenuation coefficient, then this nonlinear noise is due to beam hardening. Also nonlinearities can occur if the detectors and/or electronics do not respond linearly to the dynamic range of the signal. So care has to be taken in the proper alignment of the object with the axis of the rotating spindle. Also a complete data set (i.e., 360 degree data around the object) is needed to mitigate the noise in the CT scans.

If we define resolution purely as the ability to separate two small points separated by distance Dr, then Dr is determined strictly by the quality of the data in any one view; i.e., by the spacing between samples (i.e., rays) in any view, as well as by the width of the sampling beam and by the subsequent digital filtering of the data. This relationship between single view sampling parameters and resolution has been confirmed theoretically by earlier research. In small details tomography resolution plays an important role in being able to discern small defects and cracks. Also it can give sufficient details about the object and its significant constituents.

The number of views, becomes significant only when we require accurate (artifact-free) reconstruction over some region of interest. In practice we do not have the ideal situation described in the definition of resolution, there is
the noise factor which affects the resolution. If we use some kind of digital filtering to reduce the noise level, this comes with a sacrifice in resolution and if we try to increase the resolution than the noise level also increases. This is because when we do filtering with a high pass filter allowing only components above a certain cutoff frequency, we loose some part of the signal while eliminating the low frequency noise, which are in the low frequency range. Thus resolution is dependent on noise, because the total signal is (signal + noise), and before it was only pure signal, which we were trying to distinguish, but now it is both signal and noise, thus a removal of noise implies a loss of some signal component and thus poorer resolution with lower noise level.

2.3 Image Processing

The imaging process confounds much useful physical information into the grey-level array. In this respect, the imaging process is a collection of degenerate transformations. However, this information is not irrevocably lost, because there is much spatial redundancy: Neighboring pixels in the image have the same or nearly the same physical parameters. A collection of techniques, which will be called early processing, exploits this redundancy in order to undo the degeneracies in the imaging process. These techniques have the character of transformations for changing the image into parameter images or intrinsic images which
reflect the spatial properties of the scene. Common intrinsic parameters are edges and boundaries in an image, clusters or groups, closed contours and modelling.

Filtering is a generic name for techniques of changing the image gray levels to enhance the appearance of objects. Edge operators detect and measure very local discontinuities in intensity or its gradient. The result of an edge operator is usually the magnitude and orientation of the discontinuity.

A pyramid is a general structure for representing copies of the image at multiple resolutions. A pyramid is a utility structure which can dramatically improve the speed and effectiveness of many early processing and later segmentation algorithms.

2.3.1 Filtering

The objective of image processing is to have noise free image so that all the objects are clearly delineated and to facilitate in recognition of the objects present by clearly defined edges and boundary contours. One of the straightforward method of noise removal is by filtering techniques. There are various types of filters that are in use, like the Ramp filter, Gaussian filter, High and Low pass filters, Band pass filters and many more. The principle behind the use of filters is that, if we assume that noise and signal are separable then the noise can be reduced or removed.
Filtering takes a new meaning in the case of image processing as opposed to filtering in image reconstruction. In this case the convolution function is weighted; in image processing terms this means the transformations that make the intensity discontinuities between regions more prominent. These transformations are often dependent on gross object characteristics. For example, if the objects of interest are expected to be relatively large, the image can be blurred to erase small intensity discontinuities while retaining those of the object's boundary. Conversely, if the objects are relatively small, a transformation that selectively removes large discontinuities may be appropriate.

2.3.2 Edge Detection

An important approach to picture segmentation is based on the detection of discontinuities, i.e., places where there is more or less abrupt changes in gray level or in texture, indicating the end of one region and the beginning of another. Such discontinuity is called an edge [23].

Biological visual systems appear to make use of edge detection, but not of thresholding. When a person looks at region across which there is a gradual brightness change, the person cannot see the region as broken up into two clear-cut parts, no matter how hard the person tries to threshold. If there is an abrupt brightness change cutting across the region, on the other hand, the person immediately sees an
edge there which breaks the region apart.

Abrupt changes in gray level can take several different forms. The most common is the step edge shown in Figure (3a). This of course is an ideal example; the presence of blur and noise turns steps into noisy ramps such as in Figure (3b). There can also be an abrupt change in the rate of gray level (Figure 3c).

![Figure a](image1.png) ![Figure b](image2.png)

![Figure c](image3.png) ![Figure d](image4.png) ![Figure e](image5.png)

Figure 3  Typical edges encountered in an image

A step edge separates two regions in each of which the gray levels (or texture) is relatively uniform, with different gray values on the two sides of the edge. Another type of gray level discontinuity is the line, which is a thin strip that differs from the regions on both sides of it; it has a spike like cross section as shown in Figure (3d). Lines often occur in association with edges (highlights on edges of blocks; membrane separating parts of a cell; roads running between fields bearing different crop types); an
idealized version of the resulting combine cross section is shown in Figure (3e).

A difficulty with edge detection as an approach to picture segmentation is that the detected edges often have gaps in them, at places where the transitions between regions are not abrupt enough. Moreover, the edges may be detected at points that are part of region boundaries, if the given picture is noisy. Thus the detected edges will not necessarily form a set of closed, connected curves that surround closed, connected regions. One way to overcome these problems is to use tracking techniques to follow the edges around the regions; such techniques can be designed to tolerate gaps in edges which do not lie on a curve. Another possibility is to apply curve detection operations to the edge detector output; this rejects edge points that do not lie on curves, and can also be designed to fill gaps in edges. A method of "filling in" regions that are surrounded by broken edges, using a propagation process is also used. The tracking approach is usually the best, because of its great flexibility; in particular, it can be designed to take into account any information that may be available about the shapes of the regions whose boundaries are being tracked.

A better way of doing edge detection is to look for the first derivative operator in both x and y direction. A directional derivative or difference operator is important
from the stand point view of border. Among the higher order operators the Laplacian is extensively used. Since most of the noise can be classified by using a Gaussian function, it is logical to use a Gaussian function for determining the noise in an image. Among many filters which were researched and tried on our images, the best appears to be the Canny's filter which is the improvement over the Laplacian of the Gaussian. Digital Laplacian performs an operation on the input image equivalent to high pass spatial filtering, because the low spatial frequencies are weakened, while the high ones remain relatively intact.

As explained in Rosenfeld and Kak [24], and by others [23] [25] [27], the number of connected components of a given picture subset (say) S increases drastically if S is noisy, since every isolated point of S is a component; the perimeter of S may increase greatly if S has a ragged border; the skeleton of S is very sensitive to the presence of small holes in S. Thus S has to be extracted from the picture correctly by applying noise-removing operations in some novel ways like shrinking S slightly and re-expanding the result should remove small isolated parts of S, and should smooth out the roughness in S's border.

Let C be a component of S, and D a component of S. By the D-border of C means the set of points of C that are adjacent to points of D. In our situation C is an 8-
component and D a 4-component, and we use 4-adjacency.

An algorithm for finding all the points of the D-border of C, given an initial pair of adjacent points c, d with c in C, d in D. First check that C is not just an isolated point; if C has just the one point c, this point is C's D-border; and there is nothing else to do. Otherwise, the algorithm proceeds as follows:

1. Change the values of c and d to 3 and 2, respectively.
2. Let the 8-neighbors of c in clockwise order, starting with d and ending with the first occurrence of 1, 3, or 4, be e₁, ⋯, eₖ.
3. If c = 3, eₖ = 4, and eₕ = 2 for some h < k, change the 3 to 4, the 2 to 0, and stop.
4. Otherwise, change the value of c to 4 (if it was 1). Take eₖ as the new c and eₖ₋₁ as the new d, and return to the previous step. When the algorithm stops the 4's are exactly the points of the D-border of C.

The algorithm given above will find the boundary, of an object which is relatively noise free. If the noise cannot be suppressed adequately then, spurious edges and boundaries will start appearing. During the process of object recognition, using apriori knowledge about the physical description of the object, some of these spurious edges can be removed.
Edge detectors of some kind, have been an essential part of many computer vision systems. The edge detection process serves to simplify the analysis of images by drastically reducing the amount of data to be processed, while at the same time preserving useful information about object boundaries. There is certainly a great deal of diversity in the application of edge detection, but it is felt that many applications share a common set of requirements. These requirements lead to the mathematical theory behind edge detection, which then can be applied in many problem domains.

Edge detection has been used in several specific applications. The Binford-Horn line finder [25] used the output of an edge detector as input to a program which could isolate simple geometric solids. More recently the model based vision system ACRONYM [26] used an edge detector as the front end to a sophisticated recognition program. Shape from motion [27] can be used to infer the structure of three-dimensional objects from the motion of edge contours or edge points in the image plane. Several modern theories of stereopsis assume that images are preprocessed by an edge detector before matching is done [28]. Beattie [29] describes an edge-based labelling scheme for low level image understanding. Finally, some novel methods have been suggested for the extraction of three dimensional information from image contours, namely shape from contour [30] and shape
from texture [31].

In all of the above examples there are common criteria relevant to edge detector performance. The first and most obvious is a low error rate. It is important that edges that occur in the images should not be missed and further that there be no spurious responses. In all of the above cases, system performance will be hampered by edge detector errors. The second criteria is that edge points should be well localized. That is the distance between the points marked by the edge detector and the center of the true edge should be minimized. This is particularly true of stereo and shape from motion, where small disparities are measured between left and right images or between images produced at slightly different times. The mathematical theory utilizing these two criteria which can be used to design an edge detector will be described in the following paragraph.

It is realized that the above criteria are not good enough and a third criteria has been proposed by Canny [12]. He proposes there is a single unique shape or impulse response for an optimal edge detector, and that the tradeoff between detection and localization specified by the two original criteria can be varied by changing the spatial width of the detector. Canny also demonstrated that there is a natural uncertainty relating the detection and localization of noisy edges.
The mathematical form which is readily solvable is given below. We will first deal with signal-to-noise ratio and localization. Let the impulse response of the filter be \( f(x) \), and denote the edge itself by \( G(x) \). We will assume that the edge is centered at \( x = 0 \). Then the response of the filter to this edge at its center is given by a convolution integral,

\[
H_c = \int_{-W}^{W} G(-x)f(x)\,dx
\]

(2.14)

where we assumed the filter has a finite impulse response bounded by \([-W, W]\). The root-mean-squared response to the noise \( n(x) \) is

\[
H_n = n_0 \left[ \int_{-W}^{W} f^2(x)\,dx \right]^{1/2}
\]

(2.15)

where \( n_0^2 \) is the mean-squared noise amplitude per unit length. We define our first criterion, the output signal-to-noise ratio, as the quotient of these two responses

\[
\text{SNR} = \frac{\int_{-W}^{W} G(-x)f(x)\,dx}{n_0 \sqrt{\int_{-W}^{W} f^2(x)\,dx}}.
\]

(2.16)

For the localization criterion, we want some measure which increases as localization improves, and we will use the reciprocal of the root-mean-squared distance of the marked edge from the center of the true edge. Since we have decided
to mark edges at local maxima in the response of the operator \( f(x) \), the first derivative of the response will be zero at these points. Note also that since edges are centered at \( x = 0 \), in the absence of noise there should be a local maximum in the response at \( x = 0 \).

Let \( H_n(x) \) be the response of the filter to noise only, and \( H_0(x) \) be its response to the edge, and suppose there is a local maximum in the total response at the point \( x = x_0 \). Then we have,

\[
H_n'(x_0) + H_0'(x_0) = 0. \tag{2.17}
\]

The Taylor expansion of \( H_0'(x_0) \) about the origin gives

\[
H_0'(x_0) = H_0'(0) + H_0''(0) x_0 + O(x_0^2). \tag{2.18}
\]

Assuming \( H_0'(0) \neq 0 \), i.e., the response of the filter in the absence of noise has a local maximum at the origin, so the first term in the expansion can be ignored. The displacement \( x_0 \) of the actual maximum is assumed to be small so we will ignore quadratic and higher terms. In fact we can show that if the edge \( G(x) \) is either symmetric or antisymmetric, all even terms in \( x_0 \) vanish. Suppose \( G(x) \) is antisymmetric, and express \( f(x) \) as sum of symmetric component and an antisymmetric component. The convolution of the symmetric component with \( G(x) \) contributes nothing to the numerator of the SNR, but it does contribute to the noise component in the denominator. Therefore, if \( f(x) \) has any
symmetric component, its SNR will be worse than a purely antisymmetric filter. A dual argument holds for symmetric edges, so that if the edge \( G(x) \) is symmetric or antisymmetric, the filter \( f(x) \) will follow suit. The net result of this is that the response \( H_0(x) \) is always symmetric, and that its derivative of odd orders are zero at the origin.

Equations (2.17) and (2.18) yield

\[
H''_0(0) x_0 = -H'_0(x_0) .
\]

(2.19)

Now \( H'_n(x_0) \) is a Gaussian random quantity whose variance is the mean-squared value of \( H'_n(x_0) \) and is given by

\[
E[H'_n(x_0)^2] = n_0^2 \int_{-w}^{w} f'(x)^2 dx ,
\]

(2.20)

where \( E[y] \) is the expectation value of \( y \). Combining this result with equation (2.19) and substituting for \( H''G(0) \) gives,

\[
E[x_0^2] = \frac{n_0^2 \int_{-w}^{w} f'(x)^2 dx}{\int_{-w}^{w} G'(-x)f'(x) dx} = \delta x_0^2 ,
\]

(2.21)

where \( \delta x_0 \) is an approximation to the standard deviation of \( x_0 \). The localization is defined as the reciprocal of \( \delta x_0 \).

\[
\text{Localization} = \frac{\int_{-w}^{w} G'(-x)f'(x) dx}{n_0 \sqrt{\int_{-w}^{w} f'(x)^2 dx}}
\]

(2.22)
Equations (2.16) and (2.22) are mathematical forms for the first two criteria, and the design problem reduces to the maximization of both of these criteria simultaneously. One simple way of doing this is to maximize the product of equations (2.16) and (2.22).

\[
\frac{\int_{-W}^{+W} G(-x)f(x)dx}{\int_{-W}^{-W} f^2(x)dx} \cdot \frac{\int_{-W}^{+W} G'(-x)f'(x)dx}{\int_{-W}^{-W} f'^2(x)dx}
\]

(2.23)

In the specification of the edge detection problem, we decided that edges would be marked at local maxima in the response to a linear filter applied to the image. The detection criterion given in the last section measures the effectiveness of the filter in discriminating between signal and noise at the center of the edge. It does not take into account the behavior of the filter nearby the edge center. The first two criteria can be trivially maximized as follows. From the Schwarz inequality for integrals we can show that SNR (2.16) is bounded above by

\[
\frac{1}{n_0} \sqrt{\int_{-W}^{-W} G^2(x)dx}
\]

(2.24)

and localization (2.9) by

\[
\frac{1}{n_0} \sqrt{\int_{-W}^{-W} G'^2(x)dx}
\]

(2.25)

Both bounds are attained, and the product of SNR and
localization is maximized when \( f(x) = G(-x) \) in \([-w,w]\).

Thus, according to the first two criteria, the optimal detector for edges is a truncated step, or difference of boxes operator. The difference of boxes was used by Rosenfeld and Thurston [32], and in conjunction with lateral inhibition by Herkovitz and Binford [33]. However it has a very high bandwidth and tends to exhibit many maxima in its response to noisy edges, which is a serious problem when the imaging system adds noise or when the image itself contains textured regions. These extra edges should be considered erroneous according to the first of our criteria. However, the analytic form of this criterion was derived from the response at a single point (the center of the edge) and did not consider the interaction of the responses at several nearby points. If we examine the output of a difference of boxes edge detector we find that the response to a noisy step is a roughly triangular peak with numerous sharp maxima in the vicinity of the edge.

These maxima are so close together that it is not possible to select one as the response to the step while identifying the others as noise. We need to add to our criteria the requirement that the function \( f \) will not have too many responses to a single edge in the vicinity of the step. We need to limit the number of peaks in the response so that there will be a low probability of declaring more
than one edge. Ideally, we would like to make the distance between the peaks in the noise response approximate the width of the response of the operator to a single step. This width will be some fraction of the operator width W.

In one dimension we can characterize the position of a step edge in space with one position coordinate. In two dimension an edge also has an orientation. In this section "edge direction", will mean the direction of the tangent to the contour that the edge defines in two dimensions. We wish to detect edges of a particular orientation, and characterizing all the other edges as noise. We create a two-dimensional mask for this orientation by convolving a linear edge detection function aligned normal to the edge direction with a projection function parallel to the edge direction. This is achieved using a symmetric two-dimensional Gaussian and then differentiating normal to the edge direction. In fact we do not have to do this in every direction because the slope of a smooth surface in any direction can be determined exactly from its slope in two directions. Let G denote the Gaussian function and G' the first derivative and G'' the second derivative.

The two dimensional Gaussian is given by the equation

\[ G(x) = \exp \left( -\frac{x^2 + y^2}{2 \sigma^2} \right) \cdot \]  (2.26)
Thus the Laplacian of the Gaussian in two-dimension is

$$G''(x,y) = \left(1 - \left(\frac{x^2 + y^2}{2\sigma^2}\right)\right) \exp\left(\frac{-x^2 + y^2}{2\sigma^2}\right).$$

Equations (2.28) convolved with the image function $I$, yields the required tool for edge detection which satisfies the three criterion set by Canny. It is difficult to implement it in the mathematical form shown, so a simplification is necessitated. An implementation scheme used for this research uses the associative property of the convolution function. Thus we can convolve first with the two dimensional Gaussian and then take the second derivative. Alternately we can convolve the dimension in $y$ of image $I$, with the first derivative of the Gaussian function in $x$, and then use symmetry property of the Gaussian. The effect would be working with two one dimension functions,

$$f(x) = \sigma^2 \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

$$g(x) = -x \sigma^2 \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

We have used equations (2.29) and (2.30) to show the effect of using one dimension Gaussian. Though we have to
repeat the one-dimensional convolution twice, considerable saving in computation time can be achieved because one dimensional convolution is much faster than two-dimensional convolution. The symmetrical properties of the Gaussian and the associativity of convolution have played a key role in this simplified alternative formulation of equation (2.28) with equations (2.29) and (2.30).

The mask of our edge detection filter was convolved with the image function and an oriented edge was obtained. Thus the edge detection system, from our study was fast, elegant in that it satisfied all the Canny's criterion's and was simple to implement.

2.3.3 Segmentation

Segmentation is a process in which like groups of input data are grouped together to form a coherent mass or a homogeneous entity. The process of segmentation groups pixels to form higher-level regional image structures (which may be associated with entities in a higher level model). Through segmentation, parts or regions of an image are identified prior to determining what they represent in the 3-D object. The success of the segmentation algorithm often determines the success or failure of the overall image analysis algorithm.

Segmentation is generally considered in the context of a pattern recognition classification problem, where the
entities to be classified may be either pixels or regions (groups of pixels). Generally clustering or unsupervised learning approaches are employed under the conditions that; each resulting segmented region or pixel group is as homogeneous as possible with respect to some measure of feature similarity; pixels in different subdivisions (or different regions) are inhomogeneous; and the resulting groupings have some meaning in terms of further processing. They are not artificial groupings of pixels, but rather constitute image subregions with meaningful interpretation as part of a higher order concept such as organs, bones, skull, blood vessels, blood, Cerebra Spinal Fluid(CSF), etc.

There are two types of approaches to segmentation. In noncontextual segmentation relations among features (pixels or regions) are ignored. In contextual segmentation the segmentation process employs neighboring relations among the features. Contextual classification is often more successful, because the local image information may reinforce a classification decision. However, it is also theoretically difficult to quantify, as well as practically cumbersome to implement.

A good example of a noncontextual approach is the classification of pixels using statistical pattern recognition approaches where the feature vector is intensity based. An example is the use of simple thresholding of pixel
data to assign pixels to either ON or OFF classes in the processed image. In contrast, edge detection and region growing approaches typify contextual classification approaches in that support for a segmentation decision comes from local pixel region content rather than from an individual pixel.

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\begin{array}{cccc}
. & . & . & . \\
+ & 0 & + & 0 + 0 + \\
. & . & . & . \\
+ & 0 & + & 0 + 0 + \\
. & . & . & . \\
+ & 0 & + & 0 + 0 + \\
\end{array}
\]

- Unassigned
- Edge data
- Gray level data

Grid Structure for potential edge representation and merging

Before merge

\[ a > b \]
\[ a < b \]

After merge

\[ a > b \]
\[ a < b \]

Edge merging

Figure 4 Brice/Fenneman Edge enumeration approach for contour determination
As described before, all low-level segmentation algorithms are based on the concept of similarity of attributes. One approach to detect dissimilarity is to identify grey-level intensity discontinuities, or edges. The edge-based segmentation problem becomes more difficult if we require that an image be segmented by determining regions with edges forming closed contours in the image plane.

Any grey-level disparity between spatially adjacent pixels represents a potential edge, which may be marked with directional information. Another conceptually obvious approach to segmentation is the determination of region edge locations, followed by the agglomeration of this edge information into regions. The classical approach [34], illustrates the potential complexity of this procedure. In this approach, a discrete image matrix of intensities \( F \), of spatial dimension \( n \times n \) is mapped into a larger array of dimension \( (2n + 1) \times (2n + 1) \), denoted by \( S \). This is shown in Figure 4. For example, the intensity of pixel \((i, j)\) is mapped to location \((2i + 1, 2j + 1)\) in \( S \). For each pixel in \( S \) the intensity is compared with that of its north and east intensity neighbors (or the pixels above and to its right, respectively). The magnitude and signs of this difference are used to compute a value for the two respective "+" cells as shown. Thus, directional edge segments are produced. This approach provides a global perspective on the image.
segmentation process, since it enumerates all potential interpixel edges. The representation allows the efficient merging of edge segments into regions as shown in Figure 4.

A difficulty that arises with applying edge operators followed by agglomeration of edge data into regions is that resulting regions are not required to be connected (i.e., enclosed by a closed contour of edge segments).

An example is an image that has been preprocessed using a LOG kernel. The resultant edge locations and orientation information is insufficient, in typical image applications, to produce a closed contour of significance.

![Edge tracking and growing](image)

**Figure 5** Edge tracking and growing

A related approach for the generation of a closed contour is similar to the region growing process and involves successively growing an edge contour, given an edge seed.
This approach is referred to as edge tracking [35]. Given an image of previously extracted potential edge element features, as shown in Figure 5, with a chosen seed we proceed to grow to track the edge by finding potential paths or contours. The combinatorial explosion inherent in this type of approach is evident; however with good feature extraction methodologies and evaluation functions (which choose only reasonable paths to investigate), edge tracking attention may remain focused. Typical evaluation functions may be based on:

1. Edge magnitude and direction. For example, we may only choose to pursue edge paths using adjacent pixels whose edge strength and direction feature similarity is above a suitable threshold.

2. Curvature of a segment. For example, we may know apriori some constraints on the maximum (or minimum) curvature of the contours of the class of objects under investigation.

3. Closeness to a known apriori contour. For example, if we know a priori that we are extracting circular or elliptical contours, this information constraints possible tracking directions.

An equivalent viewpoint for edge tracking or growing approach is that potential edge paths constitute a tree of possible tracking decisions in which one or more paths through the tree are correct. Thus, search is involved.
An analogous and strongly related task is that of thinning edge-enhanced images to produce a single region edge. As before the application of additional constraints is the key to success in obtaining a thin edge image.

2.3.4 Region Growing, Splitting and Merging

A brief description of algorithms that embodies region growing, region splitting, region merging and associated combination will follow later in this section. The basic concept involves locally growing a seed to annex similar pixels into the region. Region splitting, on the other hand, starts with the entire image, which is subsequently split into smaller regions on the basis of dissimilarity.

```
  x   x   x   x

  x   +(i,j)   x   x   +(i,j)   x

  x   x   x   x   x
```

4-connected neighbors 8-connected neighbors

Figure 6 Connectedness and neighbors

The connectedness of two pixels may be defined in several ways. Most often the concept of minimum spatial distance is used. Referring to Figure 6, pixel(i,j) is
connected to its nearest N, S, E and W neighbors (4-NN) on the sampling grid, in the case of 4-connectedness.

Pixels $x_i$ and $x_j$ are said to be connected in region $R$ iff:

1. There exists a sequence $\{ x_i \ldots x_j \}$ such that adjacent pixels in the sequence are connected, and

2. All points in this sequence are in region $R$.

Region $R$ is said to be connected if every pair of points in $R$ is connected. Regions with concave boundaries may still be connected.

The objective of a region growing/splitting/merging algorithm is to partition the set of all pixels in the input image, denoted $x$, into subsets $\{ R_1 \ldots R_n \}$ such that:

1. The sets are disjoint—that is, the intersection of sets $R_i$ and $R_j$ is the empty set; and

2. The union of the $R_i$ is $X$; that is, we classify all pixels through the development of regions.

In order to group or classify pixels into a region, a measure of similarity is required. This measure as described previously, may be based on features such as intensity, color, local texture, shape or other measures.

The following approaches are developed without specific enumerations of this similarity measure. Instead, the similarity measure is embedded in a predicate function, $H$, which determines homogeneity or region uniformity via
\( H(R_i) = \text{TRUE} \) if \( R_i \) is homogeneous or \( \text{FALSE} \) otherwise.

2.3.5 Region Growing Algorithm

Region growing is a bottom-up procedure for generation of a segmented image. It is based on observation that if \( H(R_i \cup R_j) = \text{TRUE} \), then there is nothing to distinguish regions \( R_i \) and \( R_j \). Thus, they should be merged. The methodology for region growing is as follows:

Begin by choosing seeds, \( s_i \) \( i = 1,2, \ldots, n \) (these are the initial regions). Until no further classification of pixels is possible, for all current regions, \( R_i \), for each pixel in \( R_i \) that has unclassified 8-neighbors (this forces the growing of connected regions). Examine each of these unclassified 8-neighbors and assign to \( R_i \), if possible.

2.3.6 Region Splitting

Region splitting is a divisive, or top-down procedure, based on observation. Beginning with all pixels in the same region, nonhomogeneous regions are recursively divided into smaller subregions, and the homogeneity test reapplied to these regions. When all regions satisfy the homogeneity criteria, the process stops.

In addition to the choice of \( H \), two major problems with region splitting are choice of a splitting or region subdivision methodology and choice of stopping criteria. A function \( H \) that is overly restrictive, will result in a final
segmented image where each pixel is a unique region.

A combination of splitting and merging algorithms to form an iterative procedure has been the research effort of Horowitz and Pavlidis [36].

2.3.7 Low Level or Early Description

The key to segmentation of image data into regions of pixels that correspond to discernible entities is the choice of information representations. These representations may include raw pixel intensities, regional boundary models, regional features, such as geometric properties, moments, 2-D topological descriptors, region intensity statistical descriptors (e.g., co-occurrence matrices), regional frequency characteristics, as well as representations obtained by syntactically combining extracted lower-level features.

Figure 7 Typical extracted region contour for parametrization
2.3.8 Contour Based Descriptive Approaches

The shape representation of an entity presupposes that we are able to extract the boundary of the 2-D region as shown in Figure 7. The principle underlying most of contour parameterization is the conversion of a 2-D shape, represented as a closed contour, to a (periodic) 1-D signal.

2.3.9 Boundary Description and Encoding

The boundary in Figure 7 may be described or parameterized in a number of ways. Clearly, since planar curves are generally not single-valued functions, representations of the contour in forms

\[ y = f(x) \]  \hspace{1cm} (2.31)

are, at best, valid locally. An alternative representation derived from differential geometry is the use of curvature. This involves the representation of a curve parametrically in terms of its arc or path length. By considering path length of the curve, expressed in terms of a variable, \( s \), (with some reference point, denoted \( (x(s_0), y(s_0)) \)), an arbitrary curve, \( C \), may be expressed in the form

\[ C = \{x(s), y(s)\} \hspace{1cm} s \in [s_0, L] \]  \hspace{1cm} (2.32)

where \( L \) is the total path length of the curve. If \( C \) is a closed curve, \( x(s) \) and \( y(s) \) are periodic. We define the curvature, \( k \), of a curve at point \( P \) to be instantaneous rate of change of the slope (or tangent) angle \( \phi \) (measured with
respect to the coordinate system shown in Figure 2.6) with respect to curve length, \( s \): that is,

\[
k(s) = \frac{d\phi(s)}{ds}
\]

(2.33)

where

\[
ds = \sqrt{dx^2 + dy^2}
\]

(2.34)

Knowing \( k(s) \) over the interval \([0, L]\) and the reference point \((x(s_0), y(s_0))\) allows exact reconstruction of the contour function \( C \). Contour representations allow classification of regions based on this measure of shape. For example, it is possible to show that the curvature of a circle is constant, whereas that of a line is zero. Discarding the dependence of \( k(s) \) on \((x(s_0), y(s_0))\), we obtain a parameterization that is insensitive to translation since translation merely manifests itself as a variation in \((x(s_0), y(s_0))\). Similarly, if the curve length parameter is to be normalized, for example, over the interval \([0, 1]\), the representations becomes scale invariant. However, this representation is not invariant to rotation; that is due to the fact that \( k(s) \) depends on the starting value of \( s \) along the curve.

Given \( x(s) \) and \( y(s) \), \( k \) can be computed by recalling the following standard definitions of spatial derivatives

\[
y' = \frac{dy}{dx}
\]

(2.35)

\[
y'' = \frac{d^2y}{dx^2}
\]

(2.36)

from which it can be shown that
\[ k(s) = \frac{y''}{(1 + (y')^2)^{3/2}} \]  

(2.37)

\( k(s) \) is periodic with period \( L \), or \( k(s) = k(s + L) \). On this basis, the use of 1-D Fourier coefficients is appropriate. \( k(s) \) can be expanded as

\[ k(s) = \sum_{n=-\infty}^{\infty} a_n \exp[jn2\pi s/L] \]  

(2.38)

where

\[ a_n = \int_{0}^{L} k(s) \exp[-jn2\pi s/L] \, ds \]  

(2.39)

The translations of a function manifest themselves as an additive (but linear) phase term in the Fourier transform, and assuming that the first few terms of the Fourier series are adequate to represent \( k(s) \), the modification (by simply using the transform magnitude only) of several low-order Fourier series coefficients yields a translation invariant curvature description methodology.

2.3.10 Chain Codes

The curvature-based contour representation approach must be modified to allow use with the discretized pixel data encountered in actual application. One popular extension is the use of discrete boundary codes. This encoding is similar in concept to the use of curvature, in the sense that a local measure of the boundary orientation is developed, as the curve is traversed (i.e., as a function of curve length). Thus, given a set of discrete boundary orientation (and
perhaps length) primitives, a polygonal representation of the boundary can be used to generate the code. As discrete samples are being considered, the resultant parameterization is in terms of a sequence or "chain" of discrete descriptors, or "chain codes".

![Contour segment](image1)

![Chain-encoded representation](image2)

### Primitives

<table>
<thead>
<tr>
<th>Code</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8 Sample chain code contour representation**

Referring to Figure 8, a piecewise linear approximation to the contour can be developed using a set of orientation-
only primitives. Thus, the chain encoding approach is similar to generation of a syntactic description of the boundary, using the primitives as shown in the figure. An alternate mechanism for viewing this approach is derived from a "neighborhood matrix", with each neighbor coded to correspond to the primitives in the figure. This matrix appears as follows

```
3 2 1
4 . 0
5 6 7
```

Differential encoding schemes are possible, a starting point is picked and successive boundary segments codes are generated based on the relative direction and the length of the edge segments. Consecutive line segments and six possible simple relations, namely shorter, longer, equal in conjunction with left and right can be used. The descriptor "left" or "right" indicates the direction to which the current line segment turns, relative to the previous segment. As a contour is traversed, each line segment is assigned one of the six codes. For an n-sided polygon, n-digit code results. Although this approach is RST invariant, it is sensitive to the choice of the starting point. To eliminate this sensitivity, the raw code may be normalized. An example is circularly rotating the raw code until the largest code value is leftmost.
2.3.11 Histogram Transformations

A gray level histogram of an image is a function that gives the frequency of occurrence of each gray level in the image. Where the gray levels are quantized form 0 to n, the value of the histogram at a particular gray level \( p \), denoted \( h(p) \), is the number or fraction of pixels in the image with that gray level.

A histogram is useful in many different ways. It can be used as a tool to guide gray-level transformation algorithms that are akin to filtering. A very useful image transformation is called histogram equalization. Histogram equalization defines a mapping of gray levels \( p \) into gray levels \( q \) such that the distribution of gray levels \( q \) is uniform. This mapping stretches contrast for gray levels near histogram maxima and compress contrast in areas with gray levels near histogram minima. Since contrast is expanded for most of the image pixels, the transformation usually improves the detectability of many image features.

The histogram equalization mapping may be defined in terms of the cumulative histogram for the image, i.e.,

\[
g(q) \ dq = h(p) \ dp \tag{2.40}
\]

\[
g(q_2) = \frac{N^2}{M} \tag{2.41}
\]

where \( N^2 \) is the number of pixels in the image and \( M \) is the number of gray levels. Combing the above two equations and integrating,
but this is the equation for the normalized histogram.

As an example of the use of pyramidal structures in processing, consider the use of such structures in edge detection. This application, after [37], uses two pyramids, one to store the image and another to store the image edges. The idea of the algorithm is that a neighborhood in the low resolution image where the gray level values are the same is taken to imply that in fact there is no gray-level change (edge) in the neighborhood. Of course, the low resolution levels in the pyramid tend to blur the image and thus attenuate the gray-level changes that denote edges. Thus the starting level in the pyramid must be picked judiciously to ensure that the important edges are detected.

Similar kinds of edge detection strategies based on pyramids have been pursued by L Kevine [38]. The later effort is a little different in that the processing with the pyramid is bidirectional; information from edges detected at a high resolution level is projected to low resolution levels of the pyramid.

2.4 Template Matching

Template matching is recommended when the vision problem involves standard object measurement and recognition. This approach is good for an industrial setting, where the product
produced is of standard size and shape.

In template matching, the intensity profiles of the entity we seek to match the image forms a template. The segmentation process consists of searching for regions in the image where the image grey-levels regionally coincide. A very simplistic interpretation of the process, where the viewer may visually slide the template over the image until the template and image intensity levels coincide.

The template is defined in a template specific coordinate system, as \( g(x_1, x_2) \), and the image function, again in terms of a chosen coordinate system, is denoted \( f(x_1, x_2) \). Assume that \( g(x_1, x_2) \) is nonzero only over a region, denoted \( R \), in the template system.

The determination of a good measure of match, or metric, is important. The value of the metric should be large when the template and image region coincide in intensity levels over \( R \), and small otherwise. Mathematically we can write,

1. \[ m_1 = \Sigma \Sigma |f - g| \]  \hspace{1cm} (2.43)
2. \[ m_2 = \Sigma \Sigma (f-g)^2 \] \hspace{1cm} (2.44)

\( m_1 \) and \( m_2 \) will be small when \( f \) and \( g \) are identical over \( R \) and large when they are significantly different. Whereas metric from equation (2.43) is easy to compute, a closer examination of the equation (2.44) leads us to some interesting results,

\[ m_2 = \Sigma \Sigma f^2 - 2 \Sigma \Sigma f g + \Sigma \Sigma g^2 \] \hspace{1cm} (2.45)
In equation (2.45) the intensities of \( f \) and \( g \) individually contribute to \( m_2 \), through the \( \Sigma \Sigma f^2 \) and \( \Sigma \Sigma g^2 \) terms. For a fixed template, \( \Sigma \Sigma g^2 \) is a constant throughout the matching process. Conversely, the term \( \Sigma \Sigma f^2 \) varies throughout the matching process.

The \( \Sigma \Sigma f g \) term is of fundamental interest. Since, the coefficient of this term is negative, when this term is large \( m_2 \) will be small. Therefore, where \( m_2 \) provides a good measure of mismatch, the \( \Sigma \Sigma f g \) term provides a reasonable measure of match.

Assume, without loss of generality, that the coordinate system for the template and the image representations are coincident. Since the template is only nonzero in region \( \mathcal{R} \), the unnormalized correlation may be computed via

\[
C_{\text{un}}(a,b) = \Sigma \Sigma g(x_1 + a, x_2 + b) f(x_1, x_2),
\]

over all possible locations \((a, b)\) of \( g \), relative to \( f \).

The unnormalized correlation has a strong relationship with the 2-D convolution function. Due to the relation between convolution in the spatial domain and multiplication in the Fourier domain, the correlation of equation (2.46) is implemented in the discrete Fourier domain. Finally unnormalized correlation is viewed as a linear filtering operation, wherein the filter impulse response is a 180 degree rotated version of the template. This type of
filtering is referred to as matched filtering.

2.4.1 Normalized correlation

To begin, note that zero intensity values in either $f$ or $g$ contribute nothing to the unnormalized correlation measure of match in equation (2.46). This is one indication of a more serious shortcoming of the unnormalized approach. For example, in regions where both $f$ and $g$ have significant numbers of corresponding pixels with zero intensity, $C_{un}$ does not reflect this match or correspondence. This effect would be most pronounced in applying $C_{un}$ to binary images, where only 1's would contribute to the matching metric. In addition note that the local energy of the image function, $f$, over $R'$ is reflected in $C_{un}$.

For this reason a normalized correlation metric $C_{n}(a,b)$ is used

$$C_{n}(a,b) = \frac{1}{E} \Sigma R' g(x_1 + a, x_2 + b) f(x_1, x_2) \quad (2.47)$$

where

$$E = \Sigma \Sigma R' f(x_1, x_2). \quad (2.48)$$

2.4.2 Matching with Generalized Geometric Distortions

When the relative geometric orientations of the region to be matched and the template is allowed to be other than translation, the matching process becomes more complex. For example, one significant problem with regional segmentation approaches based on grey-level correlation is varying geometric orientation. We restrict our attention to 2-D
orientations. Various solutions to this problem may be pursued. For example regional features that are invariant to orientation could be used for matching. An example is the use of moments. An alternative is to store all possible rotated and scaled versions of the template and use or match metric with each of these. This approach is impractical and thus in medical images where the body of different patients have varying shapes and sizes this almost becomes an impossibility. Geometric perturbations of the template, such as rotation and dilation have significant effects on metrics that are chosen simply to indicate position.

2.4.3 Thresholding

The most common way to extract objects from a picture is to threshold the picture. If the given picture \( f \) has Gray level range \([z_1, z_K]\), and \( t \) is any number between \( z_1 \) and \( z_K \), the result of thresholding \( f \) at \( t \) is the two-valued picture \( f_t \) defined by

\[
f_t(x,y) = \begin{cases} 
1 & \text{if } f(x,y) \geq t \\
0 & \text{if } f(x,y) < t 
\end{cases}
\]  

(2.49)

Thresholding techniques are primarily designed to extract objects that have characteristics gray level ranges or texture; in other words, they yield objects that have some type of uniformity. It is not applicable to the real images obtained with variable noise content, because it gave spurious features(objects), when attempted on some of the CT
scans. Thus it was not utilized in this dissertation research because it was not a practical approach to extract objects from images.
CHAPTER III

PATTERN RECOGNITION AND IMAGE ENHANCEMENT

3.1 Introduction

Statistical pattern recognition has seen a considerable research effort for the last two decades. For example Duda Hart [39], describe supervised learning for speech and image processing tasks. The importance of this technique is fully exploited by neural networks, as described by Schalkoff [40] in the implementation of statistical pattern recognition ideas.

Image enhancement using the Hough transform is a relatively new technique and has been applied to simulated images by Dana Ballard [41], and by Kimme et al [42]. Improvements by Dana Ballard, have made the Hough transform one of the most elegant techniques in use today for shape detection and image enhancement.

A neural network is modeled on biological process. Artificial neural net models have been studied for many years by researchers like Grossberg [43] and Feldman [44], in the hope of achieving human-like performances in the fields of speech and image recognition. Neural networks are a completely different way of looking at computer systems.
These models are composed of many nonlinear computational elements operating in parallel and arranged in patterns similar to biological neural nets. Computational elements are connected via weights that are adapted during use to improve performance. An introduction to the field of neural nets is given by reviewing some important models that can be used for pattern classification. Some of the well known algorithms listed by Lippman [45], Single layer perceptron, Multi-layer perceptron, ART will be discussed in section 3.4.

3.2 Pattern Recognition

One of the most commonly used abstract models for pattern recognition is the classification model. This model according to Hartigan [46], consists of essentially three parts: a transducer, a feature extractor, and a classifier. The transducer senses the input and converts it into a form suitable for machine processing. The feature extractor also called attribute detector extracts relevant information from the input data. The classifier uses this information to assign the input data in to one of the given number of categories.

There is a thin line of separation between the feature extractor and classifier. From a theoretical viewpoint, the line between these topics is arbitrary. Generally speaking, the problem of feature extraction is much more problem
dependent than the problem of classification. The problem of classification is basically one of partitioning the feature space into regions, one region for each category. Ideally, we would like to arrange this partitioning so that none of the decisions is ever wrong. At best, we would like to minimize the probability of error.

Classification of the pattern present in a given data, can be accomplished by the process of clustering. In this process the given data is split up into a number of clusters, so that there is atleast one data in each cluster. Bayesian classifier is the method that will be used to tackle the problem of pattern recognition in this dissertation. The Bayesian classifier has the following assumptions. The number of classes is known apriori. The apriori probabilities $P(W_j)$ for each class are known, $j=1, \ldots, c$. The forms for the class-conditional probability densities $p(x_{W_j}, q_j)$ are known, $j=1, \ldots, c$, but the parameter vector $q = (q_1, \ldots, q_c)$ is not known. Part of our knowledge about $q$ is contained in a known apriori density $p(q)$. The rest of our knowledge about $q$ is contained in a set $H$ of $n$ samples $x_1, \ldots, x_n$ drawn independently from the mixture density. We can make assumptions that the apriori probabilities are known because of the fact that, the CT (Computed Tomography) numbers for a particular image of interest, is known from
previous knowledge. In particular, if we are imaging a medical object, then the CT numbers for different sections of the organ of interest has distinct CT numbers.

Clustering procedures yield a data description in terms of clusters or groups of data points that possess strong internal similarities. There are various procedures from measuring the sum of the squared distances from the cluster centers to intuitively appealing procedures that have no established properties.

The most obvious measure of similarity or dissimilarity between two samples is the distance between them. If distance is a good measure of dissimilarity, then one would expect the distance between samples in the same cluster to be significantly less than the distance between samples in the different clusters. Suppose for the moment that we say that two samples belong to the same cluster if the Euclidean distance between them is less than some threshold distance $d_0$. It is obvious that the choice of $d_0$ is very important. If $d_0$ is very large, all of the samples will be assigned to one cluster. If $d_0$ is very small, each sample will form an isolated cluster. To obtain "natural" clusters, $d_0$ will have to be greater than typical within cluster distance and less than typical between-cluster distances. Clusters defined by Euclidean distance will be invariant to translations or
rotations. However, they will not be invariant to linear transformations in general, or to other transformations that distort the distance relationship. Scaling of the data can distort this type of clustering. One of the way to achieve invariant to scaling is to normalize the data prior to clustering.

3.2.1 Hierarchical Clustering

Let us consider a sequence of partitions of the n samples into c clusters. The first of these is a partition into n clusters, each cluster containing exactly one sample. The next partition into n-1 clusters, the next a partition into n-2, and so on until the nth, in which all the samples form one cluster. We shall say that we are at level k in the sequence when \( c = n-k+1 \). Thus, level one corresponds to n clusters and level n to one. Given any two samples \( x \) and \( x' \), at some level they will be grouped together in the same cluster. If the sequence has the property that whenever two samples are in the same cluster at level k they remain together at all higher levels, then the sequence is said to be hierarchical clustering. The classical examples of this kind of clustering appear in biological taxonomy, where individuals are grouped into species, species into genera, genera into families, families into order, and so on.

Because of their simplicity, hierarchical clustering
procedures are among the best-known methods. The procedures themselves can be divided into two distinct classes, agglomerative and divisive. Agglomerative (bottom-up) procedures start with $c$ singleton clusters and form the sequence by successively merging clusters. Divisive (top-down) procedures start with all of the samples in one cluster and form the sequence by successively splitting clusters. The computation needed to go from one level to another is usually simpler for the agglomerative procedures.

The major steps in agglomerative clustering are contained in the following procedure:

Let $c = n$ and $H = \{x_i\}, i = 1, \ldots, n$. If $c < c$, stop.
Find the nearest pair of distinct clusters, say $H_i$ and $H_j$. Merge $H_i$ and $H_j$, delete $H_j$ and decrement $c$ by one. Go back and continue till all the pixels are classified.

As described this procedure terminates when the specified number of clusters has been obtained. The following distance measures, will be used:

All of these measures have a minimum-variance flavor, and they usually yield the same results if the clusters are compact and well separated. However if the clusters are close to one another, quite different results will be obtained.
3.2.2 The Nearest-Neighbor Algorithm

Consider first the behavior when $d_{\text{min}}$ is used. Suppose that we think of the data points as being nodes of a graph with edges forming a path between nodes in the same subset $H_i$. When $d_{\text{min}}$ is used to measure the distance between subsets, the nearest neighbors determine the nearest subsets. The merging of $H_i$ and $H_j$ corresponds to adding an edge between the nearest pair of nodes in $H_i$ and $H_j$. Since edges linking clusters always go between distinct clusters, the resulting graph never has any closed circuits; in the terminology of graph theory, this procedure generates a tree. If it is allowed to continue until all of the subsets are linked, the result is a spanning tree, a tree with a path from any node to any other node. Moreover, it can be shown that sum of the edge lengths of the resulting tree will not exceed the sum of the edges lengths for any other spanning tree for that set of samples. Thus with the use of $d_{\text{min}}$ as the distance measure, the agglomerative clustering procedure becomes an algorithm for generating a minimal spanning tree.

The algorithm is very sensitive to noise or slight changes in position of the data points. This effect is called chaining effect, wherein elongated clusters are formed. Thus is good for elongated clusters.
3.2.3 The Furthest-Neighbor Algorithm

When $d_{\text{max}}$ is used to measure the distance between subsets, the growth of elongated clusters is discouraged. Application of the procedure can be thought of as producing a graph in which edges connect all of the nodes in a cluster. In the terminology of graph theory, every cluster constitutes a subgraph. The distance between two clusters is determined by the most distance nodes in the two clusters. When the nearest clusters are merged, the graph is changed by adding edges between every pair of nodes in the two clusters. If we define the diameter of a cluster as the largest distance between points in the cluster, then the distance between two clusters is merely the diameter of their union. If we define the diameter of a partition as the largest diameter for clusters in the partition, then each iteration increases the diameter of the partition as little as possible. This is advantageous when the true clusters are compact and roughly equal in size. However, when this is not the case the resulting grouping can be quite meaningless.

The maximum and minimum measures represent two extremes in measuring the distance between clusters. Thus the use of averaging is an obvious way to ameliorate these problems, $d_{\text{avg}}$ and $d_{\text{mean}}$ are natural compromises. Computationally $d_{\text{mean}}$ is the simplest of all of these measures, since the other
require computing all \( n_i n_j \) pairs of distances \( x-x' \).

In this research, the theory of pattern recognition was developed, to facilitate its implementation with neural nets. The chain codes discussed in section 2.3.5 incorporates the clustering principle, whereas the backpropagation network to be described in section 3.4.5, incorporates, the metrics, discussed above.

3.3 Image Enhancement

Image enhancement is a procedure that enhances certain parts of the image, in the region of interest. The enhancement can be achieved by changing the grey levels of the region of interest. The region of interest can be organs and tumors in medical CT scans, to object boundaries in industrial scans. In this research Hough transform was used to find the regions of interest, by suitable grouping of pixels in the image by grey level values. An image was searched for the outline of liver(say), and if there was a close enough match then the boundary pixels were enhanced.

Hough method due to Hough [8], does curve and edge detection if the general shape of the internal organ imaged is known. Then the shape of the organ is broken into segments and each segment is identified by a parametric curve or a straight line. In the Hough method, the basic strategy is to compute the possible loci of reference points in
parameter space from edge point data in image space and increment the parameter points in an accumulator array.

The Hough algorithm as described in [3] is given below:

Step 0. Make a table for the shape to be located. This should have the angle measured from figure boundary to reference point and set of radii where \( r = (r, \alpha) \).

Step 1. Form an accumulator array of possible reference points \( A(x_{cmin}:x_{cmax}, y_{cmin}:y_{cmax}) \) initialized to zero.

Step 2. For each edge point do the following:
   Step 2.1 Compute \( \phi(x) \).
   Step 2.2a Calculate the possible centers; that is, for each table entry for \( \phi \), compute
   \[
   x_c := x + r(\phi) \cos(\alpha(\phi))
   \]
   \[
   y_c := y + r(\phi) \sin(\alpha(\phi))
   \]
   Step 2.2b Increment the accumulator array
   \[
   A(x_c, y_c) := A(x_c, y_c) + 1
   \]

Step 3. Possible locations for the shape are given by maxima in array \( A \).

The next step is to take this array \( A \) and then find two different row numbers for the same column and two different columns for the same row and find the distance through this measurement. Also the perimeter of the object can be found by traversing around the object using the array values as
indices.

This algorithm can be made invariant to scale and rotation by using them up as extra parameters in the array. The algorithm has to face many challenges in order to be successful, it has to find the boundary or the edge in the presence of noise. If the organ has more than one edge, than it will be difficult to find the shape of the organ, let alone its size. Non-uniformity and bad contrast also can give spurious edges and apriori knowledge is required to determine correctly the shape of the organ. This can be in the form of an expert or an expert system. The system can compare the image of the organ to the standard image of the organ and correct for the shape and contrast.

A detailed implementation of the Hough transform, was performed, that is, separate Hough transforms were developed for the recognition of straight lines, circles and ellipses. These will be discussed in chapter 4. There are other adhoc methods that can be used, such as template matching, to perform shape recognition, but they are dependent on size and thus scale of the image.

3.4 Neural Network Theory

A serial system is essentially sequential and everything happens in a deterministic sequence of operations. In contrast, a neural network is neither sequential nor
necessarily deterministic. A neural network is a computing system made of a number of simple, highly interconnected processing elements, which process information by its dynamic state response to external inputs.

A neural network is a computational structure modeled on biological processes. Here neurons become processing elements, the axons and dendrites become wires, and the synapses become variable resistors carrying weighted inputs that represent data or the sum of weights of still other processing elements. The neural network responds in parallel to the inputs presented to it. A neural network is made up of two primary elements: Processing elements and Interconnections. The processing elements are nonlinear and may be slow as compared to modern digital circuitry.

Knowledge within a neural network is not stored in a particular location but in away the processing elements are connected and in the weight of each input to the processing elements. Neural net models, are specified by the net topology, node characteristics and learning rules. Learning implies that the processing elements somehow changes its input/output behavior in response to the environment. There are many algorithms used to train neural networks and they fall into three basic categories: 1) Unsupervised training 2) Supervised training 3) Self-supervised training. Neural
nets provide a greater degree of robustness or fault tolerance because there are many more processing nodes each with primarily local connections. Adaptation or learning is the major focus of neural net research.

Inputs are summed across the resistor which is connected to an operational amplifier on which has been set a threshold so that when the sum of these inputs reach a preset threshold the processing element will fire. When the sum of inputs is below threshold, the device will have a -1 output and will not fire.

A neural network is made up of two primary elements:

1) Processing elements: Processing elements, the neural network equivalent of neurons, are generally simple devices that receive a number of input signals and, based on those inputs, either generate a single output signal or do not. The output signal of an individual processing element is sent to many other processing elements or back to itself as input signals via the interconnections between processing elements.

2) Interconnections: The structure of the neural network is defined by the interconnection structure between the processing elements, the rules that determine the firing of the processing element (the transfer function) and the rules that govern the changes of weights of individual interconnections to a processing element's input (training
Processing elements can interact in many ways by virtue of the manner in which they are interconnected:

a) Feed forward only, feedback loop

b) fully connected to all other processing elements; sparsely connected, linked only to a few others.

The nature and number of these feedback loops and connections depend on the architecture used to construct the neural network. The design of a neural network's feedback has implications for the nature of its adaptivity/trainability, while the design of a network's interconnections has implications for its parallelism.

The programmer specifies interconnections, transfer functions and the training laws of the network, then applies appropriate inputs to the network and lets it react. Neural networks "react", "self organize", "learn", and "forget". Neural networks are good at solving the kinds of problems people can solve easily.

3.4.1 Massive Parallelism

Interest in neural networks is prompted by two facts: a) the nervous system function of even a "lesser" animal can easily solve problems that are very difficult for conventional computers, including the best available, b) the
ability to model biological nervous system function using man-made machines increases understanding of that biological function.

According to Marr [47], the human cerebral cortex is comprised of approximately 100 billion \((10^{11})\) neurons with each having roughly 1000 dendrites that form 100000 billion \((10^{14})\) synapses. If the system operates at 100 Hz, it functions at some 10000 billion \((10^{16})\) interconnections per second. This capability is clearly beyond anything which can be reconstructed or modeled; but it perhaps possible to understand how the brain performs information processing. Consider a two layer network each consisting of 64 x 64 (4096) processing elements laid out in a grid. If these two layers are fully interconnected then this simple structure sorts through no less than 16 million interconnects.

3.4.2 Design of Neural Network Computers

Some of the design principles which provide the foundation for neural network research are:

1) Artificial neural networks are inspired by biological neural networks.

2) Neural networks employ distributed, parallel processing to perform computation - both memory and processing are global rather than local.

3) Computation by neural networks emerges spontaneously from
fundamental physical principles.
4) Neural networks are self-organizing systems.
5) Knowledge is stored by the strengths of the interconnections between neurons.
6) Neural networks compute significant results in a small number of steps.
7) Learning is a fundamental, essential aspect of neural networks.

3.4.3 Learning

Learning implies that the processing elements somehow changes its input/output behavior in response to the environment. Nearly every neural network has a built-in learning capability. There are many techniques (algorithms) use to train neural networks. They fall into three basic categories:

Supervised training requires the presence of an external teacher and labeling of the data used to train the network. The teacher knows the correct response wanted from the network and inputs an error signal when the network produces an incorrect response. This is sometimes called 'reinforcement learning' when the teacher only indicates whether a response was correct or incorrect but does not provide detailed error information. The error signal teaches the network the correct response and after a succession of
learning trials the network consistently produces the correct response.

Unsupervised Training is a means of training adaptive neural networks which requires unlabeled training data and no external teacher. Data is presented to the network and internal categories or clusters are formed which compress the amount of input data that must be processed at higher levels without losing important information. This clustering is sometimes called 'vector quantization'.

Self-supervised training is used by certain kinds of neural networks. The networks monitor performance internally, requiring no external teacher; an error signal is generated by the system and fed back to itself, and the correct response is produced after a number of iterations. Self-supervision is used by automata which require internal error feedback to perform some specific task.

Some of the key concepts and terms are: 1) A typical neural network contains many more interconnections than processing elements 2) Each interconnect requires one multiply/accumulate operation for summing 3) While digital computers are assessed in terms of memory (words) and speed (instructions-per-second), the neural network vernacular defines storage as the value of the input weights and measures it in terms of interconnects; neural network speed
is described in terms of interconnects-per-second within a layer or between layers.

3.4.4 Tasks

Neural networks can perform a variety of tasks, some of which are Pattern Classification, Clustering, Associative Memory, Sensory Data Processing (Vision, Speech), Computational Problems, and Non-linear Mapping.

3.4.5 Delta Rule

The most commonly used learning algorithm is the Delta rule or Least Mean Squared (LMS) training law. The network itself is called ADALINE, derived from the term ADAptive LINear element. This rule specifies how to change weights depending on whether or not a classification is correct. There is a way of telling the processing element what the correct response should be for a given input. This is done by adding another input connection called the mentor, which has a constant weight fixed at +1. The value of the mentor input $I_0$, is the value of the desired output. First the processing element has to be modified so that it can monitor its own output. Hence it is able to compare its output $y$, to the desired output signal $I_0$ and compute the error value $E$, for an input pattern.
The change in weights can be calculated using the Delta rule as follows:

$$W_{\text{new}} - W_{\text{old}} = \frac{\beta E X}{||X||^2}$$

(3.2)
aggregate statistical LMS error is minimized in the network. The error in the weights of the processing element is based on some ideal value for the weights. The current error indicates how far it is from this ideal value, for the weights for this input. The weight vector is then adjusted by computing a Delta vector that is parallel to the input vector and has a magnitude as described in the previous equation. The next question is how to determine the value of the constant $\beta$. First, $\beta$ is assumed positive so that the direction of the Delta vector is in the same direction as the ideal vector. Mathematically $\beta$ must be less than 1 or the network cannot be stabilized. $\beta$ is a measure of the speed of convergence of the weight vector to the ideal vector.

3.4.6 Supervised Classifiers

Supervised classifiers as the name indicates are trained with supervision. That is for a known input data a known output is given and the network learns the transfer function, that connects the input and the output. Single layer and multi layer perceptrons are examples of supervised learning and are explained in the next two sections.

3.4.6.1 Single Layer Perceptrons

Single-layer perceptrons were first introduced by Rosenblatt [48]. They are used in pattern classification
problems and trained with supervision. The single layer perceptron can be used with both continuous valued and binary inputs. This simple net generated lot of interest when initially developed because of its ability to learn to recognize simple patterns. A perceptron (shown below) decides whether an input belongs to one of two classes (A or B).

![Perceptron Diagram](image)

\[
y(t) = f_h \left( \sum_{i=1}^{N} w_i x_i - \theta \right) \tag{3.3}
\]

\[
y(t) = \begin{cases} +1 & \text{Class A} \\ -1 & \text{Class B} \end{cases} \tag{3.4}
\]

The single node computes a weighted sum of the input elements, subtracts a threshold (q) and passes the result through a binary-valued nonlinearity \( f_h \) such that the output \( y(t) \) is either +1 or -1. Each of the two possible outputs corresponds to a different classification response. The perceptron forms two decision regions separated by a hyperplane.

Connection weights and the thresholds in a perceptron can be fixed or adapted using a number of different
algorithms. LMS algorithm is primarily used when inputs take on continuous instead of binary values. The original perceptron convergence procedure for adjusting weights was developed by Rosenblatt [48]. First connection weights and the threshold value are initialized to small random non-zero values. Then a new input with N continuous valued elements is applied to the input and the output $y(t)$ is computed. The output $y(t)$ is compared to the desired output $d(t)$ and the weights are adjusted if there was a wrong decision.

$$W_i(t+1) = W_i(t) + \eta [d(t) - y(t)] x_i(t)$$

where $\eta$ is a positive gain fraction less than 1.

3.4.6.2 Multi-layer Perceptron & Backpropagation

A more complex structure than the Single-layer perceptron is required when classes cannot be separated by a hyperplane. Multi-layer perceptrons are feedforward networks with one or more layers of nodes between the input and output nodes. These additional layers contain hidden nodes that are not directly connected to both the input and output nodes. The capabilities of multi-layer perceptrons stem from the nonlinearities used within the nodes. If the nodes were linear elements, than a single-layer net with appropriately chosen weights could exactly duplicate those calculations performed by any multi-layer net. This network has gained popularity with the development of a new training algorithm.
called Backpropagation. This algorithm is a generalization of the LMS algorithm that uses a gradient search technique to minimize the cost function equal to the mean square difference between the desired and the actual net outputs.

3.4.7 Backpropagation Network

A learning algorithm for updating weights in a multi-layer, feedforward, mapping neural networks that minimizes mean squared mapping error. Backpropagation networks are always hierarchical; that is, they always consist of at least three layers of neurons (processing element). Hence there is an input layer, a middle (hidden) layer and an output layer. The network is constructed such that each layer is fully connected to the next layer. Hence, every neuron in the input layer will send its output to every neuron in the middle layer and every neuron in the middle layer will send its output to every neuron in the output layer. There are no connections within each layer.

Figure 10 Operation of a backpropagation network
3.4.7.1 Forward Activation Flow

The input layer receives a pattern and passes it along to each neurode in the middle layer. Each of these neurodes computes an activation \( \hat{v} \); First the summed input is determined by multiplying each input signal times the (random) weight on that interconnection.

\[
I = f \left( \sum_{i=1}^{N} w_i x_i \right)
\]

\[
f(x) = \frac{1}{1 + e^{-x - T}} \quad \text{(Sigmoidal)}
\]

\(0 < I < 1\)

The function \( f(x) \) is called the activation function of the neurode. For a backpropagation network, this function should be sigmoidal (continuous). The activation function is a function of a simple threshold \( T \) which is usually set to 0. The outputs are constrained to be in the range of 0 to 1. These outputs from the middle-layer are the inputs to the outputs layer. Each neurode here receives these signal from all middle-layers and as before an activation is computed for each output-layer neurode.

3.4.7.2 Backward Error Flow

The output of the network is compared to the desired output. If the pattern is wrong, the Delta rule is used to compute the weight changes. Logically, the output neurodes
may have generated the wrong answer, but it might also be due to the middle-layer neurodes. To assign this error, the error is back propagated for each output-layer neurode to the middle layer using the same interconnections and weights as the middle layer used to transmit outputs to the output layer. An error for each neurode in the middle layer is computed based on their portion of the blame for the output layer's error. Stronger the interconnection, larger the error. Using this error in the Delta rule, the weights are adjusted. The following formula is used:

\[ e_i = f'(I) \cdot [w_{ij} \cdot E_j] \]  

(3.8)

The function \( e_i \) is error in the ith middle-layer neurode and the sum is taken over the jth output-layer neurode. The derivative of the sigmoidal activation function is bell shaped. The derivative is used for two reasons; first, it makes the network stable as it ensures that, as outputs approach 0 and 1, only very small changes can occur. Second, it helps compensate for excessive blame attached to the middle-layer neurode.

3.4.8 Generalized Delta Rule

Backpropagation is not guaranteed to find the correct answer. The solution may be stuck in local minimum. Hence a momentum term is added in order to carry the solution 'over' instead of being stuck at the bottom. The new Delta rule
looks like:

\[ W_{\text{new}} - W_{\text{old}} = \frac{\beta \mathbf{X}}{|\mathbf{X}|^2} + \alpha (W_{\text{new}} - W_{\text{old}})_{\text{prev}} \]  \hspace{1cm} (3.9)

The momentum term is a constant \( \alpha \), multiplied by the change in weight vector of this neuronode from the previous presentation of this input pattern. Hence if the last weight change was in a particular direction, the momentum term changes in more or less the same direction.

The primary drawback of backpropagation is that it requires thousands of iterations to learn the input data and each iteration requires two passes through the network — forward and backward. Also the parameter \( b \) must be set to an approximately small value to keep the network from oscillating, but such small values can cause the network more iterations to learn. Backpropagation is one of the most important neural networks available because it is simple to implement and works extremely well for a wide variety of applications. Thus the primary uses of backpropagation are applications in which learning takes place off-line and in non-real-time environment.

3.4.9 Unsupervised Classifiers

Unsupervised classifiers as opposed to supervised classifiers, do not need a teacher or a supervisor. It can recognize the patterns that are present in the input and
classify the output with built in logic, as in Adaptive Resonance Theory (ART) as described in the following section. The ART is based on statistical (unsupervised) pattern recognition technique as described earlier in section 3.2.

3.4.9.1 ART (Adaptive Resonance Theory)

There are two distinct network models are based on ART; these are called ART-1 and ART-2 networks, and mainly due to the pioneering work of Grossberg and Carpenter [49]. ART-1 networks can process only binary input patterns and ART-2 can process gray-scale (but not analog) input data. ART networks form a new cluster (adjust weights) whenever an input pattern is sufficiently different from previously stored patterns and are trained without supervision.

The underlying traditional algorithm used in these networks is called the leader clustering program. The leader algorithm selects the first input as the exemplar for the first cluster. The next input is compared to the first cluster exemplar. It is clustered with the first if the distance to the first is less than the threshold. Otherwise it is the exemplar for a new cluster. The factor which determines the 'difference' is a global internal parameter called the vigilance parameter which must be adjusted externally to provide the desired sensitivity to differences in input patterns. This process is repeated for all
following inputs. The number of clusters thus grows with time and depends on both the threshold and distance metric used to compare inputs to clusters exemplars. Each new exemplar requires one node and 2N connections to compute the matching scores.

In the ART net feed-back connections are provided from the output nodes to the input nodes. The net is initialized by setting all exemplars to zero. In addition, the vigilance threshold $h$ is set between 0 and 1. A value near one requires a closer match. New inputs are presented sequentially at the bottom of the net. The input is compared to all stored exemplars in parallel to produce matching scores using feed-forward connections.

$$\mu_j = \sum_{i=1}^{N} b_{ij}(t) \cdot x_i$$  \hspace{1cm} \text{weight} \hspace{1cm} b_{ij}(0) = \frac{1}{1+N} \quad \text{(bottom to top)} \quad (3.10)

The exemplar with the highest score is selected using lateral inhibition. It is then compared to the input by computing the ratio of the dot product of the input and the best matching exemplar (number of 1 bits in common) divided by the number of 1 bits in the input. If this ratio is greater than the vigilance threshold, then the input is considered to be similar to the best matching exemplar.
This net is completely described using nonlinear differential equations, includes extensive feedback, and has been showed as stable.

Neural networks offer important new computational structures. Their real strength is derived from their ability to self-adapt and learn. Although there have been no practical application of neural nets, experiments and results have demonstrated the potential of the newer learning algorithms. The greatest potential of neural nets remains in the high-speed processing that could be provided through massively parallel VLSI implementations. If neural networks realize their full potential, they can be used for machine vision, speech recognition, robotics and other applications - without the need for application-specific software.

Current research is aimed at analyzing learning and self-organizing algorithms, at developing design principles and techniques to solve dynamic range and sensitivity problems which become important for large analog systems, at building complete systems for image and speech and recognition and obtaining experience and at determining which current algorithms can be implemented successfully.
CHAPTER IV

AN INTEGRATED IMAGING SYSTEM DESIGN

4.1 Introduction

The goal of this high-level image analysis is to develop and use image and representations for interpretation of image formation. This involves multilevel image analysis, and general procedures for image-based knowledge manipulation, including rule based manipulation of image data. The top level employs a great amount of non-image-related knowledge underlying the scene representation, for example, knowledge about different organ locations, their locations with influencing imaged entities and their environment. This will be the basis for our design that is explained in section 4.3.

Underlying image understanding research efforts is an implicit assumption that human-like image interpretation is a form of computation that may be identified and consequently automated. According to Schalkoff [40] the two terms identification and automation often give rise to two major impediments to success in image understanding, namely:

1. It is not possible to identify or, more specifically, quantify human capability in a form that lends itself to direct computer implementation. In other words, a "knowledge
calculus" does not presently exist. Another way to view this is to claim that what is lacking is a unified, versatile, high-level model from which algorithmic emulations may be developed.

2. Even with current computing resources, implementing even simple image understanding systems yields problems of tremendous complexity. Thus, languages and architectures that facilitate these operations are of fundamental interest.

Equivalently, it is assumed that there is another, higher level model to guide the extraction of higher level image concepts from these low-level process primitives. For example, the extraction of simple edges via a low-level process may be used to provide input (primitives) to be manipulated or combined by a grammar-based model and corresponding algorithm. In the grammar-based model, basic primitives may consist of line segments. Using these primitives, higher level scene entities residing in the image are developed.

Also there can be situations where it is not possible to clearly classify different objects present in the image, because of bad contrast at the edges. One of the technique applicable in this situation is to use the apriori knowledge of the object and the calibration of the imaging system and find the edge of the object, also CT numbers can be very
useful, if the imaging system is normalized to standard CT numbers. This is especially useful in the case of medical imaging. There could be some tolerance for the CT numbers in terms of one or two standard deviation from the average values.

4.2 Matching Invariant to Position, Scale and Orientation

The goal of any good image recognition software is to have a matching technique which is invariant to translation, rotation and scaling (T, R and S respectively). The approach is in three steps and begins with translation effects. It is easy to visualize that the magnitude of the Fourier transform $|F(u,v)|$ is invariant to translation. Computing the Fourier transform of the template and the image and ignoring the phase yields a matching approach insensitive to position. But a rotation of $f(x_1,x_2)$ rotates $|F(u,v)|$ by the same amount and a scale change in $f(x_1,x_2)$ by a scales $|F(u,v)|$ by $1/a$.

Separation of rotational and scale effects is achieved by transforming $|F(u,v)|$ from rectangular to polar coordinates. Any rotation of $f(x_1,x_2)$ manifests itself as a shift in $q$ in $F(r,\theta)$. A scale change in $f(x_1,x_2)$ of affects only the $r$ coordinate of $|F(r,\theta)|$. Thus a two-dimensional scaling of the image function is reduced to a scaling of only one coordinate in the polar representation of
One scale invariant transformation is the Mellin transform, the 2-D version of which we will consider next. The Mellin transform of image function \( f(x_1,x_2) \), denoted \( M(u,v) \) is given by

\[
M(u,v) = \iint f(x_1,x_2)x_1^{-u-1}x_2^{-v-1}\,dx_1\,dx_2 \tag{4.1}
\]

To see the scale invariance property, note the Mellin transform of \( f_2 = f(ax_1,ax_2) \), denoted by \( M_2(u,v) \) is

\[
M_2(u,v) = a^{-u-v}M(u,v) \tag{4.2}
\]

Mellin transform can be implemented by first computing an appropriate prescaling of the image function. Specifically a logarithmic scaling of the coordinates of the input function followed by a Fourier transform of its result yields the Mellin transform.

Thus, combining the Fourier and Mellin transforms with a rectangular to polar conversion yields a computationally attractive RST-invariant matching scheme. The approach makes use of the individual properties of a sequence of operators.

4.3 An Imaging System

In a typical imaging system, firstly there will be an imaging hardware like the DADR system to be described in section 5.2 or a CT machine. Secondly the projections of the object that are obtained are reconstructed utilizing one of
the image reconstruction procedures described in section 2.2 to get a 2-D slice. These slices are then processed by the image processing system with the edges and boundaries being identified using the edge detector described in section 2.4. The edge detection process depends on the type of image that is being processed. If the features(objects) are many then the edge detector is used in a low resolution mode, so that all the object boundaries are picked, then particular features(objects) are addressed by using high resolution mode for the same edge detector. If we are looking for few features in an image we can use high resolution mode and get all the object boundaries. It is to be realized that the edges detected by this edge detector in the highest resolution mode is a super set of the edges detected by the edge detector in any lower resolution modes. The edge detectors described in section 2.4 can perform both these operations. Next the edge image is fed into a contour processing system which forms closed contours and at the same time removes all the spurious edges. This processed image is fed into a segmentation system which does segmentation by grouping all the like valued data into clusters. Subsequently an image enhancement system using the Hough transform enhances the requisite clusters if they confirm to standard geometric patterns. Thus the overall image
recognizing system is comprised of an integrated set of algorithmic modules. A higher level algorithm, in conjunction with selected output from these modules, is used to guide the high level image understanding. Finally a modelling system implemented using the neural network looks for any deviation from the standard or normal model and thus identifies any deviation greater than two or three standard deviations as a major candidate for defect and if this defect is in the data base then it is recognized and displayed.

A hierarchical processing approach parallels the operation of the Human Visual System (HVS) and Mars has an excellent paper [47] describing the three stages in HVS. A brief description of the three stages in the HVS are, the primal sketch, when the eye forms edges of the image which is incident on the retina. In the second stage an intrinsic image (2.5-D sketch) is formed with cues information such as texture and motion. Only in the final stage a surface reconstruction is done to obtain a 3-D image which is recognized by the brain. All the three process take place in few milli seconds and so we are not able to identify the three process as distinct, but as a merged process in the HVS. It is intuitively obvious that in order to simulate the HVS, we have to move in discreet steps, by obtaining an edge image then the 2.5-D image and finally to a 3-D image.
The Imaging system either the DADR system or the CT machine.
The image reconstruction program and the output 2-D slice.

Low Level Processing.
A filtering system for removal of noise
The edge detector
Canny's edge detector or the LOG kernel

Intermediate Level Processing
Closed Contours and Chain Coding
Segmentation into Lower Level primitives

Higher Level Processing
Hough Transform Approach
Model Based Design

Matching and Image Description
Neural Network Pattern Recognition
Object Classification
Defect Recognition if present.

Figure 11 The Image Interpretation System
Thus the processing proceeds from a coarse descriptive level to levels of increasing refinements. The hierarchical approach is applicable to the tasks of segmentation, feature extraction, description, and matching. A benefit of this approach is the resultant computational savings.

The task of image interpretation with these models often involves the manipulation and matching of extracted features and binding of model systems (or named variables) to these extracted features, with the goal of obtaining a (model) consistent structure to achieve "unification" of the model and the observed featured data.

A total image interpretation system the result of this research is shown in Figure 11. It can be seen from this system that it has many major components which have been integrated to form a stand alone image interpretation system. Description of each of the component of this system will be explained in detail in the next few paragraphs and the need for logical order and integration then will be apparent.

We start with the imaging system which will provide images, of the objects which are to be recognized and their defects categorized. There are two types of imaging systems that are mentioned here, though we can have any number of these imaging systems in parallel. The main source of images for our image interpretation system has been from the medical
CT scanner obtained from the internet facility of medical image repository and courtesy of Dr. Vishweshwariah of Jubilee Medical CT Scanning (fourth generation CT machine), Bangalore, India [50], for image processing and recognition for medical CT scans. and the DADR system developed by Scantech [51] for image reconstruction for industrial objects. Each one of these systems create an output image of a medical organ, or an industrial object which are suitable to be processed by the next stage of our imaging system, the low level processing system. Though in the case of the medical CT scans they had to be digitized as analog images were obtained from the CT scanners. The CT scans were digitized by a modern digitizer [57]. The advantage of the DADR system was that the output was digital though the output image had to be reconstructed using the filter back projection image reconstruction procedure described earlier in chapter 2, and then fed into the next stage of processing.

The two main functions of the low level processing system are to perform filtering operation to remove noise and detect edges present in the features (objects). The need for filter has been explained in section 2.3.1. In our system design we need not specify a specific filter, because the filters are noise dependent. Also the edge detecting system in our design has a front end which has a band pass filter.
An edge detection system is necessary because edges are very important in recognizing images and as they are one of the major characteristic features of any image and also boundaries which are another important feature can also be detected using the same edge detection system. Edges convey most of the information needed to identify an object. Location of edges are obtained where the grey-levels in an image change abruptly over a window. The window is dependent on the accuracy of the edges desired. For example, a small window of one(1) pixel width will assure all edges to be visible, though there may be too many!, whereas large window of say ten(10) pixel width will locate big edges or discontinuities and is a subset of the small window edge detector. Many a time edge detection by this simple method of grey level change between adjacent pixels is insufficient to describe the required features in an image, then one looks for the next higher order, that is the slope of the grey-level image. In most cases this is sufficient to classify the pixel as an edge pixel or nonedge pixel.

There were many edge detectors that were tried, but only two; LOG (Laplacian Of the Gaussian) and Cannyesque edge detector successfully passed the criterion set in chapter 2 for a good edge detector. Also they passed the test for locating most of the important edges without introducing too
much of noise for many of the sampled images. By judiciously picking sigma values in equations (2-29) and (2-30), and thus varying the window parameter we were able to get edge detectors of varying orders, from fine (high resolution) edge detector for sigma value of one(1), to the coarse (low resolution), for a sigma value of 16. The edges detected for the test images by these edge detectors are discussed in chapter 5.

The difference of boxes edge detector was very close but it was rejected because the Cannyesque edge detector has all the advantages of the differences of boxes detector, plus it has overcome some of the disadvantages of the differences of boxes detector. The difference of boxes detector with its sharp edges, creates discontinuities and thus gives rise to more spurious edges then real edges in an image.

The intermediate level processing module accepts the output of the edge detecting system and chain codes the binary valued edge data, to closed shape contours, so that the recognition system, can recognize the feature shapes present in the image. The chain code theory and procedure has been explained in chapter 2 and the implementation of the chain code procedure is given in the next paragraph.

In the chain coding procedure the binary image from the edge detector is taken and a starting edge point is randomly
selected. Then the adjacent point is checked to see if it belongs to an edge, if so, then it is selected as the next point in the path, and this procedure continue till a closed contour is formed or if the contour has too few pixels then it is back traced and a new path (if any) is selected. This procedure is done until all closed contours are completed. Care is taken not to go on the same path by keeping a flag. All contours of less than certain predetermined pixel is rejected as noise and the rest of the image is preserved as a closed contour image.

Chain coding procedure works excellently for many of the images, but for a few cases we found that the original image data and the higher level module has to interact, with this module to get the correct interpretation of the object, that is being imaged. This is specially true when there is a lot of noise and the filtering process has also removed some of the useful signal along with the noise, for which alone it was intended.

Segmentation in to the primitives is also done in conjunction with the Hough transform from the high level processing. The Hough procedure will try to locate any contour which approximates a circle, an ellipse or a straight line. These pixels which conform closely to one of the shapes mentioned above are marked and the original grey level
image are checked by clustering techniques for uniform pixel values (within 2 standard deviations) at the suggested location by the Hough transform. If these are found as suggested then they are strongly suggestive of a feature, which conforms to this particular shape.

The higher level processing module, gets its input from three sources; the intermediate level processing module; the low level processing module; and from the original (imaged) object. This module will make the decision about the shape of the feature in the image, using the Hough transform approach, described earlier in chapter 3.

![Figure 12 Locus of Parameters with no directional information](image-url)
The idea of the Hough transform is to transform the image into Hough space where, it is easier to detect features desired, and then transform back to the image space. It exploits the fact that analytic curves if present in an image may encompass many pixels in an image space, but characterized by only a small set of parameters.

Let us consider analytic curves of the form \( F(x,a) = 0 \), where \( x \) is an image point and \( a \) is a parameter vector. Let us further suppose we are detecting circular boundaries in an image. In Cartesian co-ordinates, the equation for a circle is given by

\[
(x-a)^2 + (y-b)^2 = r^2 .
\]  \hspace{1cm} (4.3)

Suppose also that the image is the output from the low level processing unit, so that only the magnitude of the local intensity changes is known. For each edge pixel, we ask the question: if this pixel is to lie on a circle, what is the locus for the parameters of the circle? The answer is a right circular cone, as shown in Figure 12. This was obtained from equation (4.3) by treating \( x \) and \( y \) as fixed and letting \( a, b \) and \( r \) vary. The result about this locus in parameter space (Hough space) is that, if a set of edge pixels in an image are arranged on a circle with parameters \( a_0, b_0, \) and \( r_0 \), the resultant loci of parameters for each such point will pass through the same point \( (a_0, b_0, r_0) \) in
parameter space. Thus many such right circular cones will intersect at a common point.

The generalized features of this Hough algorithm as adapted in this research will be described below. We realized in our situation that if we utilize extra information, then we can reduce the number of parameters, to specify a circle. While finding edges we also found the derivative and thus the directional information. If we utilize this the number of free parameters is reduced to one. Formally, what happens is the equation

$$\frac{df}{dx}(x,a) = 0$$

(4.4)

introduces a term dy/dx which is known since

$$\frac{dy}{dx} = \tan \left[ \phi(x) - \frac{\pi}{2} \right]$$

(4.5)

where \(p(x)\) is the gradient direction.

Thus a modified Hough algorithm for analytic curves in an edge image with directional information is, the following:

1. Initialize Accumulator array \(A(a) = 0\);
2. For each edge pixel \(x\), compute all \(a\) such that \(f(x,a) = 0\) and \(df/dx(x,a) = 0\);
3. Increment \(A(a) += 1\);

After each edge pixel has been considered, local maxima in the array \(A\) correspond to curves of \(f\) in the image. A
similar argument can be made for ellipses, and other analytic curves, which follow the equation $f(x,a) = 0$.

The other part of high level processing is the model based part, that contains information about the features in the image for which we are modelling. This information is known apriori from the expert in the field, in our case Grays Anatomy [58], which describes in detail each human organ, with respect to its location and size in relation to the other prominent organs. Also salient features about an organ, with respect to its size, shape and relative position, are described in detail, which is necessary for the model based reasoning part of the system. Identification of new organs is not possible without employing a model base which has knowledge about the organ and its characteristics.

The next stage is the integration of the different modules, which ensures extraction of useful information from previous modules. This stage also abstracts knowledge from the low level (edges), intermediate level (chain coding) and higher level (model and Hough transform) systems, thus providing the neural network system with meaningful data. Input and corresponding output data are obtained from an expert (experienced radiologist). This integration and data abstraction process will be explained using an example of identification and classification of human brain scans.
The flow chart shown in figure 13 is an example that tabulates the underlying process involved in identifying and classifying human brain scans. It also helps in identifying key areas of where user input is required to assure
proper abstraction. The output vectors of the neural network are three sets of binary valued vectors which serve as identifiers for each of three cases:

1) a normal brain,
2) a haemorrhaged brain and
3) a lacunar infarcted brain.

The first two questions in Figure 13, are answered by using the Hough transform for elliptical routines. The Hough transform, which was described in Chapters 3 and 4, is checked for instances of elliptical shape in the test image. If there are more than two elliptical shapes, then the two largest shapes are used to answer the first question. The question is answered as yes(1) or no(0). The second question is answered by using the location information on the two ellipses and recording all pixel values that are located between these elliptical shapes. Pixel values that are more than a couple of standard deviations from the average pixel value and forming a cluster or a segment help in answering the second query. Isolated pixel values, that is pixel values not forming a segment within the given metric, are assumed to be noise.

The next three queries in Figure 13 require the help of an expert or a radiologist. The Hough transform method and the chain coding method which was described in Chapter 2, are
also used. Chain encoding of an image segment produces a string of numbers as its output, along with a starting and an ending location. These points will be identical for a closed contour.

The general location of the third ventricle within a quadilateral is queried from the expert. If the query is aimed at the user then previous location (default) information is also supplied. The ventricle is bounded by its four corner points. The user specifies only two diagonal points, such as \((x_1, y_1)\) and \((x_2, y_2)\). Then two more points are identified to form a quadilateral, utilizing \((x_1, y_2)\) and \((x_2, y_1)\), and thus satisfying the requirement for four points. If the shape of any segment is readily available using the Hough transform, then it is utilized. Segments are searched in the area bounded by these points by utilizing chain coded segment file to identify any segment present. The chain coded segments identified are compared with the normal chain coded segment for the third ventricle pixel by pixel. If there is a variation of more than 10% between the normal chain code and the chain code in the test image, then the chain coded segment is considered abnormal (not normal). If there is more than one chain code segment available the one that is closest to the normal chain code in terms of percent error is selected. The length of the chain code is also
checked to verify which is closer. Pixel values in the original image may also be considered at this point to resolve any ambiguity. If all these tests are inconclusive, then the answer is stored as "ambiguous" so that other variables are used to resolve this ambiguity. All chain encoded segments which answer the test are verified and their shape and size are recorded. This answers 3 questions in the query, and as before a yes is scored as 1, and a no is scored a 0 (zero).

Similarly a query is made to the user or radiologist about the location of the longitudinal fissure and a similar argument as the one used for the ventricles is employed. Any extra segments that are present after accounting for all normal segments are recorded and the last question in the query is answered.

An abstraction of the input is complete and an explicit output is required for the learning stage of the supervised neural network (backprop). The output for each of the different brain scans observed are recorded and the string "1 1 1" represents a very high probability of a brain scan being normal and the similarly the string "0 0 0", represents a very high probability of a brain scan being not normal. In our situation it has a high probability of being a lacunar infarcted or a haemorrhaged brain scan. Similarly other brain
scans were categorized during the learning stage and the performing step involved utilizing these patterns for an exact match or a close match.

The final part of the system is the Neural Network module which identifies groups (clusters) of features and identifies and labels them either as a healthy organ or an afflicted organ. The important factor for this identification process to occur depends on the decisions made in the previous stages and the expert's decision in the case of diseases (defect). An expert in the field, such as a radiologist [53], will decide which features are important for a particular organ system and the neural net will learn with this feature set and perform on similar features extracted on previous instances of normal and abnormal images.

A neural network, such as the backprop network [56] (described in chapter 3) used in this dissertation, learns only by example. A neural network needs all the possible scenarios or examples for it to perform as desired.

The output from the CT scan, edge image, intermediate level, and higher level processing stage is taken and the important features are extracted with the help of an appropriate expert, medical radiologist or industrial specialist depending on the image that is being processed.
The feature vectors were made of features identified from the edge image and their corresponding grey level values from the original image. Shapes of the closed contours as recognized by the higher level routines were also included for feature extraction. The extracted feature set had seven vectors for brain scan image which meant using a seven vector input for the neural network. There was a hidden layer in this backprop neural network and the output was 3 bits deep for a total of eight (8) possible outcomes. The output was converted into suggestions for the radiologist or the specialist. In our case an output value of seven (7) indicated a strong likelihood of the image being normal and at the other extreme zero (0), indicated a strong likelihood of the image being abnormal or defective. The other six (6) values fell between the two extreme situations of normal or defective image condition. The neural network was trained (learned) separately for each of the defect or disease encountered. A grouping of the features was a possibility only if the feature vectors were independent.

There is yet another possibility to simplify the learning process. Assume we are trying to identify brain scans with two different defects and further assume there are 3 feature vectors that are used to classify the normal brain, then the two sets of seven feature vectors can be combined to
form a single set of eleven (11) feature vectors by removing the extra features which were not providing any new information. So there is only one training process that needs to be done. We have eleven (11) feature vector as input instead of the customary seven (7), but we have eliminated a whole training process and weight matrix that is the culmination of any neural network learning process.

The next step for the neural network is performing which it embarks on by checking for all feature match (mismatch). The output of this trained neural network will be from zero (0) to seven (7). Thus the neural network output will reflect a degree of confidence in its ability to match. Also, if some of the features are stronger than others, then a suitable bias is given to the neural network during its learning phase. A neural network learns by exemplars (examples) so the network is as good as its examples.

4.4 Summary

Although the system has been modularized in design, it was not possible to really separate the modules as the high level module was necessary to gain confidence about the edges extracted and features observed in lower and intermediate processing levels.

It is to be recognized here though that the image interpretation system had to perform image reconstruction
from projections for industrial images and digitize the analog CT scans for the medical images. The interpretation system had to use different filters and edge detectors for different test images because of varying characteristics of the imaging system. Chain encoding was employed to obtain closed contours of edge images which also reduced the noise. In a few instance higher level model based knowledge was employed in the chain coding stage so that useful information was not discarded as noise. Also the neural net had to be trained separately for CT scans of different organs and regions. If the organs and their placement do not differ much, then the same weight matrix of the learning portion of neural network for some organ type (brain) can be used for performing with a different set of organs (abdominal).

In order for the neural network to perform satisfactorily, (that is > 95% success) it has to be trained exhaustively. By exhaustive we mean for 'n' input neural net since it takes only binary values, there are $2^n - 1$ possibilities. The backprop network was slow (about 30 minutes) to learn for an exhaustive combination, but quick (less than a minute) to perform. A complete set of input can be given without going to this exhaustive technique, and that was by covering all reasonable or practical possibilities. This technique took backprop 6 minutes to learn and less than
a minute to perform. The accuracy tends towards the exhaustive learning which was perfect!
CHAPTER V

EVALUATION OF THE IMAGE INTERPRETATION SYSTEM

5.1 Introduction

In medical imaging the process of data acquisition and image processing has improved enormously over recent years. In contrast with this the process of recognition and interpretation of images has not shown a similar progress in medical practice. This relatively slow progress must be attributed to the difficulty of accurate segmentation of images. This research strives to provide reliable routines for extraction of organ outlines for instance, CT-scan images, which then provides a wide range of possibilities, such as quantification of tissue characteristics.

This chapter evaluates, the ideas presented in the previous chapter, by using test images to verify the overall system design. Section 5.2 discusses the experimental apparatus, that was used to obtain the raw image from which tomograms were obtained for industrial imaging. It was the experimentation with industrial images, using the Diode Array Digital Radiographic (DADR) system that provided the motivation for this research. The medical images which were used in this research were obtained from a fourth generation CT machine [50]. The testing plan, which involved both
industrial and medical images is discussed in detail in section 5.3, with emphasis on the medical imaging part of this research. This was because, the medical images were more challenging, primarily due to the various features and different sizes encountered in a single CT scan. On the other hand the industrial images that were encountered had standard geometric features, with very little variations in a typical tomogram. Section 5.4 the results of the research experimentation, programming and ideas. The results, are represented pictorially and explanations inserted where it is required. In section 5.5 a summary of the results and an overall evaluation of this image interpretation system is discussed.

5.2 Digital Radiography and DADR System:

The X-ray image quality is a function of two basic factors: number of photons detected per unit area and the spreading of the x-ray image inherent in the detection process. A SFO based detector was incorporated into a digital radiographic system as described by Hua Shao[54]. The system is based on the patented Diode Array Digital Radiography (DADR) [51] technology.

There are three types of digital radiographic systems based on different scanning methods. In area digital radiographic systems, the entire image is exposed to an area x-ray beam at the same time with the detector covering the
entire area. This is the only true real time digital x-ray imaging method. In point detection systems, a single detector and pencil beam of x-rays scans the entire imaging point by point to build up the total image. In line scanning systems, the image is built up one line at a time by sweeping a fan beam of x-rays across one dimension of the imaging area with a narrow detector covering the length of the other dimension.

Line scan digital radiographic systems take advantage of the scatter rejection of the detector geometry and electronic detectors to greatly improve the dynamic range and low contrast detectability of their images. Radiographic images are shadows of the variations in the specimen density and atomic number. X-ray images are not only produced by x-ray absorption, but also affected by Compton scattering and pair production. The broadly spread scattering will reduce the effective contrast of the incident x-ray distribution. Line scan techniques permit the use of a collimated radial fan beam. The detector area is equal to the fan beam area cutting through the specimen. The width of the field is kept directly below the beam as the specimen through the beam and only a single narrow line of the part is illuminated with the radiation. The radiation scattered from the specimen under study is not detected. Therefore the usual background scatterer noise which degrades contrast sensitivity and
spatial resolution is eliminated and a wider dynamic range of x-ray detectors is available at the detector. The line scan technique, however, has some disadvantages. Line images are integrated as the object is scanned making the process necessarily slower than an area system. Also, the use of x-rays is inefficient due to the collimation of an area x-ray beam into a fan beam.

Diode Array Digital Radiography (DADR) is a method of radiographic imaging employing the line scan technique. DADR was invented at the University of Pittsburgh by a team lead by Sashin [55]. The intent was to develop a radiographic system which provides an image quality equal to or exceeding that of conventional film-screen radiography at a lower dose to the patient. Using the DADR technique, Scantech Corporation has developed systems for industrial Non Destructive Testing (NDT) [1].

The DADR system consists of a linear x-ray camera which is mounted beneath a shuttle table such that the x-ray beam is aligned with the image detector. The x-ray beam is slightly wider than the detector aperture so that the beam is collimated to its final detected width. The aperture is made from a dense material such as Tungsten to shield the detector from scatter. The fan beam slice through the specimen and the moderated x-ray beam interacts with a narrow strip of scintillation detector to produce a visible light line image.
Figure 14 The DADR system for the Tomographic Configuration

The line image is optically amplified by an image intensifier and, then optically coupled to linear self-
scanning photodiode arrays. The photodiode arrays convert the continuous, spatially distributed light signal into discrete electrical charges by integrating their photocurrents over a specified period. Following the line integration period, the photodiodes in an array are read out sequentially to low noise preamplifiers to form a row of pixel signals making up a line of the image. The image is then processed, amplified and stored in computer memory. In the DADR system, the specimen is linearly translated through the fan beam and an image is acquired on a line by line basis.

5.2.1 Tomography With the DADR System

The arrangement for tomography is as shown in Figure 14, which consisted of the DADR system, an x-ray machine and associated detector elements. The system has a stepping motor for rotating the object through specified angular steps in order to achieve a 2-D slice of the object using the convolution technique of reconstruction. The object under test is held on each end and rotated using the stepping motor. In Figure 14, it is shown that the object was at a distance of 933.8 mm from the x-ray source and 76.2 mm from the detector, so that the fan-beam nearly approximates a parallel beam. The object is rotated about 180 degrees about it's axis of rotation until a desired resolution is achieved.
The accuracy of the measurement, and the recording and processing time represent an engineering tradeoff. A system with relatively poor resolution will record and process the data at a rapid rate, while a high resolution system will operate more slowly, and subsequently take a longer time to process the data and require greater system memory.

The on board DADR system processor which the tomographic algorithm used was a system comparable to a micro-vax. It had the capacity to rapidly do the required mathematical analysis and rapidly display a 2-D image with quality comparable with film radiography. The on-board processor had the capability to read and write from various i/o ports like mag-tape, key-board, floppy disk etc. It also had a large memory and a buffer memory for storing raw data and tomograms.

5.2.2 The Experimental Set-Up

The object under test was rotated once fully in the axial direction with 512 steps per projection, 400 such projections were taken for the test object, so that the data array had 512 x 400 elements. The raw data which were taken for 512 pixels for 400 projections over 180 degrees, assumed for parallel beam geometry. Three hundred sixty degree data was input into the reconstruction program only for true fan beam geometry. The source to object distance was 930 mm, in almost all situations excepting the last few which were at
470 mm. The object to detector distance was always 30 mm. So the Source to Imaging detector Distance (SID) was either 1.0 meter or 0.5 meter.

5.3 Test Plan

A testing plan using a sequence of test images was developed to evaluate the image interpretation system. The image reconstruction part of this research was verified using the industrial images obtained from the DADR system. Limited image processing was done on these images because the images were well defined and their shape was recognized using the Hough transform on the grey level image without employing the edge detection routine. There were essentially two main features in these images, namely circular feature and straight line feature. The circular features were all of same radius, and thus made this a simple test image. Also classification using neural network was not attempted as there were only one class(type) of these images.

The medical images which were employed needed a different testing plan. The images were reconstructed analog images. A procedure of image digitization [57] was required to convert the analog images to digital images.

The next step was to detect edges in these images and this was performed using the low level image processing routine of the image interpretation system. The edge detection process was necessary for these images as the
features were occluded by noise. To remove the noise by only filtering was found unacceptable for these images as a large portion of useful signal was also removed. Thus edge detection using the edge detector methods described in chapter 4 was performed and then filtering and closed contour operation was used on these edge images to suppress the remaining noise.

The closed contour operation used the chain code technique as described in Chapters 2 and 4. A neural network was utilized for the classification stage by extracting interesting features with the help of a radiologist (expert) [53]. The neural network was taught using expert's knowledge and the imaging systems data abstraction module. The experts knowledge was utilized to classify a given human brain scan as normal, Haemorrhaged or Lacunar infarcted. The learned neural network performed, by classifying the images as high likelihood of being normal, to high likelihood of being not normal or afflicted. The medical images that were tested were of two different categories, brain scans and the abdominal cavity scans. The test plan for these two categories is described in the next two sub sections.

5.3.1 Brain Scan

This test plan was designed for defect recognition in brain images that results from for some of the common disease that afflict the brain such as haemorrhage and lacunar
infarction. The initial step in this plan was to consult the design strategy and perform the test according to the description in Figure 11, that was to begin with the process of obtaining an image. In this case the CT brain scans were used.

Once the image was available for further processing, appropriate actions as delineated in Chapter 4 were performed on the low level processing module. This we may recall involved finding the edge images in the CT brain scans. We used LOG as the edge detector for this purpose and tried different values of sigmas, for example sigma values of 2, 4, 6, 8 and 16.

Next we moved to the intermediate level processing module of our design strategy which used the edge images produced in the previous stage to find the best possible contour, and the filtering process to remove most of the noise in the original CT scan. Chain coding was performed at this stage to obtain an output which contained chain coded strings. The chain coded strings were useful in recognizing some standard shapes. They also provided information about the length of the string and thus the contour. The output of this stage was usually a well segmented chain encoded image which had recognizable features, such as the ventricles in the brain scans. If the edge image features were large
enough, at least more than a few pixels long, we tried the Hough transform to obtain a standard analytic curve or shape.

Finally the high level process, the model based approach and neural network approach were all combined into one step by learning about the characteristics features for a normal human brain, and also by extracting a few of these important features. This step involved interaction with an expert in radiology and neurology, and extracting the relevant information from these experts [52][53] and from the integration (data abstraction) module. Some of the queries to the expert and the data abstraction module is presented as a rule based system, which is described below.

(If the image has two concentric ellipses (1 0 )

AND

If there is a clear space between these two ellipses (1 0 )

AND

If the ventricles are clearly visible (1 0 )

AND

If the shape of the ventricle is normal (1 0 )

AND

If the size of the ventricle is normal (1 0 )

AND

If the longitudinal fissure is clearly visible (1 0 )

AND

If there are no extra features present (1 0 )
Then the image was very likely a NORMAL brain scan (1 1 1)

The above rules demonstrate a typical classification scheme that was employed for human brain scans. The input data to this rule based system, implemented as a backprop neural network, were extracted from the intermediate and high level modules as discussed in Chapter 4. Typically the Hough transform and the chain encoded segment outputs, along with the user's insight, were utilised in completing the input data set.

In the case depicted above the neural network system gave an output which was a very high likelihood for a normal brain scan. Similar rule base systems were employed to identify a haemorrhaged brain scan and a lacunar infarcted brain scan. Also information from the edge image was useful in deciding between abnormality and noise. Thus, from a few general queries to the user, a seven feature vector was obtained, which was the input to the neural network module.

The expert's knowledge was utilized in classifying the output during the learning stage of the neural network. Also learning about the magnitude of the disease that was encountered such as haemorrhage and lacunar infarction. The learning process was accomplished using seven (7) characteristics features to the neural network, which was supplied with both the normal brain characteristic features
and the afflicted brain characteristic features. The recognition was done by this neural net which tries to classify the given 7 characteristics features into a known (learned) group. To be precise, the neural net if it encounters the same feature vector as it has learned from, will recognize and classify the feature vector to the same grouping as it had learnt. The recognition in this case was 100%. If the feature vector had a deviation by more than one characteristic feature, then the recognition rate by the neural net was between 90 to 95%. If we deviated more from the learning data, the recognition rate got worse. Armed with the above information that for a neural network the learning needs to be exhaustive to achieve a recognition rate greater than 90%, we came up with an exhaustive set of learning data for the neural network which covered almost all possible practical combinations of the feature vectors.

There was an attempt made to classify some more CT brain scans as normal or defective and if defective which disease it approximates best. The output of the neural network was in the form of a suggestion to the user (an inexperienced radiologist). The second part of this testing plan involved the recognition of major organs in an image. The abdominal region was selected for this purpose and eight (8) scans, at different scan levels were tested, to identify the spleen.
5.3.1 Abdominal Scan

Abdominal scans were included in the test plan, as the brain scan in the previous section were aimed at recognizing defect in an image as opposed to extracting major features (organs) in an image. A preamble which introduces the abdominal region and gives the justification for the testing plan adopted for the recognition of the objects in the abdominal scans in this research is given in the next few paragraphs.

In image processing we can distinguish image segmentation from image recognition. Segmentation decomposes the image into regions, while the recognition process generates a description of the image by assigning labels to these regions, corresponding to a model of the imaged scene. Although it might seem advantageous to keep the segmentation process completely data directed, to remain independent of a special image domain, at present it is becoming more widely understood that in a practical system segmentation and recognition should not be independent. To compensate for poor image data the segmentation process may, for instance, be provided with context information, derived from a model describing the imaged scene. In the method which is described here this approach is utilized by using search areas to guide segmentation process. Although some use of heuristics could not be avoided the larger part of the
processing could be fitted into a structure which is generally applicable as described in chapter 4.

In this research the design method which was developed was applied to recognition of defects in a brain scan. It has also been used for detection of the spleen, and other abdominal organs CT scans. A CT image sequence provides a three-dimensional scene representing the X-ray attenuation of the anatomical structures inside the body. It might seem attractive therefore to perform the analysis in 3D space.

Figure 15 Semantic network representing a part of the abdomen
It should be noted, however, that the resolution perpendicular to the scans normally is about an order lower than in the scanning planes, perhaps even worsened by the fact that during the time between acquisition of subsequent scans the patient may have slightly moved. Furthermore it should be realized that in most cases each scan itself contains enough clues for reliable extraction of organ boundaries since it is common practice for radiologists to do this by hand.

Image understanding requires a priori knowledge about the task domain. Such knowledge may be about the spatial relationships between the imaged objects or about individual object attributes. Instead of using this knowledge only in a later stage for interpretation of regions already established by domain-independent processing, it can also be used to guide the processing. This idea has been applied in the present method. The main point is the use of knowledge about the spatial relationships between the objects in the scene, to constrain search areas for the objects to be identified. Because operations on pixel levels can be restricted to these areas, it is clear that this model guided approach avoids unnecessary processing. Moreover the use of search areas is an elegant way of incorporating global considerations about the image structure into the segmentation process, thus increasing its reliability.
The order of evaluation of the segmentation procedures is to be determined by a control structure. This order is of main importance because performance of a procedure may depend largely on the amount of knowledge already gathered in the recognition process. A natural choice of the order of evaluation will be based on the rule that the procedure which has the largest chance to be successful, based on what is known so far, should be started. Application of this rule in analysis of abdominal CT causes bone parts to be identified first, because these can be easily segmented by thresholding. The detection of ribs, spine and sternum will be followed by construction of the rib cage, which can be viewed as a general search area for the interior organs.

To perform the recognition task three types of data are accessible to the higher level processing system:
(a) the image data,
(b) positions and shapes of already labelled regions, and
(c) a list of parameter values.

The latter represents explicit knowledge about the imaged scene, for instance, describing sizes of the patient body or representing mean values of shape descriptors for particular anatomical structures. Parameter values may be set a priori or may be determined from the image data, before or during the recognition process.
In the first step the search area for the object to be identified was constructed by using the relations defined in the semantic net and the outlines of already labeled image parts. Additional parameters like sizes and reference coordinates were also used. The next step was the operation on the image data inside the search area. First an initial classification of pixels was carried out by edge finding and windowing, resulting in a binary image of object and background regions. This operation was followed by a method to separate loosely connected regions, or fill up holes and connect clusters. In the final step the resulting set of pixels was examined in order to find a region having attributes corresponding to the object to be identified. For this purpose each connected region was chosen to approximate standard geometrical patterns like a straight line, circle, ellipse, etc., so that the Hough transform routine could assist in recognition process. If object identification fails it was possible to adjust the segmentation procedure so that loosely connected regions were reinforced by using more restricted search area.

The procedure for the spinal column was subdivided into three parts. First the spinal body was isolated, then the arch and dorsally located structures were labeled, and finally the lateral area was searched for the transverse process.
A search area for the spinal body is defined relative to the already determined reference point ventrally in the spinal canal. The image was binarized by edge detection operation which was performed on the image within the search area to remove small details like the upper parts of the ribs. Then the image was converted to a segment table and features were determined for evaluation of the remaining regions. The features that were used for classification of the spinal body were center of area position, area and compactness. The latter measure was defined by the square of the perimeter divided by the area.

If there are holes in the binary image of the spinal body no consistent result could be found. This could not have been avoided by taking a lower threshold, because absorption values inside the spinal body was lower than in the surrounding tissue, especially for older patients. In this case a closing was performed on the image to fill up holes and the procedure was repeated.

The spinal arch and spineous processes are located at the dorsal side of the spinal body and could be found easily by finding the edges in those regions. For segmentation of the transverse processes those objects regions in the remaining part of the image were considered which are connected to the spinal arch. These may depict ribs, processes, or a connection of both. Two geometric features,
representing the place at which a region was connected to the arch and its lateral extension, were used for classification. In case recognition based on these regions failed an ad hoc procedure was used to extract the part of the object region more likely belonging to the spine.

The area which is enclosed between ribs, sternum and spinal column will be referred to as the rib cage [58]. Where possible its outline follows the inner side of the bone structure. Between the bone parts, where rib cage contour may be less obvious, its boundary was determined by interpolation. As the rib cage was constructed from already labeled bone parts the original image data were not used in this procedure.

First a closed contour was generated through the bone structures by using clustering (segmentation) and distance metrics. Then the region inside the contour was labeled as rib cage, with exclusion of the bone pixels. Sometimes previous scan data was used to get the proper contour. To obtain a smooth curve the continuity of the first derivative was imposed across each point, except for the fixed point inside the spinal body. In this point a discontinuity of the tangent direction allows the curve to bend sharply towards the closest ribs on both sides of the spine.

The spleen must be located inside the rib cage on the left side of the spinal column. This apriori knowledge from
Grays Anatomy [58], was used to construct the splenic search area. The mean gray level value for the spleen was searched in the region of interest. Whichever pixel satisfied the criterion which was within two standard deviations, were selected and highlighted. The general contour of the area was obtained and compared with the existing contour for the spleen. Any discrepancy was noted after accounting for the difference in size and age of the patient. If there was no good match then the criterion could be made slack or tight to achieve a good match. If there are holes inside the contour then a tumor or defect could be inferred. The case for tumor became strong if there were holes visible in the contour of the spleen in more than one scan or image.

For classification of the splenic area the region attributes area, center of area, and compactness were used, where the later was defined as the square of the perimeter divided by the area. This along with the splenic outline was used by the neural network routine in its learning and also in the recognition process.

The real problems in spleen recognition were that sometimes the scan showed the spleen connected to other organs, such as the pancreas or the stomach wall, which had about the same pixel value. Then the procedure of splitting was used, where the search procedure cleaned the image of small details and more importantly disconnected large objects.
regions in case they were only loosely connected to each other. This was the case of applying stringent conditions for clustering like pixels, and the distance metrics so chosen that pixels farther apart then the dimension of the splenic outline stay disconnected.

5.4 Results and Analysis

In the previous chapter an image interpretation system a product of this research was detailed. This section strives to verify the modelling and the accuracy by utilizing the test plan described in the previous section for the various components that form the total image interpretation system. There were four(4) normal CT brain scans and four(4) abnormal or afflicted CT brain scans which were tested. Also eight(8) abdominal CT scans were added to complete the sequence of medical test images. Also industrial images were used and because of their simplistic rendition of the whole recognition process, only a couple of them are depicted. Initially the brain scans were taken digitized and was input to the image interpretation system. The low level processing stage of this system detected the edges present in this brain scan. Various edge images were obtained utilizing different values of sigma, for the edge detector described in chapter 2 and revisited in chapter 4. This edge image was input to the next stage of the image interpretation system, and using the chain coding methodology a closed contour image was obtained.
There were various choices at this stage of processing, such as the length (number of pixels) in each contour. Also number of contours in an image was controlled by this technique.

Then the Hough transform technique was utilized to obtain all the analytic features present in the image. Finally the features were extracted using expert's knowledge and the interpretation (recognition) system activated. Since the neural network recognition system that was used was backpropogation (backprop), which required supervised learning the neural network was trained using expert's knowledge about the normal brain and the afflicted brain. In the case of the brain scan the expert's views for a normal brain scan were:

1. There has to be two concentric elliptical shapes to identify the organ as brain.
2. The Cerebro Spinal Fluid (CSF) should be visible as dark (or light in reverse video).
3. The ventricles should be visible clearly and there should not be any edges near the ventricles other than the ventricle shape. Edges near the ventricles signify dead tissue in the thallamus region.
4. Also the ventricle should not be occluded by blood, visible as a dark feature. Shape of the ventricle should be visible as normal.
5. The ventricle size should not be too large, as this indicates abnormality. The size of the ventricle was checked
by using the closed contour edge image and verifying the number of pixels, connected to the ventricle.

6. There should not be any blood in or near the longitudinal fissure, which runs across the brain.

7. There should not be any extra features visible and if present their grey level values be checked to identify any known tumor.

The neural network was taught taking this 7 valued vector as input, and the corresponding output from the supervisor (expert's suggestion). The neural network was tested for both practical and exhaustive possibilities on a trial problem. The performance by the backprop network on identical feature vectors extracted from test images after exhaustive learning was perfect. The performance by the backprop network in recognizing the input and its corresponding output on the same identical features but with practical learning was ~ 90% perfect. If identical values were given, for the test input set and the learning input set, then there was 100% match between the output of the learning and test cases.
Figure 16 Images for a normal Brain scan 1
Figure 17 Additional images for a normal Brain scan 1
Figure 18 Images for a normal Brain scan 2

Figure 19 Images for a normal Brain scan 3
Figure 20 Images for a normal Brain scan 4
<table>
<thead>
<tr>
<th>Input Vector</th>
<th>Output Vector (Normal Brain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1 1 1 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1 1 1 1 1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 1 1 1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 1 1 1 1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 1 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 1 1 0 0 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 1 1 0 1 1 1</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 1 1 0 1 1 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 1 1 0 1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 1 1</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 1 1 0 0 0 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 0 0 1 1 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 1 0 0 1 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 1 0 0 0 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 1 0 0 0 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 1 1 1 1 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 0 1 1 1 1 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 1 1 1 0 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 1 1 0 1 1</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 1 1 0 1 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>1 0 1 1 0 0 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 1 1 0 0 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 1 0 1 1 1</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 1 0 1 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 0 1 0 1 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1 0 1 0 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1 0 0 1 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1 0 0 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1 0 0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 1 1 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 1 1 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 1 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 1 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 0 1 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 0 1 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 0 0 1</td>
<td>0 0 0</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0</td>
<td>0 0 0</td>
</tr>
<tr>
<td>0 1 1 1 1 1 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>0 0 1 1 1 1 1</td>
<td>1 0 0</td>
</tr>
</tbody>
</table>

Figure 21 Backprop learning for a Normal brain
Figure 16 shows the scan for a normal healthy human brain. Since the size and shape of the brain does not vary a great deal between mature adults, we can take this sample as representative sample for the normal healthy human brain. Figure 16 (a) shows the original CT scan for a normal human adult. Figure 16(b). shows the edge image for the brain scan in Figure 16(a), obtained using a Laplacian Of Gaussian (LOG) filter using sigma =1. Similarly Figures 17(a) to 17(d) were obtained for sigma values 2, and 4, respectively. Figures 16(c), 17(b) and (d) were for chain coded edge images with emphasis on organ shape and size criterion's and any special features, in this case the ventricles, the longitudinal fissure, and the CSF in the brain scan. Figures 18, 19 and 20 are other examples for the normal human brain at different scan levels. The scan levels in this case refer to adjacent slices taken during a CT exam of a patient.

Figure 21 is the backprop learning input and output vector set for a normal brain. This data set was extracted by the procedures delineated in Chapter 4 and, earlier sections of this chapter. Some of the data was directly extracted from previous processing stages such as the Hough transform and the chain code, and a part from the assistance of an expert radiologist. Thus there was an interaction with a radiologist and relevant queries were posed. There are a total of 50 data sets used for the learning stage for the
classification of the brain scan. The output vector shown is an abstraction of the expert's knowledge. The expert radiologist explained the difference between the different brain scans. An interpretation of the expert's opinion was done for this research work. As explained before the string "1 1 1" represents very likelihood of the brain scan being normal.

The feature vector specified above was a complete set of input vectors, which handled all the possible brain scan levels. A radiologists [53], opinion was used to model the learning for the neural network as described below and then verify the suggestions given on the test images by this image interpretation and neural network system.

The normal CT brain scans were taken at the same scan level as the four unhealthy CT brain scans. The description of the normal scan was done earlier for clarity. Thus we have 4 sets of images, with the first set of images being represented by normal scan 1 and haemorrhaged scan 1. Normal scan 2 and haemorrhaged scan 2 forming the second set of images. Normal scan 3 and Lacunar infarcted scan 3 forming the third set of images and finally normal scan 4 and Lacunar infarcted scan 4 forming the final set of brain scan images.
Figure 22 Images for a Haemoraged Brain scan 1
Figure 23  Images for a Haemmoraged Brain scan 2
Figure 24 Images for a Lacunar Infarcted Brain scan
Figure 25 Images for a Lacunar Infarcted Brain scan 4
Input Vector | Output Vector (Haemorraged Brain) | 0 0 0 - very unlikely | 1 1 1 - very likely
---|---|---|---
1 1 1 1 1 1 | 0 0 0 | 1 1 1 - very likely
1 1 1 1 1 0 | 0 0 1
1 1 1 1 1 1 0 | 0 0 1
1 1 1 1 0 1 0 | 0 0 1
1 1 1 1 0 0 0 | 0 1 0
1 1 1 1 0 1 1 | 0 1 0
1 1 1 1 0 1 0 | 0 1 0
1 1 1 1 0 0 1 | 0 1 0
1 1 1 1 0 0 0 | 0 1 0
1 1 1 0 1 1 1 | 0 1 0
1 1 1 0 1 1 0 | 0 1 0
1 1 1 0 1 0 1 | 0 1 0
1 1 1 0 1 0 0 | 0 1 0
1 1 1 0 0 1 1 | 0 1 1
1 1 1 0 0 1 0 | 0 1 0
1 1 1 0 0 0 1 | 1 0 1
1 1 1 0 0 0 0 | 1 1 0
1 1 0 0 1 1 1 | 0 1 1
1 1 0 0 1 1 0 | 0 1 0
1 1 0 0 1 0 1 | 0 1 1
1 1 0 0 1 0 0 | 1 0 1
1 1 0 0 0 1 1 | 0 1 1
1 1 0 0 0 1 0 | 1 0 1
1 1 0 0 0 0 1 | 1 0 1
1 1 0 0 0 0 0 | 1 1 0
1 0 1 1 1 1 1 | 1 1 0
1 0 1 1 1 1 0 | 1 1 0
1 0 1 1 1 0 1 | 1 1 0
1 0 1 1 1 0 0 | 1 1 1
1 0 1 1 0 1 1 | 1 0 1
1 0 1 1 0 1 0 | 1 1 0
1 0 1 1 0 0 1 | 1 1 0
1 0 1 1 0 0 0 | 1 1 1
1 0 1 0 1 1 1 | 1 1 0
1 0 1 0 1 1 0 | 1 1 0
1 0 1 0 1 0 1 | 1 0 1
1 0 1 0 1 0 0 | 1 0 1
1 0 1 0 0 1 1 | 1 0 1
1 0 1 0 0 1 0 | 1 0 1
1 0 1 0 0 0 1 | 1 0 1
1 0 1 0 0 0 0 | 1 1 0
1 0 0 0 1 1 1 | 1 0 1
1 0 0 0 1 1 0 | 1 0 1
1 0 0 0 1 0 1 | 1 0 1
1 0 0 0 1 0 0 | 1 0 1
1 0 0 0 0 1 1 | 1 0 1
1 0 0 0 0 1 0 | 1 1 1
1 0 0 0 0 0 1 | 1 1 0
1 0 0 0 0 0 0 | 1 1 1
0 1 1 1 1 1 1 | 0 0 1
0 0 1 1 1 1 1 | 0 1 1

Figure 26 Backprop Learning for a Haemorraged brain
<table>
<thead>
<tr>
<th>Input Vector</th>
<th>Output (Lacunar Infarcted Brain)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0 - very unlikely</td>
</tr>
<tr>
<td>1 1 1 1 1 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 1 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 1 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 1 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 1 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 1 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 1 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 1 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 1 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 1 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 1 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 1 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 1 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 1 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 1 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 1 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 0 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 0 1 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 0 0 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1 1 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 1 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>0 0 1 1 1 1 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
</tr>
</tbody>
</table>

**Figure 27** Backprop Learning for an lacunar infarcted brain
The CT scans and the corresponding edge images for the case of thalamic haemorrhage, are shown in Figures 22 and 23. This was one of the brain condition encountered in our situation, there were few key features described by the expert as important to classify a brain scan as thalamic haemorrhage. One there may be some amount of blood in the CSF region. This region can be easily identified as the elliptical or circular region between the brain tissue and the skull. Next there may be a loss of blood in some of the brain tissue because of haemorrhage and this will manifest itself as dead tissue in the brain (near the thalamus). Thalamus is below the ventricles and the ventricles being clearly discernible in a healthy brain and occluded by blood in the afflicted brain. The cluster of dead brain cells (tissue) with a different grey-level should be found around the ventricles.

Next if any calcification has taken place, then the thalamus region will have an edge (in image processing sense) where calcification has occurred in the brain tissue. Also the longitudinal fissure should be observed for any clot of blood in this region. The longitudinal fissure is the fissure that separates the two cerebral hemispheres.

The feature vector that was employed was 7 dimensional, so that we could also test for a few features for the normal brain, as there were only 4 other (defect or disease) feature
given by the expert. The other three features were to check for the grey-level values of the brain tissue in the ventricles, the edge information about some of the prominent fissures in the brain like the longitudinal fissure. Final feature that was determined was the shape of the brain, to verify whether it conformed to the standard circular or an elliptical shape.

A similar kind of feature vector representation for a lacunar infarction another condition for an unhealthy brain was encountered in the second example as shown in Figures 24 and 25. This condition was the reverse of haemorrhage, in the sense, that the blood supply was cut off for a part of the brain and as a result of which some of the affected cells and tissue died. This again according to the radiologist (expert), manifests itself as a boat shaped feature near the ventricles.

Figures 26 and 27 is the backprop learning input and output vector set for a haemorrhaged brain and a lacunar infarcted brain scan. This data set was extracted by the procedures delineated in Chapter 4 and, earlier sections of this chapter. Some of the data was directly extracted from previous processing stages such as the Hough transform and the chain code, and a part from the assistance of an expert radiologist. Thus there was an interaction with a radiologist and relevant queries were posed. There are a
total of 50 data sets each used for the learning stage for the classification of the brain scan. The output vector shown is an abstraction of the expert's knowledge. The expert radiologist[52][53] explained the difference between the different brain scans. An interpretation of the expert's opinion was done for this research work. As explained before the string "1 1 1" represents very likelihood of the brain scan being haemorrhaged or lacunar infarcted, depending on the figure that is encountered.

The over sized ventricles though visible to a trained eye of an expert radiologist in Figure 24(a) was easily discernible in the edge images of Figures 24(b) and 24(c), even to the untrained radiologist(user). Also another scan for the same condition was given in Figure 23 and in this case Figures 25(b) and 25(c), give a much clearer indication of the large ventricle size. The oversized ventricles were an indication of the onset of lacunar infarction.

As described earlier in the section about neural nets, different disease or defect needs it's own learning vector and the vector chosen should have all representative samples of the defect and normal brain image so that it can properly distinguish between a normal healthy brain and an afflicted (not normal) brain. It is to be noted, that the input vector remains the same, and only the output vector changes depending on the defect encountered.
Figure 28 Images for Abdominal region scan 1
Figure 29 Additional images for Abdominal region scan 1:

(b) Edge image using $\sigma = 4$

(c) Edge image using $\sigma = 8$

(d) Chain coded image for $\sigma = 8$
As described in section 5.3, the abdominal region of the body has many organs, and in this research, attempt was made to identify some of the organs like the liver, stomach, duodenum, left kidney, pancreas and spleen. The eight original abdominal CT scans, and their edge images corresponding to various sigmas are depicted in Figures 28 through 36. The original CT scans have gray level values and the edge images are binary.

To determine if the shape of the image entity corresponds to the shape of the organ in the abdominal cavity, the original CT scan was taken and an edge image extracted out of it using the LOG filter. Various sigma values of 1, 2, 4, 8 and 16 were used to obtain the edge images.

There was no one good value of sigma that was applicable to all the scans. The sigmas are sensitive to noise and also to the feature size that is being extracted. For example, the shape of the left kidney was extracted from Figures 31 and 32. Figure 32(b) with sigma value of 2 and the edges further refined by the chain coding and contour following technique, gave a better matching to the shape of the kidney, than using other values of sigma. Figure 32(d) also gave a good shape representation for the left kidney, but the rib cage appeared discontinuous.
Similarly the shape of the spleen and the stomach were extracted from Figure 33. In this case again a sigma value of 2 was used and the edge image refined using the chain code. The shape of the liver was extracted from Figure 29, using a sigma value of 1. The shape of the spinal body and the spinal cord was seen in almost all of the edge images.

The duodenum as seen in Figures 31 and 37, had an "inverted three" shape, instead of the "C" shape for a normal duodenum. The expert (radiologist) considered this kind of "inverted three" shape as an abnormal condition for the pancreas. The abdominal scans which were obtained was of a patient suffering from abnormality of the liver, pancreas and duodenum. The pancreas being occluded from all the scans by a tumor also served as an indicator of the abnormality. Thus the pancreatic shape nor the pancreas were visible in any of the scans or edge images.

The neural net results are not explicitly depicted, because the output was of the form; the probability of haemorrhage is very high (>90%), probability of haemorrhage is (75%) high, there is a 50% probability of haemorrhage, probability of haemorrhage is slim(<30%), probability of this being brain image is high(>90%), and this is a unknown image. To be precise, there was no structure to the results, to either classify them in tabular form or pictorial form.
Figure 30 Images for Abdominal region scan 2

(a) Original CT scan
(b) Edge image using $\sigma = 1$
(c) Chain coded image for $\sigma = 1$
(d) Edge image using $\sigma = 2$
(e) Chain coded image for $\sigma = 2$
Figure 31 Images for Abdominal region scan 3
Figure 32 Images for Abdominal region scan 4
Figure 33 Images for Abdominal region scan 5
Figure 34 Images for Abdominal region scan 6
Figure 35 Images for Abdominal region scan 7
Figure 36 Images for Abdominal region scan 8
The neural net because of its exhaustive training did very well ~95%, correct diagnosis, for the recognition of the organ and ~90%, correct diagnosis to classify a normal brain as normal, and ~90% correct diagnosis to classify an afflicted brain as afflicted or not normal brain. The diagnosis of the correct disease for an afflicted brain was also ~90% because of the independent nature of the seven(7) dimensional learning vector for both the cases of thalamic haemorrhage and lacunar infarction. The performance verification (recognition) vector was also seven(7) dimensional and the same weight(learning) vector was used for both these cases.

In the case of abdominal scans, independent set of characteristics vectors were tested for different features on the same scan. The recognition rate here was ~90%, to correctly classify an organ(stomach) as stomach. Six scans of the eight abdominal scans were used for learning and the remaining two were used to test the learning. The system was also able to classify up to 5 features (organs) in one scan. Occlusion does create a problem when part of the image gets blocked, then we used heuristics developed earlier in chapter 5 to guide us through the recognition process. The heuristics process used the relative location(position) of the organs in the abdominal cavity and the distance between them to help in recognition. Also the grey level values and
CT numbers for the organs were used in the neural network classification to further prune the search for the correct organ.

The results and discussion in this chapter, would be incomplete if the industrial images the motivation for this research are not described. Figure 37 shows a typical industrial tomography image, obtained after performing image reconstruction on a raw image (not shown). This image was tested for recognition of few features present, circles and rectangles. The circular features were recognized using the Hough transform for radius 1 (because all were 1 cm in diameter), and the rectangular features were also recognized using the Hough transform but for four separate straight lines. There was a pixel value different than the surrounding pixel value in one of the circular features. This was a known defect of 100 microns introduced to verify the accuracy and resolution of the image reconstruction technique.

Figure 37 was a demonstration of the image reconstruction algorithm on the industrial image obtained using the DADR system. This shows a 100 x 100 output image reconstructed using a 128 x 100 input image. There were 100 projections (rotations every 3.6 degree) of 128 pixels each. Filter Back Projection (FBP), was the image reconstruction
algorithm used and the filter was a Shepp and Logan band pass filter.

Figure 37  Tomographic 2-D slice of an industrial Phantom

5.5 Summary:

Initially an introduction to the experimental apparatus used for industrial tomography was described. This work involved performing tomography from basic principles, that is from taking multiple projections to image reconstruction. Later some of the medical images (CT brain and abdominal scans), was described. Including a procedure for extracting features from these images, by the use of the image interpretation system(process). The results in figures 15-32 clearly demonstrates the viability of this procedure of image interpretation. Further various important features extracted were manually input to a neural network program (backprop).
The neural network needed supervised learning, so an input and corresponding output data set was furnished, by using similar images. A learning weight matrix was developed and then the neural network performed on the unknown images and came up with a diagnosis. The neural network performed > 90% accurate. The neural network did a recognition of 100% on test images used for learning. The backprop network can perform close to perfect accuracy if supplied with a large amount of data while learning. Learning was slow but performance was exceptionally fast.

In figures 16-36 we see a depiction of original CT scan, edges, and contours. We found that the just the edge information, was a great improvement from the original CT scan, which had its features mixed with noise. The edge image brought out the features in the original CT scan, and helped the user to identify the various features present in the image. The closed contour image further consolidated this feature representation by forming contours, which took the shape of the organs (in the case of abdominal scans). The ability of the user (radiologists) gets enhanced when they are offered the edge image and closed contour image along with the original CT scan. The ability of the expert [53], was improved when supplied with the edge and closed contour images in addition to the original CT scan. This was demonstrated by providing the expert with just the original
CT brain scan for Lacunar infarction, and recording the diagnosis (d1). Later adding the edge and closed contour images for the corresponding CT scan and again recording the diagnosis (d2). The two diagnosis d1 and d2 were compared with the original diagnosis [52] that was provided with the CT scans. In the test cases it was found that d2 was always closer than d1. Further additional images help clear any ambiguity the expert had and was not a detrimental factor.
Chapter VI

CONCLUSIONS

6.1 Introduction

The conclusions that are drawn are partially based on the results from the previous chapter. This research set out with some specific objectives and bringing together, techniques that were employed in other research studies involving imaging, image reconstruction, image processing, pattern recognition, neural network, artificial intelligence, and the Hough transform. A good image recognition system should employ more than one of these techniques in order to achieve a good image interpretation system. This research accomplished this goal by using many of these techniques listed above by; demonstrating the need of a particular technique; and implementing it in the correct sequence and thus improving the image interpretation system.

The second goal of this research was to provide the user (inexperienced radiologist) a tool, for image recognition and feature classification of the objects present in a sequence of images. This goal was also accomplished, by using the integrated module and abstracting useful information for brain scans with thalamic haemorrhage or Lacunar infarction.
condition. Also discriminating between normal brain scans from afflicted ones, was accomplished using the neural network. Also in the domain of abdominal region, where detailed procedure for recognizing spleen and splenic cross section was provided. Also this study has been used to create an image base for recognizing organs other than the brain, or brain defects. The abdominal region organs, such as the left kidney, the liver, the stomach, the pancreas, the spinal body, and the rib cage have been recognized in the second phase of testing of the medical images. To achieve an image interpretation of all these organs in an image, which recognizes all these features, at the same time, extra features for the neural network were needed. Instead of using 7 feature vectors we had to use atleast 20 feature vectors along with the semantic net information or model based information which has the relative location information about different organs of the abdominal region. Programs that were developed for this research endeavor is available using telnet from Seshadri@element.eng.ohio-state.edu. Image processing programs can be found in directory image_process, neural network backprop in the directory neural_net, images in the directory images, and display routine in the directory display. Image reconstruction routines are in directory CT. Information about connecting to element can be got e-mail to the address or to lane@element.eng.ohio-state.edu.
6.2 Applications

This research has accomplished its purpose of demonstrating the need and integration of many techniques for an image interpretation system. Applications from this research are two fold; one in applying the idea behind this technique that is to demonstrate a need and then integrate many techniques from other engineering discipline, and scientific fields to other applications in nuclear and other engineering work. Yet another application involves utilizing this image interpretation system for domains other than medical imaging, such as industrial imaging and satellite imaging, and also towards character recognition, finger print detection, and in other areas like picture recognition, in criminal and other law enforcement investigations. Also this can be utilized in airports for baggage surveillance system with proper knowledge base and samples.

6.3 Future Work

This research has addressed image interpretation for one particular modality, that is the computer tomography or CT. In order to apply this technique to other modalities, registration of the image is necessary. This is a recent (less than a year old) technique that has been developed, which identifies salient features in an organ in one
particular modality say Magnetic Resonance Imaging (MRI), and searches for an identical salient feature for the same organ, in another modality say Positron Emission Tomography (PET). Thus there is no need for scaling or rotational or translational invariant features for the imaging system. Instead each image should be registered and this registered image, included in a data base accessible to all other imaging modalities. Thus there is only one registration per organ. A standard has yet to be set for this registration technique.

Another extension is to have more than one algorithm available for each step in processing so that the user has more choices and therefore better imaging capability. Another important need is for a good graphical display capability, and this research could not address this need for lack of time and availability of necessary hardware and software. A good graphical system can be placed on the rear end of this image interpretation system and can help in displaying the image enhancements on the original and edge images. Various other techniques for image recognition may be needed if there are many non-analytic features in an image so that the Hough transform becomes ineffective. Techniques, such as Kohonen feature map or Grossberg's Art2, are elegant techniques which can help in improving recognition and these techniques could also replace the back-propogation routine.
REFERENCES


11. P. Munshi's "Error Estimates for the Convolution Backprojection Algorithm in Computer Tomography", Ph.D


24. A. Rosenfield and A. C. Kak *Digital Picture


52. Personal Communication with Dr. Vishweshwariah of Jubilee CT Scanning Institute, Bangalore, India.

53. Personal Communication with Dr. Sumathi, Dayton, Ohio.


56. Neural network software release 1.0, from SAMPL, The Ohio state university, Columbus. Ohio.

57. Digitizer at the Geodetic Sciences Department, The Ohio state university, Columbus, Ohio

58. Grays Anatomy